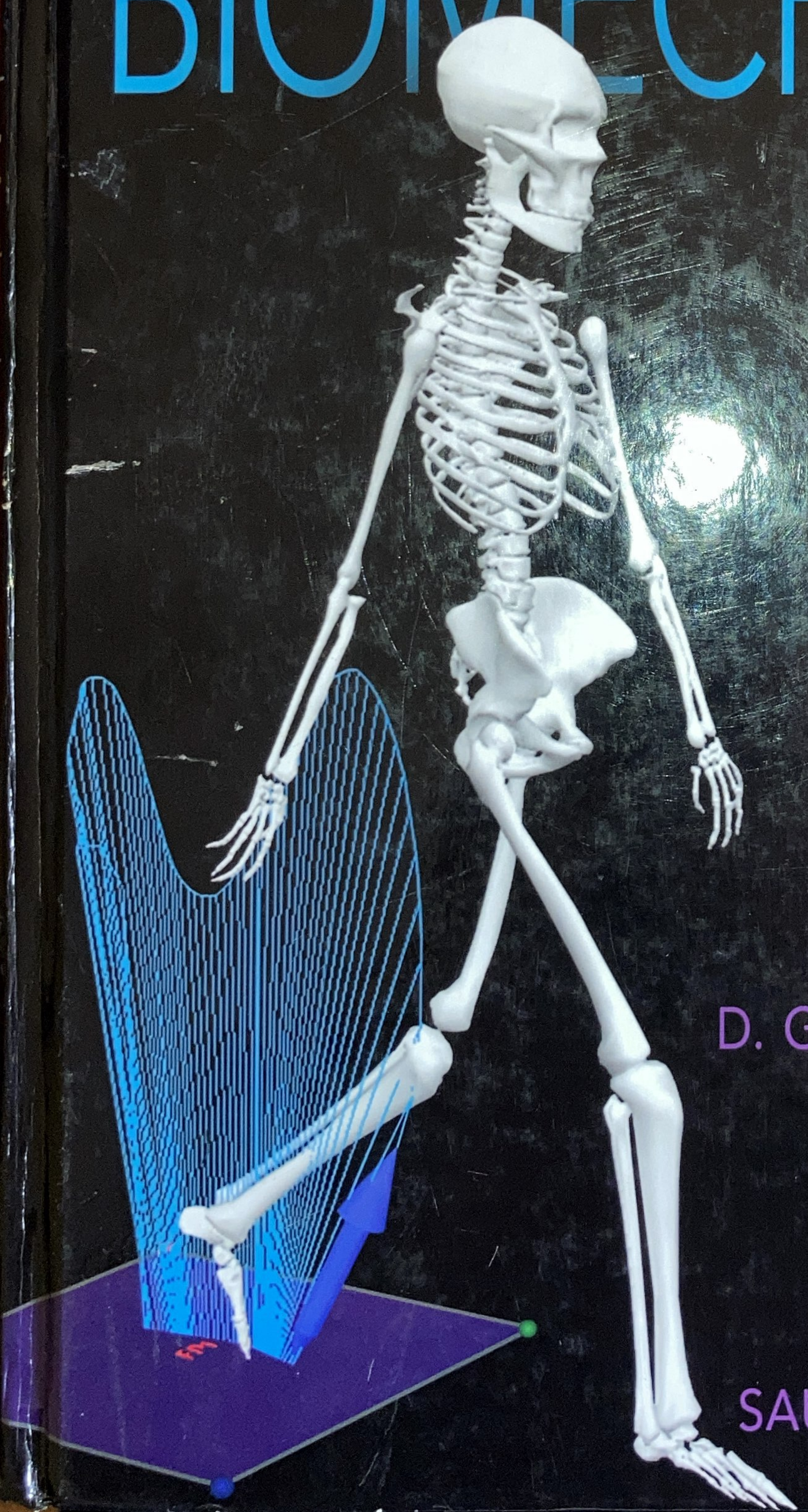
Research Methods in BIOMECHANICS



SECOND EDITION

Includes access to Visual3D Educational Edition software

D. GORDON E. ROBERTSON
GRAHAM E. CALDWELL
JOSEPH HAMILL
GARY KAMEN
SAUNDERS N. WHITTLESEY

Chapter 12

Signal Processing

Timothy R. Derrick

A **signal** is a time- or space-varying quantity that conveys information. A **waveform** is a time- or space-varying quantity that contains either information, unwanted data called **noise,** or both. A waveform may take the form of a sound wave, voltage, current, magnetic field, displacement, or a host of other physical quantities. These are examples of continuous signals (meaning that the signal is present at all instances of time or space), also called **analog signals**. For convenience and to enable manipulation by computer software, we often convert continuous signals using **analog-to-digital** converters into a series of discrete values by sampling the phenomena at specific time intervals to create an equivalent digital signal (figure 12.1). The situation can also be reversed by **digital-to-analog** converters so that a digital signal can be viewed electronically. In this chapter, we

- define how to characterize a signal or waveform,
- examine the Fourier analysis of waveforms,
- explain the sampling theorem and Nyquist frequency,
- discuss how to ensure cyclic continuity, and
- review various data-smoothing techniques for attenuating noise from waveforms.

CHARACTERISTICS OF A SIGNAL

A sinusoidal time-varying signal has four characteristics: **frequency** *(f)*, **amplitude** *(a)*, offset (a_0), and **phase angle** (θ). These characteristics are depicted in the schematics in figure 12.2. The frequency represents how rapidly the signal oscillates; it is usually measured in cycles per second or hertz. One hertz is equal to 1 cycle per second. For example, the second hand of a clock completes one cycle every 60 s. Its frequency is one cycle per 60 seconds or 1/60 Hz. The frequency of a signal (figure 12.2*a*) is

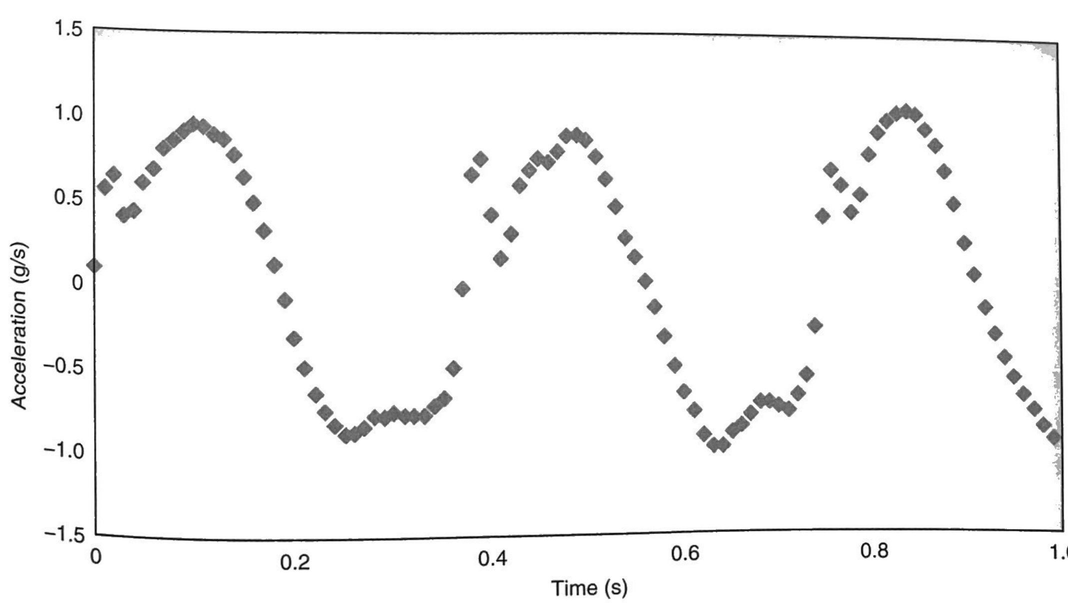


Figure 12.1 The digitized discrete representation of the acceleration of the head while a subject is running. The signal was sampled at 100 Hz (100 samples per second).

easy to determine in a single sine wave but more difficult to visualize in noncyclic signals with multiple frequencies. The amplitude of a signal (figure 12.2*b*) quantifies the magnitude of the oscillations. The offset (or direct current [DC] offset or DC bias) (figure 12.2*c*) represents the average value of the signal. The phase angle (or **phase** shift) in the signal (figure 12.2*d*) is the amount of time the signal may be delayed or time shifted.

Any time-varying signal or waveform, $w(t)$, is made up of these four characteristics. The following equation incorporates each of the four variables:

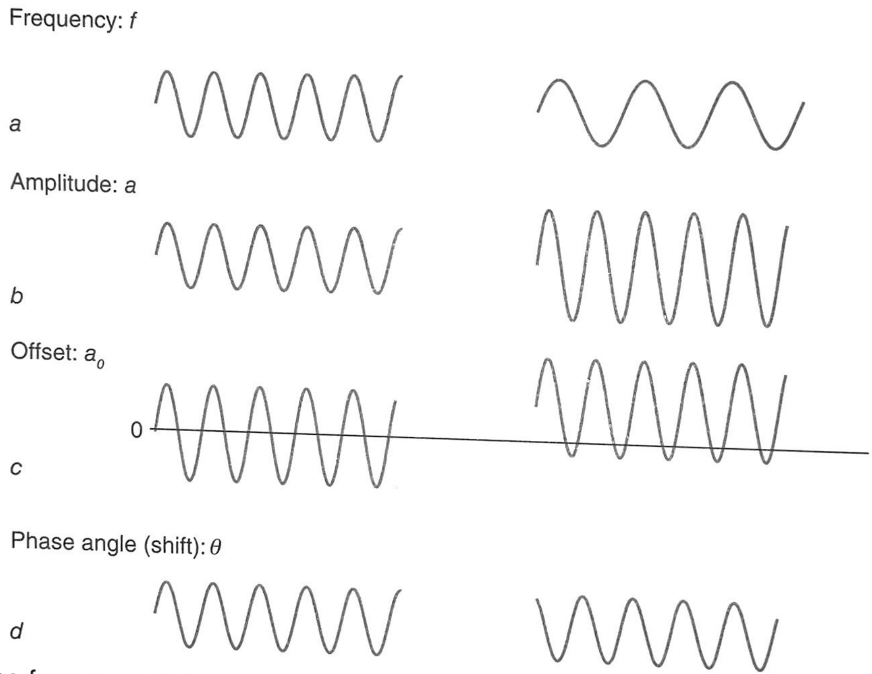
$$w(t) = a_0 + a \sin(2\pi ft + \theta) \tag{12.1}$$

Alternately, the equation can be written using the equivalent angular frequency, ω, where $\omega = 2\pi f$ (because f is in cycles per second or hertz, ω is in radians per second, and there are 2π radians in a cycle). Thus another way to write this relationship is

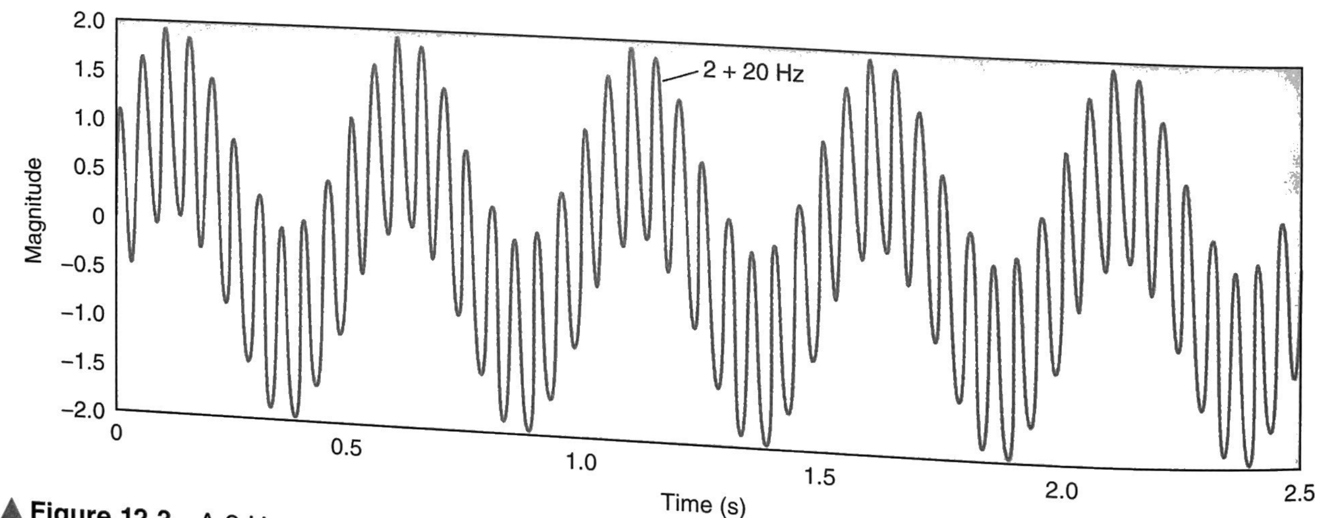
$$w(t) = a_0 + a \sin(\omega t + \theta) \tag{12.2}$$

The time (t) is a discrete time value that depends on how frequently the signal is to be sampled. If the sampling frequency is 100 Hz (100 samples per second), then the sampling interval is the inverse (1/100th or 0.01). This means that there will be a sample or datum registered every 0.01 s. So, t is one of the discrete values in the set (0, 0.01, 0.02, 0.03, . . . , T). The variable T represents the duration of the digitized signal. For example, by adding the equations for a 2 Hz sine wave to a 20 Hz sine wave, we created the following waveform (as illustrated in figure 12.3):

$$w(t) = \sin(2\pi 2t) + \sin(2\pi 20t) \tag{12.3}$$



▲ **Figure 12.2** The four essential components of a time-varying signal.



▲ **Figure 12.3** A 2 Hz and a 20 Hz sine wave summed over a 2.5 s period. The offset (a_0) and angle (θ) are zero for both waves, and the amplitudes are 1.

FOURIER TRANSFORM

Any time-varying signal can be represented by successively adding the individual frequencies present in the signal (Winter 1990). The a_n and θ_n values may be different for each frequency (f_n) and may be zero for any given frequency:

$$w(t) = a_0 + \sum a_n \sin(2\pi f_n t + \theta_n) \quad (12.4)$$

By using the cosine and sine functions, this series can be rewritten without the phase variable as

$$w(t) = a_0 + \sum [b_n \sin(2\pi f_n t) + c_n \cos(2\pi f_n t)] \quad (12.5)$$

This series is referred to as the *Fourier series.* The b_n and c_n coefficients are called the *Fourier coefficients.* They can be calculated using the following formulas:

$$a_0 = \frac{1}{T}\int_0^T w(t)dt \quad (12.6)$$

$$b_n = \frac{2}{T}\int_0^T w(t)\sin(2\pi f_n t)dt \quad (12.7)$$

$$c_n = \frac{2}{T}\int_0^T w(t)\cos(2\pi f_n t)dt \quad (12.8)$$

Here is another way of looking at it: If you want to know how much of a certain frequency (f_n) is present in a signal $w(t)$, multiply your signal by the sine wave [$\sin(2\pi f_n t)$], take the mean value, and multiply it by 2. Repeat this process for the cosine wave and then add the squares of the two together, to get how much of the signal is composed of the frequency f_n. This is called the *power* at frequency f_n. Mathematically, the process is as follows:

$$a_0 = \text{mean}\,[w(t)] \quad (12.9)$$

$$b_n = 2 \times \text{mean}\,[w(t) \times \sin(2\pi f_n t)] \quad (12.10)$$

$$c_n = 2 \times \text{mean}\,[w(t) \times \cos(2\pi f_n t)] \quad (12.11)$$

$$\text{Power}(f_n) = b_n^2 + c_n^2 \quad (12.12)$$

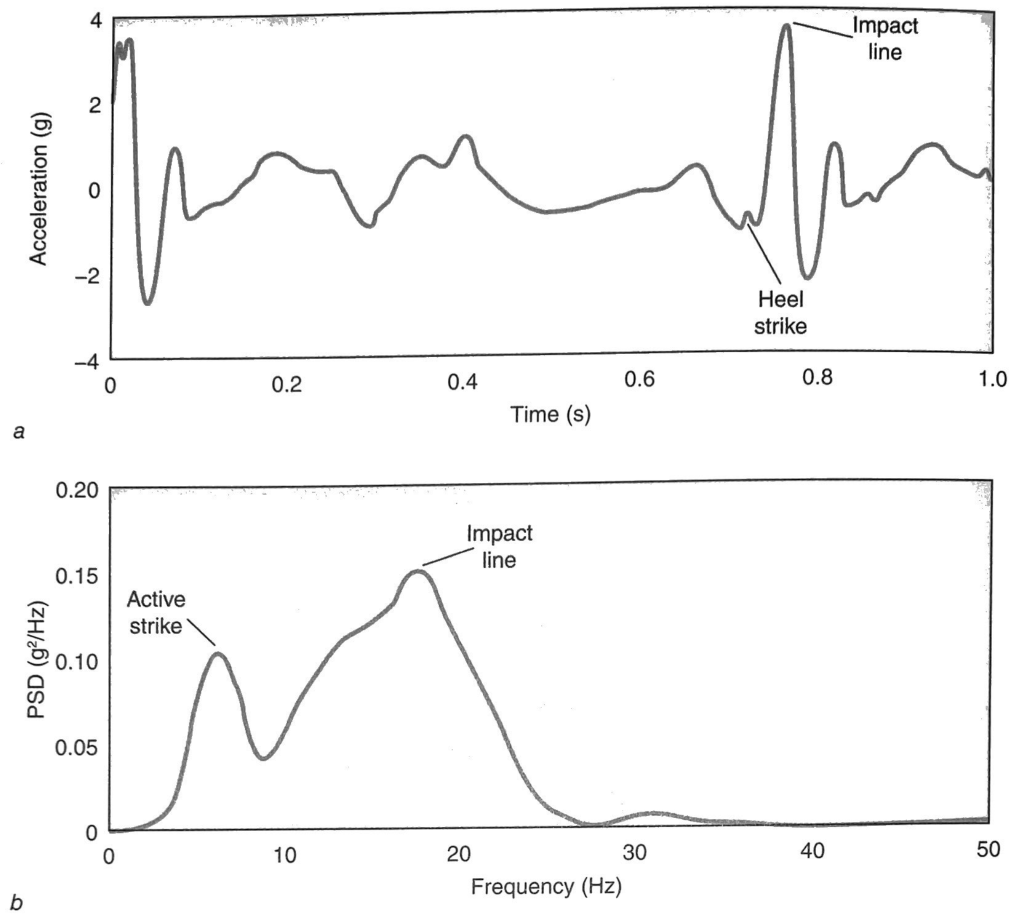
The Fourier coefficients can be calculated from the equally spaced time-varying points with the use of a discrete Fourier transformation (DFT) algorithm (appendix H). Given the Fourier coefficients, we can construct the original signal using an inverse DFT algorithm. The DFT is a calculation-intensive algorithm. Faster and more commonly used are the fast Fourier transformations (FFTs). An FFT requires that the number of original data points be a power of 2 (. . . , 64, 128, 256, 512, 1024, 2048, . . .). If this is not the case, the usual method of obtaining a "power of 2" number of samples is to pad the data with zeros (add zeros until the number of points is a power of two). This creates two problems:

- Padding reduces the power of the signal. Parseval's theorem implies that the power in the time domain must equal the power in the frequency domain (Proakis and Manolakis 1988). When you pad with zeros, you reduce the power (a straight line at zero has no power). You can restore the original power by multiplying the power at each frequency by $(N + L)/N$, where N is the number of nonzero values and L is the number of padded zeros.
- Padding can introduce discontinuities between the data and the padded zero values if the signal does not end at zero. This discontinuity shows up in the resulting spectrum as increased power in the higher frequencies. To ensure that your data start and end at zero, you can apply a windowing function or subtract the trend line before performing the transformation. Windowing functions begin at zero, rise to 1, and then return to zero again. By multiplying your signal by a windowing function, you reduce the endpoints to zero in a gradual manner. Windowing should not be performed on data unless there are multiple cycles. Subtracting a trend line that connects the first point to the last point can be used as an alternative.

Most software packages give the result of an FFT in terms of a real portion and an imaginary portion. For a real discrete signal, the real portion corresponds to the cosine coefficient and the imaginary portion corresponds to the sine coefficient of the Fourier series equation. An FFT results in as many coefficients as there are data points (N), but half of these coefficients are a reflection of the other half. Therefore, the $N/2$ points represent frequencies from zero to one-half of the sampling frequency ($f_s/2$). Each frequency bin has a width of f_s/N Hz. By increasing the number of data points (by padding with zeros or collecting for a longer period of time), you can decrease the bin width. This does not increase the resolution of the FFT; rather, it is analogous to interpolating more points from a curve.

Researchers often adjust the bin width so that each bin is 1 Hz wide. This is referred to as *normalizing the spectrum.* Adjusting the bin width changes the magnitude because the sum of the power frequency bins must equal the power in the time domain. Normalizing the spectrum allows data of different durations or sampling rates to be compared. The magnitude of a normalized spectrum is in units of (original units)^2/Hz.

A plot of the power at each frequency is referred to as the *power spectral density* (PSD) plot or simply the *power spectrum.* A PSD curve contains the same information as its time-domain counterpart, but it is rearranged to emphasize the frequencies that contain the greatest power rather than the point in time in the cycle at which the most power occurs. Figure 12.4 shows a leg acceleration curve along with its PSD.



▲ **Figure 12.4** Leg acceleration during running in the *(a)* time and *(b)* frequency domains. The time domain graph shows two ground impacts, whereas the frequency domain graph is for a single stance phase.

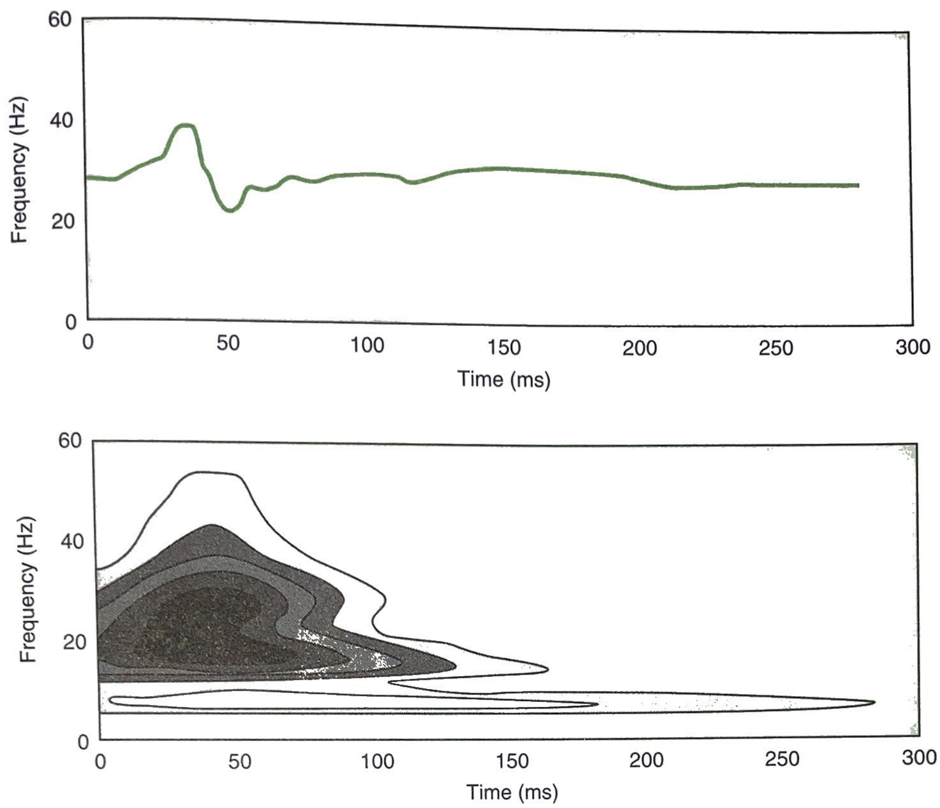
TIME-DEPENDENT FOURIER TRANSFORM

The discrete Fourier transform (DFT) has the advantage that frequencies can be separated no matter when they occur in the signal. Even frequencies that occur at the same time can be separated and quantified. A major disadvantage is that we do not know when those frequencies are present. We could overcome this difficulty by separating the signal into sections and applying the DFT to each section. We would then have a better idea of when a particular frequency occurred in the signal. This process is called a time-dependent Fourier transform.

Because we are already able to separate frequencies, we will use that technique to build an intuitive feeling for how this transform works. If we take a signal that contains frequencies from 0 to 100 Hz, the first step is to separate the frequencies into two portions, 50 Hz and below and 50 Hz and above. Next, we take these two sections and separate them into two portions each. We now have sections of 0 to 50, 50 to 100, 0 to 25, 25 to 50, 50 to 75, and 75 to 100 Hz. This procedure—called *decomposition*—continues to a predefined level. At this point, we have several time-series representations of the original signal, each containing different frequencies. We can plot these representations on a three-dimensional (3-D) graph with time on one axis, frequency on a second axis, and magnitude on the third. Figure 12.5 shows a two-dimensional version of this type of graph with the contour lines illustrating the magnitudes at each frequency and time.

SAMPLING THEOREM

The process signal must be sampled at a frequency greater than twice as high as the highest frequency present in the signal itself. This minimum sampling rate is called the *Nyquist sampling frequency* (f_N). In human locomotion, the highest voluntary frequency is less than 10 Hz, so a 20 Hz sampling rate should be satisfactory; however, in reality, biomechanists usually sample at 5 to 10 times the highest frequency in the signal. This ensures that the signal is accurately portrayed in the time domain without missing peak values.



▲**Figure 12.5** A 3-D contour map of the frequency by time values of a leg acceleration curve during running. The time domain curve is superimposed on the contour. There are two peaks in this curve. The high-frequency peak (approximately 20 Hz) occurs between 20 and 60 ms. The lower-frequency peak (approximately 8 Hz) occurs between 0 and 180 ms.

FROM THE SCIENTIFIC LITERATURE

Shorten, M.R., and D.S. Winslow. 1992. Spectral analysis of impact shock during running. *International Journal of Sport Biomechanics* 8:288-304.

The purpose of this study was to determine the effects of increasing impact shock levels on the spectral characteristics of impact shock and impact shock wave attenuation in the body during treadmill running. Three frequency ranges were identified in leg acceleration curves collected during the stance phase of running. The lowest frequencies (4-8 Hz) were identified as the active region as a result of muscular activity. The midrange frequencies (12-20 Hz) resulted from the impact between the foot and ground. There was also a high-frequency component (60-90 Hz) resulting from the resonance of the accelerometer attachment. Because these frequencies all occurred at the same time, it was impossible to separately analyze them in the time domain. Head accelerations were also calculated so that impact attenuation could be calculated from the transfer functions (TFs). TFs were calculated from the power spectral densities at the head (PSD_{head}) and the leg (PSD_{leg}) using the following formula:

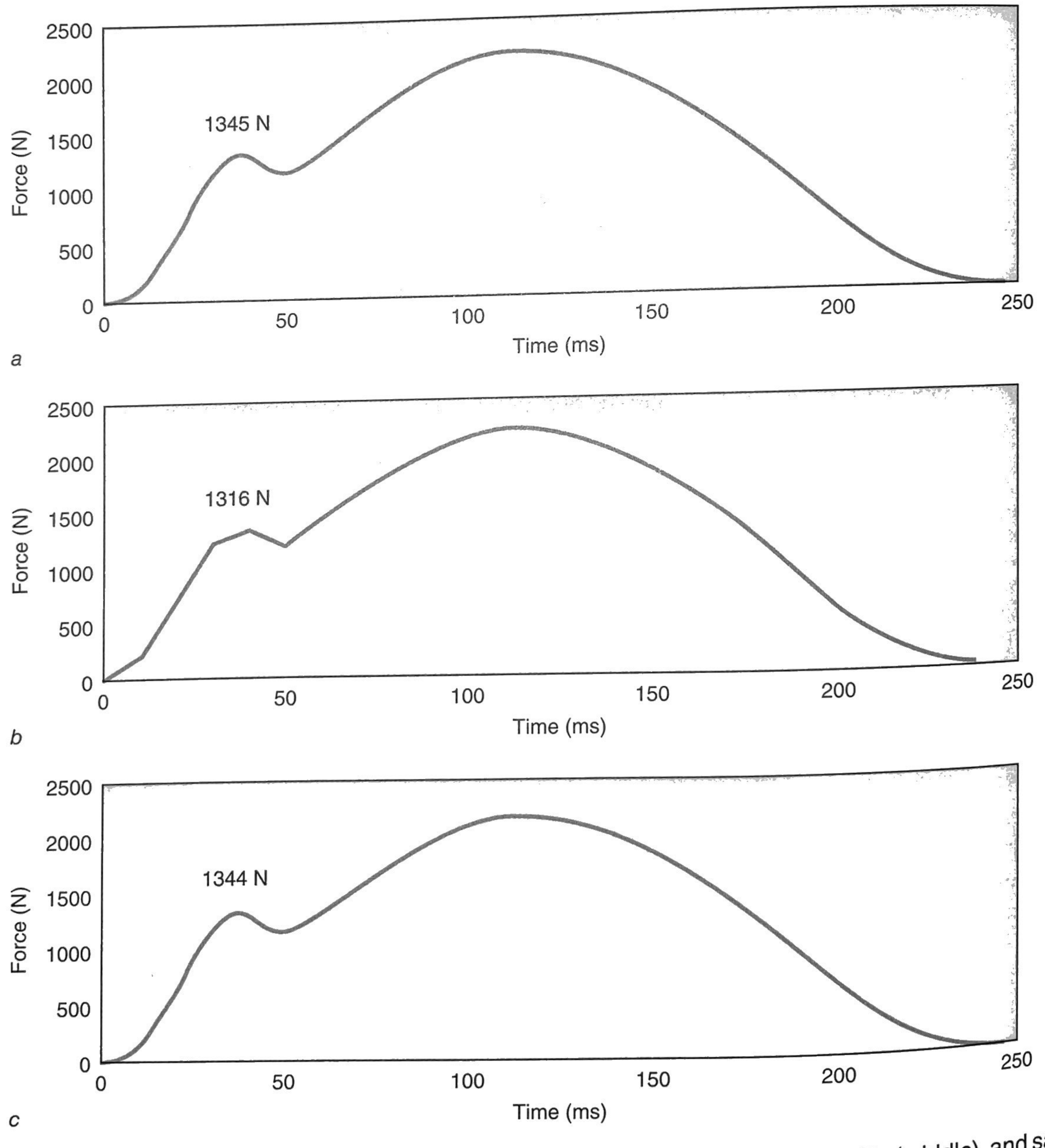
$$TF = 10 \log_{10} (PSD_{head}/PSD_{leg}) \qquad (12.13)$$

This formula resulted in positive values when there was a gain in the signal from the leg to the head and negative values when there was an attenuation of the signal from the leg to the head. The results indicated that the leg impact frequency increased as running speed increased. There was also an increase in the impact attenuation so that head impact frequencies remained relatively constant.

The sampling theorem holds that if the signal is sampled at greater than twice the highest frequency, then the signal is completely specified by the data. In fact, the original signal *(w)* is given explicitly by the following formula (also see appendix I):

$$w(t) = \Delta \sum w_n \left[\frac{\sin\left[2\pi f_c (t - n\Delta)\right]}{\pi(t - n\Delta)} \right] \quad (12.14)$$

where Δ is the sample period (1/sampling frequency), f_c is $1/(2\Delta)$, w_n is the *n*th sampled datum, and *t* is the time. By using this formula (Shannon's reconstruction formula; Hamill et al. 1997), we can collect data at slightly greater than twice the highest frequency and then apply the reconstruction formula to "resample" the data at a higher rate (Marks 1993). Figure 12.6 illustrates the use of the resampling formula to reconstruct a running vertical GRF curve. The signal was originally sampled at 1000 Hz, and the impact peak was measured at 1345 N. Every 20th point was then extracted to simulate data sampled at 50 Hz. The peak value occurred between samples, with the nearest data point at 1316 N. This also changed the time of occurrence of the impact peak. After the reconstruction formula was applied to the 50 Hz data, the peak was restored to 1344 N with the same time of occurrence as the originally sampled data. With modern computers, there is little reason to undersample



▲ **Figure 12.6** A running vertical GRF curve sampled at 1000 Hz (top), sampled at 50 Hz (middle), and sampled at 50 Hz but then reconstructed at 1000 Hz (bottom). The magnitude of the impact peak is identified in each graph. Reconstructing the signal results in a peak value very close to the original.

a signal unless the hardware is somehow limited, as is often the case when collecting kinematic data from video cameras with a sampling rate limited to 60 or 120 Hz. In many data collection systems it is necessary to collect **analog** data such as force or EMG data at sampling rates that are integral multiples of the sampling rate of the marker trajectories. For example, force and EMG may be collected at 1000 Hz when motion-capture data are collected at 200 frames per second (i.e., 5 times the video capture rate).

ENSURING CIRCULAR CONTINUITY

For the resampling formula to work correctly, the data must have circular continuity. To understand what circular continuity is, draw a curve from end to end on a piece of paper and then form a tube with the curve on the outside by rolling the paper across the curve. Circular continuity exists if there is no "discontinuity" where the start of the curve meets the end of the curve. This means that the first point on the curve must be equal to the last point. Nevertheless, the principle of circular continuity goes further: The slope of the curve at the start must equal the slope of the curve at the end. The slope of the slopes (the second derivative) must also be continuous. If you do not have circular continuity and you apply Shannon's reconstruction algorithm, you may be violating the assumption that only frequencies of less than half of the sampling frequency are present in the data. Discontinuities are by definition high-frequency changes in the data. If this occurs, you will see in the reconstructed data oscillations that have high amplitudes at the endpoints of the curve. These oscillations become smaller (damped) the farther you get from the endpoints, and they become much more evident if derivatives are calculated.

When data lack circular continuity, use the following steps (Derrick 1998) to approximate circular continuity (see figure 12.7):

1. Split the data into two halves.
2. Copy the first half of the data in reverse order and invert them. Attach this segment to the front of the original data.
3. Copy the second half of the data in reverse order and invert them. Attach this segment to the back of the original data.
4. Subtract the trend line from the first data point to the last data point.

Reversal of the first or second half of data is a procedure by which the first data point becomes the last data point of the segment, the second data point becomes the second to last, and so on. Inversion is a procedure that flips the magnitudes about a pivot point. The pivot point is the point closest in proximity to the original data. Figure 12.7 shows a schematic diagram of the data after we sum the front and back segments and before we subtract the trend line.

Step 2 ensures circular continuity at the start of the original data set. Step 3 ensures circular continuity at the end of the original data set. Steps 2 and 3 together ensure that the slopes at the start and end of the new data set are continuous, but it is still possible to have a gap between the magnitude of the first point and the magnitude of the last point of the new data set. Step 4 removes this gap by calculating the difference between the trend line and each data point. Thus, the first and last points will be equal to zero.

If you fail to heed the sampling theorem, not only will you lose the higher frequencies, but the frequencies above the $1/2f_N$ (half the Nyquist frequency) actually fold back into the spectrum. In the time domain, this is referred to as *aliasing*. An antialiasing low-pass filter with a cutoff frequency (defined in the section titled Digital Filtering) greater than $1/2f_N$ applied to a signal before processing ensures no aliasing.

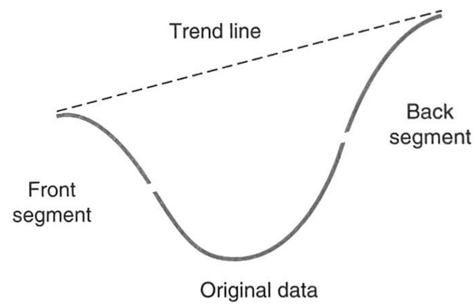
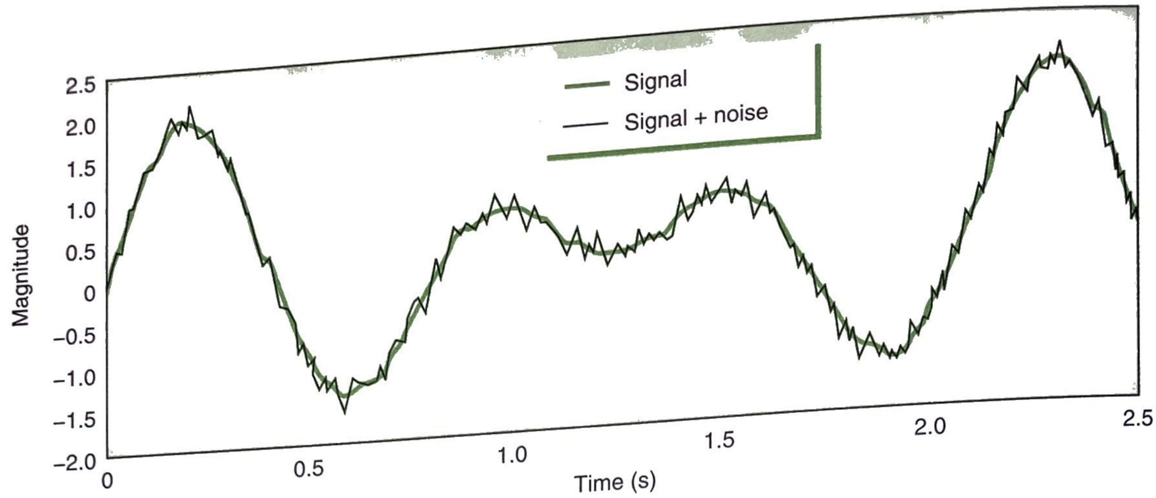


Figure 12.7 Schematic diagram of the procedure used to ensure that a signal has the property of circular continuity.

SMOOTHING DATA

Errors associated with the measurement of a biological signal may be the result of skin movement, incorrect digitization, electrical interference, artifacts from moving wires, or other factors. These errors, or "noise," often have characteristics that are different from the signal. **Noise** is any unwanted portion of a waveform. It is typically nondeterministic, lower in amplitude, and often in a frequency range different from that of the signal. For instance, errors associated with the digitizing process are generally higher in frequency than human movement. Noise that has a frequency different from those in the signal can be removed. If you were to plot the signal and the signal plus noise, it would look like figure 12.8. The



▲**Figure 12.8** A biological signal with and without noise.

goal of smoothing is to eliminate the noise but leave the signal unaffected.

There are many techniques for smoothing data to remove the influence of noise. Outlined next are a number of the most popular. Each has its own strengths and weaknesses, and none is best for every situation. Researchers must be aware of how each method affects both the signal and the noise components of a waveform. Ideally, the signal would be unaffected by the smoothing process used to remove the noise, but most smoothing techniques affect the signal component to some extent.

Polynomial Smoothing

Any n data points can be fitted with a polynomial of degree $n - 1$ of the following form:

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \ldots + a_{n-1}t^{n-1} \quad (12.15)$$

This polynomial will go through each of the n data points, so no smoothing has been accomplished. Smoothing occurs by eliminating the higher-order terms. This restricts the polynomial to lower-frequency changes and thus it will not be able to go through all of the data points. Most human movements can be described by polynomials of the ninth order or less. Polynomials produce a single set of coefficients that represent the entire data set, resulting in large savings in computer storage space. The polynomial also has the advantages of allowing you to interpolate points at different time intervals and of making the calculation of derivatives relatively easy. Unfortunately, they can distort a signal's true shape; see the article by Pezzack and colleagues (1977)—reviewed in chapter 1—for an example of this technique. In practice, avoid using polynomial fitting unless the signal is a known polynomial. For example, fitting a second-order (parabolic) polynomial to the vertical motion of the center of gravity of an airborne body is appropriate. The path of the center of gravity during walking follows no known polynomial function, however.

Splines

A spline function (deBoor 2001) consists of a number of low-order polynomials that are pieced together in such a way that they form a smooth line. Cubic (third-order) and quintic (fifth-order) splines are the most popular for biomechanics applications (Wood 1982, Vaughan 1982). Splines are particularly useful if there are missing data in the data stream that need interpolation. In many motion capture systems, the "gap filling" procedures are done by fitting splines across the gaps in trajectories when two or more cameras were unable to view a marker and therefore the 3-D trajectory could not be reconstructed. Many techniques, such as **digital** filtering (discussed subsequently), require equally time-spaced data. Splines do not have this requirement.

Fourier Smoothing

Fourier smoothing consists of transforming the data into the frequency domain, eliminating the unwanted frequency coefficients, and then performing an inverse transformation to reconstruct the original data without the noise. Hatze (1981) outlined how to apply this method for smoothing displacement data.

Moving Average

A three-point moving average is accomplished by replacing each data point (n) by the average of $n - 1$, n, and $n + 1$. A five-point moving average uses the data points $n - 2$, $n - 1$, n, $n + 1$, and $n + 2$ and results in more smoothing than does a three-point moving average. Note that there will be undefined values at the start and end of the series. This method is extremely easy to implement but is incapable of distinguishing signals from noise. It will **attenuate** valid signal components and may not affect invalid noise components. A better choice is the digital filter.

Digital Filtering

A digital filter is a type of weighted moving average. The points that are averaged are weighted by coefficients in such a manner that a cutoff frequency can be determined. The **cutoff frequency** is the frequent at which a filter reduces the frequency's power by one half (half-power point) or equivalently reduces the amplitude to $\sqrt{2}/2$ or by −3 **decibels** or to approximately 0.707 of its original amplitude. In the case of a low-pass filter, frequencies below the cutoff are attenuated whereas frequencies above the cutoff are unaffected. The type of digital filter is determined by the frequencies that are passed through without attenuation. The following digital filters are all implemented in the same manner, but the coefficients are adjusted for a particular cutoff frequency. Signals that are band-passed or band-reject filtered are run through the filter with both low-pass and high-pass cutoff frequencies (figure 12.9).

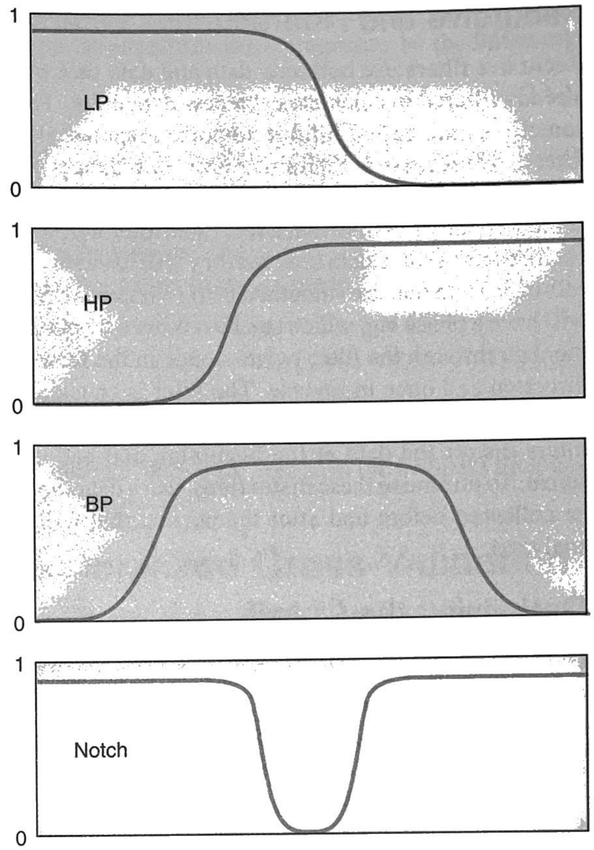


Figure 12.9 Frequency responses of different types of digital filters. The digital filter is implemented in the time domain, but it can be visualized in the frequency domain. The frequency response function is multiplied by the signal in the frequency domain and then transformed back into the time domain. LP = low-pass; HP = high-pass; BP = band-pass.

Type of Filter

Filters can be constructed to attenuate different parts of the frequency spectrum. One or sometimes two cutoff frequencies are necessary to define which part of the frequency spectrum is attenuated and which part is left "as is," or passed unattenuated.

- Low-pass: The cutoff is selected so that low frequencies are unchanged but higher frequencies are attenuated. This is the most common filter type. It is often used to remove high frequencies from digitized kinematic data and as a digital antialiasing filter.
- High-pass: The cutoff is selected so that high frequencies are unchanged but lower frequencies are attenuated. It is used as a component in band-pass and band-reject filters or to remove low-frequency movement artifacts from low-voltage signals in wires that are attached to the body (e.g., electromyographic [EMG] signals).
- Band-pass: The frequencies between two cutoff frequencies are passed unattenuated. Frequencies below the lower cutoff and frequencies above the higher cutoff are attenuated. Such a filter is often used in EMG when there is movement artifact in the low-frequency range and noise in the high-frequency range.
- Band-stop or band-reject: The frequencies between the two cutoff frequencies are attenuated. The frequencies below the lower cutoff and above the higher cutoff are passed unattenuated. This filter has little use in biomechanics.
- Notch: A narrow band or single frequency is attenuated. It is used to remove power-line noise (60 or 50 Hz) or other specific frequencies from a signal. This filter generally is not recommended for EMG signals, because they have significant power at 50 and 60 Hz (for additional information, see chapter 8).

Filter Roll-Off

Roll-off is the rate of attenuation above the cutoff frequency. The higher the order (the more coefficients) or the greater the number of times the signal is passed through the filter, the sharper the roll-off.

Recursive and Nonrecursive Filters

Recursive filters use both raw data and data that were already filtered to calculate each new data point. They sometimes are called infinite impulse response (IIR) filters. Nonrecursive filters use only raw data points and are called finite impulse response (FIR) filters. It is theoretically possible that a recursive filter will show oscillations in some data sets, but they will have sharper roll-offs. Data that are smoothed using a recursive filter will have a phase lag, which can be removed by putting the data through the filter twice—once in the forward direction and once in reverse. The filter is considered a *zero lag* filter if the net phase shift is zero. Digital filters distort the data at the beginning and end of a signal. To minimize these **distortions**, extra data should be collected before and after the portion that will be analyzed.

Optimizing the Cutoff

The selection of a cutoff frequency is very important when filtering data. This is a somewhat subjective determination based on your knowledge of the signal and the noise. A number of algorithms are used in an attempt to find a more objective criterion for determining the cutoff frequency (Jackson 1979). These optimizing algorithms typically are based on an analysis of the residuals, which are what is left over when you subtract the filtered data from the raw data. As long as only noise is being filtered, some of these values should be greater than zero and some less than zero. The sum of all of the residuals should equal zero (or at least be close). When the filter starts affecting the signal, the sum of the residuals will no longer equal zero. Some optimization routines use this fact to determine which frequency best distinguishes the signal from the noise (figure 12.10). These algorithms are not completely objective, however, because you still must determine how close to zero the sum of the residuals is before selecting the optimal cutoff frequency.

Steps for Designing a Digital Filter

The following steps create a Butterworth low-pass, recursive digital filter. Modifications for a critically-damped and a high-pass filter are also discussed. Butterworth filters are said to be optimally flat in the pass band, making them highly desirable for biomechanical variables. This means that the amplitudes of the frequencies that are to be passed are relatively unaffected by the filter. Some filters, such as the Chebyshev or elliptic, have better roll-offs than the Butterworth filter, but they distort the amplitudes of the frequencies in the pass band.

1. Convert the cutoff frequency (f_c) from Hz to rad/s.

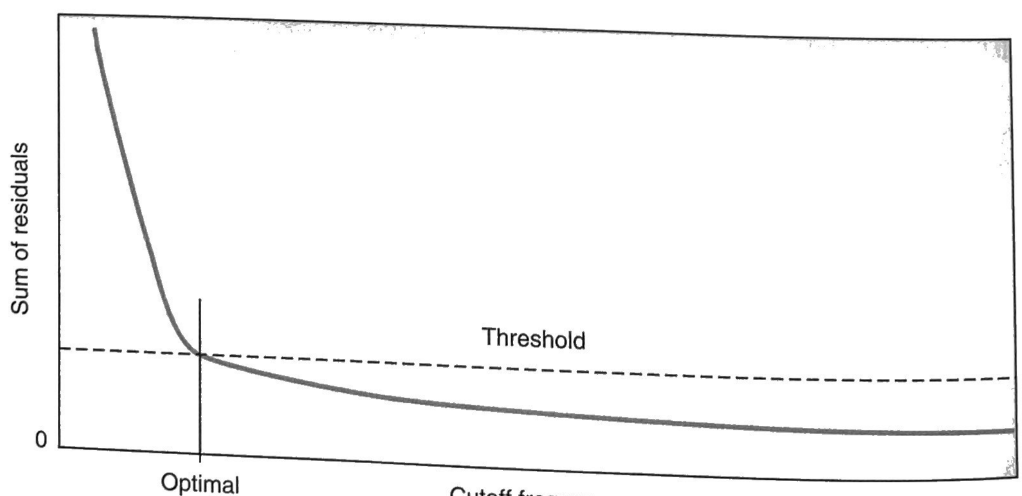
$$\omega_c = 2\pi f_c \tag{12.16}$$

2. Adjust the cutoff frequency to reduce "warping" that resulted from the bilinear transformation.

$$\Omega_c = \tan\left[\frac{\omega_c}{2\times\text{Sample rate}}\right] \tag{12.17}$$

3. Adjust the cutoff frequency for the number of passes (*P*). A pass occurs each time the data are passed through the filter. For every pass through the data, a second pass must be made in the reverse direction to correct for the phase shift. Increasing the number of passes increases the sharpness of the roll-off.

$$\Omega_N = \frac{\Omega_c}{\sqrt[4]{2^{\left(\frac{1}{P}\right)}-1}} \tag{12.18}$$



▲ **Figure 12.10** Selection of an optimal cutoff frequency using residual analysis.

4. Calculate the coefficients.

$$C_1 = \frac{\Omega_N^2}{\left(1+\sqrt{2}\,\Omega_N + \Omega_N^2\right)}$$

$$C_2 = \frac{2\Omega_N^2}{\left(1+\sqrt{2}\,\Omega_N + \Omega_N^2\right)} = 2C_1$$

$$C_3 = \frac{\Omega_N^2}{\left(1+\sqrt{2}\,\Omega_N + \Omega_N^2\right)} = C_1$$

$$C_4 = \frac{2\left(1-\Omega_N\right)^2}{\left(1+\sqrt{2}\,\Omega_N + \Omega_N^2\right)}$$

$$C_5 = \frac{\left(2\Omega_N - \Omega_N^2 - 1\right)}{\left(1+\sqrt{2}\,\Omega_N + \Omega_N^2\right)}$$

$$= 1-\left(C_1 + C_2 + C_3 + C_4\right) \tag{12.19}$$

5. Apply the coefficients to the data to implement the weighted moving average. Y_n values are filtered data and X_n values are unfiltered data. The filter is recursive because previously filtered data—Y_{n-1} and Y_{n-2}—are used to calculate the filtered data point, Y_n.

$$Y_n = C_1 X_{n-2} + C_2 X_{n-1} + C_3 X_n + C_4 Y_{n-1} + C_5 Y_{n-2} \tag{12.20}$$

This filter is underdamped (damping ratio = 0.707) and will therefore "overshoot" and "undershoot" the true signal whenever there is a rapid change in the data stream. For more information, consult the article by Robertson and Dowling (2003). A critically damped filter can be designed (damping ratio = 1) by changing to 2 in each equation in step 4. The warping function must also be altered as follows:

$$\Omega_N = \frac{\Omega_c}{\sqrt{2^{\left(\frac{1}{2P}\right)} - 1}} \tag{12.21}$$

In practice, there is little difference between the underdamped and critically damped filter. The distinction can be seen in response to a step input (a function that transitions from 0 to 1 in a single step). The Butterworth filter will produce an artificial minimum before the step and an artificial maximum after the step (Robertson and Dowling 2003). The Butterworth filter, however, has a better roll-off and therefore is better for filtering data that will be double differentiated, such as marker trajectories. In contrast, the critically damped filter is better for filtering rectified EMG data that will be used to determine onset times because this filter responds more rapidly and in the correct direction to quickly recruited muscles.

Furthermore, it is possible to calculate the coefficients so that the filter becomes a high-pass filter instead of a low-pass filter (Murphy and Robertson 1994). The first step is to adjust the cutoff frequency by the following:

$$f_c = \frac{f_s}{2} - f_{c\text{-}old} \tag{12.22}$$

where f_c is the new cutoff frequency, $f_{c\text{-}old}$ is the old cutoff frequency, and f_s is the sampling frequency. The coefficients (C_1 through C_5) are then calculated in the same way that they were for the low-pass filter and then the following adjustments are made:

$$c_1 = C_1,\ c_2 = -C_2,\ c_3 = C_3,$$
$$c_4 = -C_4, \text{ and } c_5 = C_5 \tag{12.23}$$

where c_1 through c_5 are the coefficients for the high-pass filter. The data can now be passed through the filter, forward and backward, just as in the low-pass filter case outlined previously.

Generalized Cross-Validation

Another commonly used data-smoothing technique designed by Herman Woltring (see From the Scientific Literature) called *generalized cross-validation* (GCV), occasionally described as the Woltring filter, behaves similarly to a bidirectional (zero-lag) Butterworth low-pass filter. One difference is that instead of specifying a cutoff frequency, the user may allow the software to determine its own characteristics or the user may enter a value corresponding to the mean square error (MSE) of the motion-capture system. This MSE is equivalent to the residual error computed during the calibration process of the motion capture system. In many gait laboratories this value could be 1 mm or less. Using too high a value results in oversmoothing the trajectories. This technique also has the advantage that it can fill gaps in the data stream because it does not require equidistantly sampled data (i.e., constant sampling rate).

SUMMARY

This chapter outlined the basic principles and rules for characterizing and processing signals acquired from any data collection system. Special emphasis was given to the frequency or Fourier analysis of signals and data smoothing. Data-smoothing techniques are of particular interest to biomechanists because of the frequent need to perform double time differentiation of movement patterns to obtain accelerations. As illustrated in chapter 1, small errors in digitizing create high-frequency noise that, after double differentiating, dominates true data history. Removing high-frequency noise prior to differentiation prevents this problem. The biomechanist, however, should be aware of how these smoothing processes affect data so that an appropriate method can be applied without distorting the true signal.

FROM THE SCIENTIFIC LITERATURE

Woltring, H.J. 1986. A FORTRAN package for generalized, cross-validatory spline smoothing and differentiation. *Advances in Engineering Software* 8(2):104-13.

The purpose of this technical report was to describe a software package for smoothing noisy data such as encountered by motion-capture systems. The report includes software written in FORTRAN that when given a set of data and various options computes natural B-spline functions that remove the high-frequency noise contained in the data stream. Dr. Woltring provided several means of creating the suitable smoothing function based on either a priori knowledge of the noise level of the data-capture system, the effective number of degrees of freedom in the smoothing function, or a generalized cross-validation or mean-squared prediction error criteria as described by Craven and Wahba (1979). His software assumes that the noise is uncorrelated (i.e., random) and additive and that the underlying signal is smoothly varying (i.e., has no rapid transients). Furthermore, the data stream does not need to be equidistant in time as is the case for digital filtering software. Although difficult to implement, the software is unique in that it does not require different cutoff frequencies for each marker trajectory. Several commercial software manufacturers (e.g., C-Motion, Vicon) have therefore included it as an alternative to Butterworth low-pass filters.

SUGGESTED READINGS

Antoniou, A. 2005. *Digital Signal Processing. Systems, Signals, and Filters*. Toronto: McGraw Hill Professional.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. 1989. *Numerical Recipes in Pascal*, 422-97. New York: Cambridge University Press.

Smith, S.W. 1997. *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, CA: California Technical Publishing.

Transnational College of LEX. 2012. *Who Is Fourier? A Mathematical Adventure*. 2nd ed. Trans. by Alan Gleason. Belmont, MA: Language Research Foundation.