

# Signal Processing

**THE OBJECTIVE OF THIS CHAPTER IS TO INTRODUCE SIGNAL PROCESSING. YOU WILL LEARN ABOUT TWO METHODS, OF MANY, TO EXAMINE LARGE DATA SETS. THEY ARE:**

- Harmonic analysis
- Fast Fourier transform analysis

## CONTENTS

Harmonic analysis.....	330
Fast Fourier transform (FFT).....	334

In this chapter we apply MATLAB to examine periodic data sets. Harmonic analysis is introduced in the next section to examine periodic data sets. In the subsequent section the fast Fourier transform (FFT) method is applied to examine the temporal-evolution of measured or computed data. These procedures are important methods in the field of signal processing.

A *signal* is a sequence of data that is measured by experimental methods or computed by computational methods. For example, the data  $f(x)$  is measured at relatively small intervals of a particular variable  $x$ , e.g.,  $x$  could be time or a space dimension. If the signal is periodic in a range of  $x$ , e.g.,  $-\pi$  to  $\pi$ , harmonic analysis is a useful and quite practical technique. The FFT method is a related method. Both methods are associated with the fitting of data with a Fourier series. In addition, both methods are widely used in engineering, e.g., in naval architecture, acoustics (or noise) and vibration engineering, electrical engineering, ocean engineering, and so on.

The theory of *Fourier series* is an important topic in mathematics for physicists and engineers. A reasonably thorough discussion of this topic can be found in *Mathematics of Physics and Modern Engineering*, Second Edition, by I.S. Sokolnikoff and R.M. Redheffer, a book published by McGraw-Hill (1966). Because of the importance of this topic, it should not be surprising that there are many other books that describe the theory behind the method and the practical applications

of the method. In this introductory section we give the results associated with applying the Fourier series method that are needed to examine signals.

Let us suppose  $f(x)$  is given by the *trigonometric series*:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx), \quad (15.1)$$

where  $-\pi \leq x \leq \pi$  and, hence,

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx \, dx \quad (15.2)$$

for  $n = 0, 1, 2, \dots$ , and:

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, dx \quad (15.3)$$

for  $n = 1, 2, 3, \dots$ . This assumes that the function  $f(x)$  is periodic, i.e.,  $f(x) = f(x + 2\pi)$ .

If we are given a data set  $f(x)$  and it is assumed to be periodic in the interval  $-\pi \leq x \leq \pi$ , we can compute the coefficients  $a_n$  and  $b_n$  by applying the above formulas. With this approach we say that we expanded  $f(x)$  in or say that  $f(x)$  has the *Fourier series*:

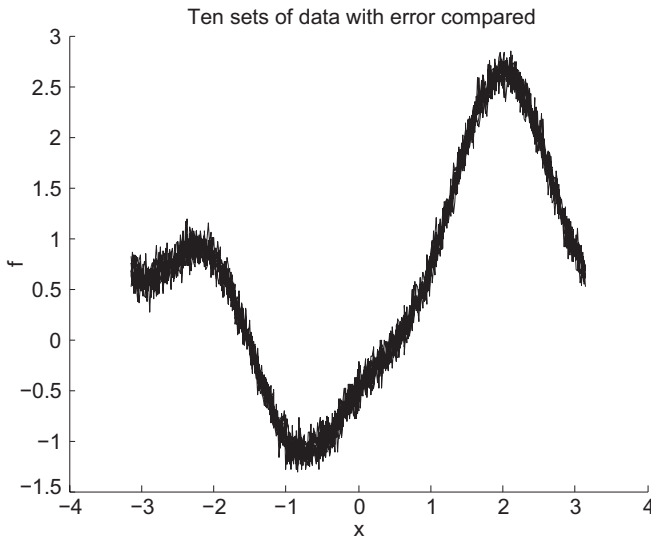
$$f(x) \sim \frac{1}{2}A_0 + \sum_{n=1}^N (A_n \cos nx + B_n \sin nx), \quad (15.4)$$

where  $N$  is a positive integer greater than or equal to 1. It is the number of terms to be evaluated to construct a Fourier series (or Fourier expansion) of  $f(x)$ . If  $N \rightarrow \infty$  and  $f(x)$  is continuous and periodic, the last equation is essentially the trigonometric expansion of  $f(x)$  and the result is exact. An important advantage of this series is that it can represent discontinuous functions as well. Of course, in this case or in cases where the trigonometric representation is not exact, it is a best fit representation of  $f(x)$  in terms of the assumed periodic function, Equation (15.4).

## 15.1 HARMONIC ANALYSIS

Let us examine the data illustrated in Figure 15.1 by fitting a trigonometric series through the data. This sample set of “observed” data was generated by using the following formula:

$$f = .6 - \cos x - .5 \cos 2x + .4 \cos 3x + \sin x + \text{error}, \quad (15.5)$$



**FIGURE 15.1** Sample of 10 repeated periodic data sets with error.

where the error was generated by the random number generator `randn`, a built-in function in MATLAB. Equation (15.5) was selected for this example because it is a five-term trigonometric series plus an error term. It was selected so that we can examine how well harmonic analysis plucks out the harmonics (i.e., the first five terms) from a signal that contains error (the error in some applications of signal processing is also called “noise”).

We want to fit a finite trigonometric sum to a set of observed data. In this case the discrete set of observed data contains the values of  $f(x)$  at  $x = -\pi, -\pi + \pi/N, -\pi + 2\pi/N, \dots, \pi - \pi/N$ . It is assumed that  $f(-\pi) = f(\pi)$ . Note that  $-\pi \leq x < \pi$  is the domain of  $x$  that the data are observed. The values of  $f(x)$  are evaluated (or observed) at equally spaced intervals of  $dx = \pi/N$ . The example numerical analysis is for  $N = 200$  where  $N_p = 2N + 1$  is the number of equally spaced values of  $f$  from 0 to  $\pi$  where the data are measured. The data obtained at these points were computed by applying Equation (15.5). Since  $f$  is prescribed on the interval  $-\pi \leq x \leq \pi$  and it is assumed that  $f(-\pi) = f(\pi)$ , harmonic analysis is a good way to examine the data to determine its *harmonic content*, i.e., to determine the coefficients of the Fourier series that we are applying to expand the data. Note that the noise in the data introduced by the random number generator introduces an error in the periodicity assumption. Hence, in the example below 10 sets of the observed data are examined by repeating the computational experiment 10 times. The results illustrate that the error (or noise), in this case, is random and can be averaged away if a sufficient number of the same experiment is performed.

It turns out that if you have  $N_p$  equally spaced data points we can fit the data with  $N_p - 1$  terms of a Fourier series. The results are exact in the sense that we have a system of linear  $N_p - 1$  equations in  $N_p - 1$  unknowns. It is reasonably well known (see, e.g., the book by Sokolnikoff and Redheffer cited above) that the solution to this problem is:

$$f = A_0 + \sum_{n=1}^N A_n \cos nx + \sum_{n=1}^{N-1} B_n \sin nx. \quad (15.6)$$

(Note that  $A_0$  is twice the value of  $a_0$  in the Fourier series given in the introduction to this chapter.) In this case, it can be shown that:

$$A_0 = \frac{1}{2N} \sum_{m=1}^{2N} f \left( -\pi + \frac{(m-1)\pi}{N} \right), \quad (15.7)$$

$$A_N = \frac{1}{2N} \sum_{m=1}^{2N} f \left( -\pi + \frac{(m-1)\pi}{N} \right) \cos((m-1)\pi), \quad (15.8)$$

$$A_n = \frac{1}{N} \sum_{m=1}^{2N} f \left( -\pi + \frac{(m-1)\pi}{N} \right) \cos \frac{(m-1)n\pi}{N}, \quad n = 1, 2, 3, \dots, N-1, \quad (15.9)$$

$$B_n = \frac{1}{N} \sum_{m=1}^{2N} f \left( -\pi + \frac{(m-1)\pi}{N} \right) \sin \frac{(m-1)n\pi}{N}, \quad n = 1, 2, 3, \dots, N-1. \quad (15.10)$$

This system of equations was applied to examine the data generated by Equation (15.5). The MATLAB script developed and applied is as follows:

```
% Harmonic analysis
clear;clc
for ir = 1:10
N = 200; % Nc = N+1 implies x = 0 (center of domain).
% Np = 2N+1 are the number of points on the
% closed interval x = [-pi, pi].
dx = pi/N; % Spacing between the points on x.
x = -pi:dx:pi;
f = [.6 - cos(x) - .5*cos(2*x) + .4*cos(3*x) ...
+ sin(x) + 0.1*randn(1,length(x))];
A0 = (1/2/N) * sum(f(1:end-1));
A(N) = (1/2/N) * sum(f(1:end-1).*cos(N*x(1:end-1)));
for jn = 1:N-1
A(jn) = (1/N)*sum(f(1:end-1).*cos(jn*x(1:end-1)));
B(jn) = (1/N)*sum(f(1:end-1).*sin(jn*x(1:end-1)));
end
```

```

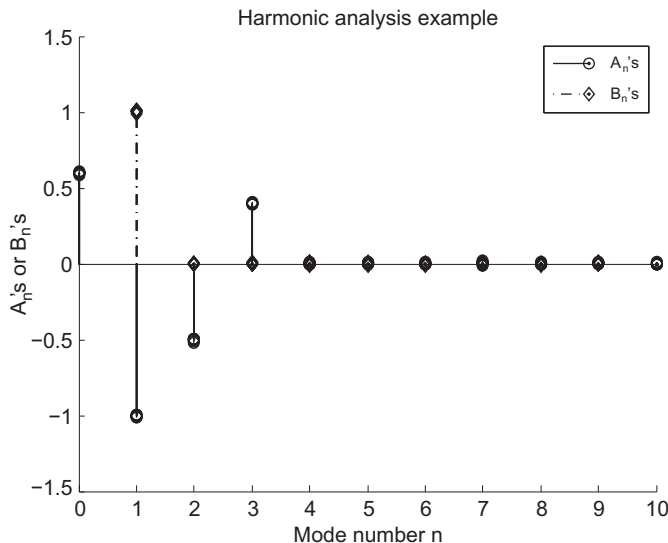
AA = [AO A];
figure(1)
hold on
stem([0:length(AA(1:11))-1],AA(1:11),'ko','LineWidth',2)
stem(B(1:9),'-kd','LineWidth',2)
legend('A_n's', 'B_n's')
xlabel('Mode number n')
ylabel('A_n's or B_n's')
title('Harmonic analysis example')
hold off
figure(2)
hold on
plot(x,f,'k')
title('Ten sets of data with error compared')
xlabel('x'),ylabel('f')
hold off
end

```

Note that the algorithm was executed 10 times to examine the influence of the random error on a converged solution. The random error is plus-or-minus 10% of unity. The 10 samples of the periodic data examined in the script are illustrated in Figure 15.1. The 10 associated harmonic-analysis results are given in Figure 15.2. As we might have expected, the results, on average, are pretty close to the exact results described next.

Since the function  $f(x)$  without error is:

$$f = .6 - \cos x - .5 \cos 2x + .4 \cos 3x + \sin x,$$



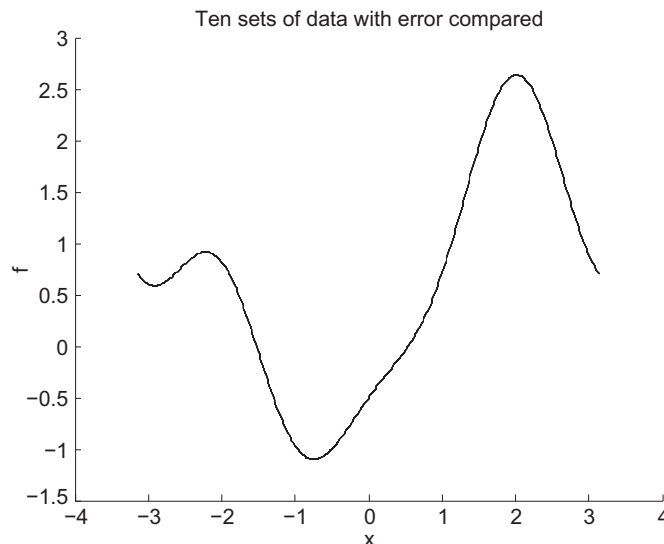
**FIGURE 15.2** Harmonic analysis of the 10 periodic data sets,  $f$ , compared.

it is a five-term trigonometric series. The harmonic analysis of this function is exact. This function and the exact results of the harmonic analysis of it are illustrated in Figures 15.3 and 15.4, respectively. The figures were generated with the same MATLAB script given above with the following changes. The coefficient of the error term was set to zero as follows: We changed `0.1*randn(1,length(x))`; to `0*randn(1,length(x))`; . Since the 10 repeated trials are identical, if you change the repeated-trials for loop, for `ir = 1:10` to `for ir = 1:1`, the results you obtain will be identical to the results in the figures for the zero error case.

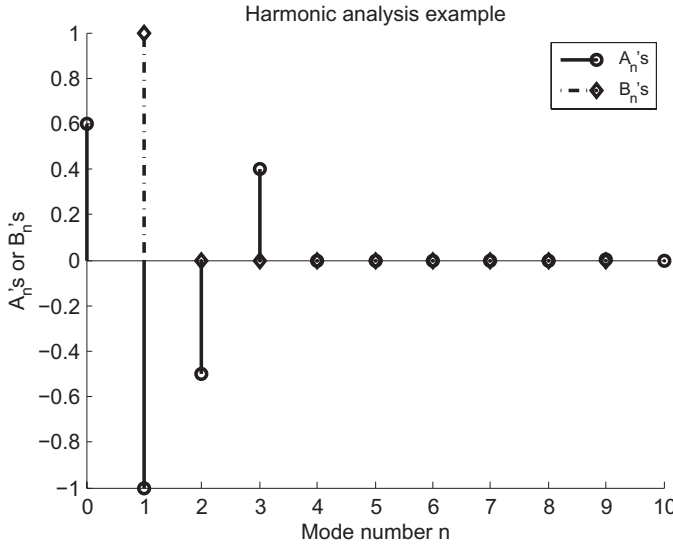
Comparing the error-less data with the noisy data generated by adding a random error, illustrates the need for making repeated measurements in the laboratory to get a sensible result for the amplitudes of the harmonics that represent the shape of the phenomenon sought, viz.,  $f(x)$  in the interval  $-\pi \leq x \leq \pi$ . Note that if a trigonometric series is used to define the input data set, the output of harmonic analysis yields an exact reproduction of the input series.

## 15.2 FAST FOURIER TRANSFORM (FFT)

James W. Cooley and John W. Tukey published *An Algorithm for the Machine Calculation of Complex Fourier Series* in 1965; it appeared in *Mathematics of Computations* 19, 297–301. This paper was cited in the book edited by Leo L. Beranek and István L. Vér titled *Noise and Vibration Control Engineering* published by John Wiley & Sons (1992). This book provides a good review of



**FIGURE 15.3** Sample of the periodic data set,  $f(x)$ , without error.



**FIGURE 15.4** Harmonic analysis of the periodic data,  $f(x)$ , without error.

signal processing as well as its practical applications in noise and vibration engineering. The discussion to follow follows closely the discussion in this book.

The Fourier transform of a time history signal,  $y(t)$ , that is measured over the time interval  $0 \leq t \leq T$  is defined for all frequencies, both positive and negative, by:

$$Y(\omega, T) = \int_0^T y(t) e^{-i2\pi\omega t} dt, \quad (15.11)$$

where  $Y$  is a function of frequency,  $\omega$ . In terms of a digital time series of  $N$  data values where  $y(t) = y(n\Delta t)$ ,  $n = 0, 1, 2, \dots, N-1$ . The Fourier transform may be written as a complex Fourier series:

$$Y(\omega, T) = Y(k\Delta\omega, N) = \Delta t \sum_{n=0}^{N-1} y(n\Delta t) \exp(-i2\pi\omega n\Delta t), \quad (15.12)$$

where the spectral components, i.e., the  $Y(k\Delta\omega, N)$ s, are generally complex valued and are defined only at  $N$  discrete frequencies:

$$\omega_k = k\Delta\omega = \frac{k}{N\Delta t}, \quad k = 0, 1, 2, \dots, N-1.$$

Hence, this is essentially the complex version of the Fourier series expansion introduced in the introduction of this chapter. To clarify this point further, let us divide Equation (15.12) by  $T = N\Delta t$ . This yields the following result:

$$C_k = \frac{Y(k\Delta\omega, N)}{N\Delta t} = \frac{1}{N} \sum_{n=0}^{N-1} x(n\Delta t) \exp(-i2\pi\omega n\Delta t) = \frac{1}{N} \sum_{n=0}^{N-1} x(n\Delta t) \exp(-i2\pi kn/N). \quad (15.13)$$

Note the analogy between this formula and the formulas for the Fourier coefficients given in the previous section on harmonic analysis. In this case, however, the coefficients are complex numbers. Since MATLAB deals with complex numbers and does complex arithmetic, applying the FFT methodology to examine signals is reasonably straight forward. This will be illustrated below by an example.

The complex Fourier coefficients defined by Equation (15.13) are for an assumed periodic function and, hence, the time history  $y(t)$  is assumed to repeat over periods equal to the sampling period,  $T$ . The Fourier components are unique only out to  $k = N/2$ , i.e., out to the frequency  $\omega_k = 1/(2\Delta t)$ , commonly known as the Nyquist frequency,  $\omega_N$ , of the digital signal. At this frequency there are only two sample values per cycle and, hence, the error known as *aliasing* is initiated; for details see, e.g., Beranek and Vér (1992) cited above. The first  $N/2 + 1$  Fourier coefficients, from  $k = 0$  to  $k = N/2$ , define the spectral components (i.e., magnitudes of the Fourier coefficients of significant magnitude) at positive frequencies, while the last  $N/2 - 1$  Fourier coefficients, from  $k = N/2 + 1$  to  $k = N - 1$ , define the spectral components at negative frequencies.

The various FFT algorithms developed since the publication of *Cooley-Tukey* algorithm are well documented in the technical literature. The details of developing an FFT algorithm is also beyond the scope of this presentation. However, since the procedure applied in the example examined in this section can be applied to other problems that may be of interest to the reader, a few of the hints that need to be considered in applying the Cooley-Tukey algorithm pointed out by Beranek and Vér are worth noting since this algorithm is the basis for the FFT algorithm applied in the example discussed in this section. They are:

- It is useful to restrict the number of equally spaced data points for each FFT to a power of 2, i.e.,  $N = 2^p$ , where  $p = 8-12$  are commonly used.
- The frequency resolution of the Fourier components will be  $1/(N\Delta t)$ .
- The Nyquist frequency where aliasing is initiated occurs at  $k = N/2$  and it is  $\omega_N = 1/(2\Delta t)$ . The sampling frequency should be at least twice the highest frequency expected to be found in the signal.



- The first  $N/2 + 1$  Fourier components up to the Nyquist frequency are related to the last  $N/2 - 1$  components above the Nyquist frequency by  $Y(k) = Y^*(N - k)$ ,  $k = 0, 1, 2, \dots, N - 1$ , where the asterisk denotes complex conjugate.
- The Fourier components that define only the positive frequencies are called a one-sided spectrum. This spectrum is given by  $Y(0)$ ,  $Y(N/2)$ , and  $2Y(k)$ ,  $k = 1, 2, \dots, N/2 - 1$ .

The one-sided spectral components for a periodic signal with zero mean are given by:

$$P(\omega) = \frac{2Y(\omega, T)}{T}, \quad \omega > 0, \text{ or } k = 1, 2, \dots, \frac{N}{2} - 1, \quad (15.14)$$

where  $Y(\omega, T)$  is defined in Equation (15.12). The magnitudes of the Fourier components,  $|P(\omega)|$ , are usually plotted as a stem plot (see example below) and, hence, it is often called a line spectrum. Because each of the Fourier components are complex, they define both a phase and an amplitude for each of the components. However, the phase information is generally utilized only in applications where there may be a need to reconstruct the time history of the signal or to determine the peak values of the signal.

Let us consider the following example to illustrate how MATLAB can be used to apply an FFT to a signal. Consider the following signal:

$$y(t) = 0.7 \sin(2\pi 50t) + \sin(2\pi 120t) + \text{noise}, \quad (15.15)$$

where the sample is taken over the interval of time  $0 \leq t \leq T$ . The sampling frequency is  $F_s = 3000$ . Thus  $T = 1/F_s$ . The power spectrum is also usually plotted to isolate the dominant frequencies in a signal. It is  $P_{YY} = (Y(\omega, T)Y^*(\omega, T))/L$ , where  $L = 2^p$  and  $p = 11$  was selected in this example. With this brief description of a signal processing problem, the following script was applied that implements the FFT methodology to examine this signal (it is based on an example in the help documentation available within MATLAB).

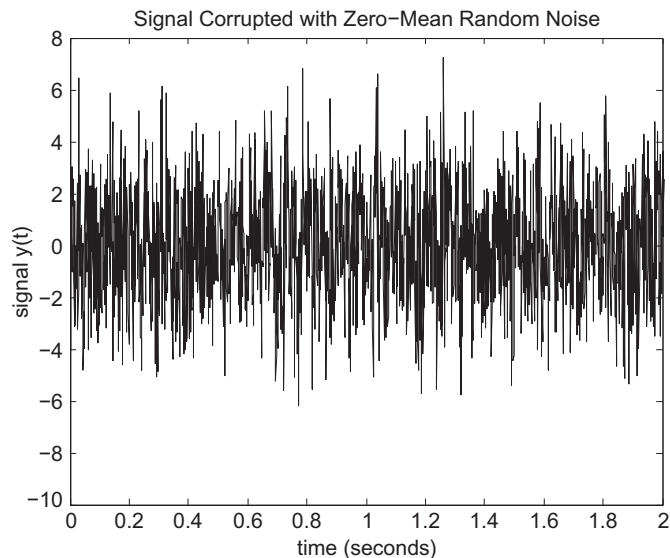
```
% -----Example of the application of FFT-----
% Example of the application of FFT
Fs = 3000;                % Sampling frequency
T = 1/Fs;                 % Sample time
pwr2 = 11;
L = power(2,pwr2);        % Length of signal
t = (0:L-1)*T;            % Time vector for FFT
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
```

```

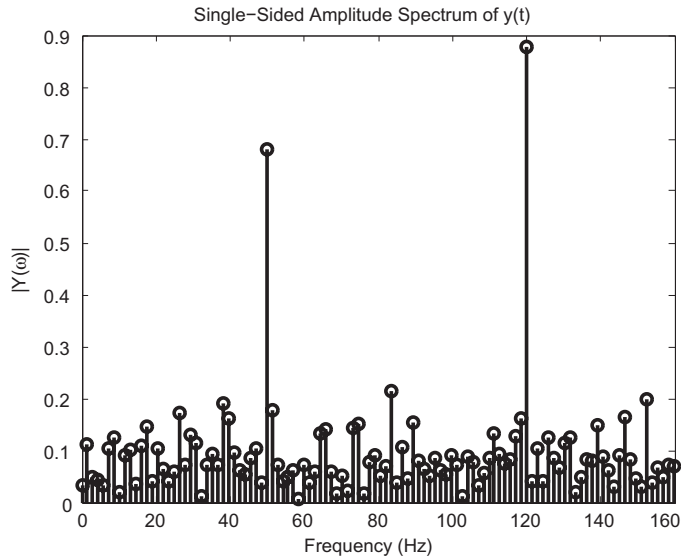
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t)); % Sinusoids plus noise at t
plot(Fs*t/1000,y,'k'),axis([0 2 -10 8])
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (seconds)')
ylabel('signal y(t)')
figure(2)
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
Pyy = Y.*conj(Y)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);
%
% -----Plot single-sided amplitude spectrum.-----
FF = 2*abs(Y(1:NFFT/2+1));
stem(f(1:110),FF(1:110),'k','Linewidth',2)
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(\omega)|')
figure(3)
stem(f(1:110),Pyy(1:110),'k','Linewidth',2)
title('Power spectral density')
xlabel('Frequency (Hz)')
ylabel('Pyy')

```

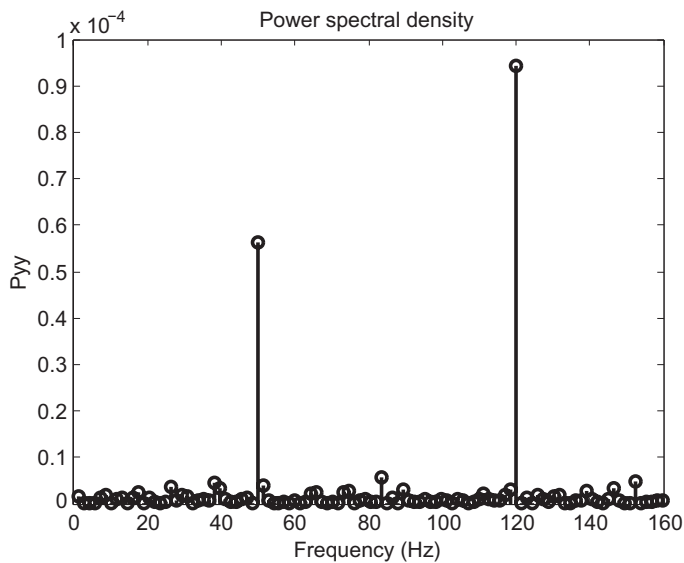
The signal to be analyzed by spectral analysis (as it is called) is illustrated in [Figure 15.5](#). The line spectrum and the power spectrum predicted for this signal are given in [Figures 15.6 and 15.7](#), respectively. There are two things to



**FIGURE 15.5** Example of signal to be analyzed by spectral analysis.



**FIGURE 15.6** Line spectrum of the signal in Figure 15.5.



**FIGURE 15.7** Power spectrum of the signal in Figure 15.5.

note. One is that the dominant frequencies in the signal are identified clearly in the spectra. The second is that the power spectrum isolates the dominant modes in the signal more clearly.