

Biosignal Processing

Monty Escabí, PhD

O U T L I N E

11.1	Introduction	668	11.7	Signal Averaging	721
11.2	Physiological Origins of Biosignals	668	11.8	The Wavelet Transform and the Short-Time Fourier Transform	727
11.3	Characteristics of Biosignals	671	11.9	Artificial Intelligence Techniques	732
11.4	Signal Acquisition	674	11.10	Exercises	741
11.5	Frequency Domain Representation of Biological Signals	679	Suggested Readings	745	
11.6	Linear Systems	700			

AT THE CONCLUSION OF THIS CHAPTER, STUDENTS WILL BE ABLE TO:

- Describe the different origins and types of biosignals.
- Distinguish between deterministic, periodic, transient, and random signals.
- Explain the process of A/D conversion.
- Define the sampling theorem.
- Describe the main purposes and uses of the Fourier transforms.
- Define the Z-transform.
- Describe the basic properties of a linear system.
- Describe the concepts of filtering and signal averaging.
- Explain the basic concepts and advantages of fuzzy logic.
- Describe the basic concepts of artificial neural networks.

11.1 INTRODUCTION

Biological signals, or biosignals, are space, time, or space-time records of a biological event such as a beating heart or a contracting muscle. The electrical, chemical, and mechanical activity that occurs during this biological event often produces signals that can be measured and analyzed. Biosignals, therefore, contain useful information that can be used to understand the underlying physiological mechanisms of a specific biological event or system and that may be useful for medical diagnosis.

Biological signals can be acquired in a variety of ways—for example, by a physician who uses a stethoscope to listen to a patient's heart sounds or with the aid of technologically advanced biomedical instruments. Following data acquisition, biological signals are analyzed in order to retrieve useful information. Basic methods of signal analysis, such as amplification, filtering, digitization, processing, and storage, can be applied to many biological signals. These techniques are generally accomplished with simple electronic circuits or with digital computers. In addition to these common procedures, sophisticated digital processing methods are quite common and can significantly improve the quality of the retrieved data. These include signal averaging, wavelet analysis, and artificial intelligence techniques.

11.2 PHYSIOLOGICAL ORIGINS OF BIOSIGNALS

11.2.1 Bioelectric Signals

Nerve and muscle cells generate bioelectric signals that are the result of electrochemical changes within and between cells (see Chapter 5). If a nerve or muscle cell is stimulated by a stimulus that is strong enough to reach a necessary threshold, the cell will generate an action potential. The action potential, which represents a brief flow of ions across the cell membrane, can be measured with intracellular or extracellular electrodes. Action potentials generated by an excited cell can be transmitted from one cell to adjacent cells via its axon. When many cells become activated, an electric field is generated that propagates through the biological tissue. These changes in extracellular potential can be measured on the surface of the tissue or organism by using surface electrodes. The electrocardiogram (ECG), electrogastrogram (EGG), electroencephalogram (EEG), and electromyogram (EMG) are all examples of this phenomenon ([Figure 11.1](#)).

11.2.2 Biomagnetic Signals

Different organs, including the heart, brain, and lungs, also generate weak magnetic fields that can be measured with magnetic sensors. Typically, the strength of the magnetic field is much weaker than the corresponding physiological bioelectric signals. Biomagnetism is the measurement of the magnetic signals that are associated with specific physiological activity and that are typically linked to an accompanying electric field from a specific tissue or organ. With the aid of very precise magnetic sensors or SQUID (superconducting quantum interference device) magnetometers, it is possible to directly monitor

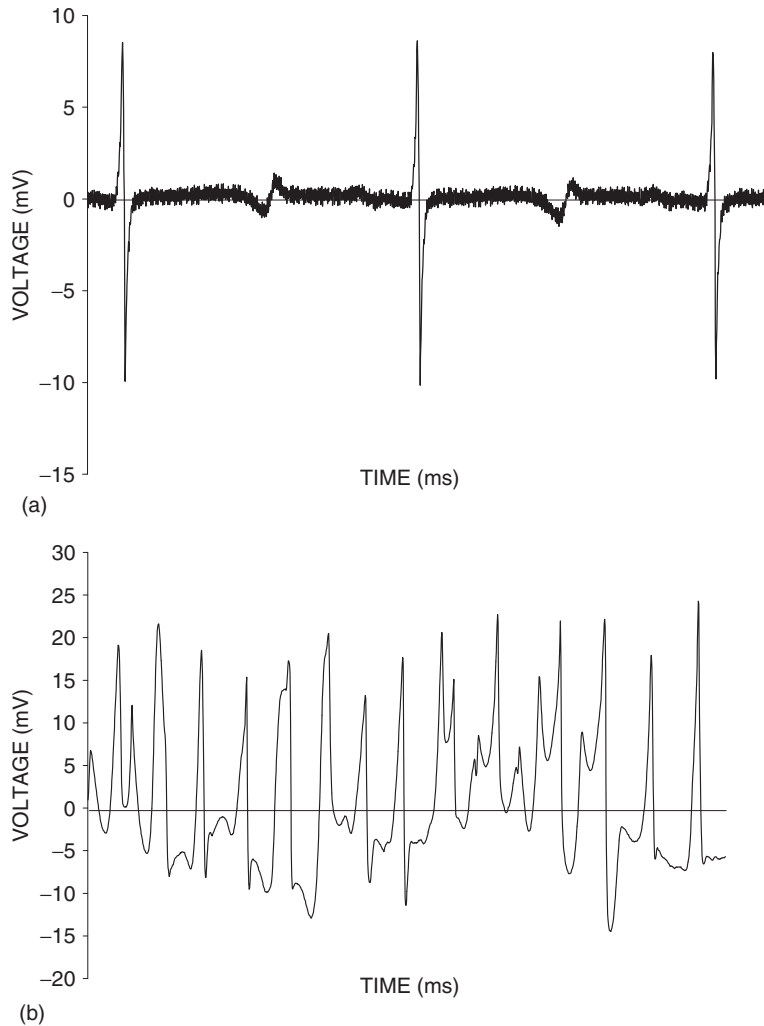


FIGURE 11.1 (a) Electrogram recorded from the surface of a pig's heart during normal sinus rhythm. (b) Electrogram recorded from the surface of the same pig's heart during ventricular fibrillation (VF). (Sampled at 1,000 samples/s.)

magnetic activity from the brain (magnetoencephalography, MEG), peripheral nerves (magnetoneurography, MNG), gastrointestinal tract (magnetogastrography, MGG), and the heart (magnetocardiography, MCG).

11.2.3 Biochemical Signals

Biochemical signals contain information about changes in concentration of various chemical agents in the body. The concentration of various ions, such as calcium and potassium, in cells can be measured and recorded. Changes in the partial pressures of oxygen (P_{O_2}) and

carbon dioxide (P_{CO_2}) in the respiratory system or blood are often measured to evaluate normal levels of blood oxygen concentration. All of these constitute biochemical signals. These biochemical signals can be used for a variety of purposes, such as determining levels of glucose, lactate, and metabolites and providing information about the function of various physiological systems.

11.2.4 Biomechanical Signals

Mechanical functions of biological systems, which include motion, displacement, tension, force, pressure, and flow, also produce measurable biological signals. Blood pressure, for example, is a measurement of the force that blood exerts against the walls of blood vessels. Changes in blood pressure can be recorded as a waveform (Figure 11.2). The upstrokes in the waveform represent the contraction of the ventricles of the heart as blood is ejected from the heart into the body and blood pressure increases to the systolic pressure, the maximum blood pressure (see Chapter 3). The downward portion of the waveform depicts ventricular relaxation as the blood pressure drops to the minimum value, better known as the diastolic pressure.

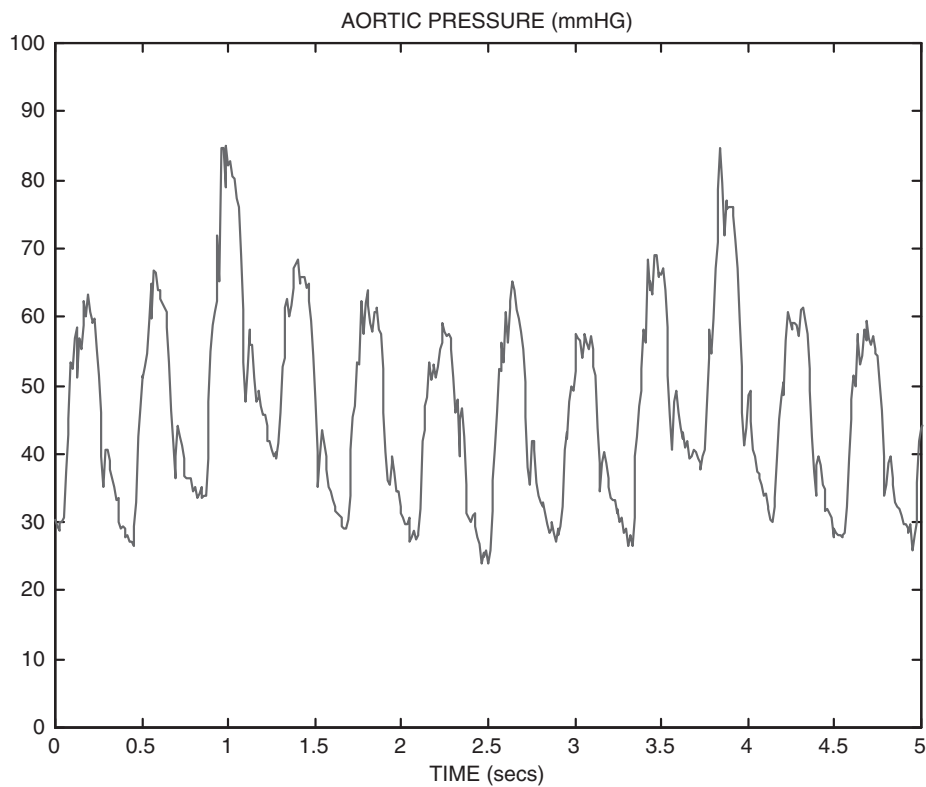


FIGURE 11.2 Blood pressure waveform recorded from the aortic arch of a 4-year-old child. (Sampled at 200 samples/s.)

11.2.5 Bioacoustic Signals

Bioacoustic signals are a special subset of biomechanical signals that involve vibrations (motion). Many biological events produce acoustic noise. For instance, the flow of blood through the valves in the heart has a distinctive sound. Measurements of the bioacoustic signal of a heart valve can be used to determine whether it is operating properly. The respiratory system, joints, and muscles also generate bioacoustic signals that propagate through the biological medium and can often be measured at the skin surface by using acoustic transducers such as microphones and accelerometers.

11.2.6 Biooptical Signals

Biooptical signals are generated by the optical or light induced attributes of biological systems. Biooptical signals can occur naturally, or in some cases, the signals may be introduced to measure a biological parameter with an external light medium. For example, information about the health of a fetus may be obtained by measuring the fluorescence characteristics of the amniotic fluid. Estimates of cardiac output can be made by using the dye dilution method that involves monitoring the concentration of a dye as it recirculates through the bloodstream. Finally, red and infrared light are used in various applications, such as to obtain precise measurements of blood oxygen levels by measuring the light absorption across the skin or a particular tissue.

EXAMPLE PROBLEM 11.1

What types of biosignals would the muscles in your lower legs produce if you were to sprint across a paved street?

Solution

Motion of the muscles and the external forces imposed as your feet hit the pavement produce biomechanical signals. Muscle stimulation by nerves and the contraction of muscle cells produce bioelectric signals. Metabolic processes in the muscle tissue could be measured as biochemical signals.

11.3 CHARACTERISTICS OF BIOSIGNALS

Biological signals can be classified according to various characteristics of the signal, including the waveform shape, statistical structure, and temporal properties. Two broad classes of signals that are commonly encountered include continuous and discrete signals. *Continuous* signals are defined over a continuum of time or space and are described by continuous variables. The notation $x(t)$ is used to represent a signal, x , that varies as a function of continuous time, t . Signals that are produced by biological phenomena are almost always continuous signals. Some examples include voltage measurements from the heart (see [Figure 11.1](#)), arterial blood pressure measurements (see [Figure 11.2](#)), and measurements of electrical activity from the brain.

Discrete signals are also commonly encountered in today's clinical setting. Unlike continuous signals, which are defined along a continuum of points in space or time, discrete signals are defined only at a subset of regularly spaced points in time and/or space. Discrete signals are therefore represented by arrays or sequences of numbers. The notation, $x(n)$, is used to represent a discrete sequence, x , that exists only at a subset of points in discrete time, n . Here, $n = 0, 1, 2, 3 \dots$ is always an integer that represents the n th element of the discrete sequence. Although most biological signals are not discrete per se, discrete signals play an important role due to today's advancements in digital technology. Sophisticated medical instruments are commonly used to convert continuous signals from the human body to discrete digital sequences (see Chapter 7) that can be analyzed and interpreted with a computer. Computer axial tomography (CAT) scans, for instance, take digital samples from continuous x-ray images of a patient that are obtained from different perspective angles (see Chapter 15). These digitized or discrete image slices are then digitally enhanced, manipulated, and processed to generate a full three-dimensional computer model of a patient's internal organs. Such technologies are indispensable tools for clinical diagnosis.

Biological signals can also be classified as being either *deterministic* or *random*. Deterministic signals can be described by mathematical functions or rules. *Periodic* and *transient* signals make up a subset of all deterministic signals. Periodic signals are usually composed of the sum of different sine waves or sinusoid components and can be expressed as

$$x(t) = x(t + kT) \quad (11.1)$$

where $x(t)$ is the signal, k is an integer, and T is the period. The period represents the distance along the time axis between successive copies of the periodic signal. Periodic signals have a basic waveshape with a duration of T units that repeats indefinitely. Transient signals are nonzero or vary only over a finite time interval and subsequently decay to a constant value as time progresses. The sine wave, shown in Figure 11.3a, is a simple example of a periodic signal, since it repeats indefinitely with a repetition interval of 1 second. The product of a decaying exponential and a sine wave, as shown in Figure 11.3b, is a transient signal, since the signal amplitude approaches zero as time progresses.

Real biological signals almost always have some unpredictable noise or change in parameters and, therefore, are not entirely deterministic. The ECG of a normal beating heart at rest is an example of a signal that appears to be almost periodic but has a subtle unpredictable component. The basic waveshape consists of the P wave, QRS complex, and T wave and repeats (see Figure 3.22). However, the precise shapes of the P waves, QRS complexes, and T are somewhat irregular from one heartbeat to another. The length of time between QRS complexes, which is known as the R-R interval, also changes over time as a result of heart rate variability (HRV). HRV is used as a diagnostic tool to predict the health of a heart that has experienced a heart attack. The extended outlook for patients with low HRV is generally worse than it is for patients with high HRV.

Random signals, also called *stochastic* signals, contain uncertainty in the parameters that describe them. Because of this uncertainty, mathematical functions cannot be used to precisely describe random signals. Instead, random signals are most often analyzed using statistical techniques that require the treatment of the random parameters of the signal with probability distributions or simple statistical measures such as the mean and standard deviation. The electromyogram (EMG), an electrical recording of electrical activity in skeletal muscle that is

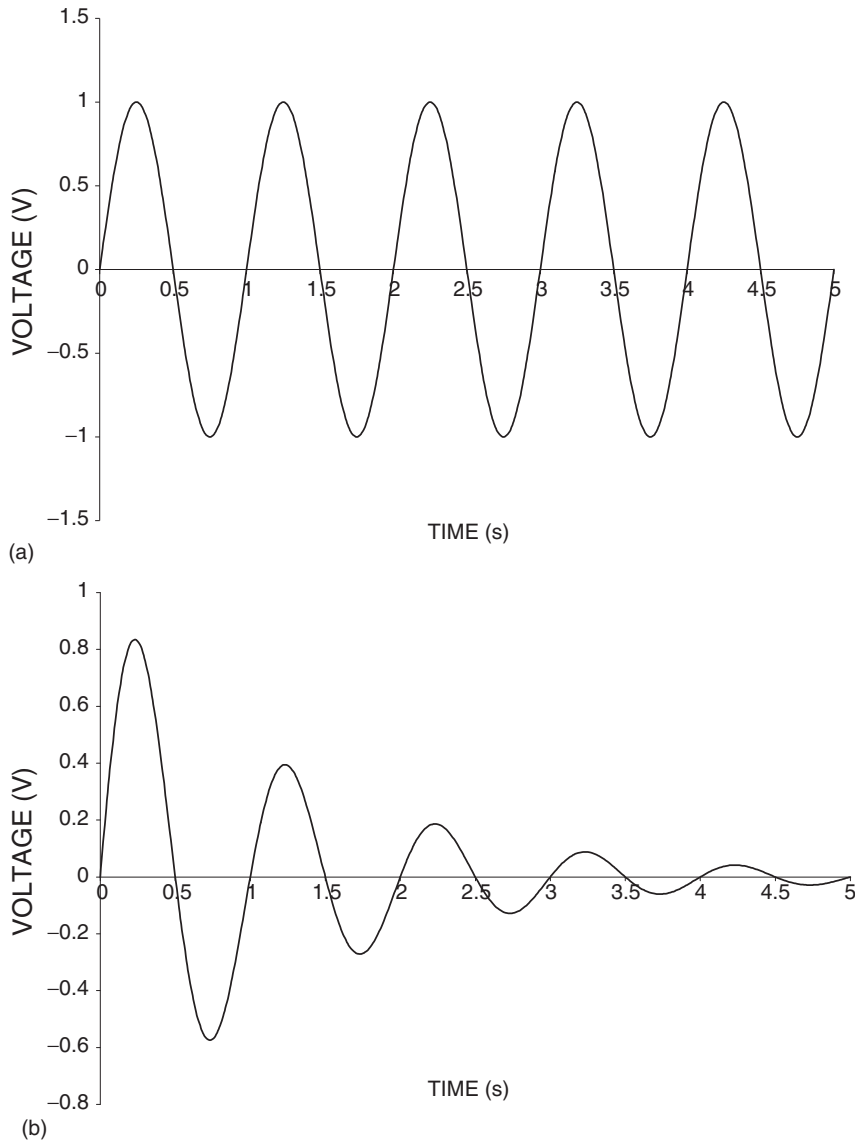


FIGURE 11.3 (a) Periodic sine wave signal $x(t) = \sin(\omega t)$ with period of 1 Hz. (b) Transient signal $y(t) = e^{-0.75t} \sin(\omega t)$ for the same 1 Hz sine wave.

used for the diagnosis of neuromuscular disorders, is a random signal. Stationary random signals have statistical properties, such as a mean and variance, that remain constant over time. Conversely, nonstationary random signals have statistical properties that vary with time. In many instances, the identification of stationary segments of random signals is important for proper signal processing, pattern analysis, and clinical diagnosis.

EXAMPLE PROBLEM 11.2

Ventricular fibrillation (VF) is a cardiac arrhythmia in which there are no regular QRS complexes, T waves, or rhythmic contractions of the heart muscle (see [Figure 11.1b](#)). VF often leads to sudden cardiac death, which is one of the leading causes of death in the United States. What type of biosignal would most probably be recorded by an ECG when a heart goes into VF?

Solution

An ECG recording of a heart in ventricular fibrillation will be a random, continuous, bioelectric signal.

11.4 SIGNAL ACQUISITION

11.4.1 Overview of Biosignal Data Acquisition

Biological signals are often very small and typically contain unwanted interference or noise. Such interference has the detrimental effect of obscuring relevant information that may be available in the measured signal. Noise can be extraneous in nature, arising from sources outside the body, such as thermal noise in sensors or 60-cycle noise in the electronic components of the acquisition system. Noise can also be intrinsic to the biological media, meaning it can arise from adjacent tissues or organs. ECG measurements from the heart, for instance, can be affected by bioelectric activity from adjacent muscles.

In order to extract meaningful information from biological signals sophisticated data acquisition techniques and equipment are commonly used. High-precision low-noise equipment is often necessary to minimize the effects of unwanted noise. [Figure 11.4](#) shows the basic components in a bioinstrumentation system.

Throughout the data acquisition procedure, it is critical that the information and structure of the original biological signal of interest be faithfully preserved. Since these signals are often used to aid the diagnosis of pathological disorders, the procedures of amplification, analog filtering, and A/D conversion should not generate misleading or untraceable distortions. Distortions in a signal measurement could lead to an improper diagnosis.

11.4.2 Sensors, Amplifiers, and Analog Filters

Signals are first detected in the biological medium, such as a cell or on the skin's surface, by using a sensor (see Chapter 6). A sensor converts a physical measurand into an electric output and provides an interface between biological systems and electrical recording instruments. The type of biosignal determines what type of sensor will be used. ECGs, for example, are measured with electrodes that have a silver-silver chloride (Ag-AgCl) interface attached to the body that detects the movement of ions. Arterial blood pressure is measured with a sensor that detects changes in pressure. It is very important that the sensor used to detect the biological signal of interest does not adversely affect the properties and characteristics of the signal it is measuring.

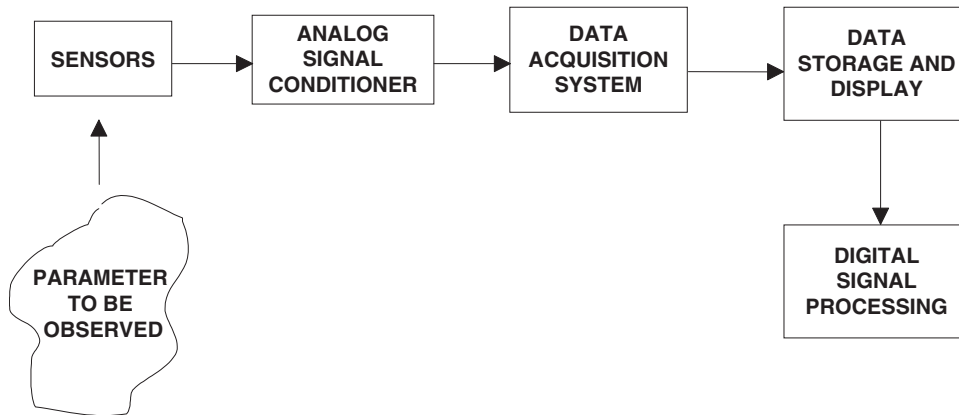


FIGURE 11.4 Sensors adapt the signal that is being observed into an electrical analog signal that can be measured with a data acquisition system. The data acquisition system converts the analog signal into a calibrated digital signal that can be stored. Digital signal processing techniques are applied to the stored signal to reduce noise and extract additional information that can improve understanding of the physiological meaning of the original parameter.

After the biosignal has been detected with an appropriate sensor, it is usually amplified and filtered. Operational amplifiers are electronic circuits that are used primarily to increase the amplitude or size of a biosignal. Bioelectric signals, for instance, are often faint and require up to a thousand-fold boosting of their amplitude with such amplifiers. An analog filter may then be used to remove noise or to compensate for distortions caused by the sensor. Amplification and filtering of the biosignal may also be necessary to meet the hardware specifications of the data acquisition system. Continuous signals may need to be limited to a certain band of frequencies before the signal can be digitized with an analog-to-digital converter, prior to storing in a digital computer.

11.4.3 A/D Conversion

Analog-to-digital (A/D) converters are used to transform biological signals from continuous analog waveforms to digital sequences. An A/D converter is a computer-controlled voltmeter, which measures an input analog signal and gives a numeric representation of the signal as its output. [Figure 11.5a](#) shows an analog signal, and [Figure 11.5b](#) shows a digital version of the same signal. The analog waveform, originally detected by the sensor and subsequently amplified and filtered, is a continuous signal. The A/D converter transforms the continuous, analog signal into a discrete, digital signal. The discrete signal consists of a sequence of numbers that can easily be stored and processed on a digital computer. A/D conversion is particularly important because storage and analysis of biosignals are becoming increasingly computer based.

The digital conversion of an analog biological signal does not produce an exact replica of the original signal. The discrete, digital signal is a digital approximation of the original, analog signal that is generated by repeatedly sampling the amplitude level of the original signal at fixed time intervals. As a result, the original, analog signal is represented as a sequence of numbers: the digital signal.

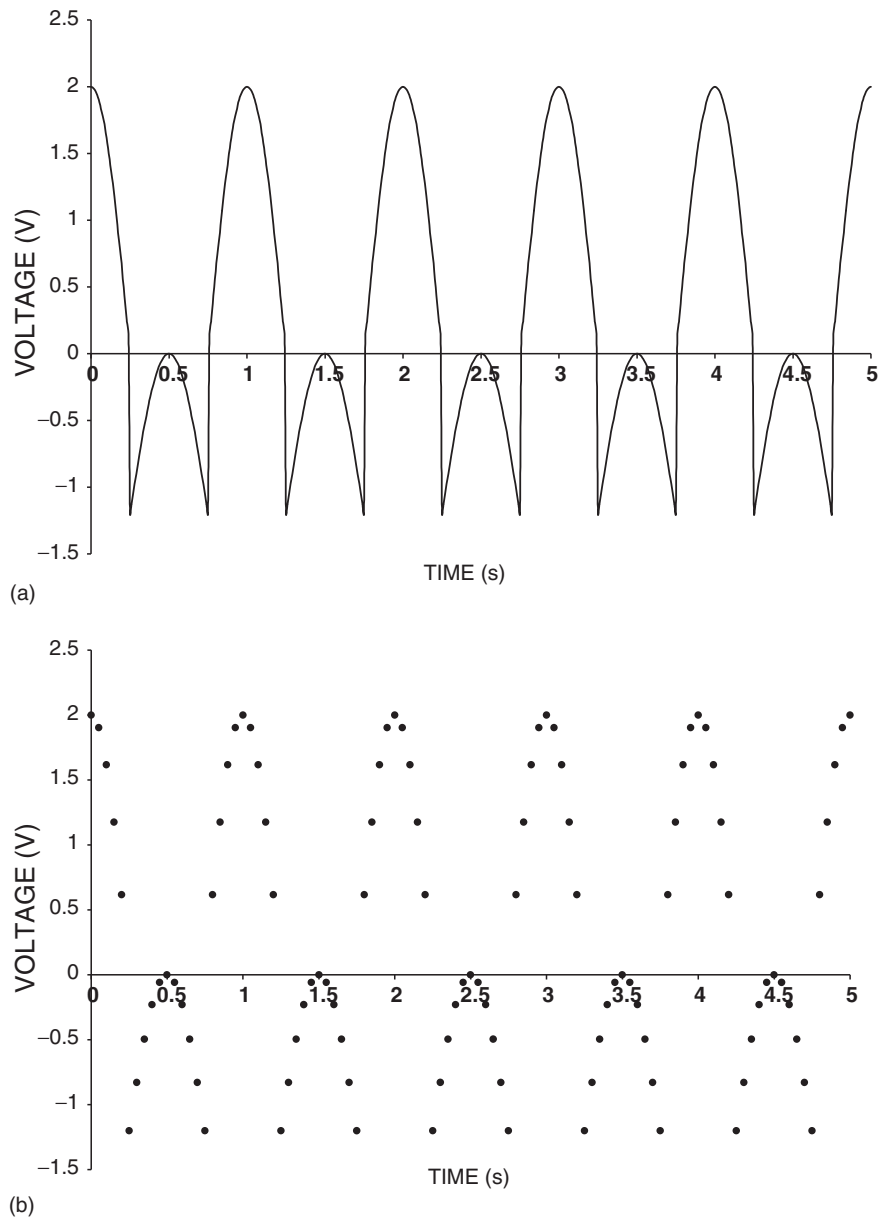


FIGURE 11.5 (a) Analog version of a periodic signal. (b) Digital version of the analog signal.

The two main processes involved in A/D conversion are *sampling* and *quantization*. Sampling is the process by which a continuous signal is first converted into a discrete sequence in time. If $x(t)$ is an analog signal, sampling involves recording the amplitude value of $x(t)$ every T seconds. The amplitude value is denoted as $x(kT)$ where $k = 0, 1, 2, 3, \dots$ is an integer that denotes the position or the sample number from the sample set or data sequence.

T represents the sampling interval or the time between adjacent samples. In real applications, finite data sequences are generally used in digital signal processing. Therefore, the range of a data points is $k = 0, 1, \dots, N-1$, where N is the total number of discrete samples. The sampling frequency, f_s , or the sampling rate, is equal to the inverse of the sampling period, $1/T$, and is measured in units of Hertz (s^{-1}).

The following are digital sequences that are of particular importance:

The unit-sample of impulse sequence:

$$\delta(k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}$$

The unit-step sequence:

$$u(k) = \begin{cases} 1 & \text{if } k > 0 \\ 0 & \text{if } k < 0 \end{cases}$$

The exponential sequence:

$$a^k u(k) = \begin{cases} a^k & \text{if } k > 0 \\ 0 & \text{if } k < 0 \end{cases}$$

The sampling rate used to discretize a continuous signal is critical for the generation of an accurate digital approximation. If the sampling rate is too low, distortions will occur in the digital signal. Nyquist's theorem states that the minimum sampling rate used, f_s , should be at least twice the maximum frequency of the original signal in order to preserve all of the information of the analog signal. The Nyquist rate is calculated as

$$f_{\text{nyquist}} = 2 \cdot f_{\text{max}} \quad (11.2)$$

where f_{max} is the highest frequency present in the analog signal. The Nyquist theorem therefore states that f_s must be greater than or equal to $2 \cdot f_{\text{max}}$ in order to fully represent the analog signal by a digital sequence. Practically, sampling is usually done at five to ten times the highest frequency, f_{Max} .

The second step in the A/D conversion process involves signal quantization. Quantization is the process by which the continuous amplitudes of the discrete signal are digitized by a computer. In theory, the amplitudes of a continuous signal can be any of an infinite number of possibilities. This makes it impossible to store all the values, given the limited memory in computer chips. Quantization overcomes this by reducing the number of available amplitudes to a finite number of possibilities that the computer can handle.

Since digitized samples are usually stored and analyzed as binary numbers on computers, every sample generated by the sampling process must be quantized. During quantization, the series of samples from the discretized sequence are transformed into binary numbers. The resolution of the A/D converter determines the number of bits that are available for storage. Typically, most A/D converters approximate the discrete samples with 8, 12, or 16 bits. If the number of bits is not sufficiently large, significant errors may be incurred in the digital approximation.

A/D converters are characterized by the number of bits that they use to generate the numbers of the digital approximation. A quantizer with N bits is capable of representing a total of 2^N possible amplitude values. Therefore, the resolution of an A/D converter

increases as the number of bits increases. A 16-bit A/D converter has better resolution than an 8-bit A/D converter, since it is capable of representing a total of 65,536 amplitude levels, compared to 256 for the 8-bit converter. The resolution of an A/D converter is determined by the voltage range of the input analog signal divided by the numeric range (the possible number of amplitude values) of the A/D converter.

EXAMPLE PROBLEM 11.3

Find the resolution of an 8-bit A/D converter when an input signal with a 10 V range is digitized.

Solution

$$\frac{\text{input voltage range}}{2^N} = \frac{10 \text{ V}}{256} = 0.0391 \text{ V/bit} = 39.1 \text{ mV/bit}$$

EXAMPLE PROBLEM 11.4

The frequency content of an analog EEG signal is 0.5–100 Hz. What is the lowest rate at which the signal can be sampled to produce an accurate digital signal?

Solution

Highest frequency in analog signal = 100 Hz.

$f_{\text{nyquist}} = 2 \cdot f_{\text{max}} = 2 \cdot 100 \text{ Hz} = 200 \text{ samples/second.}$

Another problem often encountered is determining what happens if a signal is not sampled at a rate high enough to produce an accurate representation of the signal. A direct result of the sampling theorem is that all frequencies of the form $[f - kf_s]$, where $-\infty \leq k \leq \infty$ and $f_s = 1/T$, look the same once they are sampled.

EXAMPLE PROBLEM 11.5

A 360 Hz signal is sampled at 200 samples/second. What frequency will the “aliased” digital signal contain?

Solution

According to the preceding formula, $f_s = 200$, and the pertinent set of frequencies that look alike is in the form of $[360 - k 200] = [\dots 360 \ 160 \ -40 \ -240 \ \dots]$. The only signal in this group that will be accurately sampled is 40 Hz, since the sampling rate is more than twice this value. Note that for real signals -40 Hz and $+40 \text{ Hz}$ are equivalent—that is, $\cos(-\omega t) = \cos(\omega t)$ and $\sin(-\omega t) = -\sin(\omega t)$. Thus, the sampled signal will exhibit a period of 40 Hz. The process is shown in [Figure 11.6](#).

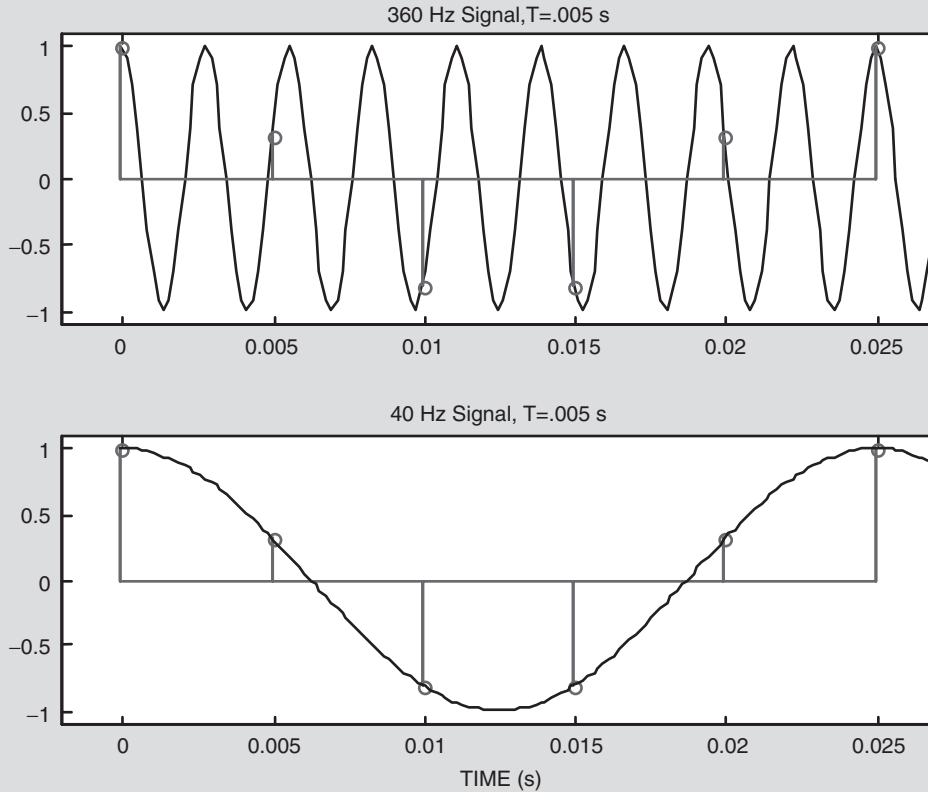


FIGURE 11.6 A 360 Hz sine wave is sampled every 5 ms—that is, at 200 samples/s. This sampling rate will adequately sample a 40 Hz sine wave but not a 360 Hz sine wave.

11.5 FREQUENCY DOMAIN REPRESENTATION OF BIOLOGICAL SIGNALS

In the early nineteenth century, Joseph Fourier laid out one of the most important theories on the field of function approximation. At the time, his result was applied toward the problem of heat transfer in solids, but it has since gained a much broader appeal. Today, Fourier's findings provide a general theory for approximating complex waveforms with simpler functions that has numerous applications in mathematics, physics, and engineering. This section summarizes the Fourier transform and variants of this technique that play an important role in the analysis and interpretation of biological signals.

11.5.1 Periodic Signal Representation: The Trigonometric Fourier Series

As an artist mixes oil paints on a canvas, a scenic landscape is meticulously recreated by combining various colors on a palette. It is well known that all shades of the color spectrum

can be recreated by simply mixing primary colors (red, green, and blue, or RGB) in the correct proportions. Television and computer displays often transmit signals as RGB, and these signals are collated together to create colors much as a master painter would on a canvas. In fact, the human visual system takes exactly the opposite approach. The retina decomposes images and scenery from the outside world into purely red-green-blue signals that are independently analyzed and processed by our brains. Despite this, we perceive a multitude of colors and shades.

This simple color analogy is at the heart of Fourier's theory, which states that a complex waveform can be approximated to any degree of accuracy with simpler functions. In 1807, Fourier showed that an arbitrary periodic signal of period, T , can be represented mathematically as a sum of trigonometric functions. Conceptually, this is achieved by summing or mixing sinusoids while simultaneously adjusting their amplitudes and frequency, as shown for a square wave function in [Figure 11.7](#).

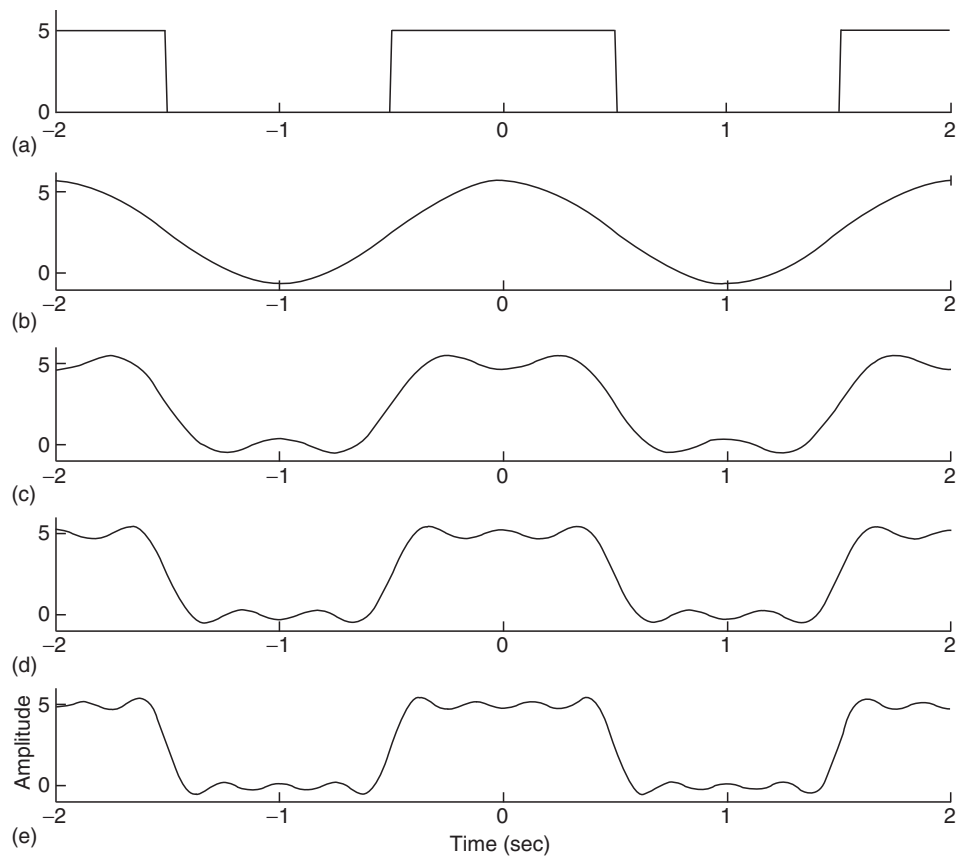


FIGURE 11.7 A square wave signal (a) is approximated by adding sinusoids (B–E). (b) 1 sinusoid, (c) 2 sinusoids, (d) 3 sinusoids, (e) 4 sinusoids. Increasing the number of sinusoids improves the quality of the approximation.

If the amplitudes and frequencies are chosen appropriately, the trigonometric signals add constructively, thus recreating an arbitrary periodic signal. This is akin to combining prime colors in precise ratios to recreate an arbitrary color and shade. RGB are the building blocks for more elaborate colors, much as sinusoids of different frequencies serve as the building blocks for more complex signals. All of these elements (the color and the required proportions; the frequencies and their amplitudes) have to be precisely adjusted to achieve a desired result. For example, a first-order approximation of the square wave is achieved by fitting the square wave to a single sinusoid of appropriate frequency and amplitude. Successive improvements in the approximation are obtained by adding higher-frequency sinusoid components, or *harmonics*, to the first-order approximation. If this procedure is repeated indefinitely, it is possible to approximate the square wave signal with infinite accuracy.

The Fourier series summarizes this result as

$$x(t) = a_0 + \sum_{m=1}^{\infty} (a_m \cos m\omega_0 t + b_m \sin m\omega_0 t) \quad (11.3a)$$

where $x(t)$ is the periodic signal to be approximated, $\omega_0 = 2\pi/T$ is the fundamental frequency of $x(t)$ in units of radians/s, and the coefficients a_m and b_m determine the amplitude of each cosine and sine term at a specified frequency $\omega_m = m\omega_0$. Equation (11.3a) tells us that the periodic signal, $x(t)$, is precisely replicated by summing an infinite number of sinusoids. The frequencies of the sinusoid functions always occur at integer multiples of ω_0 and are referred to as “harmonics” of the fundamental frequency. If we know the coefficients a_m and b_m for each of the corresponding sine or cosine terms, we can completely recover the signal $x(t)$ by evaluating the Fourier series. How do we determine a_m and b_m for an arbitrary signal?

The coefficients of the Fourier series correspond to the amplitude of each sine and cosine. These are determined as

$$a_0 = \frac{1}{T} \int_T x(t) dt \quad (11.3b)$$

$$a_m = \frac{2}{T} \int_T x(t) \cos(m\omega_0 t) dt \quad (11.3c)$$

$$b_m = \frac{2}{T} \int_T x(t) \sin(m\omega_0 t) dt \quad (11.3d)$$

where the integrals are evaluated over a single period, T , of the waveform.

EXAMPLE PROBLEM 11.6

Find the trigonometric Fourier series of the square wave signal shown in Figure 11.7A, and implement the result in MATLAB for the first ten components. Plot the time waveform and the Fourier coefficients.

Continued

Solution

First note that

$$T = 2 \quad \text{and} \quad \omega_0 = \frac{2\pi}{T} = \pi$$

To simplify the analysis, integration for a_m and b_m is carried out over the first period of the waveform (from -1 to 1)

$$a_0 = \frac{1}{T} \int_{-1}^1 x(t) dt = \frac{1}{2} \int_{-1/2}^{1/2} 5 dt = \frac{5}{2}$$

$$\begin{aligned} a_m &= \frac{2}{T} \int_{-1}^1 x(t) \cos(m\omega_0 t) dt = \int_{-1/2}^{1/2} 5 \cdot \cos(m\pi t) dt \\ &= -5 \frac{\sin(m\pi t)}{m\pi} \bigg|_{-1/2}^{1/2} = -5 \frac{\sin(m\pi/2)}{m\pi/2} = 5 \cdot \text{sinc}(m\pi/2) \end{aligned}$$

$$b_m = \frac{2}{T} \int_{-1}^1 x(t) \sin(m\omega_0 t) dt = \int_{-1/2}^{1/2} 5 \cdot \sin(m\pi t) dt = -5 \cdot \frac{\cos(m\pi t)}{m\pi} \bigg|_{-1/2}^{1/2} = 0$$

where by definition $\text{sinc}(x) = \sin(x)/x$. Substituting the values for a_0 , a_m , and b_m into [Eq. \(11.3a\)](#) gives

$$x(t) = \frac{5}{2} + 5 \cdot \sum_{m=1}^{\infty} \frac{\sin(m\pi/2)}{m\pi/2} \cos(m\pi t)$$

MATLAB implementation:

```
%Plotting Fourier Series Approximation
subplot(211)
time=-2:0.01:2; %Time Axis
x=5/2; %Initializing Signal
for m=1:10
    x=x+5*sin(m*pi/2)/m/pi*2*cos(m*pi*time);
end
plot(time,x,'k') %Plotting and Labels
xlabel('Time (sec)')
ylabel('Amplitude')
set(gca,'Xtick',[-2:2])
set(gca,'Ytick',[0 5])
set(gca,'Box','off')

%Plotting Fourier Magnitudes
subplot(212)
m=1:10;
```



```
Am=[5/2 5*sin(m*pi/2)./m/pi*2]; %Fourier Magnitudes
Faxis=(0:10)*.5; %Frequency Axis
plot(Faxis,Am,'k.') %Plotting
axis([0 5 -2 4])
set(gca,'Box','off')
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitudes')
```

Note that the approximation of summing the first ten harmonics (Figure 11.8a) closely resembles the desired square wave. The Fourier coefficients, a_m , for the first ten harmonics are shown as a function of the harmonic frequency in Figure 11.8b. To fully replicate the sharp transitions of the square wave, an infinite number of harmonics are required.

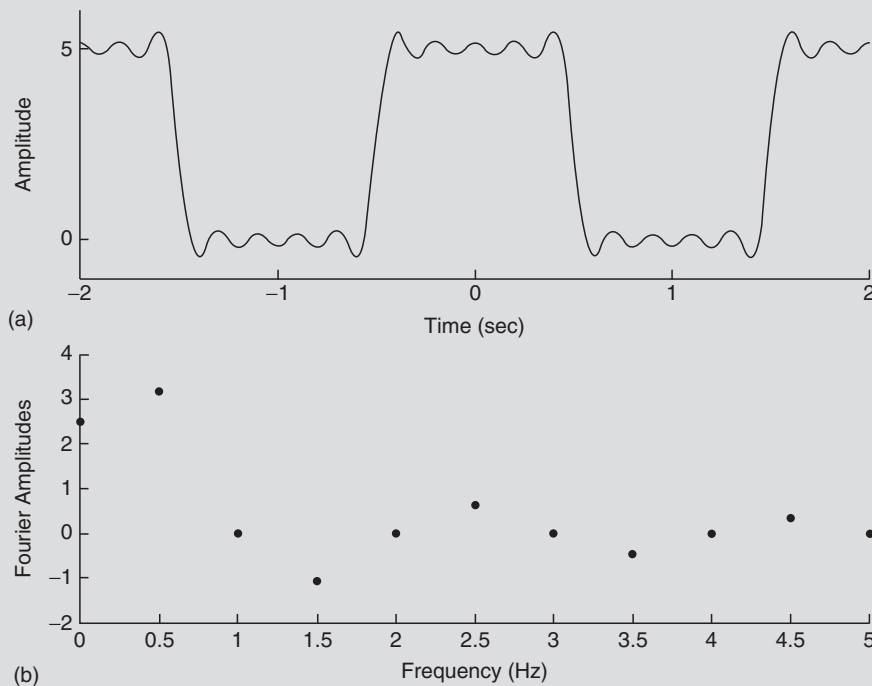


FIGURE 11.8 (a) MATLAB result showing the first ten terms of Fourier series approximation for the square wave. (b) The Fourier coefficients are shown as a function of the harmonic frequency.

11.5.2 Compact Fourier Series

The trigonometric Fourier series provides a direct approach for fitting and analyzing various types of biological signals, such as the repetitive beating of a heart or the cyclic oscillations produced by the vocal folds as one speaks. Despite its utility, alternate forms of the Fourier series are sometimes more appealing because they are easier to work with

and because signal measurements can often be interpreted more readily. The most widely used counterparts for approximating and modeling biological signals are the *exponential* and *compact* Fourier series.

The compact Fourier series is a close cousin of the standard Fourier series. This version of the Fourier series is obtained by noting that the sum of sinusoids and cosines can be rewritten by a single cosine term with the addition of a phase constant $a_m \cos m\omega_0 t + b_m \sin m\omega_0 t = A_m \cos(m\omega_0 t + \phi_m)$, which leads to the compact form of the Fourier series:

$$x(t) = \frac{A_0}{2} + \sum_{m=1}^{\infty} A_m \cos(m\omega_0 t + \phi_m). \quad (11.4a)$$

The amplitude for each cosine, A_m , is related to the Fourier coefficients through

$$A_m = \sqrt{a_m^2 + b_m^2} \quad (11.4b)$$

and the cosine phase is obtained from a_m and b_m as

$$\phi_m = \tan^{-1}\left(\frac{-b_m}{a_m}\right). \quad (11.4c)$$

EXAMPLE PROBLEM 11.7

Convert the standard Fourier series for the square pulse function of [Example Problem 11.5](#) to compact form and implement in MATLAB.

Solution

We first need to determine the magnitude, A_m , and phase, ϕ_m , for the compact Fourier series. The magnitude is obtained as

$$A_m = \sqrt{a_m^2 + b_m^2} = \sqrt{(5 \cdot \text{sinc}(m\pi/2))^2 + (0)^2} = 5 \frac{|\sin(m\pi/2)|}{\pi m/2}$$

Since

$$|\sin(m\pi/2)| = \begin{cases} 1 & m = \text{odd} \\ 0 & m = \text{even} \end{cases}$$

we have

$$A_m = \begin{cases} 10/m\pi & m = \text{odd} \\ 0 & m = \text{even} \end{cases}.$$

Unlike a_m or b_m in the standard Fourier series, note that A_m is strictly a positive quantity for all m . The phase term is determined as

$$\phi_m = \tan^{-1}\left(\frac{-b_m}{a_m}\right) = \tan^{-1}\left(\frac{0}{5 \cdot \text{sinc}(m\pi/2)}\right) = \begin{cases} 0 & \text{for } m = 0, 1, 4, 5, 8, 9 \dots \\ \pi & m = 2, 3, 6, 7, 10, 11 \dots \end{cases}$$

Combining results

$$x(t) = \frac{5}{2} + \sum_{m=1}^{\infty} \frac{10}{m\pi} \cos(m\omega_0 t + \phi_m)$$

where ϕ_m is as just defined. An interesting point regards the similarity of standard and compact versions of the Fourier series for this square wave example. In the standard form, the coefficient a_m alternates between positive and negative values, while for the compact form the Fourier coefficient, A_m , is identical in magnitude to a_m , but it is always a positive quantity. The sign (+ or -) of the standard Fourier coefficient is now consumed in the phase term, which alternates between 0 and π . This forces the cosine to alternate in its external sign because $-\cos(x) = \cos(x + \pi)$. The two equations are therefore mathematically identical, differing only in the way that the trigonometric functions are written out.

MATLAB implementation:

```
%Plotting Fourier Series Approximation
time=-2:0.01:2; %Time Axis
x=5/2; %Initializing Signal
m=1:10;
A=(10*sin(m*pi/2)./m/pi); %Fourier Coefficients
P=angle(A); %Phase Angle
A=abs(A); %Fourier Magnitude
for m=1:10
    x=x+A(m)*cos(m*pi*time+P(m));
end
subplot(211)
plot(time,x,'k') %Plotting and Labels
xlabel('Time (sec)')
ylabel('Amplitude')
set(gca,'Xtick',[-2:2])
set(gca,'Ytick',[0 5])
set(gca,'Box','off')

%Plotting Fourier Magnitudes
subplot(212)
m=1:10;
A=[5/2 A]; %Fourier Magnitudes
Faxis=(0:10)*.5; %Frequency Axis
plot(Faxis,A,'k.') %Plotting
axis([0 5 -2 4])
set(gca,'Box','off')
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitudes')
```

The results are identical to those shown in [Figures 11.8a and b](#).

11.5.3 Exponential Fourier Series

The main result from the Fourier series analysis is that an arbitrary periodic signal can approximate by summing individual cosine terms with specified amplitudes and phases. This result serves as much of the conceptual and theoretical framework for the field of signal analysis. In practice, the Fourier series is a useful tool for modeling various types of quasi-periodic signals.

An alternative and somewhat more convenient form of this result is obtained by noting that complex exponential functions are directly related to sinusoids and cosines through Euler's identities: $\cos(\theta) = (e^{j\theta} + e^{-j\theta})/2$ and $\sin(\theta) = (e^{j\theta} - e^{-j\theta})/2j$, where $j = \sqrt{-1}$. By applying Euler's identity to the compact trigonometric Fourier series, an arbitrary periodic signal can be expressed as a sum of complex exponential functions:

$$x(t) = \sum_{m=-\infty}^{+\infty} c_m e^{jk\omega_o t} \quad (11.5a)$$

This equation represents the exponential Fourier series of a periodic signal. The coefficients c_m are complex numbers that are related to the trigonometric Fourier coefficients

$$c_m = \frac{a_m - jb_m}{2} = \frac{A_m}{2} e^{j\phi_m} \quad (11.5b)$$

The proof for this result is beyond the scope of this text, but it is important to realize that the trigonometric and exponential Fourier series are intimately related, as can be seen by comparing their coefficients. The exponential coefficients can also be obtained directly by integrating $x(t)$,

$$c_m = \frac{1}{T} \int_T x(t) e^{-jm\omega_o t} dt \quad (11.5c)$$

over one cycle of the periodic signal. As for the trigonometric Fourier series, the exponential form allows us to approximate a periodic signal to any degree of accuracy by adding a sufficient number of complex exponential functions. A distinct advantage of the exponential Fourier series, however, is that it requires only a single integral (Eq. (11.5c)), compared to the trigonometric form, which requires three separate integrations.

EXAMPLE PROBLEM 11.8

Find the exponential Fourier series for the square wave of [Figure 11.7a](#) and implement in MATLAB for the first ten terms. Plot the time waveform and the Fourier series coefficients.

Solution

Like [Example Problem 11.6](#), the Fourier coefficients are obtained by integrating from -1 to 1 . Because a single cycle of the square wave signal has nonzero values between $-1/2$ and $+1/2$, the integral can be simplified by evaluating it between these limits:

$$\begin{aligned}
 c_m &= \frac{1}{T} \int_T x(t) e^{-jm\omega_0 t} dt = \frac{1}{2} \int_{-1/2}^{1/2} 5e^{-jm\pi t} dt = \frac{5}{2} \cdot \frac{e^{-jm\pi t}}{-jm\pi} \bigg|_{-1/2}^{1/2} \\
 &= \frac{5}{2} \cdot \frac{e^{+jm\pi/2} - e^{-jm\pi/2}}{jm\pi} = \frac{5}{2} \cdot \frac{\sin(m\pi/2)}{m\pi/2}.
 \end{aligned}$$

Therefore,

$$x(t) = \sum_{m=-\infty}^{+\infty} c_m e^{jk\omega_0 t} = \sum_{m=-\infty}^{\infty} \frac{5}{2} \cdot \frac{\sin(m\pi/2)}{m\pi/2} \cdot e^{jm\pi t}$$

MATLAB implementation:

```
%Plotting Fourier Series Approximation
subplot(211)
time=-2:0.01:2; %Time Axis
x=0; %Initialize Signal
for m=-10:10
    if m==0
        x=x+5/2; %Term for m=0
    else
        x=x+5/2*sin(m*pi/2)/m/pi*2*exp(j*m*pi*time);
    end
end
plot(time,x,'k') %Plotting and Labels
xlabel('Time (sec)')
ylabel('Amplitude')
set(gca,'Xtick',[-2:2])
set(gca,'Ytick',[0 5])
set(gca,'Box','off')

%Plotting Fourier Magnitudes
subplot(212)
m=(-10:10)+1E-10;
A=[5/2*sin(m*pi/2)./m/pi*2]; %Fourier Magnitudes
Faxis=(-10:10)*.5; %Frequency Axis
plot(Faxis,A,'k.') %Plotting
axis([-5 5 -2 4])
set(gca,'Box','off')
xlabel('Frequency (Hz)')
ylabel('Fourier Amplitudes')
```

Note that we now require positive and negative frequencies in the approximation. Results showing the MATLAB output are shown in [Figure 11.9](#).

Continued

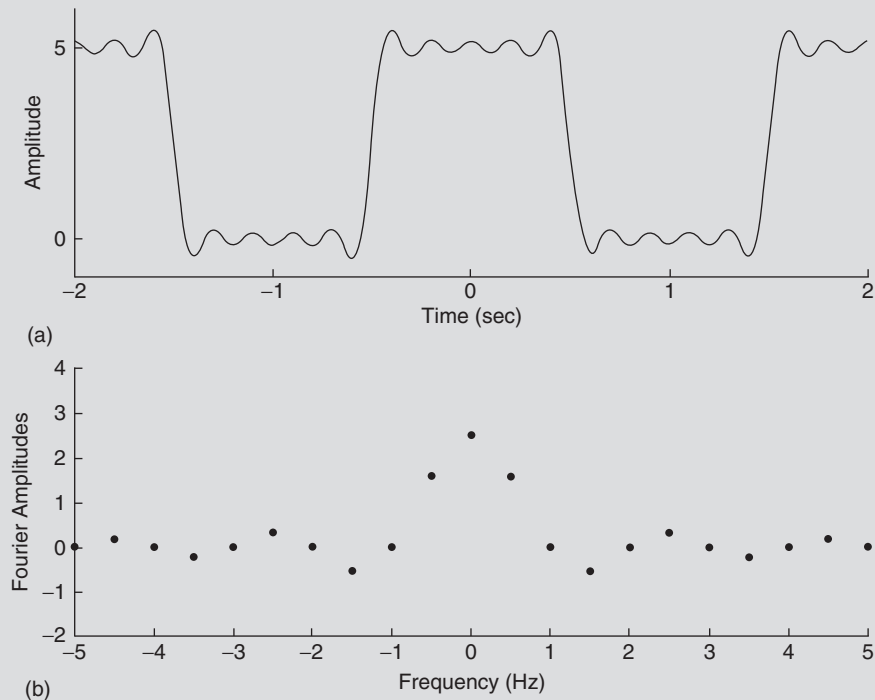


FIGURE 11.9 (a) MATLAB result showing the first ten terms of exponential Fourier series approximation for the square wave. (b) The compact Fourier coefficients are shown as a function of the harmonic frequency. Note that both negative and positive frequencies are now necessary to approximate the square wave signal.

In practice, many periodic or quasi-periodic biological signals can be accurately approximated with only a few harmonic components. [Figures 11.10](#) and [11.11](#) illustrate a harmonic reconstruction of an aortic pressure waveform obtained by applying a Fourier series approximation. [Figure 11.10](#) plots the coefficients for the cosine series representation as a function of the harmonic number. Note that the low-frequency coefficients are large in amplitude, whereas the high-frequency coefficients contain little energy and do not contribute substantially to the reconstruction. The amplitude coefficients, A_m , are plotted on a \log_{10} scale so the smaller values are magnified and are therefore visible. [Figure 11.11](#) shows several levels of harmonic reconstruction. The mean plus the first and second harmonics provide the basis for the general systolic and diastolic shape, since the amplitudes of these harmonics are large and contribute substantially to the reconstructed waveform. Additional harmonics add fine details but do not contribute significantly to the raw waveform.

11.5.4 Fourier Transform

In many instances, conceptualizing a signal in terms of its contributing cosine or sine functions has various advantages. The concept of frequency domain is an abstraction that

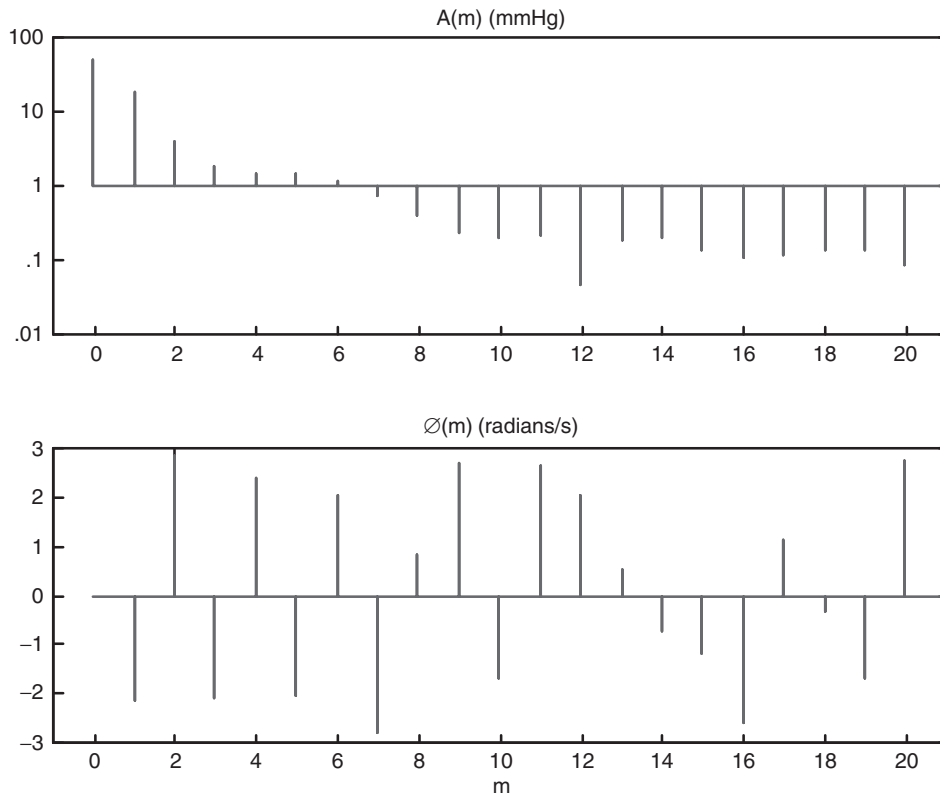


FIGURE 11.10 Harmonic coefficients of the aortic pressure waveform shown in Figure 11.2.

is borne out of the Fourier series representation for a periodic signal. A signal can be expressed either in the “time-domain” by the signal’s time function, $x(t)$, or alternatively in the “frequency-domain” by specifying the Fourier coefficient and phase, A_m and ϕ_m , as a function of the signal’s harmonic frequencies, $\omega_m = m\omega_0$. Thus, if we know the Fourier coefficients and the frequency components that make up the signal, we can fully recover the periodic signal $x(t)$.

One of the disadvantages of the Fourier series is that it applies only to periodic signals, and many biological signals are not periodic. In fact, a broad class of biological signals includes signals that are continuous functions of time but that never repeat in time. Luckily, the concept of Fourier series can also be extended for signals that are not periodic. The Fourier integral, also referred to as the Fourier transform, is used to decompose a continuous aperiodic signal into its constituent frequency components

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (11.6)$$

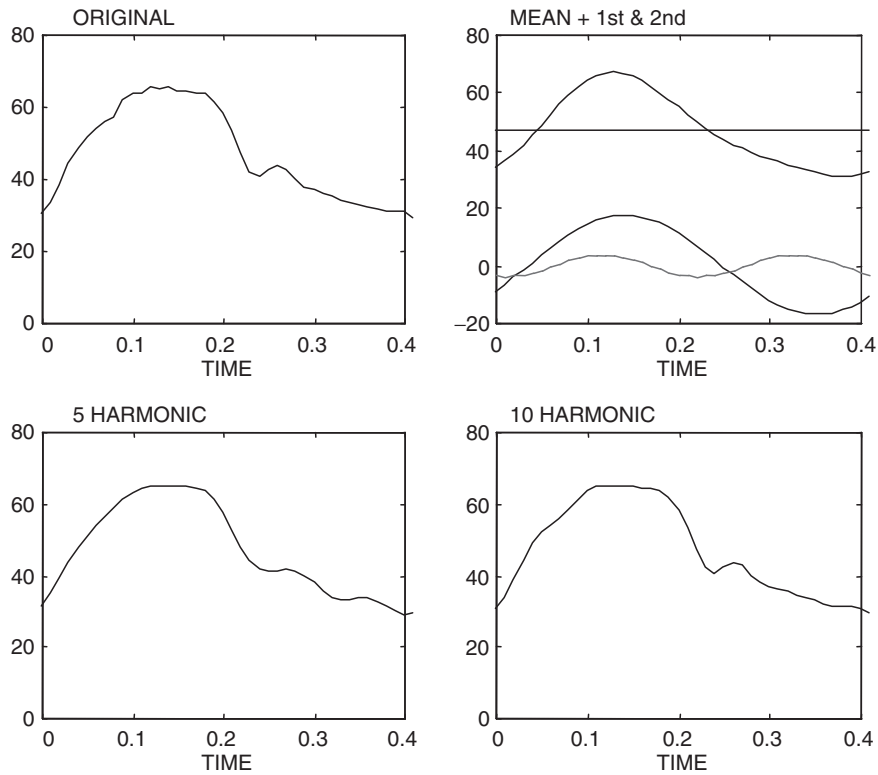


FIGURE 11.11 Harmonic reconstruction of the aortic pressure waveform shown in Figure 11.2.

much as the Fourier series decomposes a periodic signal into its corresponding trigonometric components. $X(\omega)$ is a complex valued function of the continuous frequency, ω , and is analogous to the coefficients of the complex Fourier series, c_m . A rigorous proof for this relationship is beyond the scope of this text, but it is useful to note that the Fourier integral is derived directly from the exponential Fourier series by allowing the period, T , to approach infinity. The coefficients c_m of the trigonometric series approach $X(\omega)$ as $T \rightarrow \infty$. Conceptually, a function that repeats at infinity can be considered as aperiodic, since you will never observe it repeating. Tables of Fourier transforms for many common signals can be found in most signals and systems or signal processing textbooks.

As for the Fourier series, a procedure for converting the frequency-domain version of the signal, $X(\omega)$, to its time-domain expression is desired. The time-domain signal, $x(t)$, can be completely recovered from the Fourier transform with the inverse Fourier transform (IFT)

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega. \quad (11.7)$$

These two representations of a signal are interchangeable, meaning that we can always go back and forth between the time-domain version of the signal, $x(t)$, and the frequency-domain

version obtained with the Fourier transform, $X(\omega)$. The frequency domain expression therefore provides all of the necessary information for the signal and allows one to analyze and manipulate biological signals from a different perspective.

EXAMPLE PROBLEM 11.9

Find the Fourier Transform (FT) of the rectangular pulse signal

$$x(t) = 1, |t| < a$$

$$0, |t| > a$$

Solution

Equation (11.6) is used.

$$X(\omega) = \int_{-a}^a e^{-j\omega t} dt = \left. \frac{e^{-j\omega t}}{-j\omega} \right|_{-a}^a = \frac{2 \sin \omega a}{\omega}$$

As for the Fourier series representation of a signal, the magnitude and the phase are important attributes of the Fourier transform. As stated previously, $X(\omega)$ is a complex valued function, meaning that it has a real, $\text{Re}\{X(\omega)\}$, and imaginary, $\text{Im}\{X(\omega)\}$, component and can be expressed as

$$X(\omega) = \text{Re}\{X(\omega)\} + j\text{Im}\{X(\omega)\}. \quad (11.8)$$

As for the Fourier series, the magnitude determines the amplitude of each complex exponential function (or equivalent cosine) required to reconstruct the desired signal, $x(t)$, from its Fourier transform

$$|X(\omega)| = \sqrt{\text{Re}\{X(\omega)\}^2 + \text{Im}\{X(\omega)\}^2} \quad (11.9)$$

In contrast, the phase determines the time shift of each cosine signal relative to a reference of time zero. It is determined as

$$\theta(\omega) = \tan^{-1} \left(\frac{\text{Im}\{X(\omega)\}}{\text{Re}\{X(\omega)\}} \right). \quad (11.10)$$

Note the close similarity for determining the magnitude and phase from the trigonometric and compact forms of the Fourier series (Eqs. (11.4a, b, c)). The magnitude of the Fourier transform, $|X(\omega)|$, is analogous to A_m , whereas a_m and b_m are analogous to $\text{Re}\{X(\omega)\}$ and $\text{Im}\{X(\omega)\}$, respectively. The equations are identical in all other respects.

EXAMPLE PROBLEM 11.10

Find the magnitude and phase of the signal with the Fourier transform

$$X(\omega) = \frac{1}{1 + j\omega}$$

Continued

Solution

The signal has to be put in a recognizable form similar to Eq. (11.8). To achieve this,

$$X(\omega) = \frac{1}{1+j\omega} \cdot \frac{1-j\omega}{1-j\omega} = \frac{1-j\omega}{1+\omega^2} = \frac{1}{1+\omega^2} - j \frac{\omega}{1+\omega^2}.$$

Therefore

$$\operatorname{Re}\{X(\omega)\} = \frac{1}{1+\omega^2} \quad \text{and} \quad \operatorname{Im}\{X(\omega)\} = -\frac{\omega}{1+\omega^2}.$$

Using Eqs. (11.9) and (11.10), the magnitude is

$$|X(\omega)| = \frac{1}{1+\omega^2}$$

and the phase

$$\theta(\omega) = \tan^{-1}(-\omega) = -\tan^{-1}(\omega).$$

11.5.5 Properties of the Fourier Transform

In practice, computing Fourier transforms for complex signals may be somewhat tedious and time consuming. When working with real-world problems, it is therefore useful to have tools available that help simplify calculations. The FT has several properties that help simplify frequency domain transformations. Some of these are summarized following.

Let $x_1(t)$ and $x_2(t)$ be two signals in the time domain. The FTs of $x_1(t)$ and $x_2(t)$ are represented as $X_1(\omega) = F\{x_1(t)\}$ and $X_2(\omega) = F\{x_2(t)\}$.

Linearity

The Fourier transform is a linear operator. Therefore, for any constants a_1 and a_2 ,

$$F\{a_1x_1(t) + a_2x_2(t)\} = a_1X_1(\omega) + a_2X_2(\omega) \quad (11.11)$$

This result demonstrates that the scaling and superposition properties defined for a linear system also hold for the Fourier transform.

Time Shifting/Delay

If $x_1(t - t_0)$ is a signal in the time domain that is shifted in time, the Fourier transform can be represented as

$$F\{x_1(t - t_0)\} = X(\omega) \cdot e^{-j\omega \cdot t_0} \quad (11.12)$$

In other words, shifting a signal in time corresponds to multiplying its Fourier transform by a phase factor, $e^{-j\omega t_0}$.

Frequency Shifting

If $X_1(\omega - \omega_0)$ is the Fourier transform of a signal, shifted in frequency, the inverse Fourier transform is

$$F^{-1}\{X_1(\omega - \omega_0)\} = x(t) \cdot e^{-j\omega_0 t} \quad (11.13)$$

Convolution Theorem

The convolution between two signals, $x_1(t)$ and $x_2(t)$, in the time domain is defined as

$$c(t) = \int_{-\infty}^{\infty} x_1(\tau)x_2(t - \tau)d\tau = x_1(t) * x_2(t) \quad (11.14)$$

where $*$ is shorthand for the convolution operator. The convolution has an equivalent expression in the frequency domain

$$C(\omega) = F\{c(t)\} = F\{x_1(t) * x_2(t)\} = X_1(\omega) \cdot X_2(\omega). \quad (11.15)$$

Convolution in the time domain, which is relatively difficult to compute, is a straightforward multiplication in the frequency domain.

Next, consider the convolution of two signals, $X_1(\omega)$ and $X_2(\omega)$, in the frequency domain. The convolution integral in the frequency domain is expressed as

$$X(\omega) = \int_{-\infty}^{\infty} X_1(v)X_2(\omega - v)dv = X_1(\omega) * X_2(\omega) \quad (11.16)$$

It can be shown that the inverse Fourier transform (IFT) of $X(\omega)$ is

$$x(t) = F^{-1}\{X(\omega)\} = F^{-1}\{X_1(\omega) * X_2(\omega)\} = 2\pi \cdot x_1(t) \cdot x_2(t) \quad (11.17)$$

Consequently, the convolution of two signals in the frequency domain is 2π times the product of the two signals in the time domain. As we will see subsequently for linear systems, convolution is an important mathematical operator that fully describes the relationship between the input and output of a linear system.

EXAMPLE PROBLEM 11.11

What is the FT of $3 \sin(25t) + 4 \cos(50t)$? Express your answer only in a symbolic equation. Do not evaluate the result.

Solution

$$F\{3 \sin(25t) + 4 \cos(50t)\} = 3F\{\sin(25t)\} + 4F\{\cos(50t)\}$$

11.5.6 Discrete Fourier Transform

In digital signal applications, continuous biological signals are first sampled by an analog-to-digital converter (see [Figure 11.4](#)) and then transferred to a computer, where they can be further analyzed and processed. Since the Fourier transform applies only to continuous signals of time, analyzing discrete signals in the frequency domain requires that we first modify the Fourier transform equations so they are structurally compatible with the digital samples of a continuous signal.

The discrete Fourier transform (DFT)

$$X(m) = \sum_{k=0}^{N-1} x(k) e^{-j \frac{2\pi mk}{N}}; m = 0, 1, \dots, N/2 \quad (11.18)$$

provides the tool necessary to analyze and represent discrete signals in the frequency domain. The DFT is essentially the digital version of the Fourier transform. The index m represents the digital frequency index, $x(k)$ is the sampled approximation of $x(t)$, k is the discrete time variable, N is an even number that represents the number of samples for $x(k)$, and $X(m)$ is the DFT of $x(k)$.

The inverse discrete Fourier transform (IDFT) is the discrete-time version of the inverse Fourier transform. The inverse discrete Fourier transform (IDFT) is represented as

$$x(k) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) e^{j \frac{2\pi mk}{N}}; k = 0, 1, \dots, N-1 \quad (11.19)$$

As for the FT and IFT, the DFT and IFT represent a Fourier transform pair in the discrete domain. The DFT allows one to convert a set of digital time samples to its frequency domain representation. In contrast, the IDFT can be used to invert the DFT samples, allowing one to reconstruct the signal samples $x(k)$ directly from its frequency domain form, $X(m)$. These two equations are thus interchangeable, since either conveys all of the signal information.

EXAMPLE PROBLEM 11.12

Find the discrete Fourier transform of the signal $x(k) = 0.25^k$ for $k = 0:15$

$$X(m) = \sum_{k=0}^{N-1} x(k) e^{-j \frac{2\pi mk}{N}} = \sum_{k=0}^{15} 0.25^k e^{-j \frac{2\pi mk}{16}} = \sum_{k=0}^{15} \left(0.25 \cdot e^{-j \frac{2\pi m}{N}} \right)^k = \sum_{k=0}^{15} a^k$$

Note that the preceding is a geometric sum in which $a = 0.25 \cdot e^{-j \frac{2\pi m}{N}}$. Since for a geometric sum

$$\sum_{k=M}^N a^k = \frac{a^{N+1} - a^M}{a - 1}$$

we obtain

$$X(m) = \frac{a^{16} - a^0}{a - 1} = \frac{0.25^{16} e^{-j \frac{32m\pi}{N}} - 1}{0.25 e^{-j \frac{2m\pi}{N}} - 1}$$

An efficient computer algorithm for calculating the DFT is the fast Fourier transform (FFT). The output of the FFT and DFT algorithms are the same, but the FFT has a much faster execution time than the DFT (proportional to $N \cdot \log_2(N)$ versus N^2 operations). The ratio of computing time for the DFT and FFT is therefore

$$\frac{\text{DFT computing time}}{\text{FFT computing time}} = \frac{N^2}{N \cdot \log_2 N} = \frac{N}{\log_2 N} \quad (11.20)$$

In order for the FFT to be efficient, the number of data samples, N , must be a power of two. If $N = 1024$ signal samples, the FFT algorithm is approximately $1024/\log_2(1024) = 10$ times faster than the direct DFT implementation. If N is not a power of two, alternate DFT algorithms are usually used.

Figures 11.12 and 11.13 show two signals and the corresponding DFT, which was calculated using the FFT algorithm. The signal shown in Figure 11.12a is a sine wave

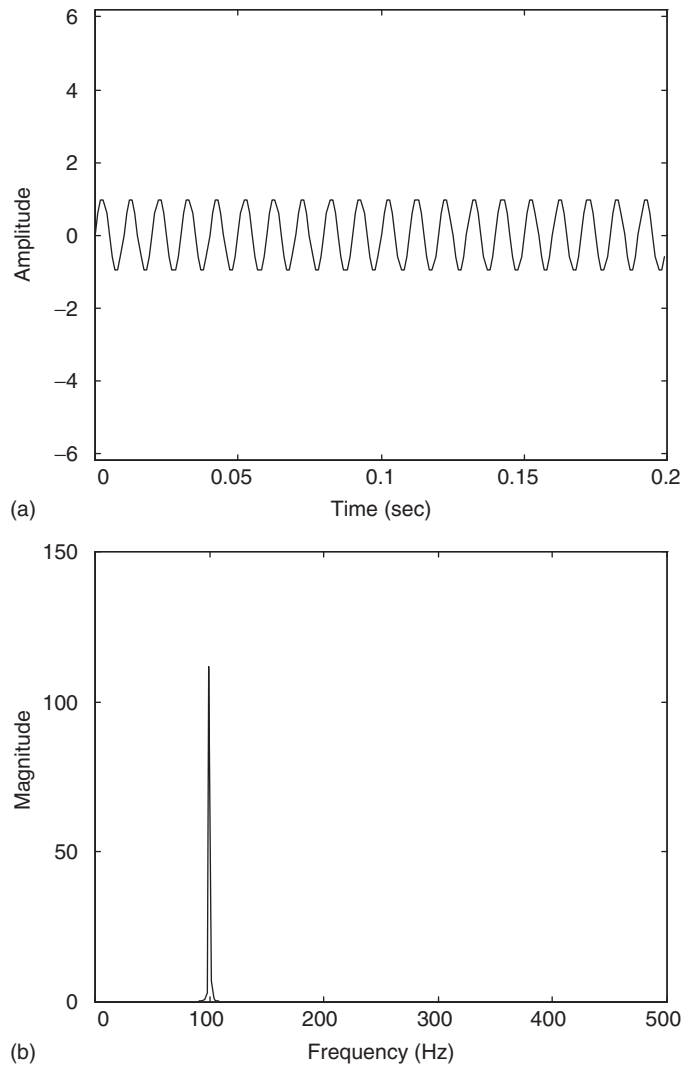


FIGURE 11.12 (a) 100 Hz sine wave. (b) Fast Fourier transform (FFT) of 100 Hz sine wave.

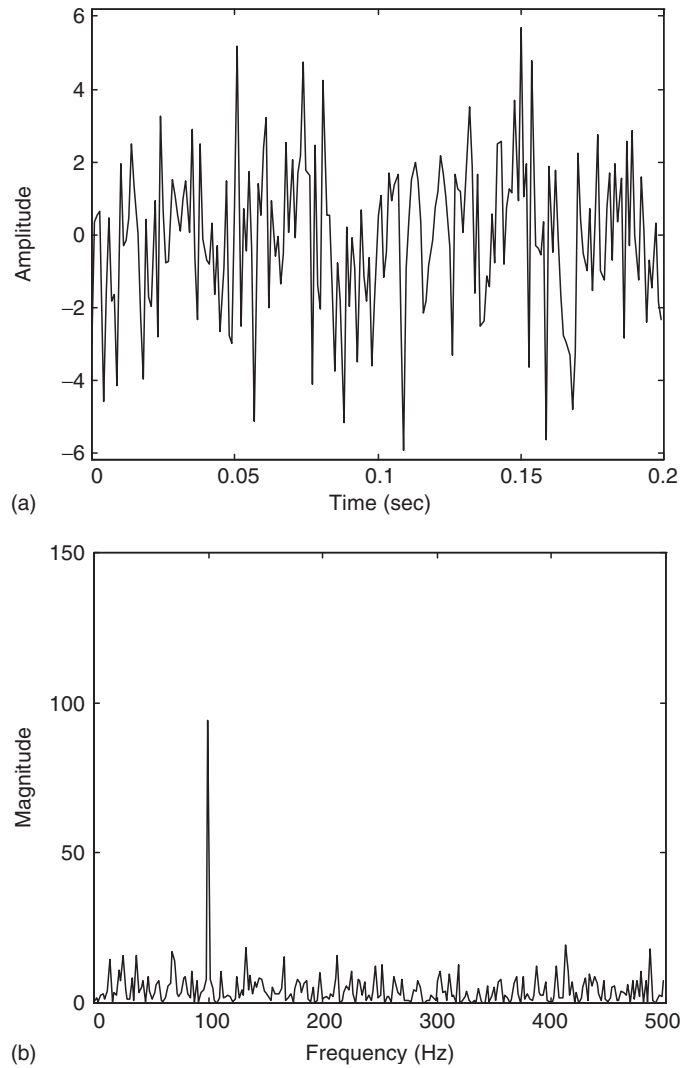


FIGURE 11.13 (a) 100 Hz sine wave corrupted with noise. (b) Fast Fourier transform (FFT) of noisy 100 Hz sine wave.

of frequency of 100 Hz. [Figure 11.12b](#) shows the FFT of the 100 Hz sine wave. Notice that the peak of the FFT occurs at 100 Hz frequency, indicating that all of the energy is confined to this frequency. [Figure 11.13a](#) shows a 100 Hz sine wave corrupted with random noise that was added to the waveform. The frequency of the signal is not distinct in the time domain. After transforming this signal to the frequency domain, the signal ([Figure 11.13b](#)) reveals a definite component at 100 Hz frequency, which is marked by the large peak in the FFT.

EXAMPLE PROBLEM 11.13

Find and plot the magnitude of the discrete Fourier transform of the signal $x(n) = \sin(\pi/4 \cdot n) + 2 \cdot \cos(\pi/3 \cdot n)$ in MATLAB.

Solution

```
n=1:1024; %Discrete Time Axis
x=sin(pi/4*n)+2*cos(pi/3*n); %Generating the signal
X=fft(x,1024*16)/1024; %Computing 16k point Fast Fourier Transform
Freq=(1:1024*16)/(1024*16)*2*pi; %Normalizing Frequencies
between 0-2*pi
plot(Freq,abs(X),'k') %Plotting
axis([0.7 1.15 0 1.2])
xlabel('Frequency (rad/s)')
ylabel('Fourier Magnitude')
```

Results are shown in [Figure 11.14](#).

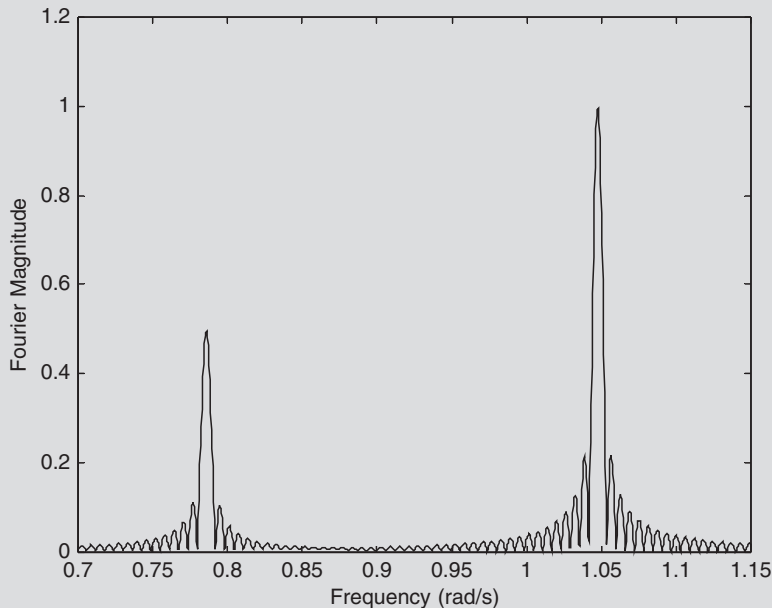


FIGURE 11.14 Fast Fourier transform magnitude for the sum of two sinusoids. Dominant energy peaks are located at the signal frequencies $\pi/3$ and $\pi/4$ rad/s.

11.5.7 The z -Transform

The z -transform provides an alternative tool for analyzing discrete signals in the frequency domain. This transform is essentially a variant of the DFT, where we allow $z = e^{-j\frac{2\pi m}{N}}$.

In most applications, the z -transform is somewhat easier to work with than the DFT because it does not require the use of complex numbers directly. The z -transform plays a similar role for digital signals as the Laplace transform does for the analysis of continuous signals.

If a discrete sequence $x(k)$ is represented by x_k , the (one-sided) z -transform of the discrete sequence is expressed by

$$X(z) = \sum_{k=0}^{\infty} x_k z^{-k} = x_0 + x_1 z^{-1} + x_2 z^{-2} + \dots + x_k z^{-k} \quad (11.21)$$

Note that the z -transform can be obtained directly from the DFT by allowing $N \rightarrow \infty$ and replacing $z = e^{-j\frac{2\pi m}{N}}$ in Eq. (11.18). In most practical applications, sampled biological signals are represented by a data sequence with N samples so the z -transform is estimated for $k = 0 \dots N-1$ only. Tables of common z -transforms and their inverse transforms can be found in most digital signal processing textbooks.

After a continuous signal has been sampled into a discrete sequence, its z -transform is found quite easily. Since the data sequence of a sampled signal is represented as

$$\mathbf{x} = [x(0), x(T), x(2T), \dots, x(kT)] \quad (11.22)$$

its z -transform is obtained by applying Eq. (11.21) to its samples

$$X(z) = x(0) + x(T)z^{-1} + x(2T)z^{-2} + \dots + x(kT)z^{-k} \quad (11.23)$$

where T is the sampling period or sampling interval.

A sampled signal is a data sequence with each sample separated from its neighboring samples by precisely one sampling period. In the z -transform, the value of the multiplier, $x(kT)$, is the value of the data sample. The terms z^{-k} have an intuitive graphical explanation. The power, k , corresponds to the number of sampling periods following the start of the sampling process at time zero; z^{-k} can therefore be thought of as a “shift operator” that delays the sample by exactly k sampling periods or kT . The variable z^{-1} , for instance, represents a time separation of one period, T , following the start of the signal at time zero. In Eq. (11.18), $z(0)$ is the value of the sampled data at $t = 0$, and $x(T)$ is the value of the sampled data that was obtained after the first sampling period. The z -transform is an important method for describing the sampling process of an analog signal.

EXAMPLE PROBLEM 11.14

The discrete unit impulse function is represented as the sequence $\mathbf{x} = [1, 0, 0, 0, \dots, 0]$. Find the z -transform of this sequence.

Solution

$$X(z) = 1 + 0z^{-1} + 0z^{-2} + \dots + 0z^{-k} = 1 + 0 + 0 + \dots + 0 = 1$$

EXAMPLE PROBLEM 11.15

An A/D converter is used to convert a recorded signal of the electrical activity inside a nerve into a digital signal. The first five samples of the biological signal are $[-60.0, -49.0, -36.0, -23.0, -14.0]$ mV. What is the z -transform of this data sequence? How many sample periods after the start of the sampling process was the data sample -23.0 recorded?

Solution

$$Y(z) = -60.0 - 49.0z^{-1} - 36.0z^{-2} - 23.0z^{-3} - 14.0z^{-4}$$

The value of the negative exponent of the -23.0 mV z -term is 3. Therefore, the data sample with the value of -23.0 was recorded 3 sampling periods after the start of sampling.

11.5.8 Properties of the z -Transform

The z -transform obeys many of the same rules and properties that we've already shown for the Fourier transform. These properties can significantly simplify the process of evaluating z -transforms for complex signals. The following are some of the properties of the z -transform. Note the close similarity to the properties for Eqs. (11.11), (11.12), and (11.14). Let $x_1(k)$ and $x_2(k)$ be two digital signals with corresponding z -transforms $X_1(z)$ and $X_2(z)$.

Linearity:

The z -transform is a linear operator. For any constants a_1 and a_2 ,

$$Z\{a_1x_1(k) + a_2x_2(k)\} = \sum_{k=0}^{\infty} [a_1x_1(k) + a_2x_2(k)]z^{-k} = a_1X_1(z) + a_2X_2(z) \quad (11.24)$$

Delay:

Let $x_1(k - n)$ be the original signal that is delayed by n samples. The z -transform of the delayed signal is

$$Z\{x_1(k - n)\} = \sum_{k=0}^{\infty} x_1(k - n)z^{-k} = \sum_{k=0}^{\infty} x_1(k)z^{-(k+n)} = z^{-n}X_1(z) \quad (11.25)$$

As described previously, note that the operator z^{-n} represents a shift of n samples or precisely nT seconds.

Convolution:

Let $x(k)$ be the discrete convolution between $x_1(k)$ and $x_2(k)$,

$$x(k) = x_1(k) * x_2(k)$$

$X(z)$, the z -transform of $x(k)$, is calculated as

$$X(z) = Z\{x(k)\} = X_1(z)X_2(z) \quad (11.26)$$

As with the Fourier transform, this result demonstrates that convolution between two sequences is performed by simple multiplication in the z -domain.

11.6 LINEAR SYSTEMS

A system is a process, machine, or device that takes a signal as an *input* and manipulates it to produce an *output* that is related to, but is distinctly different from, its input. Figure 11.15 shows a system block diagram.

Biological systems and organs are very often modeled as systems. The heart, for instance, is a large-scale system that takes oxygen-deficient blood from the veins (the *input*) and pumps it through the lungs. This produces a blood *output* via the main arteries of the heart that is rich in oxygen content. Neurons in the brain can also be thought of as a simple microscopic system that takes electrical nerve impulses from various neurons as the input and sums these impulses to produce action potentials: the output. Linear systems are a special class of systems with a unique set of properties that make them easy to analyze.

11.6.1 Linear System Properties

While biological systems are not per se linear, very often they can be approximated by a linear system model. This is desired because it makes their analysis and the subsequent interpretation more tractable.

All linear systems are characterized by the principles of *superposition* (or *additivity*) and *scaling*. The superposition property states that the sum of two independent inputs produces an output that is the sum or superposition of the outputs for each individual input. The scaling property tells us that a change in the size of the input produces a comparable change at the output. Mathematically, if we know the outputs for two separate inputs—that is,

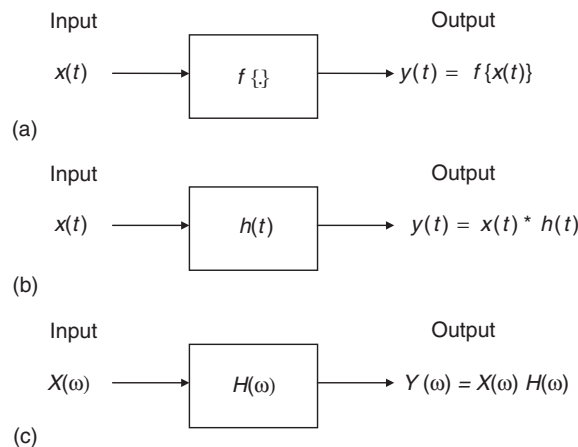


FIGURE 11.15 (a) Block diagram representation of a system. The input signal, $x(t)$, passes through the system transformation $f\{\cdot\}$ to produce an output, $y(t)$. (b) Time domain representation of a linear system. The output of the linear system is represented by the convolution of the input and impulse response. (c) Frequency domain representation of a linear system. The output corresponds to the product of the input and the system transfer function.

Input		Output
$x_1(t)$	\rightarrow	$y_1(t)$
$x_2(t)$	\rightarrow	$y_2(t),$

we can easily determine the output of a linear system to any arbitrary combination of these inputs. More generally, a linear superposition and scaling of the input signals produces a linear superposition and scaling of the output signals

$$\begin{array}{ccc} \text{Input} & & \text{Output} \\ k_1 \cdot x_1(t) + k_2 \cdot x_2(t) & \rightarrow & k_1 \cdot y_1(t) + k_2 \cdot y_2(t) \end{array} \quad (11.27)$$

where k_1 and k_2 are arbitrary amplitude scaling constants. These constants scale the input amplitudes by making them larger ($k > 1$) or smaller ($k < 1$). This produces a comparable change in the net outputs, which are likewise scaled by the same constants.

EXAMPLE PROBLEM 11.16

The following information is given for a linear system

Input		Output
$x_1(t) = \cos(t)$	\rightarrow	$y_1(t) = \cos(t + \pi/2)$
$x_2(t) = \cos(t) + \sin(2t)$	\rightarrow	$y_2(t) = \cos(t + \pi/2) + 5 \sin(2t)$
$x_3(t) = \cos(3t)$	\rightarrow	$y_3(t) = 2 \cos(3t)$

Find the output if the input is: $x(t) = 3 \sin(2t) + 1/2 \cos(3t)$

Solution

The input is represented as a superposition of x_1 , x_2 , and x_3

$$x(t) = 3(x_2(t) - x_1(t)) + 1/2 x_3(t) = 3x_2(t) - 3x_1(t) + 1/2 x_3(t)$$

Applying the superposition and scaling properties produces an output

$$\begin{aligned} y(t) &= 3y_2(t) - 3y_1(t) + 1/2 y_3(t) = 3(\cos(t + \pi/2) + 5 \sin(2t)) - 3(\cos(t + \pi/2)) + 1/2(2 \cos(3t)) \\ &= 15 \cdot \sin(2t) + \cos(3t) \end{aligned}$$

EXAMPLE PROBLEM 11.17

Consider the system given by the expression

$$y(t) = f\{x(t)\} = A \cdot x(t) + B.$$

Determine if this is a linear system.

Solution

To solve this problem, consider a superposition of two separate inputs, $x_1(t)$ and $x_2(t)$, that independently produce outputs $y_1(t)$ and $y_2(t)$. Apply the input $x(t) = k_1 \cdot x_1(t) + k_2 \cdot x_2(t)$. If the system is linear, the output obeys

Continued

$$\begin{aligned} y_{Lin}(t) &= k_1 \cdot y_1(t) + k_2 \cdot y_2(t) = k_1 \cdot (A \cdot x_1(t) + B) + k_2 \cdot (A \cdot x_2(t) + B) \\ &= A(k_1 \cdot x_1(t) + k_2 \cdot x_2(t)) + (k_1 + k_2)B \end{aligned}$$

The true system output, however, is determined as

$$y(t) = f\{x(t)\} = f\{k_1 \cdot x_1(t) + k_2 \cdot x_2(t)\} = A \cdot (k_1 \cdot x_1(t) + k_2 \cdot x_2(t)) + B$$

We need to compare our expected linear system output, $y_{lin}(t)$, with the true system output, $y(t)$. Note that $y(t) \neq y_{lin}(t)$, and therefore the system is not linear.

The superposition principle takes special meaning when applied to periodic signals. Because periodic signals are expressed as a sum of cosine or complex exponential functions with the Fourier series, their output must also be expressed as a sum of cosine or exponential functions. Thus, if a linear system is stimulated with a periodic signal, its output is also a periodic signal with identical harmonic frequencies. The output, $y(t)$, of a linear system to a periodic input, $x(t)$, is related by

$$\begin{array}{ccc} \text{Input} & & \text{Output} \\ x(t) = \frac{A_0}{2} + \sum_{m=-\infty}^{+\infty} A_m \cos(m\omega_0 t + \phi_m) & \Rightarrow & x(t) = \frac{B_0}{2} + \sum_{m=-\infty}^{+\infty} B_m \cos(m\omega_0 t + \theta_m) \end{array} \quad (11.28)$$

meaning the input and output contain cosines with identical frequencies, $m\omega_0$, and are expressed by equations with similar form. A similar form of this expression is also obtained for the exponential Fourier series:

$$\begin{array}{ccc} \text{Input} & & \text{Output} \\ x(t) = \sum_{m=-\infty}^{+\infty} c_m e^{jm\omega_0 t} & \Rightarrow & y(t) = \sum_{m=-\infty}^{+\infty} b_m e^{jm\omega_0 t} \end{array} \quad (11.29)$$

where the input and output coefficients, c_m and b_m , are explicitly related to A_m and B_m via Eq. (11.5b).

From Eqs. (11.28) and (11.29) the input and output of a linear system to a periodic input differ in two distinct ways. First the amplitudes of each cosine are selectively scaled by different constants, A_m for the input and B_m for the output. These constants are uniquely determined by the linear system properties. Similarly, the phases angle of the input components, ϕ_m , are different from the output components, θ_m , meaning that the input and output components are shifted in time in relationship to each other. As for the amplitudes, the phase difference between the input and output is a function of the linear system. Thus, if we know the mathematical relationship of how the input components are amplitude scaled and phase shifted between the input and output, we can fully describe the linear system. This relationship is described by the system *transfer function*, H_m . The transfer function fully describes how the linear system manipulates the amplitude and phases of the input to produce a specific output. This transformation is described by two separate components: the *magnitude* and the *phase*.

The magnitude of H_m is given by the ratio of the output to the input for the m -th component

$$|H_m| = \frac{B_m}{A_m} \quad (11.30)$$

Note that if we know the input magnitudes, we can determine the output Fourier coefficients by multiplying the transfer function magnitude by the input Fourier coefficients: $B_m = |H_m| \cdot A_m$. The phase angle of the transfer function describes the phase relationship between the input and output for the m -th frequency component

$$\angle H_m = \theta_m - \phi_m. \quad (11.31)$$

If we know the input phase, the output phase is determined as $\theta_m = \angle H_m + \phi_m$. Equations (11.30) and (11.31) are the two critical pieces of information that are necessary to fully describe a linear system. If these two properties of the transformation are known, it is possible to determine the output for any arbitrary input.

11.6.2 Time Domain Representation of Linear Systems

The relationship between the input and output of a linear system can be described by studying its behavior in the time domain (Figure 11.15b). The *impulse response* function, $h(t)$, is a mathematical description of the linear system that fully characterizes its behavior. As we will see subsequently, the impulse response of a linear system is directly related to the system transfer function as outlined for the periodic signal. If one knows $h(t)$, one can readily compute the output, $y(t)$, to any arbitrary input, $x(t)$, using the *convolution integral*

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau)d\tau. \quad (11.32)$$

The symbol $*$ is shorthand for the convolution between the input and the system impulse response. Integration is performed with respect to the dummy integration variable τ . For the discrete case, the output of a discrete linear system is determined with the convolution sum

$$y(k) = h(k) * x(k) = \sum_{m=-\infty}^{\infty} h(m)x(k - m) \quad (11.33)$$

where $h(m)$ is the impulse response of the discrete system. A detailed treatment of the convolution integral is found in many signal processing textbooks and is beyond the scope of this text. As shown in a subsequent section, a simpler treatment of the input-output relationship of a linear system is obtained by analyzing it in the “frequency-domain.”

EXAMPLE PROBLEM 11.18

A cytoplasmic current injection $i(t) = u(t)$ to a cell membrane produces an intracellular change in the membrane voltage, $v(t)$. The membrane of a cell is modeled as a linear system with impulse response $h(t) = A \cdot e^{-t/\tau} \cdot u(t)$, where A is a constant in units V/s/A and τ is the cell membrane time constant (units: seconds). Find the cell membrane voltage output in closed form. Simulate in MATLAB, for $A = 100$ and $\tau = 0.01$ sec. Compare the closed form solution to the simulated results.

Continued

Solution

The input, $i(t)$, and output, $v(t)$, are related by the convolution integral (Eq. (11.32)):

$$v(t) = h(t) * i(t) = \int_{-\infty}^{\infty} h(\varsigma) i(t - \varsigma) d\varsigma = \int_{-\infty}^{\infty} A \cdot e^{-\varsigma/\tau} u(\varsigma) u(t - \varsigma) d\varsigma$$

where we use a dummy integration variable, ς , to distinguish it from the cell time constant, τ . The unit step functions inside the integral take values of one or zero—in which case they do not contribute to the integral. $u(\varsigma) = 1$ if $\varsigma > 0$ and $u(t - \varsigma) = 1$ if $t - \varsigma > 0$. Combining these two inequalities, we have that $u(t - \varsigma) \cdot u(\varsigma) = 1$ whenever $0 < \varsigma < t$, and we can therefore change the limits of integration and replace the unit step function with 1:

$$v(t) = \int_0^t A \cdot e^{-\varsigma/\tau} d\varsigma = A \cdot \tau \cdot (1 - e^{-t/\tau}).$$

MATLAB Solution

```
%Theoretical Result
dt=0.0001; %Sampling Resolution
Fs=1/dt; %Sampling Rate
time=0:dt:5; %5-second time-axis
Tau=0.1; %Cell Membrane Time Constant
A=10;
y=A*Tau*(1-exp(-time/Tau)); %Closed Form Output Equation
plot(time,y,'color',[.75 .75 .75],'linewidth',3) %Plotting
Theoretical Output
hold on

%Simulated Output
h=A*exp(-time/Tau); %Impulse Response
x=[zeros(1,Fs) ones(1,Fs*4)]; %Step Input
y=conv(h,x)*dt; %Step Response, obtained by convolving: y=h*x

%Plotting Simulated Results
time=(0:length(x)-1)*dt-1;
plot(time,x,'k-.') %Plotting Input
hold on
time=(0:length(y)-1)*dt-1;
plot(time,y,'k') %Plotting Output
axis([-1 1 0 1.2])
xlabel('time (sec)','fontsize',14)
ylabel('Amplitude','fontsize',14)
set(gca,'XTick',[-1 -0.5 0 0.5 1])
set(gca,'YTick',[0:.2:1.2])
```

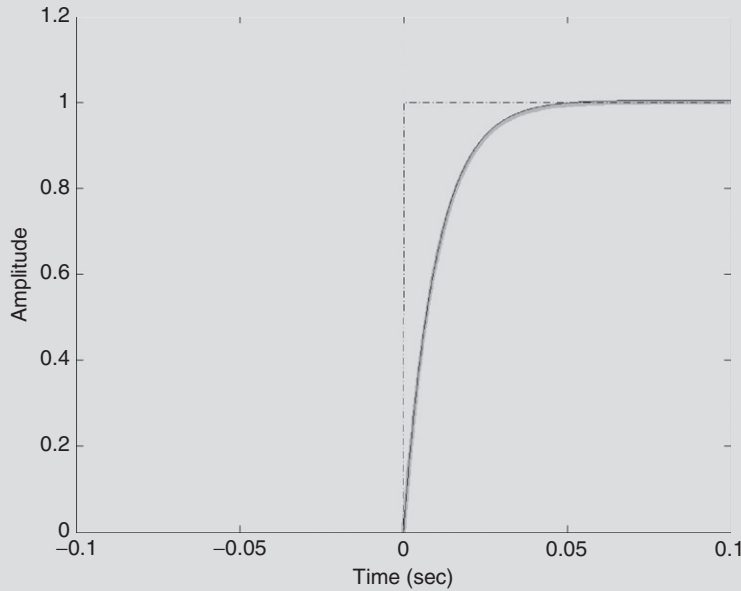


FIGURE 11.16 MATLAB results showing the step response of a linear cell membrane. Dotted lines represent the step input. Continuous gray line shows the theoretical step response. Black line shows the simulated step response. Note that the theoretical solution and simulated results are superimposed.

The results for the cell membrane step response are shown in [Figure 11.16](#). Note that the simulated (black) and theoretical (gray) outputs are precisely matched.

11.6.3 Frequency Domain Representation of Linear Systems

We have already considered the special case of linear systems in the frequency domain for periodic inputs. Recall that the system output of a linear system to a periodic stimulus is fully described by the system transfer function. The output of a linear system is also expressed in the time domain by the convolution integral ([Figure 11.15b](#)). The impulse response is the mathematical model that describes the linear system in the time domain. These two descriptions for the input-output relationship of a linear system are mutually related. Notably, for the aperiodic signal case, the transfer function is the Fourier transform of the impulse response

$$\begin{array}{ccc} \text{Time Domain} & & \text{Frequency Domain} \\ h(t) & \Leftrightarrow & H(\omega) \end{array} \quad (11.34)$$

where $H(\omega)$ is the system transfer function. Since the impulse response is a complete model of a linear system and since the Fourier transform is invertible (we can always go back and forth between $h(t)$ and $H(\omega)$), the transfer function contains all the necessary information to fully describe the system. The advantage of the transfer function comes in its simplicity of use. Rather than performing a convolution integral, which can be quite intricate in many

applications, the output of a linear system in the frequency domain is expressed as a product of the system input and its transfer function: $Y(\omega) = X(\omega)H(\omega)$. This result is reminiscent of the result for the Fourier series (Eqs. (11.30) and (11.31)). Specifically, the *convolution property* of the Fourier transform states that a convolution in time corresponds to a multiplication in the frequency domain

$$\begin{array}{ccc} \text{Time Domain} & & \text{Frequency Domain} \\ y(t) = x(t) * h(t) & \Leftrightarrow & Y(\omega) = X(\omega)H(\omega) \end{array} \quad (11.35)$$

and thus the output of a linear system, $Y(\omega)$, is expressed in the frequency domain by the product of $X(\omega)$ and $H(\omega)$ (Figure 11.15c). Note that this result is essentially the convolution theorem (Eqs. (11.14) and (11.15); for proof, see Example Problem 11.19) applied to the output of a linear system (Eq. (11.32)). In many instances, this is significantly easier to compute than a direct convolution in the time domain.

EXAMPLE PROBLEM 11.19

Prove the convolution property of the Fourier transform.

Solution

$$Y(\omega) = FT\{y(t)\} = \int y(t)e^{-j\omega t} dt = \int \int x(\tau)h(t - \tau)d\tau e^{-j\omega t} dt$$

Make a change of variables, $u = t - \tau$, $du = dt$

$$= \int \int x(\tau)h(u)d\tau e^{-j\omega(u+\tau)} du = \int x(\tau)e^{-j\omega\tau} d\tau \cdot \int h(u)e^{-j\omega u} du = X(\omega)H(\omega)$$

11.6.4 Analog Filters

Filters are a special class of linear systems that are widely used to manipulate the properties of a biological signal. Conceptually, a filter allows the user to selectively remove an undesired signal component while preserving or enhancing some component of interest. Although most of us are unaware of this, various types of filters are commonplace in everyday settings. Sunblock, for instance, is a type of filter that removes unwanted ultraviolet light from the sun in order to minimize the likelihood of sunburn and potentially reduce the risk of skin cancer. Filters are also found in many audio applications. Treble and bass controls in an audio system are a special class of filter that the user selectively adjusts to boost or suppress the amount of high-frequency (treble) and low-frequency (bass) sound to a desired level and quality.

Filters play an important role in the analysis of biological signals, in part because signal measurements in clinical settings are often confounded by undesirable noise. Such noise distorts the signal waveform of interest, making it difficult to obtain a reliable diagnosis. Ideally, if one could completely remove unwanted noise, one could significantly improve the quality of a signal and thus minimize the likelihood of an incorrect diagnosis.

Practically, most filters can be subdivided into three broad classes, according to how they modify the frequency spectrum of the desired signal. These broad classes include *low-pass*, *high-pass*, and *band-pass* filters. Low-pass filters work by removing high frequencies from a signal while selectively keeping the low frequencies (Figure 11.17a). This allows the low frequencies of the signal to pass through the filter uninterrupted, hence the name low-pass. In some instances, low frequencies could be accentuated further by magnifying them while selectively removing the high frequencies. High-pass filters perform exactly the opposite function of a low-pass filter (Figure 11.17b). They selectively pass the high frequencies but remove the low frequencies of the signal. The treble control in an audio system is a form of high-pass filter that accentuates the high frequencies, thus producing crisp and rich sound. In contrast, the bass control is a form of low-pass filter that selectively enhances low frequencies or the “bass,” creating a “warmer” sound quality. Band-pass filters fall somewhere in between the low-pass and high-pass filter. Rather than simply removing the low or high frequencies, band-pass filters remove both high and low frequencies but selectively keep a small “band” of frequencies (Figure 11.17c), hence the name. The function of a band-pass filter could be achieved by simply combining a low-pass and high-pass filter, as we will see subsequently.

Since filters are linear systems, the output of a filter is expressed by the convolution between the input and the filter’s impulse response (Eq. (11.32)). Conversely, if the output is determined in the frequency domain, the output corresponds to the product of the filter

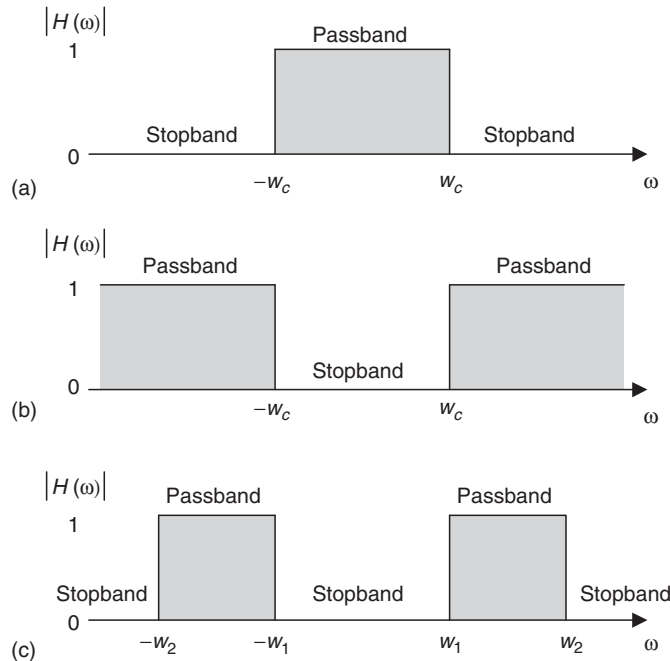


FIGURE 11.17 Frequency domain magnitude response plot, $|H(\omega)|$, of the ideal (a) low-pass filter, (b) high-pass filter, and (c) band-pass filter. Signals in the shaded region, the passband, are preserved at the output, whereas signals in the stopband are selectively removed from the output.

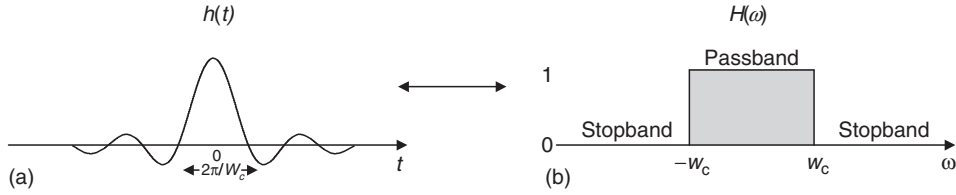


FIGURE 11.18 Time and frequency domain representation of the ideal low-pass filter. (a) The impulse response of the ideal low-pass filter, $h(t)$. (b) Transfer function of the ideal low-pass filter, $H(\omega)$.

transfer function and the input Fourier transform (Eq. (11.35)). The impulse response and transfer function of the ideal low-pass filter are shown in Figure 11.18. The transfer function of this filter takes a value of one within the filter passband and zero within the stopband.

Since the output of a linear system in the frequency domain is given as the product of the input Fourier transform and the signal transfer function, $Y(\omega) = H(\omega)X(\omega)$, any signal presented to this filter within its passband will pass through to the output uninterrupted because the frequency components are multiplied by one. In contrast, signals in the stopband are removed at the output of the filter, since the frequency components are multiplied by zero. The impulse response of the ideal low-pass analog filter is

$$h_{LP}(t) = \frac{W_c}{\pi} \text{sinc}(W_c t). \quad (11.36)$$

where $W_c = 2\pi f_c$ is the filter cutoff frequency. In the frequency domain, the ideal low-pass filter transfer function is

$$H_{LP}(\omega) = \begin{cases} 1 & |\omega| < W_c \\ 0 & |\omega| > W_c \end{cases}. \quad (11.37)$$

This dual time and frequency domain representation of the ideal low-pass filter is shown in Figure 11.18. Note that the transfer function takes a value of one only within the passband. At the cutoff frequency, the filter transfer function transitions from a value of one in the passband to a value of zero in the stopband.

In the frequency domain, the ideal high-pass filter performs the exact opposite function of the low-pass filter

$$H_{HP}(\omega) = \begin{cases} 0 & |\omega| < W_c \\ 1 & |\omega| > W_c \end{cases}, \quad (11.38)$$

that is, the passband exists for frequencies above the cutoff frequency, whereas the stopband exists for frequencies below the filter's cutoff. This filter therefore preserves high-frequency signal components (above the cutoff frequency) and selectively removes low-frequency signals. The ideal high-pass filter transfer function can be easily derived from the ideal low-pass filter as

$$H_{HP}(\omega) = 1 - H_{LP}(\omega) \quad (11.39)$$

In the time domain, the ideal high-pass filter impulse response is obtained as

$$h_{HP}(t) = \delta(t) - h_{LP}(t) = \delta(t) - \frac{W_c}{\pi} \text{sinc}(W_c t) \quad (11.40)$$

A schematic depiction of the ideal high-pass filter transfer function is shown in [Figure 11.17b](#).

The final class of filter we will consider is the band-pass filter. The prototypical band-pass filter is somewhat more complex than the low-pass and high-pass filters, since it requires the definition of a lower and upper cutoff frequency, W_1 and W_2 . [Figure 11.17c](#) illustrates the magnitude response of the ideal band-pass filter transfer function and impulse response. Only signals between the two cutoff frequencies are allowed to pass through to the output. All other signals are rejected. The transfer function of the ideal band-pass filter is given by

$$H_{BP}(\omega) = \begin{cases} 0 & W_1 < |\omega| < W_2, \\ 1 & \text{otherwise} \end{cases} \quad (11.41)$$

In the frequency domain, the ideal band-pass filter can be obtained by combining a high-pass filter with cutoff W_1 and a low-pass filter with cutoff W_2 . The band-pass filter transfer function can therefore be expressed as the product of transfer functions for a low-pass and high-pass filter:

$$H_{BP}(\omega) = H_{HP}(\omega) \cdot H_{LP}(\omega). \quad (11.42)$$

In the time domain, the band-pass filter impulse response is obtained by the inverse Fourier transform of the filter transfer function

$$h_{BP}(t) = h_{HP}(t) * h_{LP}(t). \quad (11.43)$$

This is done by applying the convolution theorem ([Eqs. \(11.14\)](#) and [\(11.15\)](#)) to [Eq. \(11.42\)](#).

EXAMPLE PROBLEM 11.20

Consider the cell membrane cytoplasmic current injection for [Example Problem 11.18](#). Find the cell's output voltage in the Fourier domain. Also, compute and plot in MATLAB the transfer function magnitude of the cell membrane. What type of filter is this?

Solution

The Fourier transform of the step current input is

$$I(\omega) = \int i(t) e^{-j\omega t} dt = \int_0^{\infty} 1 \cdot e^{-\omega t} dt = \left. \frac{e^{-\omega t}}{-j\omega} \right|_0^{\infty} = \frac{1}{j\omega}$$

The transfer function is determined as the Fourier transform of the impulse response

$$H(\omega) = \text{FT}\{h(t)\} = \int A \cdot e^{-t/\tau} \cdot u(t) dt = \int_0^{\infty} A \cdot e^{-t/\tau} e^{-j\omega t} dt = \frac{A}{j\omega + 1/\tau}$$

Continued

The cell's voltage output in the frequency domain is determined as

$$V(\omega) = H(\omega)I(\omega) = \frac{A}{j\omega + 1/\tau} \cdot \frac{1}{j\omega} = \frac{A}{j\omega} - \frac{A}{j\omega + 1/\tau}$$

Next, using the derived transfer function, the transfer function magnitude is obtained as

$$|H(\omega)| = \sqrt{H(\omega) \cdot H(\omega)^*} = \sqrt{\frac{A}{j\omega + 1/\tau} \cdot \frac{A}{-j\omega + 1/\tau}} = \frac{A}{\sqrt{\omega^2 + 1/\tau^2}}$$

where $H(\omega)^*$ is the complex conjugate transfer function. In MATLAB the transfer function magnitude is plotted as follows:

```
%Cell Membrane Magnitude Response
tau=0.01;
A=1/tau^2;
w=0:0.1:500;
H=A./(w.^2+1/tau^2);
plot(w/2/pi,H,'k')
axis([0 500/2/pi 0 1.2])
xlabel('Frequency (Hz)')
ylabel('Magnitude')
```

The results are shown in [Figure 11.19](#). Note that the transfer function tends to preserve the low frequencies while rejecting the high frequencies. Thus, the cell membrane behaves like a low-pass filter.

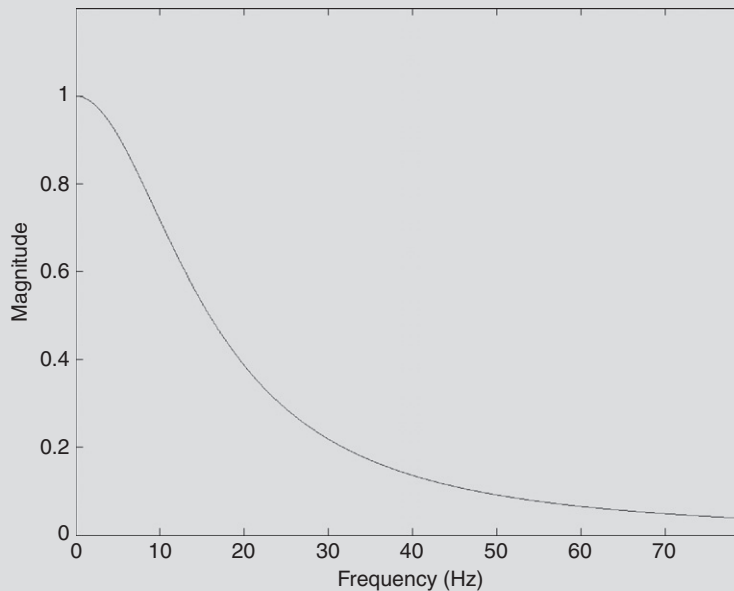


FIGURE 11.19 Cell membrane transfer function magnitude. Note that the cell behaves like a low-pass filter.

EXAMPLE PROBLEM 11.21

An electromyographic (EMG) signal contains energy within the frequencies 25 and 100 Hz. Design a filter to remove unwanted noise.

Solution

We need to design a band-pass filter with passband frequencies 25 and 100 Hz. First determine the cutoff frequencies in rad/s. Since $W_c = 2\pi f_c$,

$$W_1 = 50\pi$$

$$W_2 = 200\pi$$

Next, we find the impulse response of the corresponding low-pass and high-pass filters:

$$h_{HP}(t) = \delta(t) - \frac{W_1}{\pi} \text{sinc}(W_1 t) = \delta(t) - 50 \text{sinc}(50\pi t)$$

$$h_{LP}(t) = \frac{W_2}{\pi} \text{sinc}(W_2 t) = 200 \text{sinc}(200\pi t)$$

The band-pass filter impulse response is

$$h_{BP}(t) = h_{BP}(t) * h_{LP}(t) = [\delta(t) - 50 \text{sinc}(50\pi t)] * 200 \text{sinc}(200\pi t)$$

The described ideal analog filters provide a conceptual reference for various filter design applications. In practice, real analog filters cannot be implemented to achieve the strict specifications of the ideal filter because the impulse response of ideal filters is of infinite duration (extends from $-\infty$ to $+\infty$). Thus, the ideal filters require that one integrate over an infinite amount of time to produce an output. Typically, most analog filters are designed with simple electronic circuits. Various approximations to the ideal low-pass, high-pass, and band-pass filter can be derived that are well suited for a variety of applications, including signal analysis of biomedical signals.

11.6.5 Digital Filters

Digital systems are described by difference equations, just like analog systems are described by differential equations. Difference equations are essentially discretized differential equations that have been sampled at a particular sampling rate. The general form of a real-time digital filter/difference equation is

$$y(k) = \sum_{m=0}^M b_m x(k-m) - \sum_{m=1}^N a_m y(k-m) \quad (11.44)$$

where the discrete sequence $x(k)$ corresponds to the input and $y(k)$ represents the output sequence of the discrete system. For instance, if $M = 2$ and $N = 2$, then

$$y(k) = b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) - a_1 y(k-1) - a_2 y(k-2)$$

where $x(k)$ and $y(k)$ represent the input and output at time k , $x(k-1)$ and $y(k-1)$ represent the input and output one sample into the past, and similarly, $x(k-2)$ and $y(k-2)$ correspond to the input and output two samples into the past.

Digital systems, like analog systems, can also be defined by their impulse responses, $h(k)$, and the convolution sum (Eq. (11.33)). If the response has a finite number of nonzero points, the filter is called an FIR filter or a “finite impulse response filter.” If the response has an infinite number of nonzero points, the filter is called an IIR or “infinite impulse response filter.” One positive quality of digital filters is the ease with which the output for any input can be calculated.

EXAMPLE PROBLEM 11.22

Find the impulse response for the digital filter

$$y(k) = \frac{1}{2}x(k) + \frac{1}{2}y(k-1)$$

Solution

Assume the system is at rest before input begins—that is, $y(n) = 0$ for $n < 0$.

$$y(-2) = \frac{1}{2}\delta(-2) + \frac{1}{2}y(-3) = 0 + 0 = 0$$

$$y(-1) = \frac{1}{2}\delta(-1) + \frac{1}{2}y(-2) = 0 + 0 = 0$$

$$y(0) = \frac{1}{2}\delta(0) + \frac{1}{2}y(-1) = \frac{1}{2} + 0 = \frac{1}{2}$$

$$y(1) = \frac{1}{2}\delta(1) + \frac{1}{2}y(0) = 0 + \frac{1}{2}\left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^2$$

$$y(2) = \frac{1}{2}\delta(2) + \frac{1}{2}y(1) = 0 + \frac{1}{2}\left(\frac{1}{2}\right)^2 = \left(\frac{1}{2}\right)^3$$

$$y(3) = \frac{1}{2}\delta(3) + \frac{1}{2}y(2) = 0 + \frac{1}{2}\left(\frac{1}{2}\right)^3 = \left(\frac{1}{2}\right)^4$$

...

$$y(k) = \left(\frac{1}{2}\right)^{k+1} u(k)$$

The impulse response for the filter is an exponential sequence. This is an IIR filter because the impulse response is of infinite duration.

EXAMPLE PROBLEM 11.23

Find the impulse response for the digital filter

$$y(k) = \frac{1}{3}x(k) + \frac{1}{3}x(k-1) + \frac{1}{3}x(k-2)$$

Solution

Assume the system is at rest before input begins—that is, $y(n) = 0$ for $n < 0$.

$$y(-2) = \frac{1}{3}\delta(-2) + \frac{1}{3}\delta(-3) + \frac{1}{3}\delta(-4) = 0 + 0 + 0 = 0$$

$$y(-1) = \frac{1}{3}\delta(-1) + \frac{1}{3}\delta(-2) + \frac{1}{3}\delta(-3) = 0 + 0 + 0 = 0$$

$$y(0) = \frac{1}{3}\delta(0) + \frac{1}{3}\delta(-1) + \frac{1}{3}\delta(-2) = \frac{1}{3} + 0 + 0 = \frac{1}{3}$$

$$y(1) = \frac{1}{3}\delta(1) + \frac{1}{3}\delta(0) + \frac{1}{3}\delta(-1) = 0 + \frac{1}{3} + 0 = \frac{1}{3}$$

$$y(2) = \frac{1}{3}\delta(2) + \frac{1}{3}\delta(1) + \frac{1}{3}\delta(0) = 0 + 0 + \frac{1}{3} = \frac{1}{3}$$

$$y(3) = \frac{1}{3}\delta(3) + \frac{1}{3}\delta(2) + \frac{1}{3}\delta(1) = 0 + 0 + 0 = 0$$

$$y(4) = \frac{1}{3}\delta(4) + \frac{1}{3}\delta(3) + \frac{1}{3}\delta(2) = 0 + 0 + 0 = 0$$

...

$$y(k) = 0; k \geq 3$$

This is an FIR filter with only three nonzero coefficients.

IIR filters are particularly useful for simulating analog systems. The main advantage of an IIR filter is that the desired job can usually be accomplished with fewer filter coefficients than for an FIR filter; in other words, IIR filters tend to be more efficient. The main disadvantage of an IIR filter is that signals may be distorted in an undesirable way. FIR filters can be designed with symmetry to prevent undesired signal distortion. Methods for dealing with the distortion problem in FIR filters are outside our discussions here.

Digital filters, as the name implies, are most often designed to perform specific “filtering” operations: low-pass filters, high-pass filters, band-pass filters, bandstop filters, notch filters, and so on. However, digital filters can be used to simulate most analog systems—for example, to differentiate and to integrate. Many textbooks have been written on digital filter design. The key components of the process are illustrated following.

From Digital Filter to Transfer Function

The transfer function for the digital system, $H(z)$, can be obtained by rearranging the difference equation (Eq. (11.23)) and applying Eq. (11.21). $H(z)$ is the quotient of the z -transform of the output, $Y(z)$, divided by the z -transform of the input, $X(z)$.

$$\begin{aligned}
 y(k) + a_1y(k-1) + a_2y(k-2) \dots + a_Ny(k-N) &= b_0x(k) + b_1x(k-1) + \dots + b_Mx(k-M) \\
 Y(z) + a_1z^{-1}Y(z) + a_2z^{-2}Y(z) \dots + a_Nz^{-N}Y(z) &= b_0X(z) + b_1z^{-1}X(z) + b_2z^{-2}X(z) \dots + b_Mz^{-M}X(z) \\
 Y(z)(1 + a_1z^{-1} + a_2z^{-2} \dots + a_Nz^{-N}) &= X(z)(b_0 + b_1z^{-1} + b_2z^{-2} \dots + b_Mz^{-M}) \\
 H(z) = \frac{Y(z)}{X(z)} &= \frac{b_0 + b_1z^{-1} + b_2z^{-2} \dots + b_Mz^{-M}}{1 + a_1z^{-1} + a_2z^{-2} \dots + a_Nz^{-N}}
 \end{aligned}
 \tag{11.45}$$

From Transfer Function to Frequency Response

The frequency response ($H'(\Omega)$) of a digital system can be calculated directly from $H(z)$, where Ω is in radians. If the data are samples of an analog signal as previously described, the relationship between ω and Ω is $\Omega = \omega T$:

$$H'(\Omega) = H(z)|_{z=e^{j\Omega}} \tag{11.46}$$

For a linear system, an input sequence of the form

$$x(k) = A \sin(\Omega_0 k + \Phi)$$

will generate an output whose steady-state sequence will fit into the following form:

$$y(k) = B \sin(\Omega_0 k + \emptyset)$$

Values for B and \emptyset can be calculated directly:

$$\begin{aligned}
 B &= A|H'(\Omega_0)| \\
 \emptyset &= \Phi + \text{angle}(H'(\Omega_0))
 \end{aligned}$$

EXAMPLE PROBLEM 11.24

The input sequence for the digital filter used in [Example Problem 11.23](#) is

$$x(k) = 100 \sin\left(\frac{\pi}{2}k\right)$$

What is the steady-state form of the output?

Solution

$$y(k) - \frac{1}{2}y(k-1) = \frac{1}{2}x(k)$$

The difference equation is first converted into the z-domain:

$$Y(z) - \frac{1}{2}Y(z)z^{-1} = Y(z)\left[1 - \frac{1}{2}z^{-1}\right] = \frac{1}{2}X(z).$$

Solving for $H(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{1}{2}}{1 - \frac{1}{2}z^{-1}}$$

gives the filter transfer function. To determine the output, the transfer function is evaluated at the frequency of the input sinusoid ($z = e^{j\frac{\pi}{2}}$)

$$H'\left(\frac{\pi}{2}\right) = H\left(e^{j\frac{\pi}{2}}\right) = \frac{\frac{1}{2}}{1 - \frac{1}{2}e^{-j\frac{\pi}{2}}} = \frac{\frac{1}{2}}{1 + \frac{1}{2}j} = 0.4 - j0.2 = 0.45e^{-j0.15\pi}.$$

This transfer function tells us that the output is obtained by scaling the input magnitude by 0.45 and shifting the signal by a phase factor of 0.15π rads. Therefore, the output is

$$y(k) = 45 \sin\left(\frac{\pi}{2}k - .15\pi\right)$$

Filter design problems begin with identifying the frequencies that are to be kept versus the frequencies that are to be removed from the signal. For ideal filters, $|H'(\Omega_{\text{keep}})| = 1$ and $|H'(\Omega_{\text{remove}})| = 0$. The filters in [Example Problems 11.23](#) and [11.24](#) can both be considered as low-pass filters. However, their frequency responses ([Figure 11.20](#)) show that neither is a particularly good low-pass filter. An ideal low-pass filter that has a cutoff frequency of $\pi/4$ with $|H'(\Omega)| = 1$ for $\Omega < \pi/4$ and $|H'(\Omega)| = 0$ for $\pi/4 < \Omega < \pi$ is superimposed for comparison.

Windowed FIR Filter Design

The ideal filters in [Section 11.6.4](#) provide a general framework from which to build a variety of filter functions to meet specific design criteria for both analog and discrete systems. Unfortunately, the ideal low-pass filter is not physically realizable, as we will see following. Using the ideal low-pass filter impulse response as a starting reference, we will develop a modified filter function that overcomes this limitation. The design of a windowed FIR filter is illustrated following for the case of a low-pass filter, but the same procedures and concepts apply for high-pass and band-pass filters.

There are two practical limitations associated with the ideal low-pass filter. First, note that the impulse response of the ideal low-pass filter has infinite duration, extending from $-\infty$ to $+\infty$. Thus, implementing an ideal low-pass filter requires an infinite amount of time. The simplest way to overcome this limitation is to truncate the impulse response over a finite time interval from $-T$ to T ([Figure 11.21a](#); shown for $T = 0.1$ sec). However, truncation leads to a second undesired effect. The sharp transitions in the impulse

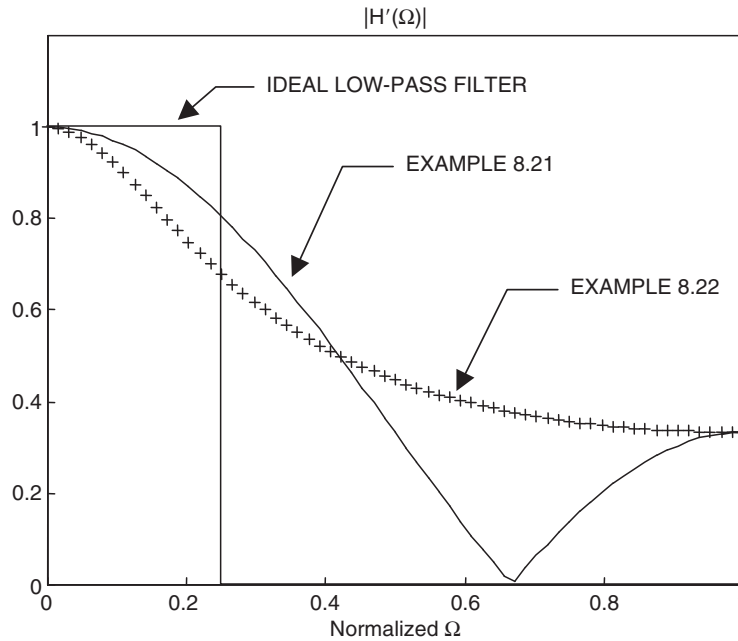


FIGURE 11.20 A frequency domain comparison of low-pass filters described in [Example Problems 11.23](#) and [11.24](#). An ideal low-pass filter with a cutoff frequency at $\pi/4$ rads or 0.25 when normalized by π radians is superimposed for comparison. The cutoff frequency of a low-pass filter is usually defined as the frequency at which the amplitude is equal to $1/\sqrt{2}$ or approximately 0.71, which matches [Example Problem 11.12](#). Both digital filters have the same amplitude at f_{\max} —that is, where normalized $\Omega = 1$.

response to the selected interval do not allow for convergence of the Fourier integral (Gibbs Phenomena). In the frequency domain, the effects of the truncation can be seen in the transfer function magnitude of the truncated low-pass filter ([Figure 11.21b](#)). The truncated filter exhibits errors in the passband and stopband regions. These errors are undesirable because they will distort the signal in the passband while allowing signals in the stopband from passing through.

One way to overcome the limitation is to gradually truncate the filter impulse response with a smooth window function, $w(t)$. The modified low-pass filter is expressed as the product of the ideal filter and the window function:

$$h(t) = h_{LP}(t)w(t) = \frac{W_c}{\pi} \text{sinc}(W_c t)w(t) \quad (11.47)$$

where $w(t)$ is restricted to the interval $-T$ to T ([Figure 11.21c](#)). The windowing procedure is illustrated in [Figure 11.21d](#), which shows the product of the window with the ideal filter impulse response. The window allows for a smooth truncation of the impulse response, thus allowing for convergence of the Fourier integral. As can be seen, the resulting stopband and passband errors of the windowed filter ([Figure 11.21e](#)) are substantially smaller than for the truncated ideal filter ([Figure 11.21b](#)). Details of the design of window functions

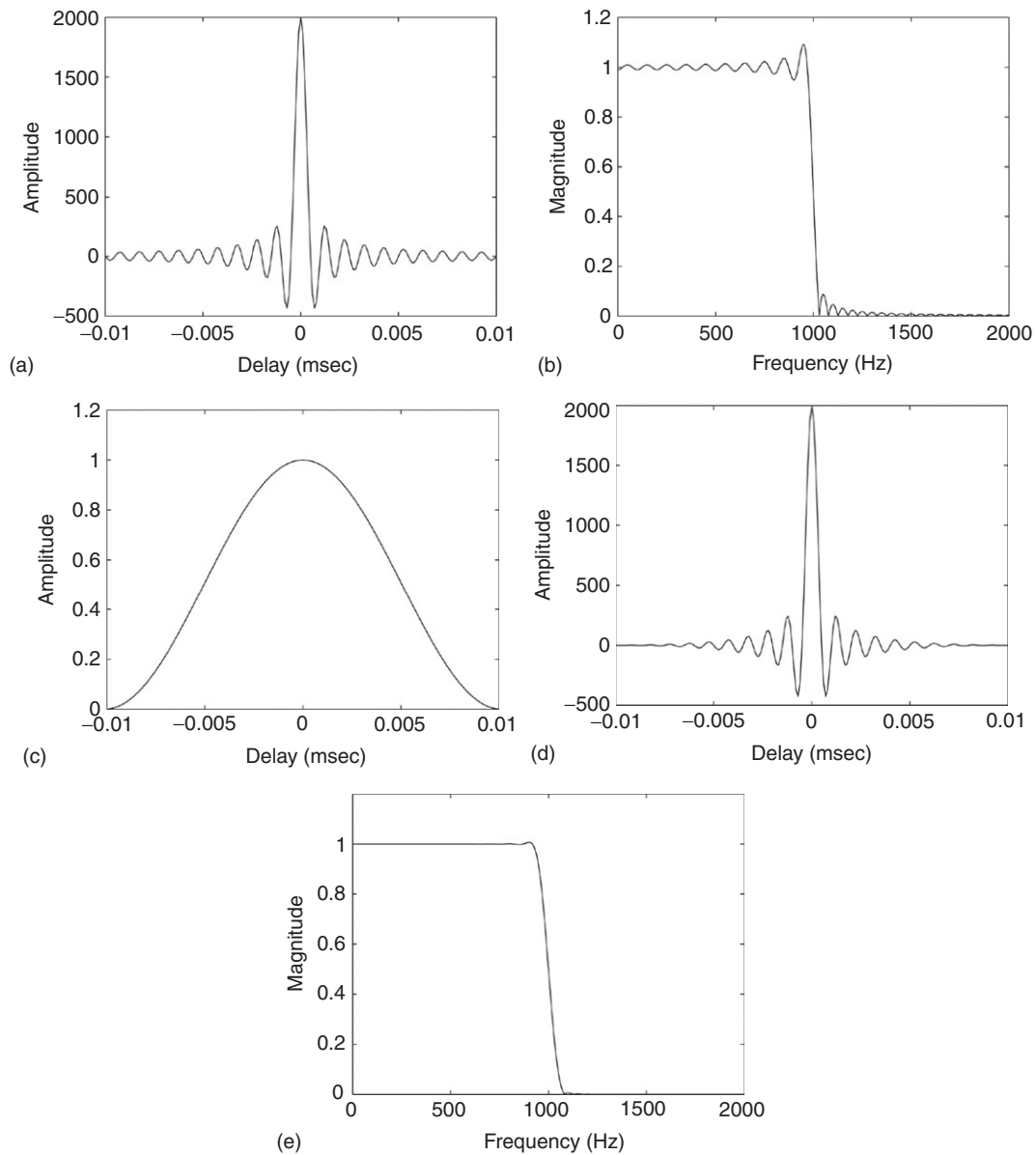


FIGURE 11.21 Filter design by windowing. (a) Truncated ideal low-pass filter impulse response. (b) Magnitude response for the ideal truncated filter. Note the errors in the passband and stopband regions. (c) Window function. (d) Windowed impulse response. (e) Magnitude response of the windowed filter exhibits substantially smaller passband and stopband errors.

are beyond the scope of this text and can be found in many signal processing textbooks. However, numerous window functions have been developed for a variety of applications. Some of the most celebrated window functions include the Kaiser, Hanning, and Hamming windows.

So far, we have illustrated the windowing method for a continuous filter, but we would like to apply this procedure to develop a discrete FIR filter. To achieve this, the continuous filter impulse response is sampled by allowing $t = k/F_s$, for integer k . The resulting discrete impulse response is expressed as

$$h[k] = \frac{W_c}{\pi} \operatorname{sinc}\left(\frac{W_c}{F_s}k\right) w(k) = \frac{W_c}{\pi} \operatorname{sinc}(\Omega_c k) w(k) \quad (11.48)$$

where $\Omega_c = W_c/F_s$ is the digital cutoff frequency of the filter. Note that according to the Nyquist sampling theorem, $0 < W_c < F_s/2$, and thus $0 < \Omega_c < \pi$.

EXAMPLE PROBLEM 11.25

In MATLAB, implement a 201 sample digital low-pass and high-pass filter using a Hanning window. The cutoff frequency of both filters is 1000 Hz. Plot the impulse response and transfer function magnitude.

Solution

```
Wc=2*pi*1000; %Cutoff Frequency in Radians / Sec
Fs=10000; %Sampling Rate in Hz
T=1/Fs;

%Ideal Filters
N=100; %Filter Order
t=(-N:N)/Fs; %Time Axis Sampled at Fs
h_lp=Wc/pi*sinc(1/pi*Wc*t); %Sampled Ideal Low-pass Filter
Impulse Response
delta=[zeros(1,N) Fs zeros(1,N)]; %Discrete Diract Impulse
Function
h_hp=(delta-h_lp); %Sampled Ideal High-pass Filter Impulse
Response

%Hanning Filters
W=hanning(2*N+1)'; %Hanning Window
h_lp=h_lp.*W; %Hanning Low-pass Filter Impulse Response
h_hp=h_hp.*W; %Hanning High-pass Filter Impulse Response
NFFT=1024*8; %FFT number of samples
faxis=(0:NFFT-1)/NFFT*Fs; %Frequency Axis
H_lp=abs(fft(h_lp,NFFT))/Fs; %Hanning Low-pass Filter Transfer
Function Magnitude
H_hp=abs(fft(h_hp,NFFT))/Fs; %Hanning High-pass Filter Transfer
Function Magnitude
```

```

%Plotting Results
subplot(221)
plot(t,h_lp,'k')
xlabel('Time (sec)')
ylabel('Amplitude')

subplot(222)
plot(faxis,H_lp,'k');
axis([0 Fs/2 0 1.2])
xlabel('Frequency (Hz)')
ylabel('Magnitude')

subplot(223)
plot(t,h_hp,'k')
xlabel('Time (sec)')
ylabel('Amplitude')

subplot(224)
plot(faxis,H_hp,'k');
axis([0 Fs/2 0 1.2])
xlabel('Frequency (Hz)')
ylabel('Magnitude')

```

The results are shown in [Figure 11.22](#).

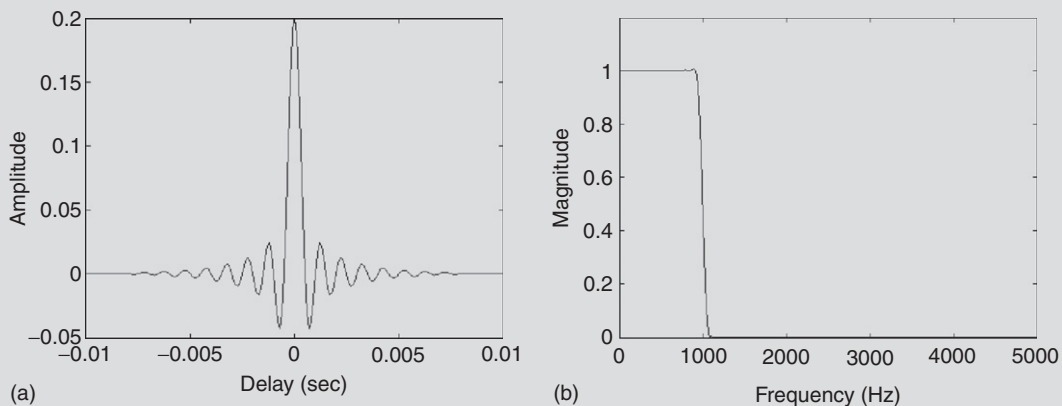


FIGURE 11.22 Low-pass and high-pass FIR filter MATLAB simulation. (a) Hanning low-pass filter impulse response and magnitude response (b).

Continued

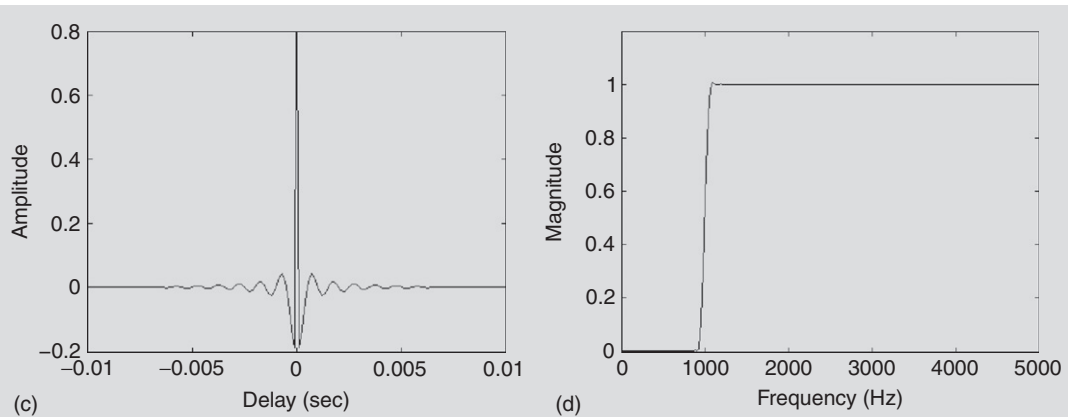


FIGURE 11.22, cont'd (c) Hanning high-pass filter impulse response and magnitude response (d).

EXAMPLE PROBLEM 11.26

Using the digital low-pass and high-pass filters of [Example Problem 11.25](#), filter a white noise (flat spectrum) signal. Plot the magnitude spectrum of the input and output signals.

Solution

```
%Filtering the Random Noise Signal
X=randn(1,Fs); %1 second of Random Noise
Y_lp=conv(X,h_lp); %Low-pass Filtered Noise
Y_hp=conv(X,h_hp); %High-pass Filtered Noise

%Plotting Results
subplot(221)
psd(X,1024,Fs); %Input Power spectrum Magnitude
hold on
psd(Y_lp,1024,Fs); %Low-pass Output Power spectrum Magnitude
ch=get(gca,'children')
set(ch(1),'color','k')
set(ch(2),'color',[0.5 0.5 0.5])

subplot(222)
psd(X,1024,Fs); %Input Power spectrum Magnitude
hold on
psd(Y_hp,1024,Fs); %High-pass Output Power spectrum Magnitude
ch=get(gca,'children')
ch=get(gca,'children')
set(ch(1),'color','k')
set(ch(2),'color',[0.5 0.5 0.5])
```

The results are shown in [Figure 11.23](#).

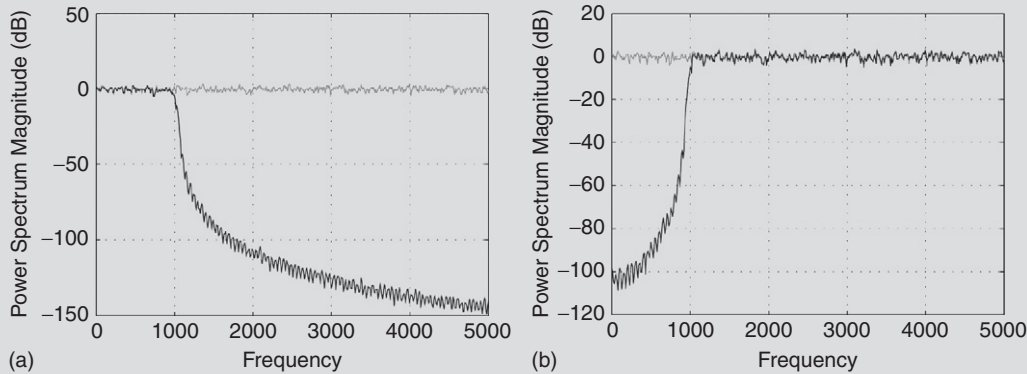


FIGURE 11.23 Low-pass and high-pass filtered noise signal. (a and b) Gray lines show the power spectrum of the input signal. (a) Low-pass filtered noise power spectrum is shown as black. (b) High-pass filtered noise power spectrum is shown as black.

11.7 SIGNAL AVERAGING

Biological signal measurements are often confounded by measurement noise. Variability in the measurement of a signal often makes it difficult to determine the signal characteristics, making it nearly impossible to obtain a reliable clinical diagnosis.

Many classes of biological signals are modeled as the sum of an ideal noiseless signal component, $x(t)$, and separate independent noise term, $n(t)$:

$$x_i(t) = x(t) + n_i(t). \quad (11.49)$$

The signal, $x_i(t)$, corresponds to the “measured” i -th trial or i -th measurement of the signal. Note that the i -th measurement contains both a deterministic component, $x(t)$, and a random or stochastic noise term, $n_i(t)$. Although the deterministic component of the signal is fixed from trial to trial, the noise term represents intrinsic variability, which may arise from a number of separate sources. The i -th measurement can therefore exhibit significant trial-to-trial variability because the random component, $n_i(t)$, is different across consecutive trials. As an example, a measurement ECG (electrocardiogram) electrode can pick up extraneous signals from the muscles, lungs, and even the internal electronics of the recording devices (e.g., 60 cycle noise from the power supply). The activity of these signals is unrelated to the activity of the beating heart, and it therefore shows up in the signal measurement as noise. Other unpredictable changes in the activity of the heartbeat, such as from the caffeine jolt after drinking an espresso, could also show up in a measurement and be interpreted as noise to an uninformed observer.

We have already examined one possible way to separate out the signal term from the noise term by filtering the signal with an appropriately designed filter. Appropriate filtering allows one to clean up the signal, thus improving the quality of signal and the diagnostic reliability in clinical settings. If the spectrum of the noise and signal components do not overlap in the frequency domain, one can simply design a filter that keeps or enhances the desired signal term, $x(t)$, and discards the unwanted noise term, $n_i(t)$. While this is a simple and useful way of cleaning up a signal, this approach does not work in many instances because the biological signal and noise spectrums overlap.

Many biological signals are approximately periodic in nature. Signals associated with the beating heart—blood pressure, blood velocity, electrocardiogram (ECG)—fall into this category. However, due to intrinsic natural variability, noise, and/or the influence of other functions, such as respiration, beat-to-beat differences are to be expected. Figure 11.2 is an example of a blood pressure signal that has all of the described variability.

Blood pressure signals have many features that clinicians and researchers use to determine a patient's health. Some variables that are often measured include the peak pressure while the heart is ejecting blood (systolic phase), the minimum pressure achieved while the aortic valve is closed (diastolic phase), the peak derivative (dP/dt) during the early part of the systolic phase (considered an indication of the strength of the heart), and the time constant of the exponential decay during diastole (a function of the resistance and compliance of the blood vessels). One way to determine variables of interest is to calculate the variables or parameters for each beat in a series of beats and then report the means. This is often not possible because noise from individual measurements makes it very difficult to accurately determine the relevant biological parameters. An alternate approach is to first average the signal measurements from separate trials,

$$\bar{x}(t) = \frac{1}{N} \sum_{i=1}^N x_i(t), \quad (11.50)$$

such that a representative beat is obtained. If the signal is discrete, this average is represented by

$$\bar{x}(k) = \frac{1}{N} \sum_{i=1}^N x_i(k) \quad (11.51)$$

Here, $x_i(t)$, or $x_i(k)$ for the discrete case, represents the i -th measured heartbeat signal out of a total of N measurements. The signal $\bar{x}(t)$, $\bar{x}(k)$ for the discrete case, represents the mean or average waveform obtained following the averaging procedure. Substituting Eq. (11.50) into (11.49) leads to

$$\bar{x}(t) = x(t) + \frac{1}{N} \sum_{i=1}^N n_i(t) = x(t) + \varepsilon(t). \quad (11.52)$$

If the noise term, $n_i(t)$, is random and independent from trial-to-trial, it can be shown that the measurement error term in Eq. (11.52), $\varepsilon(t)$, which contains the influence of the noise, approaches 0 as $N \rightarrow \infty$. Thus, $\bar{x}(t) \approx x(t)$ for very large N , where $\varepsilon(t)$ tends to be small. This is a very powerful result! It tells us that we can effectively remove the noise by simply averaging measurements from many trials. Essentially, if we average a sufficiently large number of signal trials, the averaged signal closely approximates the true noiseless signal waveform.

Many biological acquisition systems are designed to calculate signal averages (Eqs. (11.50) and (11.51)) as data are collected. The summation process is triggered by a signal or a signal-related feature. The ECG signal, which has many sharp features, is often used for heartbeat-related data. Figure 11.24 shows a signal-averaged blood pressure waveform for the data shown in Figure 11.2. Figure 11.25 shows the signal-averaging procedure for an auditory brainstem response (ABR) EEG measurement.

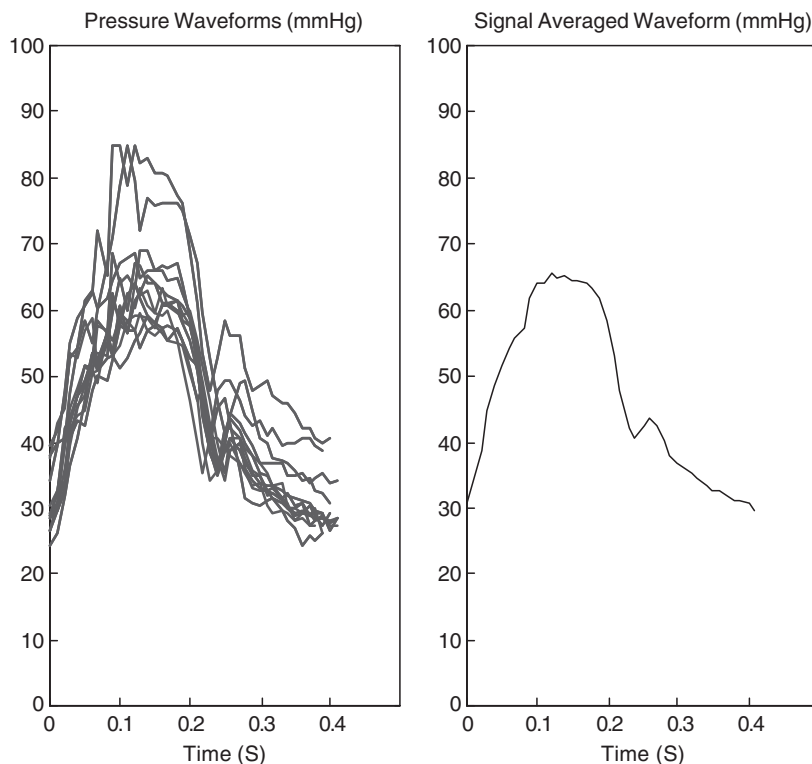


FIGURE 11.24 A signal-averaged pressure waveform for the data shown in Figure 11.2.

In both cases, note that the averaging helps to preserve the relevant signal features and remove undesirable noise disturbances.

The blood pressure and ABR examples illustrate signal averaging in the time domain. For signals that are random in nature, signal averaging in the frequency domain is sometimes preferable. Figure 11.26 shows an EEG signal sampled over the occipital lobe of a patient. The sampling rate was 16 kHz. EEG analysis is usually done in the frequency domain, since the presence of different frequencies is indicative of different brain states, such as sleeping, resting, alertness, and so on. The power at each frequency estimate, which can be approximated by the square of the Fourier transform, is the measurement of choice.

If a DFT is performed on the data to estimate the power of the frequencies in the signal, the expected noise in the measurement is of the same size as the measurement itself. To reduce the noise variance, a statistical approach must be undertaken. One popular approach is known as the “Welch” or “periodogram averaging” method. The signal is broken into L sections (disjoint if possible) of N points each. A DFT is performed on each of the L sections. The final result for the N frequencies is then the average at each frequency for the L sections.

The N data points in the i -th segment are denoted as

$$x_i(k) = x(k + (i - 1)N) \quad 0 \leq k \leq N - 1, 1 \leq i \leq L$$

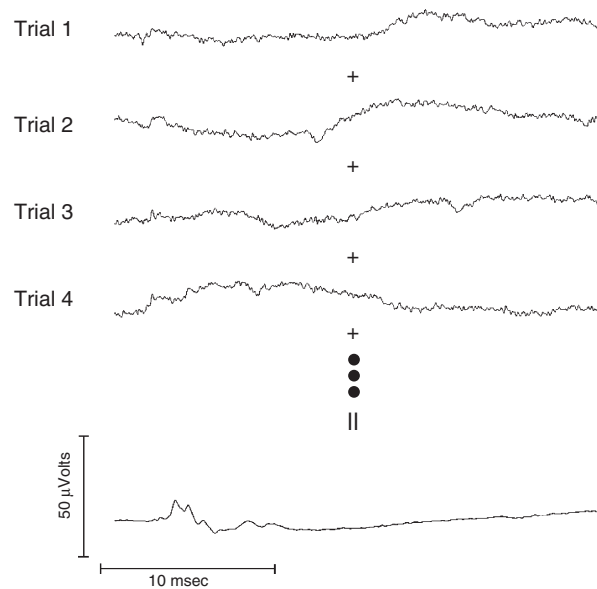


FIGURE 11.25 Single trials from an auditory evoked response to a brief sound pulse (at time zero) were measured on the temporal lobe. The auditory response from individual trials is obscured by random noise (shown first 4 out of 1,000). Averaged response of 1,000 trials reveals the auditory response component (bottom trace).

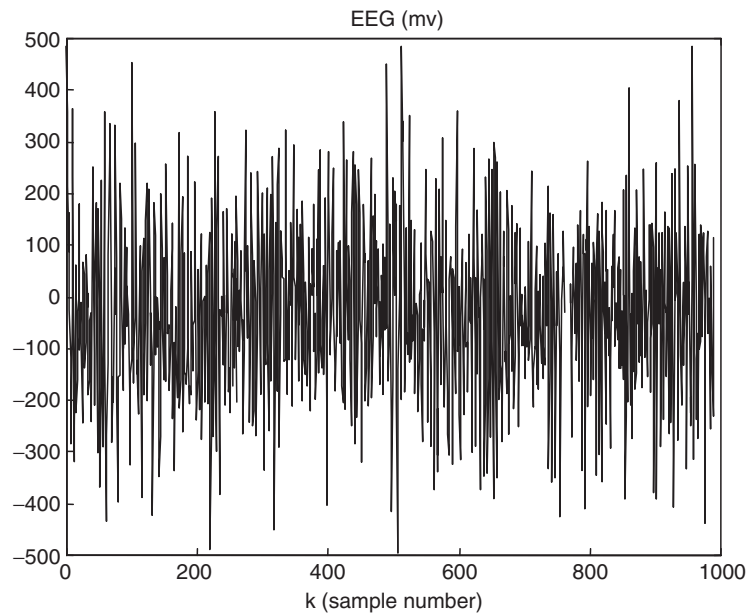


FIGURE 11.26 An EEG signal containing 1,000 samples sampled at 16 kHz from the occipital.

if the segments are consecutive and disjoint. The power estimate based on the DFT of an individual segment “ i ” is

$$\hat{P}_i(m) = \frac{1}{N} \left| \sum_{k=0}^{N-1} x_i(k) e^{-j \frac{2\pi mk}{N}} \right|^2 \text{ for } 0 \leq m \leq N-1 \quad (11.53)$$

where m is associated with the power at a frequency of $\Omega = 2\pi m/N$ radians. The averaged signal spectrum is calculated by taking the mean at each frequency

$$\hat{P}(m) = \frac{1}{L} \sum_{i=1}^L \hat{P}_i(m) \quad (11.54)$$

The selection of N is very important, since N determines the resolution in the frequency domain. For example, if data are sampled at 500 samples/s and the resolution is desired at the 1 Hz level, at least 1 second or 500 samples ($N = 500$) should be included in each of the L sections. If resolution at the 10 Hz level is sufficient, only 0.1 seconds or 50 data points need to be included in each section. Note that the reduction in the frequency resolution leads to improvement in the certainty of the measurement because one can now average more signal segments. The averaging procedure decreases the variance by a factor of $1/L$. This averaging process is demonstrated for the EEG data in [Figure 11.27](#). Modifications to

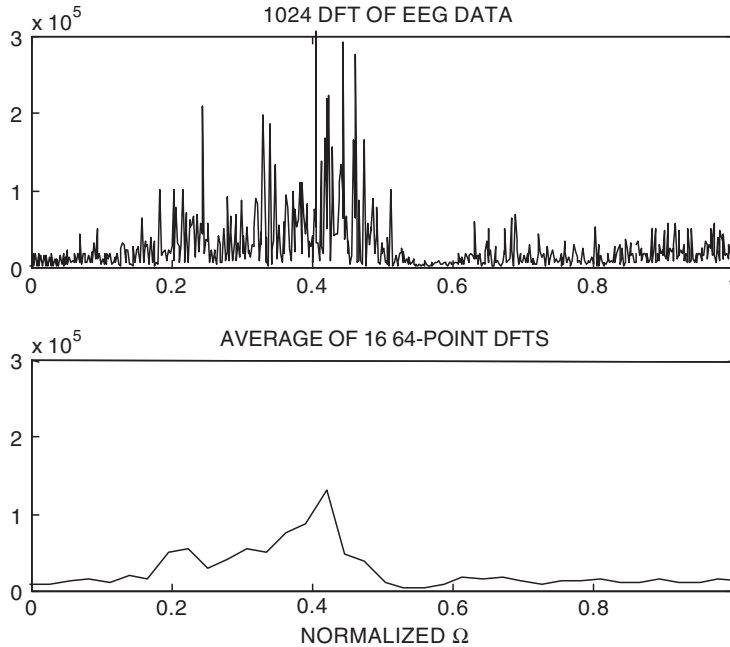


FIGURE 11.27 DFT averaging of an EEG. The top trace shows the raw DFT. The bottom trace shows the periodogram-averaged DFT obtained with 16 64-point segments of the data.

the procedure may include using overlapping segments if a larger value for L is needed and the number of available data points is not sufficient and/or multiplying each section by a window that forces continuity at the end points of the segments.

EXAMPLE PROBLEM 11.27

Consider the sinusoid signal

$$x(k) = \sin(\pi/4k) + n(k)$$

that is corrupted by random noise, $n(k)$. Using MATLAB, show that averaging the signal removes the noise component and reveals the deterministic component. Show results for 1, 10, and 100 averages.

Solution

```
k=1:64; %Discrete Time Axis
for i=1:100 %Generating 100 signal Trials
    x(i,:)=sin(pi/4*k)+randn(1,64); %i-th trial
end
X1=x(1,:); %1 Averages
X10=mean(x(1:10,:)); %10 Averages
X100=mean(x); %100 Averages

subplot(311) %Plotting Results, 1 Average
plot(k,X1,'k')
axis([1 64 -3 3])
title('1 Average')
ylabel('Amplitude')

subplot(312) %Plotting Results, 10 Averages
plot(k,X10,'k')
axis([1 64 -3 3])
title('10 Averages')
ylabel('Amplitude')

subplot(313) %Plotting Results, 100 Averages
plot(k,X100,'k')
axis([1 64 -3 3])
title('100 Averages')
xlabel('Discrete Time')
ylabel('Amplitude')
```

The results are shown in [Figure 11.28](#).

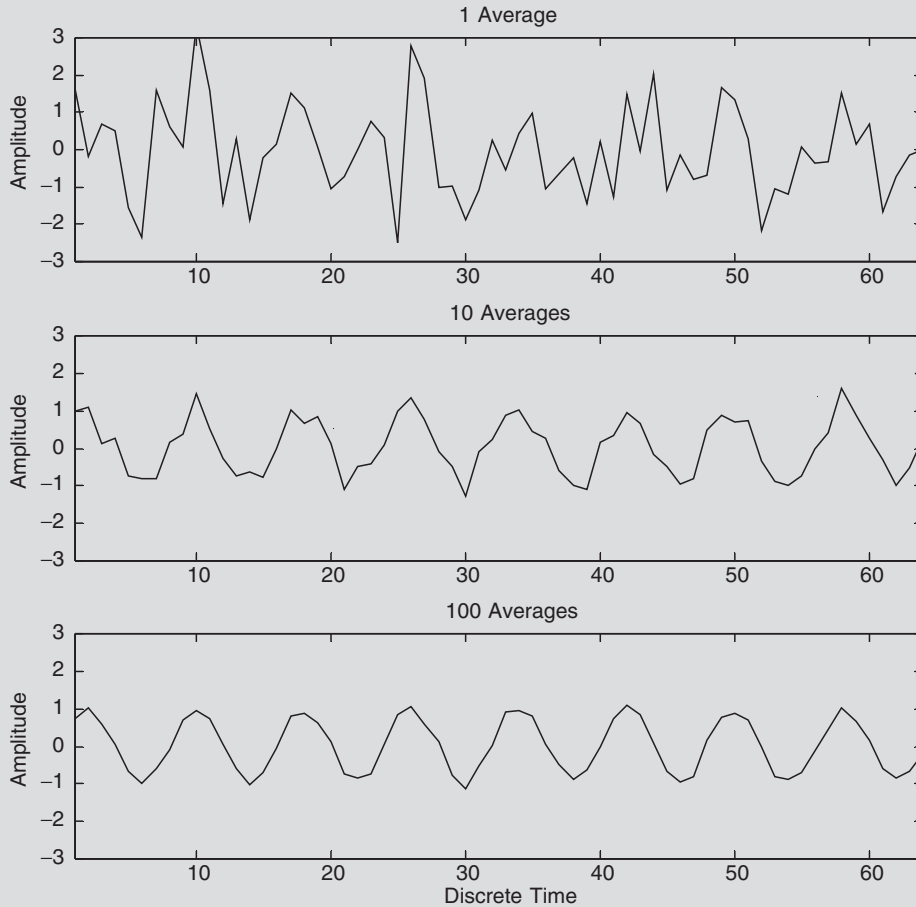


FIGURE 11.28 MATLAB results showing noise removal by averaging a noisy sinusoid signal. Shown for 1, 10, and 100 averages.

11.8 THE WAVELET TRANSFORM AND THE SHORT-TIME FOURIER TRANSFORM

The Fourier transform (Eq. (11.6)) is a well-known signal processing tool for breaking a signal into constituent sinusoidal waveforms of different frequencies. For many applications, particularly those that change little over time, knowledge of the overall frequency content may be all that is desired. The Fourier transform, however, does not delineate how a signal changes over time.

The short-time Fourier transform (STFT) and the wavelet transform (WT) have been designed to help preserve the time-domain information. The STFT approach is to perform a Fourier transform on only a small section (window) of data at a time, thus mapping

the signal into a 2D function of time and frequency. The transform is described mathematically as

$$X(\omega, a) = \int_{-\infty}^{\infty} x(t)g(t-a)e^{-j\omega t} dt \quad (11.55)$$

where $g(t)$ may define a simple box or pulse function. The inverse of the STFT is given as

$$X(\omega, a) = K_g \iint X(\omega, a)g(t-a)e^{j\omega t} dt da \quad (11.56)$$

where K_g is a function of the window used.

To avoid the “boxcar” or “rippling” effect associated with a sharp window, the window may be modified to have more gradually tapered sides. Both designs are shown in Figure 11.29. The windows are superimposed on a totally periodic aortic pressure signal. For clarity, the windows have been multiplied by a factor of 100.

The STFT amplitudes for three box window sizes, $\frac{1}{2}$ period, 1 period, and 2 periods, are illustrated in Figure 11.30. The vertical lines in the top figure are indicative of longer periodicities than the window. The solid-colored horizontal lines in the bottom two figures indicate that the frequency content is totally independent of time at that window size. This is expected, since the window includes either one or two perfect periods. The dark (little or no frequency content) horizontal lines interspersed with the light lines in the bottom figure indicate that multiple periods exist within the window.

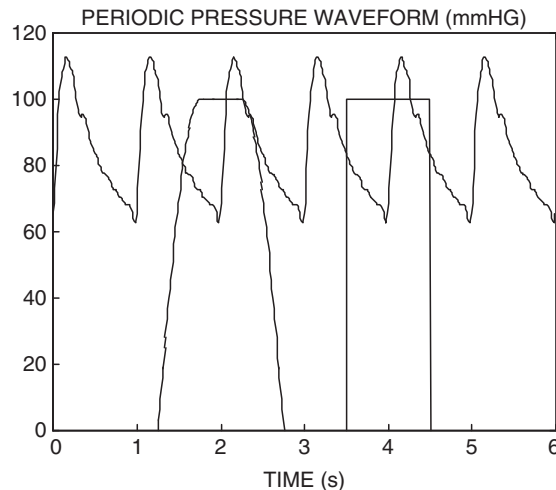


FIGURE 11.29 An example of two windows that might be used to perform an STFT on a perfectly periodic aortic pressure waveform. Each window approximates the width of one pulse. The tapered window on the left can help avoid the “boxcar” or “rippling” effect associated with the sharp window on the right. For clarity, the windows have been multiplied by a factor of 100.

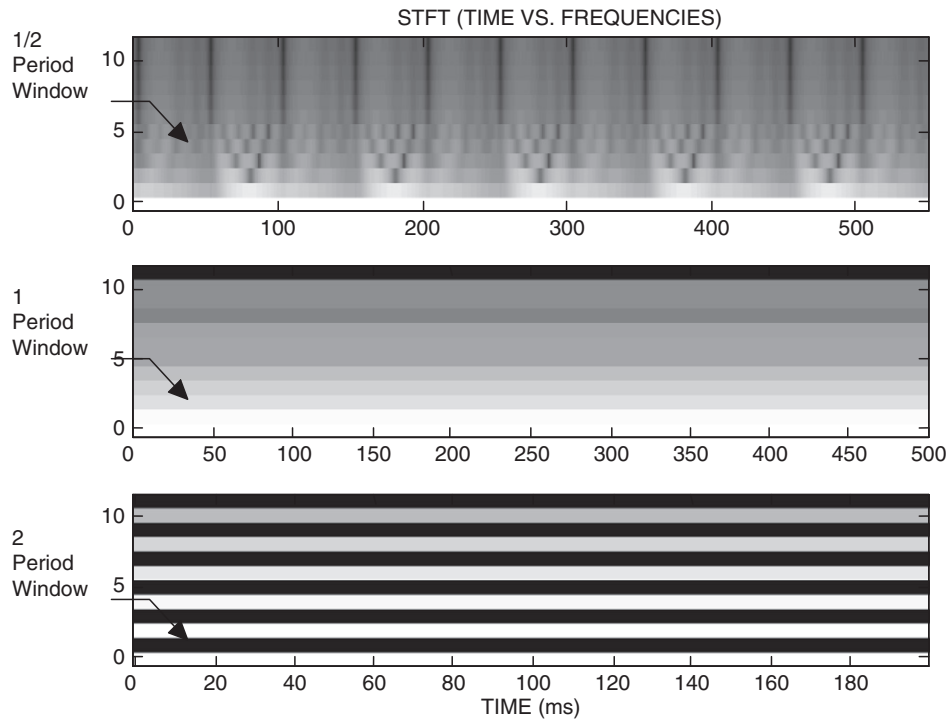


FIGURE 11.30 A two-dimensional rendering of the STFT amplitude coefficients for three box window sizes— $\frac{1}{2}$ period, 1 period, and 2 periods—applied to the perfectly periodic data shown in Figure 11.17. The lighter the color, the larger the amplitude. For example, the 0th row corresponds to the mean term of the transform, which is the largest in all cases. Higher rows correspond to harmonics of the data, which, in general, decrease with frequency. The vertical lines in the top figure are indicative of longer periodicities than the window. The solid-colored horizontal lines in the bottom two figures indicate that the frequency content is totally independent of time at that window size. The dark (little or no frequency content) horizontal lines interspersed with the light lines in the bottom figure indicate that multiple periods exist within the window.

In contrast, Figure 11.31 shows an amplitude STFT spectrum for the aperiodic pressure waveform shown in Figure 11.2, with the window size matched as closely as possible to the heart rate. The mean has been removed from the signal so the variation in the lowest frequencies (frequency level 0) reflects changes with respiration. The level of the heart rate (level 1) is most consistent across time, and the variability increases with frequency.

The main disadvantage of the STFT is that the width of the window remains fixed throughout the analysis. Wavelet analysis represents a change from both the FT and STFT in that the constituent signals are no longer required to be sinusoidal and the windows are no longer of fixed length. In wavelet analysis, the signals are broken up into shifted and scaled versions of the original or “mother” wavelet, $\psi(t)$. Figure 11.32 shows examples of two wavelets: the Haar on the left and one from the Daubechies (db2) series on the right. Conceptually, these mother wavelet functions are analogous to the impulse response of a band-pass filter. The sharp corners enable the transform to match up with local details that are not possible to observe using a Fourier transform.

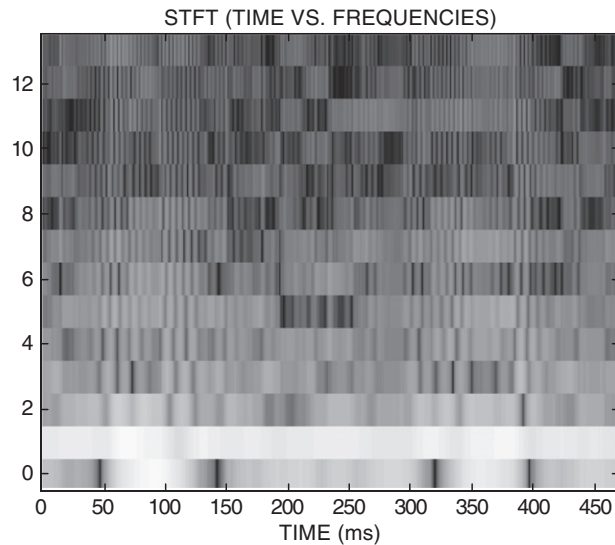


FIGURE 11.31 A two-dimensional rendering of the STFT of the aperiodic aortic pressure tracing shown in [Figure 11.2](#). The window size was matched as closely as possible to the heart rate. The mean was removed from the signal so the variation in the lowest frequencies—that is, frequency level 0—reflects changes with respiration.

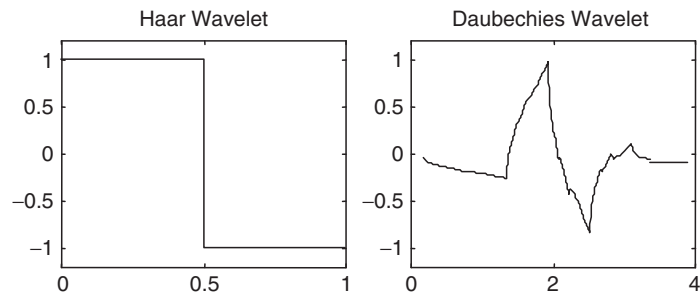


FIGURE 11.32 The general shape of two wavelets commonly used in wavelet analysis. The sharp corners enable the transform to match up with local details not possible to observe when using a Fourier transform that matches only sinusoidal shapes.

The notation for the 2D WT is

$$C(a, s) = \int_{-\infty}^{\infty} x(t) \varphi(a, s, t) dt \quad (11.57)$$

where a = scale factor and s = the position factor. C can be interpreted as the correlation coefficient between the scaled, shifted wavelet and the data. [Figure 11.33](#) shows the db2 ($\varphi(t)$) wavelet at different scales and positions—for example, $\varphi(2, -100, t) = \varphi(2t - 100)$. The inverse wavelet transform

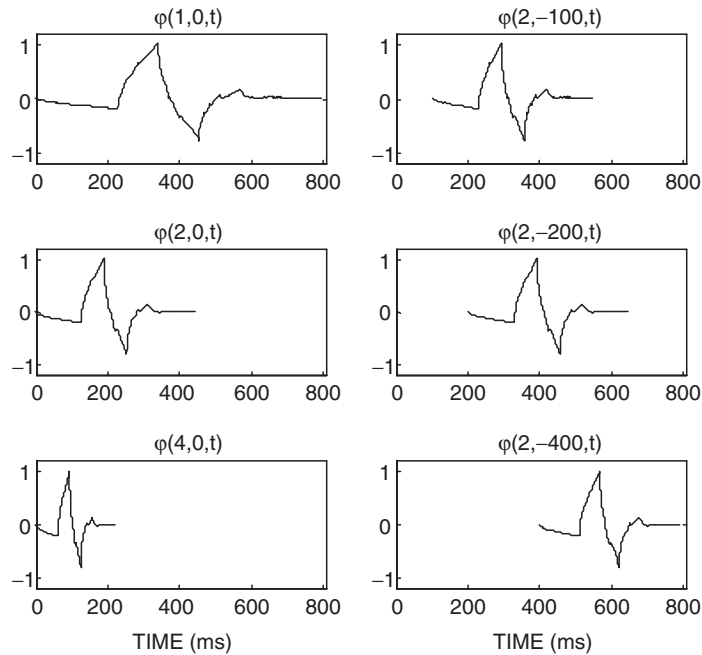


FIGURE 11.33 Illustrations of the db2 wavelet at several scales and positions. The upper left-hand corner illustrates the basic waveform $\varphi(t)$. The notation for the illustrations is given in the form $\varphi(\text{scale}, \text{delay}, t)$. Thus, $\varphi(t) = \varphi(1,0,t)$, $\varphi(2t-100) = \varphi(2,-100,t)$, and so on.

$$x(t) = K_{\varphi} \iint C(a,s) \varphi(a,s,t) dt ds \quad (11.58)$$

can be used to recover the original signal, $x(t)$, from the wavelet coefficients, $C(a,s)$. K_{φ} is a function of the wavelet used.

In practice, wavelet analysis is performed on digitized signals using a subset of scales and positions (see MATLAB's Wavelet Toolbox). One computational process is to recursively break the signal into low-frequency ("high-scale" or "approximation") and high-frequency ("low-scale" or "detail") components using digital low-pass and high-pass filters that are functions of the mother wavelet. The output of each filter will have the same number of points as the input. In order to keep the total number of data points the same at each level, every other data point of the output sequences is discarded. This process is known as "downsampling." Using "upsampling" and a second set of digital filters, called reconstruction filters, the process can be reversed, and the original data set is reconstructed. Remarkably, the inverse discrete wavelet transform does exist!

While this process will rapidly yield wavelet transform coefficients, the power of discrete wavelet analysis lies in its ability to examine waveform shapes at different resolutions and to selectively reconstruct waveforms using only the levels of approximation and detail that are desired. Applications include detecting discontinuities and breakdown points, detecting long-term evolution, detecting self-similarity (e.g., fractal trees), identifying pure

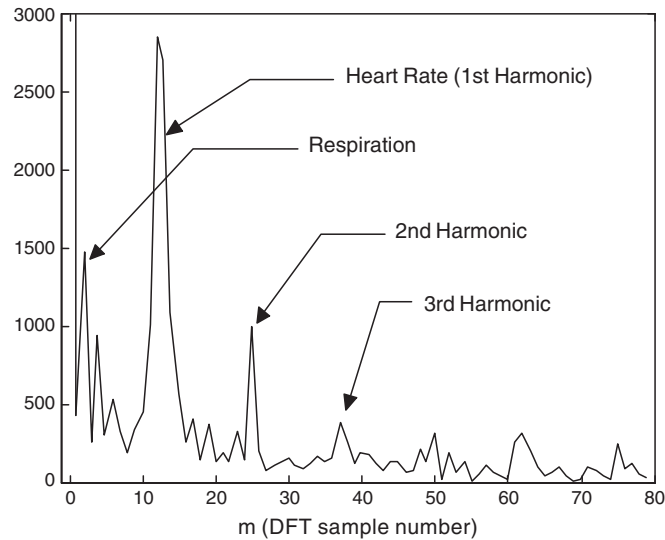


FIGURE 11.34 DFT of pressure data from Figure 11.2. The first, second, and third harmonics of the heart rate are clearly visible.

frequencies (similar to Fourier transform), and suppressing, denoising, and/or compressing signals.

For comparison purposes, discrete Fourier transforms and discrete wavelet transforms are illustrated for the pressure waveforms shown in Figure 11.2. Figure 11.34 shows details of the DFT on the entire record of data. The beat-to-beat differences are reflected by the widened and irregular values around the harmonics of the heart rate. The respiration influence is apparent at the very low frequencies.

Finally, an example from the MATLAB Wavelet Toolbox is shown that uses the same pressure waveform. Figure 11.35 is a 2D rendering of the wavelet transform coefficients. The x -axis shows the positions and the y -axis shows the scales, with the low scales on the bottom and the high scales on the top. The top scale clearly shows the two respiratory cycles in the signal. More informative than the transform coefficients, however, is a selective sample of the signal details and approximations. As the scale is changed from a_1 to a_7 , the approximation goes from emphasizing the heart rate components to representing the respiration components. The details show that the noise at the heart rate levels is fairly random at the lower scales but moves to being quite regular as the heart rate data become the noise (Figure 11.36).

11.9 ARTIFICIAL INTELLIGENCE TECHNIQUES

Artificial intelligence (AI) is a broad field that focuses on the application of computer systems that exhibit intelligent capabilities. AI systems can be built from a number of separate technologies, including fuzzy logic, neural networks, and expert systems. The principal aim of AI is to create intelligent machines that can function under adverse and unpredictable circumstances. The term *intelligent* as it applies to machines indicates

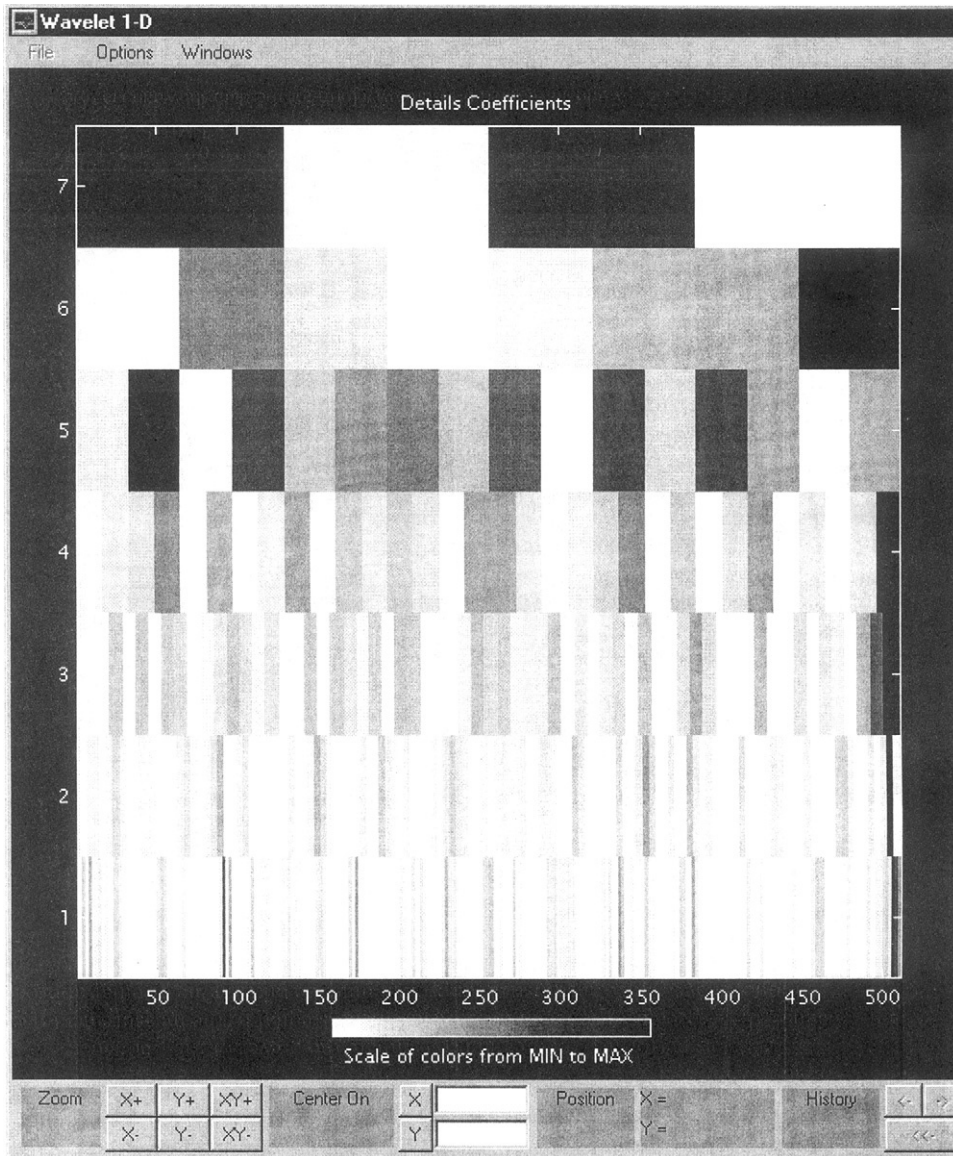


FIGURE 11.35 MATLAB was used to produce a two-dimensional rendering of the wavelet transform coefficients, with the Daubechies wavelet applied to the aortic pressure tracing in [Figure 11.2](#). The x-axis shows the positions and the y-axis shows the scales with the low scales on the bottom and the high scales on the top. The associate waveforms at selected levels of these scales are shown in [Figure 11.33](#).

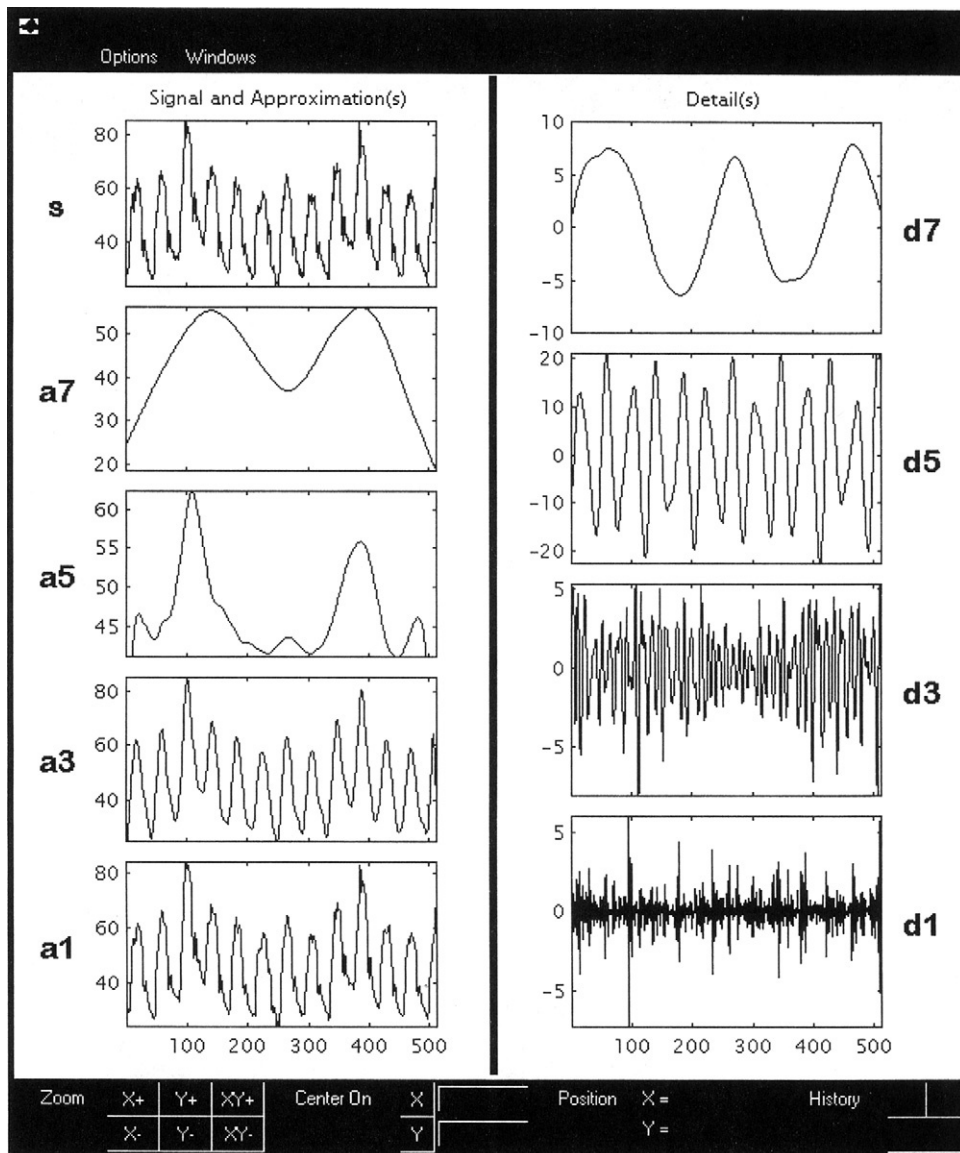


FIGURE 11.36 A selective sample of the signal details and approximations generated by MATLAB as part of the wavelet transform process.

computer-based systems that can interact with their environment and adapt to changes in the environment. The adaptation is accomplished through self-awareness and perceived models of the environment that are based on qualitative and quantitative information. In other words, the basic goal of AI techniques is to produce machines that are more capable of human-like reasoning, decision making, and adaptation.

The machine intelligence quotient (MIQ) is a measure of the intelligence level of machines. The higher a machine's MIQ, the higher the capacity of the machine for automatic reasoning and decision making. The MIQ of a wide variety of machines has risen significantly during the past few years. Many computer-based consumer products, industrial machinery, and biomedical instruments and systems are using more sophisticated artificial intelligence techniques. Advancements in the development of fuzzy logic, neural networks, and other soft computing techniques have contributed significantly to the improvement of the MIQ of many machines.

Soft computing is an alliance of complementary computing methodologies. These methodologies include fuzzy logic, neural networks, probabilistic reasoning, and genetic algorithms. Various types of soft computing often can be used synergistically to produce superior intelligent systems. The primary aim of soft computing is to allow for imprecision, since many of the parameters that machines must evaluate do not have precise numeric values. Parameters of biological systems can be especially difficult to measure and evaluate precisely.

11.9.1 Fuzzy Logic

Fuzzy logic is based on the concept of using words rather than numbers for computing, since words tend to be much less precise than numbers. Computing has traditionally involved calculations that use precise numerical values, while human reasoning generally uses words. Fuzzy logic attempts to approximate human reasoning by using linguistic variables. Linguistic variables are words that are used to describe a parameter. For body temperature, linguistic variables that might be used are high fever, above normal, normal, below normal, and frozen. The linguistic variables are more ambiguous than the number of degrees Fahrenheit, such as 105.0, 98.9, 98.6, 97.0, and 27.5.

In classical mathematics, numeric sets called crisp sets are defined, while the basic elements of fuzzy systems are fuzzy sets. An example of a crisp set is $A = [0, 20]$. Crisp sets have precisely defined, numeric boundaries. Fuzzy sets do not have sharply defined bounds. Consider the categorization of people by age. Using crisp sets, the age groups could be divided as $A = [0, 20]$, $B = [30, 50]$, and $C = [60, 80]$. [Figure 11.37a](#) shows the characteristic function for the sets A, B, and C. The value of the function is either 0 or 1, depending on whether the age of a person is within the bounds of set A, B, or C. The scheme using crisp sets lacks flexibility. If a person is 25 years old or 37 years old, he or she is not categorized.

If the age groups were instead divided into fuzzy sets, the precise divisions between the age groups would no longer exist. Linguistic variables, such as young, middle-aged, and old, could be used to classify the individuals. [Figure 11.37b](#) shows the fuzzy sets for age categorization. Note the overlap between the categories. The words are basic descriptors, not precise measurements. A 30-year-old woman may seem old to a 6-year-old boy but quite young to an 80-year-old man. For the fuzzy sets, a value of 1 represents a 100 percent degree of membership to a set. A value of 0 indicates that there is no membership in the set. All numbers between 0 and 1 show the degree of membership to a group. A 35-year-old person, for instance, belongs 50 percent to the young set and 50 percent to the middle-aged set.

As with crisp sets from classical mathematics, operations are also defined for fuzzy sets. The fuzzy set operation of intersection is shown in [Figure 11.38a](#). [Figure 11.38b](#) shows the

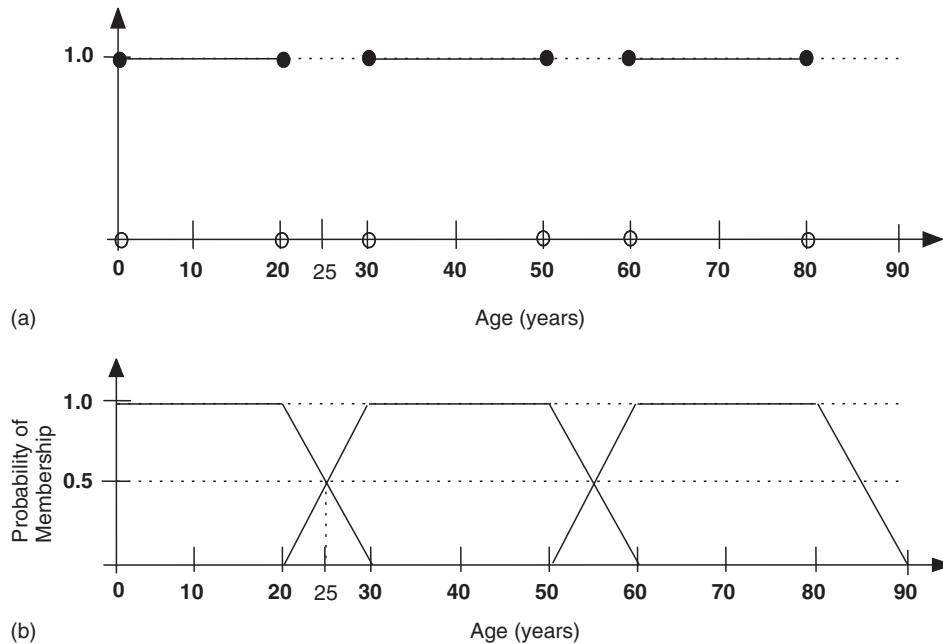


FIGURE 11.37 (a) Crisp sets for the classification of people by age. (b) Fuzzy sets for the classification of people by age.

fuzzy union operator, and [Figure 11.38c](#) shows the negation operator for fuzzy sets. The solid line indicates the result of the operator in each figure.

Although it is easy to form fuzzy sets for a simple example such as age classification, fuzzy sets for more sophisticated applications are derived by using sophisticated calibration techniques. The linguistic variables are formulated mathematically and then can be processed by computers. Once the fuzzy sets have been established, rules are constructed. Fuzzy logic is a rule-based logic. Fuzzy systems are constructed by using a large number of rules. Most rules used in fuzzy logic computing are if-then statements that use linguistic variables. Two simple rules that use the fuzzy sets for age classification might be the following:

If the patient is YOUNG, then use TREATMENT A.

If the patient is MIDDLE-AGED or OLD, then use TREATMENT B.

The degree of membership in a group helps to determine which rule will be used and, consequently, the type of action that will be taken or, in the preceding example, the sort of treatment that will be used. Defuzzification methods are used to determine which rules will be used to produce the final output of the fuzzy system.

For many applications, fuzzy logic has significant advantages over traditional numeric computing methods. Fuzzy logic is particularly useful when information is too limited or too complex to allow for numeric precision, since it tolerates imprecision. If an accurate mathematical model cannot be constructed, fuzzy logic may prove valuable. However, if

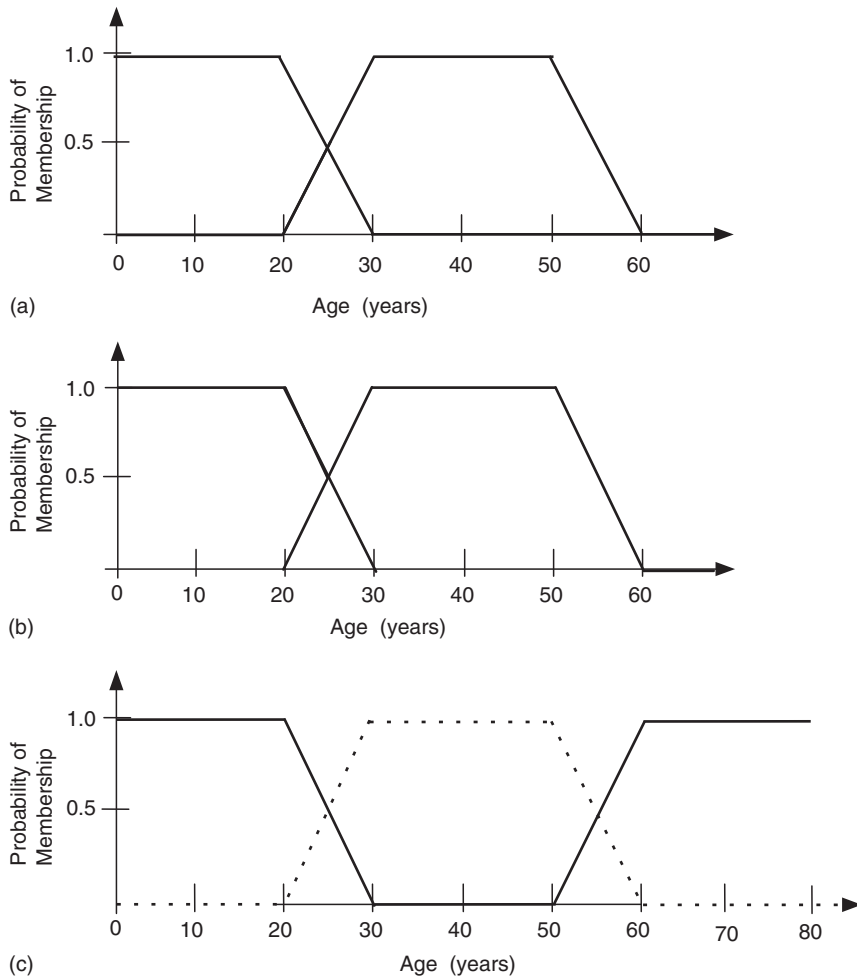


FIGURE 11.38 (a) Intersection of fuzzy sets: YOUNG AND MIDDLE-AGED. (b) Union of fuzzy sets: MIDDLE-AGED OR OLD. (c) Negation of fuzzy sets: NOT OLD.

a process can be described or modeled mathematically, then fuzzy logic will not generally perform better than traditional methods.

Biomedical engineering applications, which involve the analysis and evaluation of bio-signals, often have attributes that confound traditional computing methods but are well suited to fuzzy logic. Biological phenomena are often not precisely understood and can be extremely complex. Biological systems also vary significantly from one individual to the next individual. In addition, many key quantities in biological systems cannot be measured precisely due to limitations in existing sensors and other biomedical measuring devices. Sensors may have the capability to measure biological quantities intermittently or in combination with other parameters but not independently. Blood glucose sensors,

for example, are sensitive not only to blood glucose but also to urea and other elements in the blood. Fuzzy logic can be used to help compensate for the limitations of sensors.

Fuzzy logic is being used in a variety of biomedical engineering applications. Closed loop drug delivery systems, which are used to automatically administer drugs to patients, have been developed by using fuzzy logic. In particular, fuzzy logic may prove valuable in the development of drug delivery systems for anesthetic administration, since it is difficult to precisely measure the amount of anesthetic that should be delivered to an individual patient by using conventional computing methods. Fuzzy logic is also being used to develop improved neuroprosthetics for paraplegics. Neuroprosthetics for locomotion use sensors controlled by fuzzy logic systems to electrically stimulate necessary leg muscles and will ideally enable the paraplegic patient to walk.

EXAMPLE PROBLEM 11.28

A fuzzy system is used to categorize people by heart rates. The system is used to help determine which patients have normal resting heart rates, bradycardia, or tachycardia. Bradycardia is a cardiac arrhythmia in which the resting heart rate is less than 60 beats per minute, while tachycardia is defined as a cardiac arrhythmia in which the resting heart rate is greater than 100 beats per minute. A normal heart rate is considered to be in the range of 70–80 beats per minute. What are three linguistic variables that might be used to describe the resting heart rates of the individuals?

Solution

A variety of linguistic variables may be used. The names are important only in that they offer a good description of the categories and problem. Slow, normal, and fast might be used. Another possibility is simply bradycardia, normal, and tachycardia.

11.9.2 Artificial Neural Networks

Artificial neural networks (ANN) are the theoretical counterpart of real biological neural networks. The human brain is a highly sophisticated biological neural network, consisting of billions of brain cells (i.e., neurons) that are highly interconnected among one another. Such a highly interconnected architecture of neurons allows for immense computational power, typically far beyond our most sophisticated computers. The brains of humans, mammals, and even simple invertebrate organism (e.g., a fly) can easily learn from experience, recognize relevant sensory signals (e.g., sounds and images), and react to changes in the organism's environment. Artificial neuronal networks are designed to mimic and attempt to replicate the function of real brains.

ANNs are simpler than biological neural networks. A sophisticated ANN contains only a few thousand neurons with several hundred connections. Although simpler than biological neural networks, the aim of ANNs is to build computer systems that have learning, generalized processing, and adaptive capabilities resembling those seen in real brains. Artificial neural networks can learn to recognize certain inputs and to produce a particular output for a given input. Therefore, artificial neural networks are commonly used for pattern detection and classification of biosignals.

ANNs consist of multiple interconnected neurons. Different types of neurons can be represented in an ANN. Neurons are arranged in a layer, and the different layers of neurons are connected to other neurons and layers. The manner in which the neurons are interconnected determines the architecture of the ANN. There are many different ANN architectures, some of which are best suited for specific applications. Figure 11.39 shows a schematic of a simple ANN with three layers of neurons and a total of six neurons. The first layer is called the *input layer* and has two neurons, which accept the input to the network. The middle layer contains three neurons and is where much of the processing occurs. The *output layer* has one neuron that provides the result of the ANN.

Mathematical equations are used to describe the connections between the neurons. The diagram in Figure 11.40 represents a single neuron and a mathematical method for determining the output of the neuron. The equation for calculating the total input to the neuron is

$$x = (\text{Input}_1 \bullet \text{Weight}_1) + (\text{Input}_2 \bullet \text{Weight}_2) + \text{Bias Weight} \quad (11.59)$$

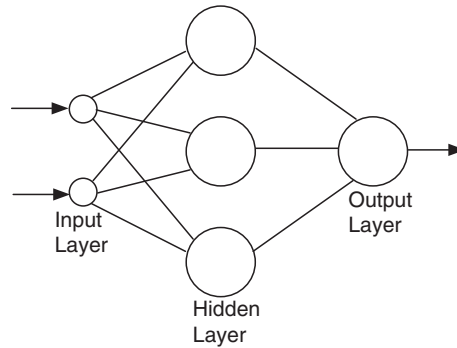


FIGURE 11.39 A simple artificial neural network (ANN) with six neurons and three layers.

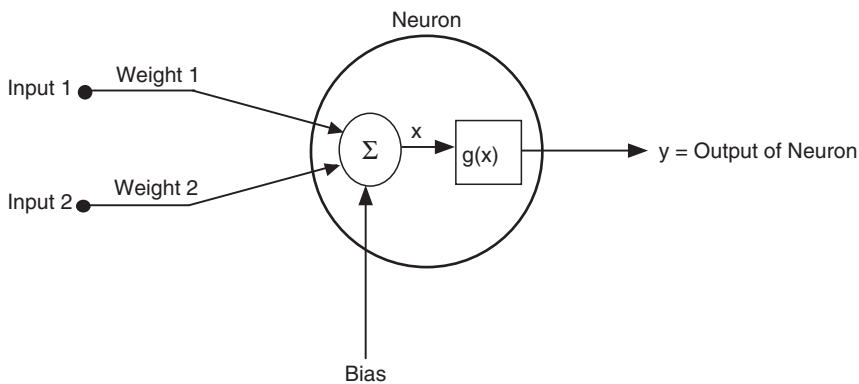


FIGURE 11.40 A single neuron showing mathematical input and output relationships.

The output for the neuron is determined by using a mathematical function, $g(x)$. Threshold functions and nonlinear sigmoid functions are commonly used. The output y of a neuron using the sigmoid function is calculated from the following simple equation:

$$y = 1/(1 + e^{-x}) \quad (11.60)$$

In biosignal processing applications, the inputs to the first layer or input layer of the ANN can be raw data, a preprocessed signal, or extracted features from a biosignal. Raw data are generally a sample from a digitized signal. Preprocessed signals are biosignals that have been transformed, filtered, or processed using some other method before being input to the neural network. Features can also be extracted from biosignals and used as inputs for the neural network. Extracted features might include thresholds; a particular, reoccurring waveshape; or the period between waveforms.

The ANN must learn to recognize the features or patterns in an input signal, but this is not the case initially. In order for the ANN to learn, a training process must occur in which the user of the ANN presents the neural network with many different examples of important input. Each example is given to the ANN many times. Over time, after the ANN has been presented with all of the input examples several times, the ANN learns to produce particular outputs for specific inputs.

There are a variety of types of learning paradigms for ANNs. Learning can be broadly divided into two categories: unsupervised learning and supervised learning. In unsupervised learning, the outputs for the given input examples are not known. The ANN must perform a sort of self-organization. During unsupervised learning, the ANN learns to recognize common features in the input examples and produces a specific output for each different type of input. Types of ANNs with unsupervised learning that have been used in biosignal processing include the Hopfield network and self-organizing feature maps networks.

In supervised learning, the desired output is known for the input examples. The output that the ANN produces for a particular input or inputs is compared against the desired output or output function. The desired output is known as the target. The difference between the target and the output of the ANN is calculated mathematically for each given input example. A common training method for supervised learning is backpropagation. The multilayered perceptron trained with backpropagation is a type of a network with supervised learning that has been used for biosignal processing.

Backpropagation is an algorithm that attempts to minimize the error of the ANN. The error of the ANN can be regarded as simply the difference between the output of the ANN for an input example and the target for that same input example. Backpropagation uses a gradient-descent method to minimize the network error. In other words, the network error is gradually decreased down an error slope that is in some respects similar to how a ball rolls down a hill. The name *backpropagation* refers to the way by which the ANN is changed to minimize the error. Each neuron in the network is "credited" with a portion of the network error. The relative error for each neuron is then determined, and the connection strengths between the neurons are changed to minimize the errors. The weights, such as those that were shown in [Figure 11.40](#), represent the connection strengths between neurons. The calculations of the neuron errors and weight changes propagate backward through the ANN from the output neurons to the input neurons. Backpropagation is the method of finding the optimum weight values that produce the smallest network error.

ANNs are well suited for a variety of biosignal processing applications and may be used as a tool for nonlinear statistical analysis. They are often used for pattern recognition and classification. In addition, ANNs have been shown to perform faster and more accurately than conventional methods for signals that are highly complex or contain high levels of noise. ANNs also have the ability to solve problems that have no algorithmic solution—in other words, problems for which a conventional computer program cannot be written. Since ANNs learn, algorithms are not required to solve problems.

As advances are made in artificial intelligence techniques, ANNs are being used more extensively in biosignal processing and biomedical instrumentation. The viability of ANNs for applications ranging from the analysis of ECG and EEG signals to the interpretation of medical images and the diagnosis of a variety of diseases has been investigated. In neurology, research has been conducted by using ANNs to characterize brain defects that occur in disorders such as epilepsy, Parkinson's disease, and Alzheimer's disease. ANNs have also been used to characterize and classify ECG signals of cardiac arrhythmias. One study used an ANN in the emergency room to diagnose heart attacks. The results of the study showed that, overall, the ANN was better at diagnosing heart attacks than the emergency room physicians were. ANNs have the advantage of not being affected by fatigue, distractions, or emotional stress. As artificial intelligence technologies advance, ANNs may provide a superior tool for many biosignal processing tasks.

EXAMPLE PROBLEM 11.29

A neuron in a neural network has three inputs and uses a sigmoid function to calculate the output of the neuron. The three values of the inputs are 0.1, 0.9, and 0.1. The weights associated with these three inputs are 0.39, 0.72, and 0.26, and the bias weight is 0.48 after training. What is the output of the neuron?

Solution

Using Eq. (11.32) to calculate the relative sum of the inputs gives

$$\begin{aligned} x &= (\text{Input}_1 \bullet \text{Weight}_1) + (\text{Input}_2 \bullet \text{Weight}_2) + (\text{Input}_3 \bullet \text{Weight}_3) + \text{Bias Weight} \\ &= (0.1) 0.39 + (0.9) 0.72 + (0.1) 0.24 + 0.48 \\ &= 1.19 \end{aligned}$$

The output of the neuron is calculated using Eq. (11.60):

$$y = 1/(1 + e^{-x}) = 1/(1 + e^{-1.19}) = 0.77$$

11.10 EXERCISES

1. What types of biosignals would the nerves in your legs produce during a sprint across the street?
2. What types of biosignals can be recorded with an EEG? Describe in terms of both origins and characteristics of the signal.

Continued

3. Describe the biosignal that the electrical activity of a normal heart would generate during a bicycle race.
4. A 16-bit A/D converter is used to convert an analog biosignal with a minimum voltage of -30 mV and a maximum voltage of 90 mV. What is the sensitivity?
5. An EMG recording of skeletal muscle activity has been sampled at 200 – 250 Hz and correctly digitized. What is the highest frequency of interest in the original EMG signal?
6. Two signals, $x_1(t)$ and $x_2(t)$, have the magnitude spectrum shown in [Figure 11.41](#). Find the Nyquist rate for:
 - (a) $x_1(t)$
 - (b) $x_2(t)$
 - (c) $x(t) = x_1(t) * x_2(t)$
 (Hint: Apply the convolution theorem.)

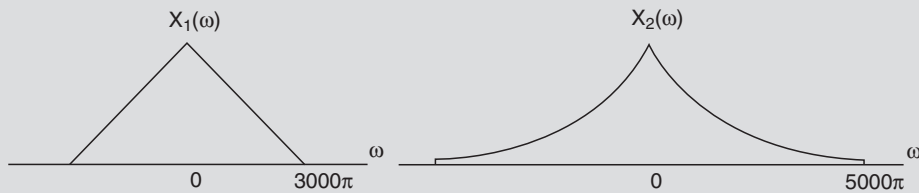


FIGURE 11.41

7. Consider the signal

$$x(t) = 3 + \sin(2\pi 100t) + \cos(2\pi 250t + \pi/3)$$

Find the Nyquist frequency.

8. A sinusoid with the frequency of 125 kHz is sampled at $70,000$ samples per second. What is the apparent frequency of the sampled signal?
9. An electroencephalographic (EEG) signal has a maximum frequency of 300 Hz. The signal is sampled and quantized into a binary sequence by an A/D converter.
 - (a) Determine the sampling rate if the signal is sampled at a rate 50 percent higher than the Nyquist rate.
 - (b) The samples are quantized into $2,048$ levels. How many binary bits are required for each sample?
10. Find the exponential Fourier series for the signal shown in [Figure 11.42a](#).
11. Find the exponential Fourier series for the signal shown in [Figure 11.42b](#).
12. $f(t)$ is a periodic signal shown in [Figure 11.43](#). Find its trigonometric Fourier series.
13. Consider the following trigonometric Fourier series:

$$f(t) = 3 + 3 \cos(t) + 2 \cos(2t) + 4 \sin(2t) - 4 \left(\frac{e^{j4t} + e^{-j4t}}{2} \right).$$

Write $f(t)$ in its compact trigonometric Fourier series form.

14. Explain why the exponential Fourier series requires negative frequencies.

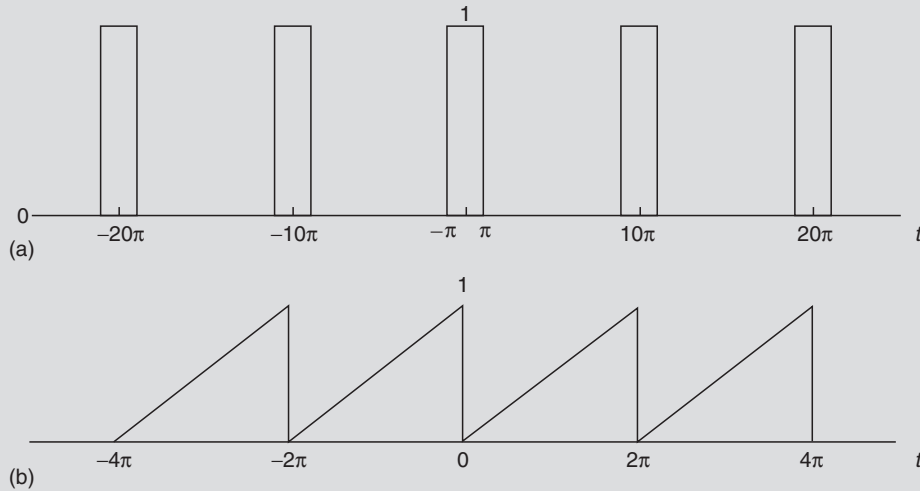


FIGURE 11.42

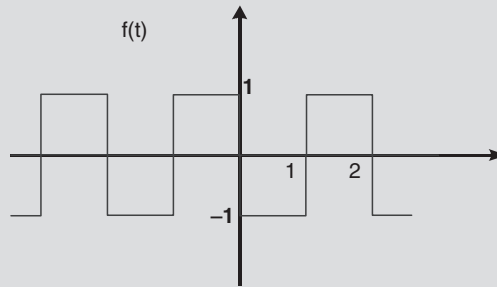


FIGURE 11.43

15. Find the Fourier transform of
 - (a) $u(t)$
 - (b) $e^{-at}u(t)$
 - (c) $\cos(at)u(t)$
16. Find the Fourier transform of $f_1(t) = e^{-3t}u(t)$.
17. Find the Fourier transform of $f(t) = e^{-3|t|}$ and sketch its time and frequency domain representations. (Hint: Find a few points on the curve by substituting values for the variable.)
18. Prove the shift property of the Fourier transform.
19. Given $x(t) = e^{-at}u(t)$ and $h(t) = e^{-bt}u(t)$, where a and b are constants greater than zero, explain why it would be easier to evaluate the convolution $x(t) * h(t)$ in the frequency domain.
20. A brief current pulse of duration 50 ms and amplitude 1 mA is presented to a cell membrane with time constant 10 ms. Find the cell membrane voltage output.

Continued

21. The ion exchange process of a cell is estimated to have the following impulse response:
 $h(t) = e^{-4t}u(t)$.
- Explain what type of general information would be available to the researcher if this estimation of $h(t)$ were accurate.
 - If sodium ions are injected into the system for two seconds in the form of a brief pulse approximated by the equation $x(t) = 3u(t) - 3u(t - 2)$, how would the cell respond to (e.g., pump out ions) such input? Find the answer using time domain procedures. (Hint: Convolve the input and the impulse response.)
22. An ECG recording of the electrical activity of the heart during ventricular fibrillation is digitized, and the signal begins with the data sequence $[-90.0, 10.0, -12.0, -63.0, 7.0, -22.0]$. The units of the data sequence are given in mV. What is the z-transform of this data sequence of the biosignal?
23. For the systems described by the following equations, determine which of the systems is linear and which is not.
- $\frac{dy}{dt} + 2y(t) = f^2(t)$
 - $\frac{dy}{dt} + 3ty(t) = t^2f(t)$
 - $\frac{dy}{dt} + 2y(t) = f(t)\frac{df}{dt}$
 - $y(t) = \int_{-\infty}^t f(\tau)d\tau$

24. Examine the characteristics of the digital filter

$$y(k) = \frac{1}{4}x(k) + \frac{1}{4}x(k-1) + \frac{1}{2}y(k-1).$$

Find the impulse response, $H(z)$ and $H'(\Omega)$. Use MATLAB to calculate and plot $|H'(\Omega)|$ for $0 < \Omega < \pi$. Observe the difference between this filter and the filter in Example Problem 10.12. Why is this a better low-pass filter? What is the output if the input sequence is $x(k) = 100\sin\left(\frac{\pi}{2}k + \frac{\pi}{8}\right)$? What is the output if the input sequence is $x(k) = 100u(k)$?

25. Find the z-transform of
- $x(k) = u(k)$
 - $x(k) = a^k u(k)$
 - $x(k) = \cos(b \cdot k)u(k)$
26. Find the z-transform of the following:
- $x[k] = \left(\frac{1}{2}\right)^k u(k)$
 - $x[k] = (\cos \Omega k)u[k]$
27. Find the first four outputs of the discrete system

$$y[k] - 3y[k-1] + 2y[k-2] = f[k-1]$$

if $y[-1] = 2$, $y[-2] = 3$, and $f[k] = 3^k u[k]$.

28. Find the first four outputs of the discrete system

$$y[k] - 2y[k-1] + 2y[k-2] = f[k-2]$$

if $y[-1] = 1$, $y[-2] = 0$, and $f[k] = u[k]$.

29. In MATLAB, design a routine to show that averaging random noise across many trials approaches zero as the number of trials increases.
30. Accurate measurements of blood glucose levels are needed for the proper treatment of diabetes. Glucose is a primary carbohydrate that circulates throughout the body and serves as an energy source for cells. In normal individuals the hormone insulin regulates the levels of glucose in the blood by promoting glucose transport out of the blood to skeletal muscle and fat tissues. Diabetics suffer from improper management of glucose levels, and the levels of glucose in the blood can become too high. Describe how fuzzy logic might be used in the control of a system for measuring blood glucose levels. What advantages would the fuzzy logic system have over a more conventional system?
31. Describe three different biosignal processing applications for which artificial neural networks might be used. Give at least two advantages of artificial neural networks over traditional biosignal processing methods for the applications you listed.
32. The fuzzy sets in [Example Problem 11.28](#) have been calibrated so a person with a resting heart rate of 95 beats per minute has a 75 percent degree of membership in the normal category and a 25 percent degree of membership in the tachycardia category. A resting heart rate of 65 beats per minutes indicates a 95 percent degree of membership in the normal category. Draw a graph of the fuzzy sets.

Suggested Readings

- M. Akay, *Biomedical Signal Processing*, Academic Press, Inc., San Diego, CA, 1994.
- M. Akay (Ed.), *Time Frequency and Wavelets in Biomedical Signal Processing*, IEEE Press, New York, NY, 1998.
- P. Bauer, S. Nouak, R. Winkler, *A Brief Course in Fuzzy Logic and Fuzzy Control*, Fuzzy Logic Laboratorium Linz-Hagenberg, Linz, Austria, 1996. <http://www.flll.uni-linz.ac.at/fuzzy>.
- C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press Inc., New York, NY, 1995.
- E.N. Bruce, *Biomedical Signal Processing and Signal Modeling*, Wiley-Interscience, New York, NY, 2000.
- E.J. Ciaccio, S.M. Dunn, M. Akay, *Biosignal Pattern Recognition and Interpretation Systems: Part 1 of 4: Fundamental Concepts*, IEEE Eng. Med. Biol. Mag. 12 (1993) 810–897.
- E.J. Ciaccio, S.M. Dunn, M. Akay, *Biosignal Pattern Recognition and Interpretation Systems: Part 2 of 4: Methods for Feature Extraction and Selection*, IEEE Eng. Med. Biol. Mag. 12 (1993) 106–113.
- E.J. Ciaccio, S.M. Dunn, M. Akay, *Biosignal Pattern Recognition and Interpretation Systems: Part 3 of 4: Methods of Classification*, IEEE Eng. Med. Biol. Mag. 12 (1994) 269–279.
- E.J. Ciaccio, S.M. Dunn, M. Akay, *Biosignal Pattern Recognition and Interpretation Systems: Part 4 of 4: Review of Applications*, IEEE Eng. Med. Biol. Mag. 13 (1994) 269–273.
- A. Cohen, *Biomedical Signal Processing: Volume I Time and Frequency Domain Analysis*, CRC Press, Boca Raton, FL, 1986.
- A. Cohen, *Biomedical Signal Processing: Volume II Compression and Automatic Recognition*, CRC Press, Boca Raton, FL, 1986.
- J. Dempster, *Computer Analysis of Electrophysiological Signals*, Academic Press Inc., San Diego, CA, 1993.
- S.R. Devasahayam, *Signals and Systems in Biomedical Engineering: Signal Processing and Physiological Systems Modeling*, Kluwer Academic, New York, NY, 2000.

- S. Haykin, *Neural Networks—A Comprehensive Foundation*, Macmillan College Publishing Co., Inc., New York, NY, 1994.
- R.B. Northrop, *Signals and Systems Analysis in Biomedical Engineering*, CRC Press, Boca Raton, FL, 2003.
- B. Onaral (Ed.), VI. Biomedical signal analysis, in: J.D. Bronzino (Ed.), *The Biomedical Engineering Handbook*, CRC Press, Boca Raton, FL, 1995.
- A.V. Oppenheim, R.W. Schaffer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.
- A.V. Oppenheim, A.S. Willsky, I.T. Young, *Signals and Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.
- R.A. Roberts, C.T. Mullis, *Digital Signal Processing*, Addison-Wesley Publishing Co., Reading, MA, 1987.
- M. Smith, *Neural Networks for Statistical Modeling*, International Thomson Computer Press, Boston, MA, 1996.
- S.D. Stearns, R.A. David, *Signal Processing Algorithms in Fortran and C*, P T R Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.
- W.J. Thompkins, *Biomedical Digital Signal Processing*, P T R Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.
- C.S. Williams, *Designing Digital Filters*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.
- L.A. Zadeh, *Fuzzy Sets and Applications*, John Wiley & Sons, New York, NY, 1987.
- R.E. Ziemer, W.H. Tranter, D.R. Fannin, *Signals and Systems: Continuous and Discrete*, third ed., Macmillan Publishing Co., New York, NY, 1993.