CHAPTER 6

Data Acquisition and Signal Processing

Chapter Outline

6.1 Introduction

Sensors to measure variables like temperature, pressure, and strain are an essential part of measurement systems. The later chapters (Chapter 14 onward) provide a comprehensive coverage of the sensors available for measuring a range of physical variables. Although it is possible to use oscilloscopes or multimeters to monitor these variables (generally in conjunction with added *signal conditioning* devices), it is usually preferable to use a personal computer (PC) to view and record the data through the use

Measurement and Instrumentation. http://dx.doi.org/10.1016/B978-0-12-800884-3.00006-X

of a Data Acquisition (DAQ) device (typically a specialized card within the PC or a USB-based device external to the PC). One particular advantage of using PCs in this respect is that data can easily be stored and converted to a format that can be used by *spreadsheets* (such as Microsoft Excel) or other *software* packages such as Matlab for more extensive analysis. Another advantage is that significant *digital processing* of the data can be performed in real time via the same platform that is used to acquire the data. This can significantly improve the process of performing an experiment by making the real-time data interpretable and useful for subsequent actions (e.g., decisions that have to be made on the basis of the acquired data).

It is a prerequisite of DAQ that the measurement system is designed to maximize the quality of sensor outputs and minimize the occurrence of systematic error, as discussed in Chapter 3. However, it is usually impossible to eliminate all errors, and it is often difficult to achieve sensors outputs of a suitable amplitude. Thus, in general, signals acquired via sensors must be processed or conditioned to make them suitable for subsequent analysis or for integration within control loops. Since signals acquired via common sensors are often *analog*; that is, continuous in time and in range of values, some level of analog signal processing is generally required. This is to (1) amplify weak sensor signals, (2) filter the inherent noise that is usually mixed in with the original signal, and (3) eliminate high-frequency components in the signal that would cause *aliasing*; that is, presence of dubious counterparts of the original signal when sampled below a certain sampling rate commonly referred to as the *Nyquist rate*. Signal processing is also usually necessary to deal with induced noise that arises when signals are transmitted from sensors to other parts of the measurement system, as discussed in Section 3.6 of Chapter 3, even when efforts have been made to minimize the magnitude of this induced noise as far as possible.

Analog signal processing is therefore essential to the process of DAQ even if one is largely reliant on a (digital) computer-based system for a significant part of the overall DAQ process. With this in mind, in this chapter, we shall initially consider the essential aspects of *analog* signal processing with a view toward those techniques that are meant to address the three aforementioned concerns (amplification, filtering, and anti-aliasing). We shall subsequently consider how *digital* signal processing techniques, in their abstract form as well as via computer-based tools such as Matlab/Simulink and Microsoft Excel, can be used to implement more sophisticated filtering and data analysis techniques. Clearly, for real-time implementation of digital signal processing techniques, additional software tools are necessary, as for instance the Matlab Real-time Workshop, in conjunction with digital signal processing hardware such as those manufactured by dSpace, Inc., National Instruments, among others. These implementation issues are not as much discussed in this chapter, however, since they require additional expertise beyond that which can be reasonably covered in an introductory course. Later in the book, however, we will consider how real-time implementation of digital signal processing techniques can be accomplished via LabVIEW©, which is perhaps the standard tool for computer-based DAQ systems.

With the above mind, the following basic concepts are discussed in the current chapter:

- DAQ process and its components
- Definition of frequency spectrum of an analog signal
- Discrete time representation of analog signals: sampling and quantization
- Aliasing phenomenon: Nyquist frequency and Nyquist rate
- Analog filters and their implementation in passive and active form
- Digital filters and their implementation in Matlab and Excel

In completing this chapter, the reader should be able to understand the basic processes that are required in transforming a sensor signal from its raw form into a form that is suitable for subsequent analysis in a digital computer.

6.2 Preliminary Definitions

As stated above, common sensor (or transducer) signals are often *analog*; that is, continuous in time and in range of values as schematically depicted in Figure 6.1.

The signal shown in the figure is given by

$$x(t) = 1 + \sin(0.15\pi t) + 0.5\,\sin(0.25\pi t) \tag{6.1}$$

where t represents time in seconds and x denotes the amplitude of the signal. This particular signal may be associated with a sensor reading with virtually no noise present. It has a constant (or DC) offset of 1 unit, and is additionally composed of two sinusoidal components of frequencies $\omega_1 = 0.15\pi$ rad/s and $\omega_2 = 0.25\pi$ rad/s. These frequencies are

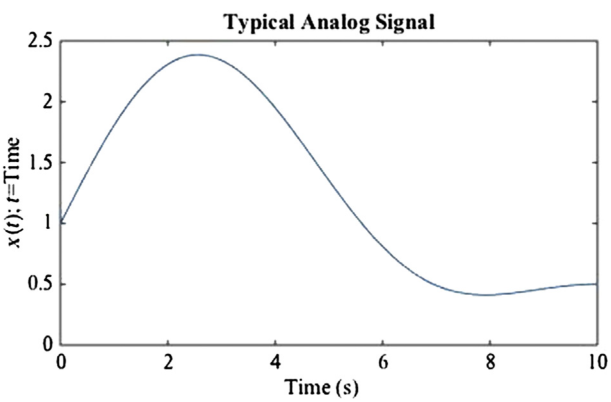

Figure 6.1
Typical analog signal.

more commonly stated as $f_1 = \omega_1/2\pi = 0.075$ Hz (Cycles/s) and $f_2 = \omega_2/2\pi = 0.125$ Hz. Note that $x(t)$ is a simple two-component signal with a maximum frequency of $f_2 = 0.125$ Hz. Note also that ω is typically used to denote the frequency of a given signal in radians per second (rad/s) while f is used to denote the frequency in cycles per second or Hertz (Hz).

We consider the above signal, $x(t)$, to be *band-limited* with a *bandwidth* of 0.125 Hz where the term *band* (or *bandwidth*) signifies the spread of the signal over a given frequency range. In addition, the term *spectrum*, used shortly, refers to the frequency content of the signal over the given range. In this case, the *spectrum* of the signal is *discrete* (or made up of individual components) as shown in Figure 6.2 where it is evident that the amplitude of the signal is zero at all frequencies except at 0.075 and 0.125 Hz.

In realistic situations, sensor signals (as shown for instance in the subsequent section) are much more complex. If a signal is truly periodic, it can be represented as a (possibly infinite) sum of sinusoidal components of different amplitudes (its Fourier Series) as for instance described below:

$$x(t) = X_1 \sin(\omega t + \phi_1) + X_2 \sin(2\omega t + \phi_2) + X_3(3\omega t + \phi_3) + \ldots \tag{6.2}$$

where X_i, $i = 1, 2, \ldots$ are the *amplitudes* of the components of the signal at frequencies that are multiples of the *fundamental frequency* (the lowest frequency or the overall periodic rate) of the original signal, ω. The spectrum of the signal would be discrete as the case above but with components at each of the so-called *harmonics* (multiples of the fundamental frequency). A typical such signal is depicted in Figure 6.3.

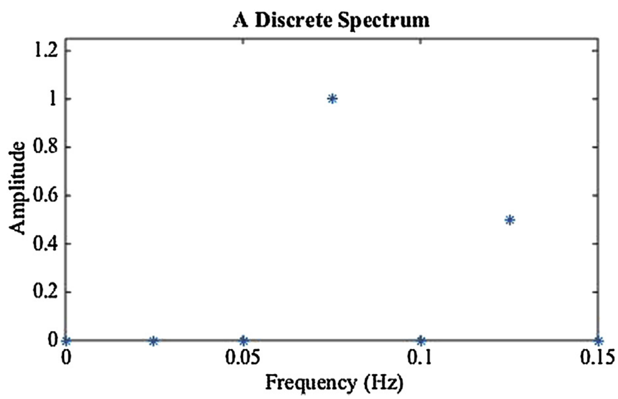

Figure 6.2
Discrete spectrum of a simple signal.

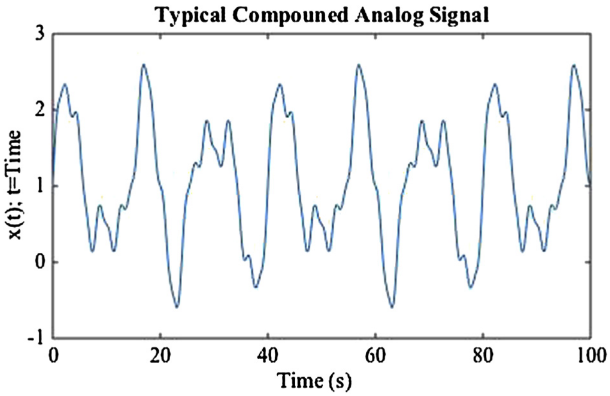

Figure 6.3
Typical compound sinusoidal signal.

The signal in this case is given by

$$x(t) = 1 + \sin(0.125\pi t) + 0.5\,\sin(0.25\pi t) + 0.25\,\sin(0.5\pi t) + 0.125\,\sin(\pi t) \tag{6.3}$$

which is the sum of four sinusoids. The fundamental frequency of the signal is $\omega = 0.125\pi$ rad/s or $f = \omega/2\pi = 1/16$ Hz. The harmonics at 0.25π rad/s or 1/8 Hz, 0.5π rad/s or 1/4 Hz, and finally at π rad/s or 1/2 Hz are also present in the signal. As complex as this signal looks, it is still periodic with a period of $T = 2\pi/\omega = 2\pi/0.125\pi = 16$ s and remains band limited. In general, a given signal may not even be periodic nor band limited, in which case it must be represented by a *Fourier Transform*, which is an *integral* as opposed to a sum. In this case, the spectrum of the signal will be continuous, albeit with certain peaks at significant frequencies. We shall generally assume that the given signal is indeed band limited with a frequency spectrum that is capped at a certain maximum value, called the *bandwidth* of the signal.

6.3 Sensor Signal Characteristics

In general, signals acquired from sensors are either weak and small in magnitude or contaminated with noise or both. For instance, the original voltage reading generated by a thermocouple is in the range of microvolts, making it difficult to process via a DAQ system without appropriate amplification and other types of preconditioning. Similarly, the signal generated by an accelerometer is typically very noisy as it is, for instance, depicted in Figure 6.4. In the figure, a three-axis accelerometer attached to a stationary object shows variations that are quite consistent with the nature of this type of measurement. The expected signal is constant and all the variations are in this case due to sensor noise (although this is not purely due to electrical noise as in many other cases). The so-called *spectrum* of this type of noise is often *wide*, necessitating preconditioning the signal using

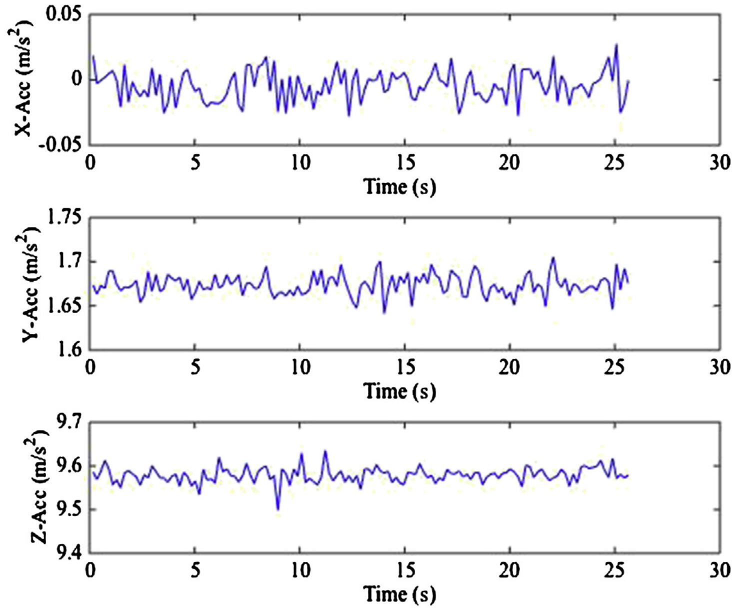

Figure 6.4
Typical accelerometer signal in three channels.

an analog filter prior to acquisition with a digital DAQ. This has to do with the so-called *aliasing* effect that can seriously undermine any digital signal processing system if it is not properly sampled. In general, aliasing occurs if the bandwidth of the analog signal that is being sampled by the system at discrete intervals exceeds the so-called *Nyquist frequency* (1/2 the sampling rate) of the sampling system (or digitizer, as for instance the DAQ card used to acquire the signal). This issue is discussed below at more length.

6.4 Aliasing

When a high-frequency signal is sampled by a computer DAQ, only a limited amount of information is captured (once per sample). If the sampling rate is sufficiently high, the loss of fidelity is not significant. However, as the sampling rate is reduced, the signal looks less and less like the original. This is depicted in Figure 6.5 where a sinusoidal function of 1 Hz (or 1 cycle/s) frequency is sampled initially at 50 samples/s, later at 5 samples/s, and finally at 2 samples/s. Clearly the discretized representation of the original analog signal is very realistic at high sampling rates and degrades perceptibly as the sampling rate is reduced. Below the final limit of 2 samples/s, which in this case corresponds to the so-called *Nyquist rate* of the signal (twice its maximum frequency), the sampled signal can no longer be guaranteed to represent its original analog version. Formally, the Nyquist rate of a band-limited signal is twice the maximum frequency of that signal. Alternatively one can view

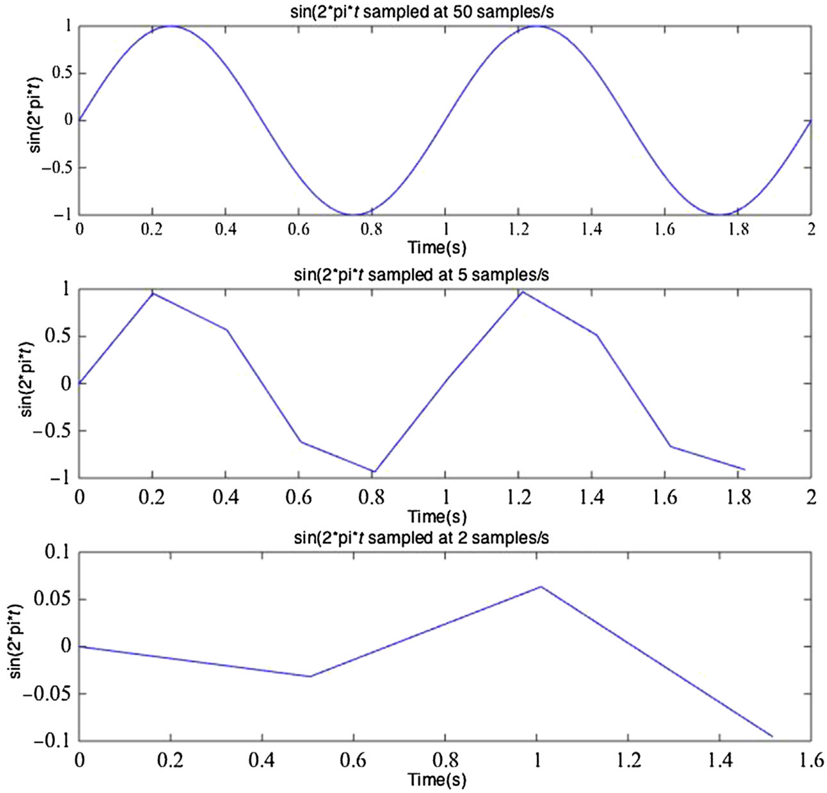

Figure 6.5
Sampling a signal at different rates.

the situation from the perspective of the DAQ and consider the *Nyquist frequency*, which is half the sampling rate and which serves as the upper limit for the maximum frequency of a band-limited analog signal that can be accurately[1] captured by a discrete-time sampling process.

In more specific terms, a signal that is sampled at a rate that is below its Nyquist rate will appear to include low-frequency *aliases* of the higher frequency components that are contained in the original signal and exceed the Nyquist frequency of the sampling system. This can be made more precise by considering the *digital frequency* of a signal. Assuming that we are dealing with a continuous time domain signal of the form

$$x(t) = \sin(2\pi ft + \phi) \tag{6.4}$$

[1] Accuracy in this context is theoretical. In practice, a sampling rate of higher than the Nyquist rate is necessary to capture the signal effectively.

In discrete time (or sampled data) form this signal is stated as follows:

$$x[k] = \sin(2\pi Fk + \phi) \tag{6.5}$$

where k is the sample count; that is, effectively $t = kT_s$ where T_s is the sampling interval. The relationship between the original analog frequency, f, and the digital frequency, F, is obtained by a comparison of the above equations while noting that $f_s = 1/T_s$ is the sampling frequency of the system. Starting with the fact that

$$ft = Fk \tag{6.6}$$

we have

$$F = \frac{ft}{k} = \frac{fkT_s}{k} = fT_s = \frac{f}{f_s} \tag{6.7}$$

Given the property of the sin() function that adding any multiples of 2π (say $2\pi m$ where m is any integer[2]) to its argument does not change the value of the function, we have the following result for the sampled signal $x[k]$:

$$x[k] = \sin(2\pi Fk + \phi) = \sin(2\pi Fk + 2\pi Nk + \phi) = \sin(2\pi(F + N)k + \phi) \tag{6.8}$$

This implies that a signal of frequency $F + N$ is captured as if it were of frequency F. In other words a high-frequency signal is aliased at a lower frequency. In general, the digital frequency, F, is less than $^1\!/_2$ since the analog frequency, f, must be less than $^1\!/_2 f_s$ for the signal to be accurately represented in discrete-time form. For reasons that are not discussed at length here, the more complete technical assumption is that

$$-\frac{1}{2} \le F \le \frac{1}{2} \tag{6.9}$$

This means that F can be negative although this fact has little bearing on the discussion at hand. The range between $-^1\!/_2$ and $^1\!/_2$ is called the *principal range* for F. If an analog frequency, f, is such that its digital equivalent, F, falls outside this principal range, then it would have aliases within the principal range. For instance if an 8 Hz signal is sampled at 6 Hz, that is, well below the required Nyquist rate of 16 Hz, we will have

$$F = \frac{8}{6} = \frac{4}{3} > \frac{1}{2} \tag{6.10}$$

In other words, since F falls outside the principal range, it will be aliased at a lower frequency that would fall within the principal range. The nature of the alias is determined as follows. In order to place the alias within the principal range, we will need to subtract a whole integer, $N = 1$, from 4/3, which produces $F' = 1/3$ (within the principal range).

[2] Later we will write $m = Nk$ where N is a fixed integer and k, also an integer, varies from zero to infinity as a sample-time index.

The resulting frequency is in effect a much lower frequency alias of the original signal, that is, its equivalent analog frequency is

$$f' = F'f_s = \frac{1}{3}6 = 2\ \text{Hz} \tag{6.11}$$

In other words, the 8 Hz signal is "equivalent" to a 2 Hz signal in its discrete-time representation. One can illustrate this point via a Matlab/Simulink model as depicted in Figure 6.6.

In this case two signals, one low frequency (2 Hz) and the other, high frequency (8 Hz), are discretized at the same rate (6 Hz) and both produce the same discrete-time signal as illustrated. The original analog signals (actually discretized at high sample rates) are shown on the left. The same signals sampled at 6 Hz are shown at the bottom of the figure. The relative offset is intentionally introduced to distinguish the two signals whereas the frequencies and amplitudes are clearly the same.

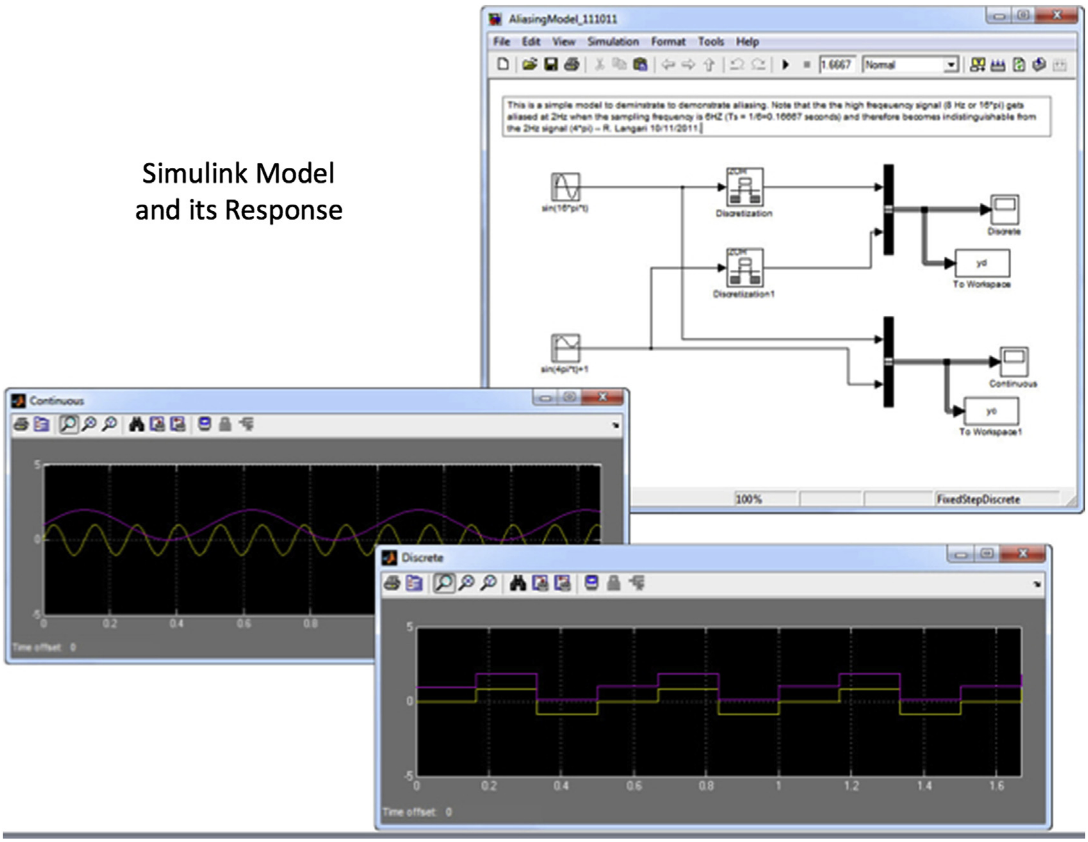

Figure 6.6
Illustration of aliasing effect via Matlab.

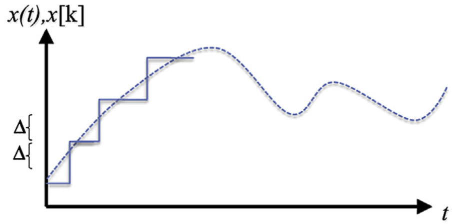

Figure 6.7
Quantization effect.

6.5 Quantization

Another impact of discretization of analog signals is quantization whereby the given signal is represented at a finite resolution given the ability of the analog to digital conversion process. For instance a possible situation is depicted in Figure 6.7 where the analog signal is quantized at the resolution of Δ, which depends on the resolution of the A/D conversion process. In particular if a 12-bit A/D converter has a range of ± 10 V, then its quantization level is given by

$$\Delta = \frac{20}{2^{12}} = 0.005 \text{ V} \tag{6.12}$$

or 5 mV. In other words, signal values that fall between the levels defined by this quantization are rounded up or down to the next level, which leads to the schematic graph shown in Figure 6.7. As shown, the analog signal is rounded up or down to the nearest multiple of the resolution of the A/D converter (i.e., Δ).

6.6 Analog Signal Processing

As noted earlier, the concepts of frequency spectrum and bandwidth are essential in understanding the nature of sensor signals. These notions are not only fundamental to the way sensor signals are interpreted, but also essential to the manner in which sensor signal processing mechanisms are devised. For instance, to eliminate noise, which has a high-frequency spectrum, a *low-pass* filter is used. There also instances such as in processing linear variable differential transducer (LVDT) signals that a high-pass filter is utilized. Finally there are instances where the desired spectrum is constrained within a certain range in which case a *band-pass* filter is used. Finally, there are cases where a specific undesirable frequency such as the 60 Hz power line frequency must be eliminated in which case a *notch* filter is used.

In the sequel, we will discuss the design of *passive* and *active* filters (and amplifiers). Passive filters, as the name implies, are made up of passive components such as resistors and capacitors. They are simple to design and implement. However, they can excessively load the source circuit and thereby undermine the signal processing process. In addition, passive filters do not provide for a *robust* design as their frequency response characteristics (their functional bandwidth) varies with the characteristics of the circuitry in which they are integrated.

Active circuits (filters and amplifiers), on the other hand, incorporate operational amplifiers or transistors, and are robust and more effective as signal processing devices. They do require a power source, which may or may not be an issue depending on the circumstances in which they are deployed. Active circuits are also more complex to design and implement and are more costly. Nevertheless in most practical signal processing applications, active filters (as filters or as amplifier/filters) are preferred to passive ones for the reasons noted above.

6.7 Passive Filters

Passive filters can be implemented with a few simple electronic components (resistors and capacitors). The basic *low-pass* filter, depicted in Figure 6.8, can be used attenuate high-frequency noise in the original signal, in this case denoted by v_i. The underlying assumption here is that any function of time can be viewed as a combination of sinusoidal functions as alluded to earlier. More exactly any *periodic* function of time is approximated by an infinite series of sinusoids at frequencies that are multiples of the so-called fundamental frequency of the original signal (the so-called *spectral content* of the signal). Nonperiodic functions can be viewed in essentially the same way but we must allow for a continuous spectrum. For example, the original signal, v_i, may be approximated by

$$v_i = V_{i,1}\sin(\omega t) + V_{i,2}\sin(2\omega t) + V_{i,3}\sin(3\omega t) + \ldots \tag{6.13}$$

where $V_{i,1}$, $V_{i,2}$, $V_{i,3}$,... are the amplitudes of the sequentially higher frequency components or (super) *harmonics* of the original signal.

These higher frequency components may represent fluctuations (or in many cases, electrical noise) that we may wish to attenuate to prevent aliasing and generally to present a clean signal to the DAQ system. The filter in Figure 6.8 produces an output, v_0, which has the *same set* of components (in terms of the respective frequencies) as the original signal v_i but at reduced amplitude:

$$v_0 = V_{0,1}\sin(\omega t) + V_{0,2}\sin(2\omega t) + V_{0,3}\sin(3\omega t) + \ldots \tag{6.14}$$

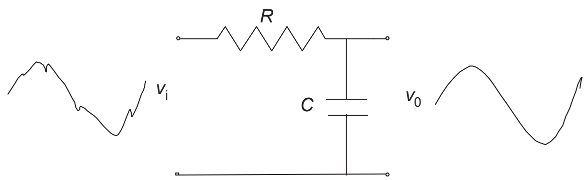

Figure 6.8
Passive low-pass filter.

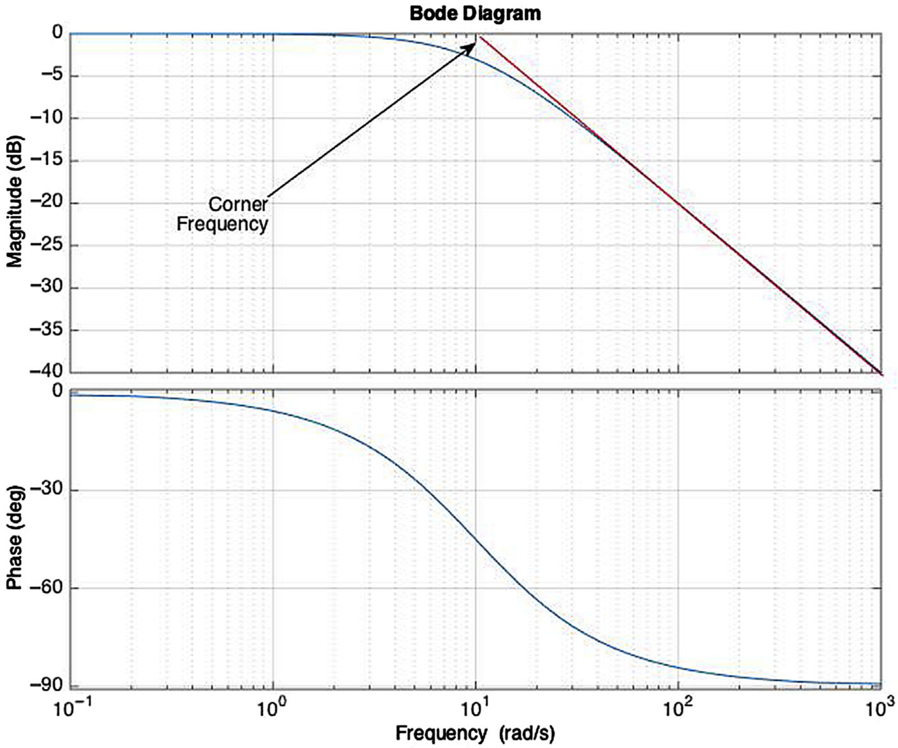

Figure 6.9
Bode plot of the passive low-pass filter.

where $V_{0,1}, V_{0,2}, V_{0,3}, \ldots$ are the amplitudes of the sinusoidal components in v_0. In general, $V_{0,1}, V_{0,2}, V_{0,3}, \ldots$ are *smaller* than their counterparts in the input signal, $V_{i,1}, V_{i,2}, V_{i,3}, \ldots$. For instance, depending on the values of R and C in the filter shown in Figure 6.8, $V_{0,1}$ may be very close to $V_{i,1}$, say 98% of this value, but $V_{0,2}$ may be about 70% of $V_{i,2}$ and so forth. The reason for this is that the given low-pass filter attenuates each signal according to its frequency; the higher the frequency, the larger the attenuation (hence the smaller the amplitude of the given component in the output signal). The precise amount of attenuation can be found from the so-called frequency response graph of the filer (commonly referred to as its *bode* plot) as for instance is depicted in Figure 6.9.

6.7.1 Filter Transfer Function

The specific equation for the above filter from the application of Kirchoff's current law at the output node (assuming an open circuit or no load at the output) is given by

$$\frac{v_i - v_0}{R} + C\frac{\mathrm{d}}{\mathrm{d}t}(v_0) = 0 \tag{6.15}$$

This can be simplified into

$$RC\frac{dv_0}{dt} + v_0 = v_i \tag{6.16}$$

As a first-order differential equation, this equation can be transformed into *transfer function* form, using Laplace Transform, as

$$\frac{\hat{v}_0}{\hat{v}_i} = \frac{1}{\tau s + 1} \tag{6.17}$$

where $\tau = RC$ is the time constant of the filter; that is, essentially the time it takes for the filter to respond to a step input function by reaching 63% (almost 2/3) of its steady state output.

The inverse of τ, that is, $\omega_c = 1/\tau$ is known as the *corner frequency* of the filter, that is, the frequency above which the filter starts to attenuate its input. In Figure 6.9, whose derivation is discussed in Section 6.7.2, this is set to 10 rad/s (or $\sim$ 1.6 Hz). In practice this may be too low a corner frequency for practical signal processing applications where we typically have corner frequencies of 10 Hz or even 100 Hz and above since viable range of the signal is usually several hertz or tens of hertz or even hundreds of hertz.

6.7.2 Filter Bode Plot

In order to formally develop the *bode plot* (or frequency response plot) of the filter, we need to consider its response to generic sinusoidal signals. Let us assume the input takes the form of sinusoidal function of the form $v_i = V_i(\omega t)$, whose Laplace Transform is given by

$$\hat{v} = \frac{\omega V_i}{s^2 + \omega^2} \tag{6.18}$$

Substituting in (6.17), taking partial fractions and simplification (including dropping the transient response term as detailed in the Appendix) the steady state output would be $v_0 = V_0 \sin(\omega t + \phi)$ where

$$\frac{V_0}{V_i} = \frac{1}{\sqrt{(\tau\omega)^2 + 1}} \quad \text{and} \quad \phi = -\tan^{-1}(\tau\omega) \tag{6.19}$$

In more standard form, we have

$$\left.\frac{V_0}{V_i}\right|_{\text{dB}} = 20\log\frac{1}{\sqrt{(\tau\omega)^2 + 1}} = 20\log 1 - 10\log\left((\tau\omega)^2 + 1\right) \tag{6.20}$$

$$\left.\frac{V_0}{V_i}\right|_{\text{dB}} = \begin{cases} 0 & \tau\omega << 1 \\ -20\log\tau\omega & 1 << \tau\omega \end{cases} \tag{6.21}$$

The graph of the $V_0/V_i|_{\mathrm{dB}}$ is precisely what was depicted earlier in Figure 6.9 where it is evident that the higher frequencies, particularly those higher than the corner frequency are attenuated at an increasing rate, leading to the reduction of high-frequency components in the filtered signal. In the figure, the corner frequency is 10 rad/s (approximately 1.6 Hz), and so signals at frequencies of 100 rad/s ($\sim$ 16 Hz) are attenuated by -20 dB (or a factor of 10).

The Matlab implementation of Figure 6.9 is stated below:

```
>> s = tf('s');          % Create a basic transfer function to simplify the
                         % definition of complex transfer functions
>> g=1/(0.1*s+1);        % Define a first order transfer function g(s)=k/(τs+1)
                         % with k = 1 and τ = 0.1.
>> bode(g);              % Create a bode phase and magnitude plot for g.
>> grid on;              % Define the grid in the graph for clarity
```

As noted above, the corner frequency of this low-pass filter is also its bandwidth, which is presently 10 rad/s ($\sim$1.6 Hz). In practice, the bandwidth of actual filters we use in DAQ process may be in the range of 10 Hz−1 kHz, depending on the application domain. The filter structure itself may be more complex as well but this example illustrates the very basic idea of a low-pass passive filter.

6.7.3 Passive High-Pass Filter

High-pass filters can also be developed using a simple resistor, capacitor pair as shown previously in Figure 6.8 by replacing the positions of the resistor and capacitor. The resulting filter would be given by

$$\frac{\widehat{v}_0}{\widehat{v}_i} = \frac{\tau s}{\tau s + 1} \tag{6.22}$$

As it is evident from Figure 6.10., the filter attenuates low-frequency signals and passes through the high-frequency components of a given signal. This filter is defined by the following Matlab script below:

```
>> s = tf('s');          % Create a basic transfer function to allow for definition
                         % of more complex one.
>> g=0.1*s/(0.1*s+1);    % Define a first order transfer function g(s)= τs /(τs+1)
                         % with τ = 0.1.
>> bode(g);              % Create a bode phase and magnitude plot for g.
>> grid on;              % Define the grid in the graph for clarity
```

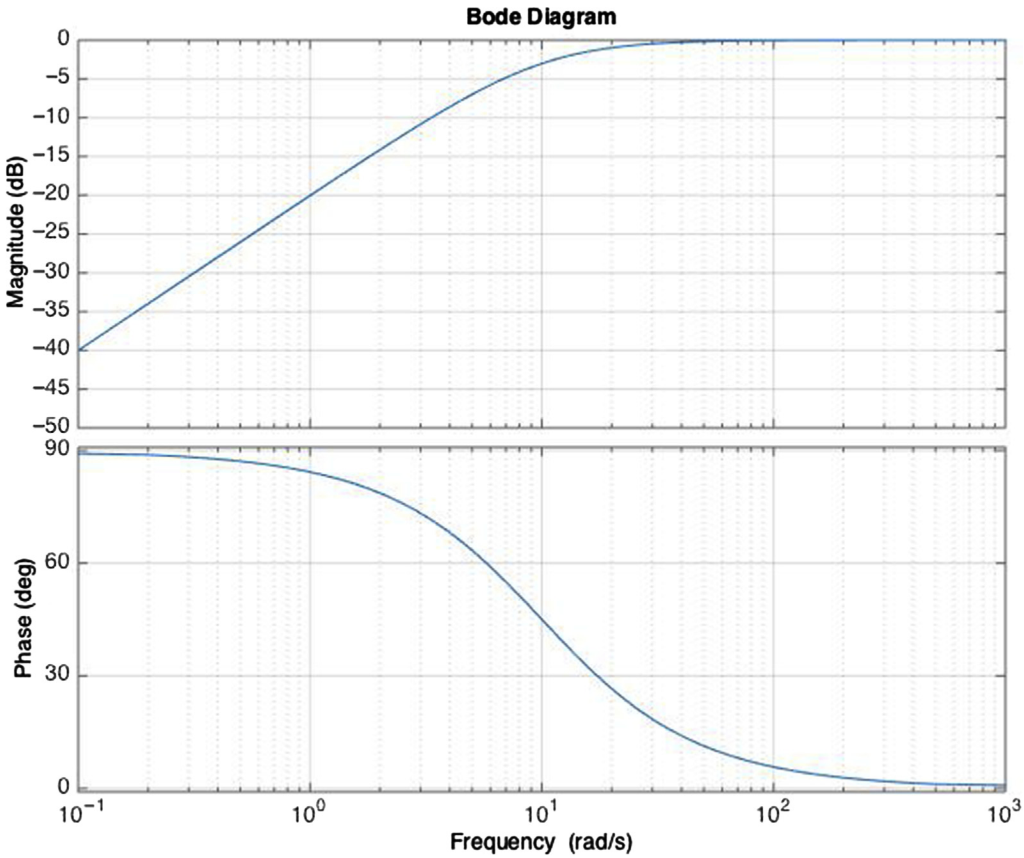

Figure 6.10
High-pass filter response.

The corner frequency of the above filter is once again 10 rad/s (∼1.6 Hz). Signals at frequencies of 1 rad/s (∼0.16 Hz) are attenuated by 20 dBs or a factor of 10 while signals of frequencies 0.1 rad/s (0.016 Hz) are attenuated by 40 dBs or a factor of 100 and so forth. Clearly true DC signals are fully attenuated, which makes this filter useful in eliminating the DC offset of some types of signals.

6.8 Active Filters Using Op-Amps

The passive filter shown above is simple and can in principle be used to filter out undesirable components of a given signal. However, since it is made up of entirely passive components (resistors and capacitors) it has to draw current from the input and will in addition "load" the circuit that is connected to the output of the filter.

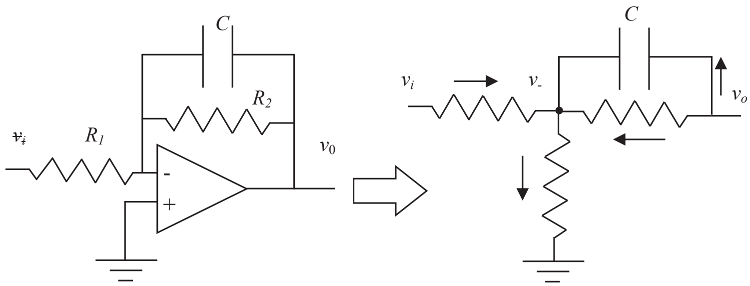

Figure 6.11
Active low-pass filter.

Operational amplifiers, or Op-Amps, can eliminate this problem since the current that is drawn from the input stage is very small (since op-amps have large input resistances or impedances; of the order of 10 MΩ). Likewise, as active devices, op-amps supply current to drive their output and hence minimize the impact of the filter on any output circuit such as the DAQ card and thereby less drastically affect the reading of the acquired signal. With this in mind, op-amps are often used in conjunction with resistors and capacitors to create an active filter. A sample configuration is given in Figure 6.11.

To be able to utilize this circuit, we must determine the relation between the input v_i and the output v_0. The summation of currents at the inverting input is given by

$$\frac{v_i - v_-}{R_1} + \frac{v_0 - v_-}{R_2} + C\frac{\mathrm{d}}{\mathrm{d}t}(v_0 - v_-) = 0 \tag{6.23}$$

Now, in a negative feedback configuration as shown above, $v_- = v_+ = 0$ (since the input impendence of the amplifier is very high) we can simplify this to

$$\frac{v_i}{R_1} + \frac{v_0}{R_2} + C\frac{\mathrm{d}}{\mathrm{d}t}(v_0) = 0 \tag{6.24}$$

We can rewrite this as

$$v_0 + R_2 C\frac{\mathrm{d}}{\mathrm{d}t}(v_0) = -\frac{R_2}{R_1}v_i \tag{6.25}$$

This is very similar to a passive filter equation, with the exception of the "gain" on v_i. The negative sign means that the filtered signal will lag at least 180° behind the unfiltered one. (This issue can be resolved via an inverting filter of gain 1.) We can call this gain k; so $k = R_2/R_1$. In addition, we see that the time constant is given by $\tau = R_2 C$. So, we have two degrees of freedom in our filter configuration. We may

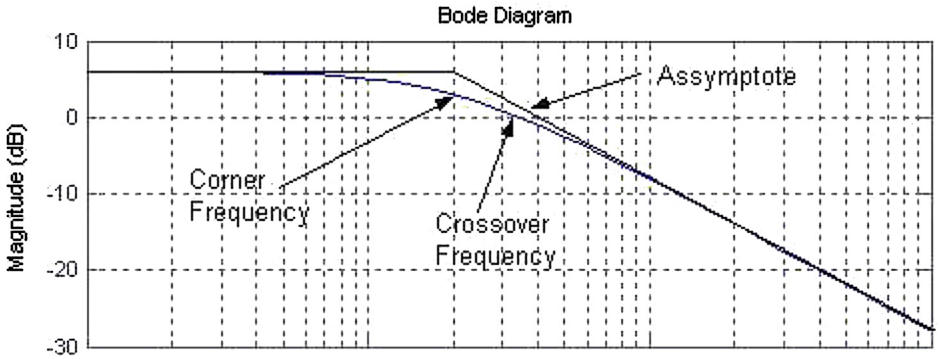

Figure 6.12
Bode diagram of the low-pass filter.

adjust the gain constant, k, to amplify low-frequency signals. However, this will also raise the value of the crossover frequency and therefore allow noise to have a higher amplification. In addition, we can change the corner frequency ($\omega_c = 1/\tau$) by adjusting the relevant parameters. This will have effects on the gain as well that must be taken into account. These can be better understood by examining the system in the frequency domain. In the frequency domain, our ratio of the output to input voltage is given by

$$\left.\frac{V_0}{V_i}\right|_{\mathrm{d}B} = 20\log\frac{k}{\sqrt{(\tau\omega)^2+1}} = 20\log k - 10\log\left((\tau\omega)^2+1\right)$$

$$\left.\frac{V_0}{V_i}\right|_{\mathrm{d}B} = \begin{cases} 20\log k & \tau\omega << 1 \\ 20\log k - 20\log\tau\omega & 1 << \tau\omega \end{cases}$$

We can understand better how the system works by examining the bode plot of the filter, which is depicted in Figure 6.12. As discussed earlier, the graph is meant to illustrate how the filter passes through signals of frequency up to its corner frequency and gradually attenuates those beyond this level as depicted in the figure. There is an accompanying phase shift that is not shown but is similar to that in Figure 6.9.

6.9 Signal Amplification

Signal amplification is carried out when the typical signal output level of a sensor is considered to be too low (as for instance in the case of a weak thermocouple signal, which is of the order of microvolts). A sample configuration is given in Figure 6.13.

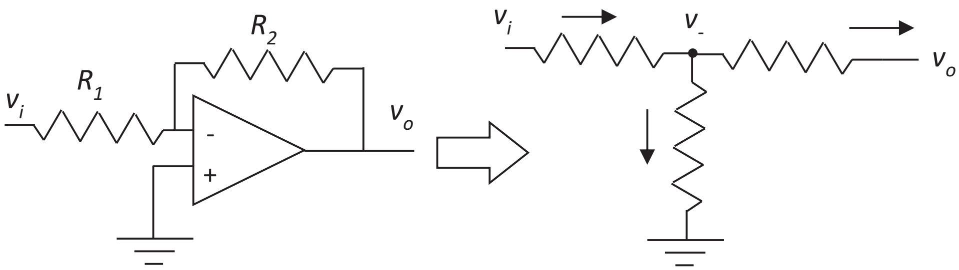

Figure 6.13
Active inverting amplifier.

To be able to utilize this circuit, we must determine the relation between the input, v_i, and the output, v_0. The summation of currents at the inverting input (v_-) is given by

$$\frac{v_i - v_-}{R_1} = \frac{v_- - v_0}{R_2} \tag{6.26}$$

Once again, in a negative feedback configuration as shown above, $v_- = v_+ = 0$, we can simplify this to

$$v_0 = -\frac{R_2}{R_1} v_i \tag{6.27}$$

This implies that the above circuit acts as an inverting amplifier with a "gain" of R_2/R_1. In other words, the circuit can amplify (and simultaneously invert) its input signal. The amplification ratio is determined by the designer through the selection of the respective resistance values, R_1 and R_2. Typically these values are chosen in reference to the internal resistance of the amplifier. If the op-amp input impedance is of the order of $10^8\ \Omega$ then R_1 and R_2 will need to be of the order of 10^4–$10^6\ \Omega$, offering a factor of up to 100 amplification. It is typically not very sensible to use these simple op-amp circuits for amplifications beyond this range, particularly since noise can also be magnified but also because the internal circuitry of the op-amp is not set up for very high amplification.

As it is evident in the above circuit, the simple inverting amplifier is a subset of the low-pass filter in Figure 6.11. This fact itself suggests that the active low-pass filter can be tailored to act as a combined filter/amplifier (albeit with an appropriate choice of resistance and capacitance values). We should also point out that the inverting feature of both circuits presents a challenge in certain situation where the sign of the original signal must be preserved (although if the signal is digitally acquired, this may not be an issue since it is always possible to perform the sign inversion inside the PC). In such cases, one must either tag a second inverting amplifier (with a gain of 1) onto the first-stage filter/amplifier or devise a single noninverting amplifier

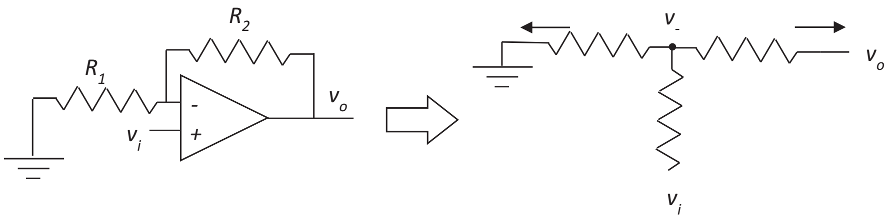

Figure 6.14
Active non-inverting amplifier.

(or low-pass filter) as necessary. A simple noninverting amplifier is shown in Figure 6.14.

To be able to utilize this circuit, we must determine the relation between the input v_i and the output v_0. We make note of the fact that due to the high input impedance of the op-amp, $v_i = v_-$. Subsequently we have

$$\frac{v_i}{R_1} + \frac{v_i - v_0}{R_2} = 0 \tag{6.28}$$

We can simplify this to

$$\frac{v_0}{R_2} = v_i\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \tag{6.29}$$

$$\frac{v_0}{v_i} = 1 + \frac{R_2}{R_1} \tag{6.30}$$

This implies that the above circuit acts as a non-inverting amplifier with a “gain” of $1 + R_2/R_1$ and so the (nonunique) choice of the resistance values, R_1 and R_2, determines the amplification factor.

6.9.1 Differential Amplification

Figure 6.15 shows a common amplifier configuration that is used to amplify the small difference that may exist between two voltage signals V_A and V_B.

These may represent, for example, the pressures on either side of an obstruction device put in a pipe to measure the volume flow rate of fluid flowing through it. The output voltage V_0 is given by

$$V_0 = \frac{R_3}{R_1}(V_B - V_A) \tag{6.31}$$

A differential amplifier is also very useful for removing *common-mode noise* voltages. Suppose V_A and V_B in Figure 6.15 are set up such that $V_A = +V_s$ V and $V_B = 0$ V. Let us

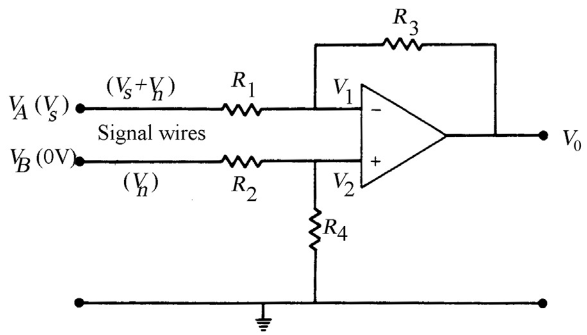

Figure 6.15
Active differential amplifier.

assume that the measurement circuit has been corrupted by a common-mode noise voltage V_n such that the voltages on the $+V_s$ and 0 V signal wires become $(V_s + V_n)$ and (V_n). The inputs to the amplifier V_1 and V_2 and the output V_0 can then be written as

$$V_1 = \frac{R_3}{R_1}(V_s + V_n) \tag{6.32}$$

$$V_2 = \frac{R_4}{R_2 + R_4} V_n \tag{6.33}$$

$$V_0 = V_2\left(1 + \frac{R_3}{R_1}\right) - V_1 \tag{6.34}$$

Hence:

$$\begin{aligned} V_0 &= \left(\frac{R_4}{R_2 + R_4} V_n\right)\left(1 + \frac{R_3}{R_1}\right) - \frac{R_3}{R_1}(V_s + V_n) \\ &= V_n\left(\frac{R_4}{R_2 + R_4} + \frac{R_3 R_4}{R_1(R_2 + R_4)}\right) - \frac{R_3}{R_1}(V_s + V_n) \end{aligned} \tag{6.35}$$

Subsequently we have

$$V_0 = V_n\left(\frac{R_4(1 + R_3/R_1)}{R_2(1 + R_4/R_2)} - \frac{R_3}{R_1}\right) - \frac{R_3}{R_1} V_s \tag{6.36}$$

If the resistance values are chosen carefully such that $R_4/R_2 = R_3/R_1$, the above simplifies to

$$V_0 = -\frac{R_3}{R_1} V_s \tag{6.37}$$

that is, the noise voltage V_n has been removed and we have an inverting amplifier of "gain," R_3/R_1.

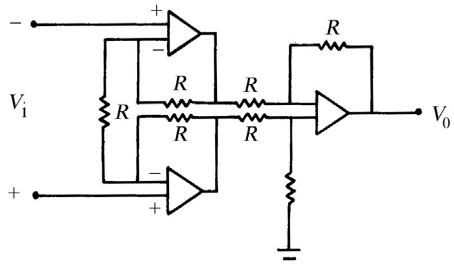

Figure 6.16
Instrumentation amplifier.

6.9.2 Instrumentation Amplifier

For applications requiring the amplification of very low-level signals, a special type of amplifier known as an *instrumentation amplifier* is used. This consists of a circuit containing three standard operational amplifiers, as shown in Figure 6.16. The advantage of the instrumentation amplifier compared with a standard operational amplifier is that its differential input impedance is much higher. In consequence, its common-mode rejection capability is much better. This means that, if a twisted wire pair is used to connect a transducer to the differential inputs of the amplifier, any induced noise will contaminate each wire equally and will be rejected by the common-mode rejection capacity of the amplifier.

6.9.3 Other Op-Amp-Based Filters and Amplifiers

There are numerous other variants of low-pass filters and other amplifiers can be implemented using op-amps and simple resistance and capacitor elements. Several examples are depicted in Figure 6.17. It is important to note that high-pass filters are often used to remove DC offset in many types of signals as necessary.

6.10 Digital Filters

Digital filters are generally implemented via a microprocessor or a digital signal processor or microcontroller. The simplest digital filter takes the form of

$$y_k = \alpha u_k + (1 - \alpha) u_{k-1} \tag{6.38}$$

Note that we can write $\alpha_0 \equiv \alpha$ and $\alpha_1 \equiv 1 - \alpha$ and hence write the above as

$$y_k = \alpha_0 u_k + \alpha_1 u_{k-1} \tag{6.39}$$

and thus further extend this formulation to more complex filters such as

$$y_k = \alpha_0 u_k + \alpha_1 u_{k-1} + \alpha_2 u_{k-2} + \ldots + \alpha_n u_{k-n} \tag{6.40}$$

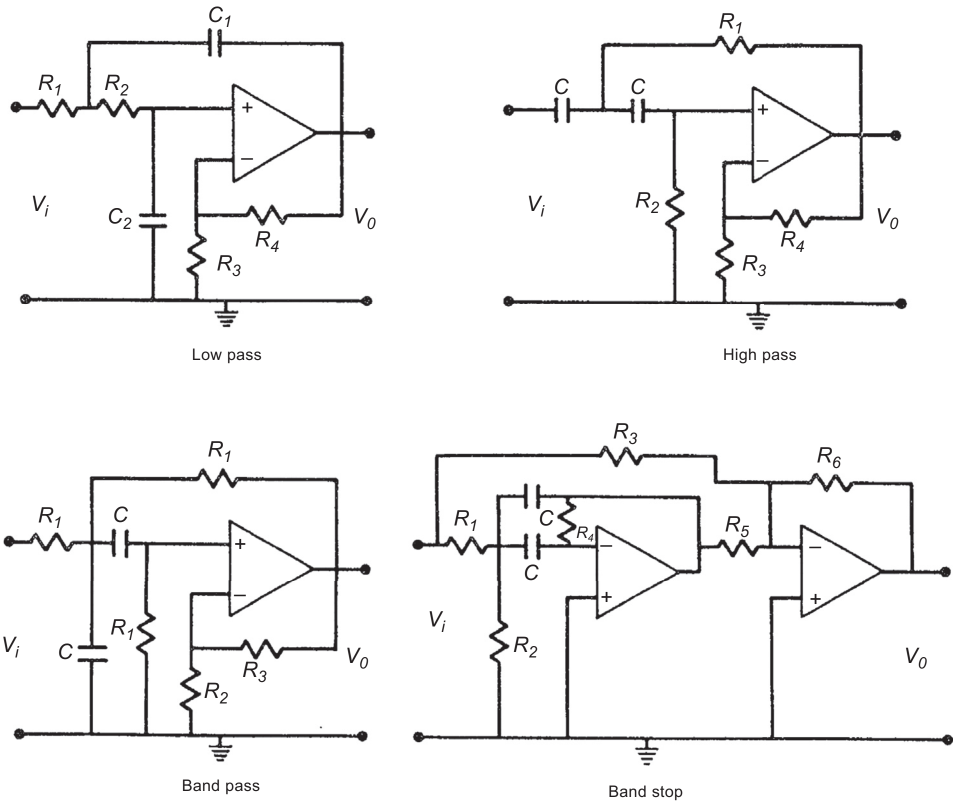

Figure 6.17
Other amplifiers and filters.

This general form is called a *moving average* or finite impulse response filter, since it in effect averages current and past values of the input signal, each with its respective weight. Selection of these weights is often an issue and can be formalized although in the present context we will deal with simpler cases in which an intuitive approach to selecting these values can be used.

6.10.1 Filter with Memory

In the filter with memory, previously filtered values will be used to adjust the new output. This filter will take the form

$$y_k = \alpha u_k + (1 - \alpha) y_{k-1}, \tag{6.41}$$

where α is the weight on the current value of the unfiltered signal, u_k. The remainder is from the previous value of the filtered signal, y_{k-1}. Varying α will change the extent to which that the input signal is filtered. In particular, a relatively large α weighs in the current value of the input signal while a small α weighs in the past (filtered) signal, y_k. Normally $\alpha \leq 1$. This is evident in the example given below.

Table 6.1: Data for digital filtering example

k	u_k
0	0.10
1	1.05
2	1.92
3	3.90
4	4.02
5	4.94

6.10.2 Example

A set of data points is measured from a continuous signal as given in the Table 6.1.

A simple input averaging filter with α values of 0.25, 0.5, 0.75, and 1.0 is used to filter these values as depicted in Figure 6.18. The filters produce a range of response patterns indicating that no single value of α is necessarily the best. However, one can argue, based on proximity to the general pattern of the input signal, that $\alpha = 0.5$ maybe a reasonable choice. In practice one has to fine-tune α (or similar parameters of a given filter) to fit the application in mind. There are as we shall see below formal methods (based on digital signal processing) that allow the user to select the filter parameter to affect the frequency response of the filter (similar to the analog case). These mathematical techniques underlying these tools are beyond the scope of this course although their application will be discussed below.

6.10.3 ARMA and Infinite Impulse Response Filters

While the simple filter in the previous section works reasonably well, one can build more effective filters using a combination of autoregressive terms and moving average terms, via

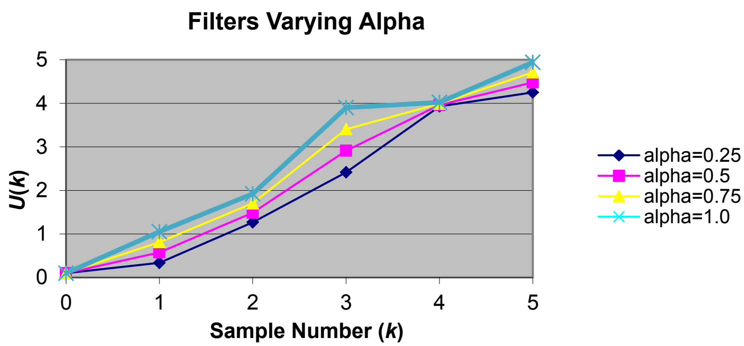

Figure 6.18
A simple digital filter response.

an AutoRegressive Moving Average (ARMA) model, also referred to as an Infinite Impulse Response (IIR) filter. The general equation for such a filter is given below

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - a_3 y_{k-3} + \ldots + b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + b_3 u_{k-3} + \ldots \quad (6.42)$$

Note that a_i and b_i must be properly chosen for stable and effective performance. This is not a trivial task and requires advanced techniques that extend beyond the scope of the present text. The essential idea is to place the so-called poles and zeros (the roots of the denominator and numerator of the corresponding discrete-time transfer functions) in reasonable locations in the complex plane. These are not readily clear from the brief discussion in this chapter and we do not plan to delve into the underlying concepts in these short few pages. It is possible, however, to produce the filter coefficients in Matlab or a similar tool and use a similar technique as in the previous section to implement the given filter.

For this reason there are many other digital filtering approaches. Many revolve around on designing an analog filter and approximating it as a digital filter. There are also well-known design patterns for digital filters. This is the case for the so-called *Butterworth* filter. Luckily, this design process can be automated using MATLAB. MATLAB uses a command called *butter*() to generate the coefficients for a filter with a certain order and cutoff frequency. A sample command is given below:

order $2f_c/f_s$

```
>> [b,a]=butter(3,0.1)
b = 0.0029    0.0087    0.0087    0.0029
```

b_0 b_1 b_2 b_3

```
a = 1.0000   -2.3741    1.9294   -0.5321
```

a_0 a_1 a_2 a_3

The command given above generates a third-order Butterworth filter. The first term in the command (3 in this example) is the order of the filter. By setting this term, you can control how many past data points the filter uses. The second term in the argument is the ratio of the cutoff (or corner) frequency to the Nyquist frequency. The Nyquist frequency, as it was noted before, is one half of the sampling rate, so this ratio, r, is given by,

$$r = \frac{f_c}{f_s/2} \quad (6.43)$$

Alternatively, r is the ratio of the cutoff frequency, f_c to the Nyquist frequency of sampling system (i.e., $^1/_2$ the sampling rate, f_s).

These frequencies are both given in hertz, which makes r a unitless quantity. It is important to note that digital filters only filter frequencies relative to the sampling frequency. If we increase the cutoff frequency and increase the sampling frequency by the same amount, we will achieve the same result. In addition, the maximum value of r is 1.0. However, this would correspond to an unfiltered response.

Finally, from an implementation standpoint, we note that the equation for the filter is given by

$$y_k = \frac{1}{a_0}\left(- a_1 y_{k-1} - a_2 y_{k-2} - a_3 y_{k-3} + b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + b_3 u_{k-3}\right) \tag{6.44}$$

The response of the above filter to the compound analog signal we discussed earlier is shown below. The solid line is the original signal and the dotted line represents the filtered signal but they overlap. We should point out that the sampling frequency of the system is set at 100 Hz so the Nyquist frequency is 50 Hz. The filter's corner frequency is 0.1, which means the corner frequency of the filter is 5 Hz, well beyond the bandwidth of the signal (0.5 Hz). This means the signal is not noticeably attenuated or otherwise affected by the filter, which is the reason why the solid and dashed lines are not distinguishable in the figure (Figure 6.19).

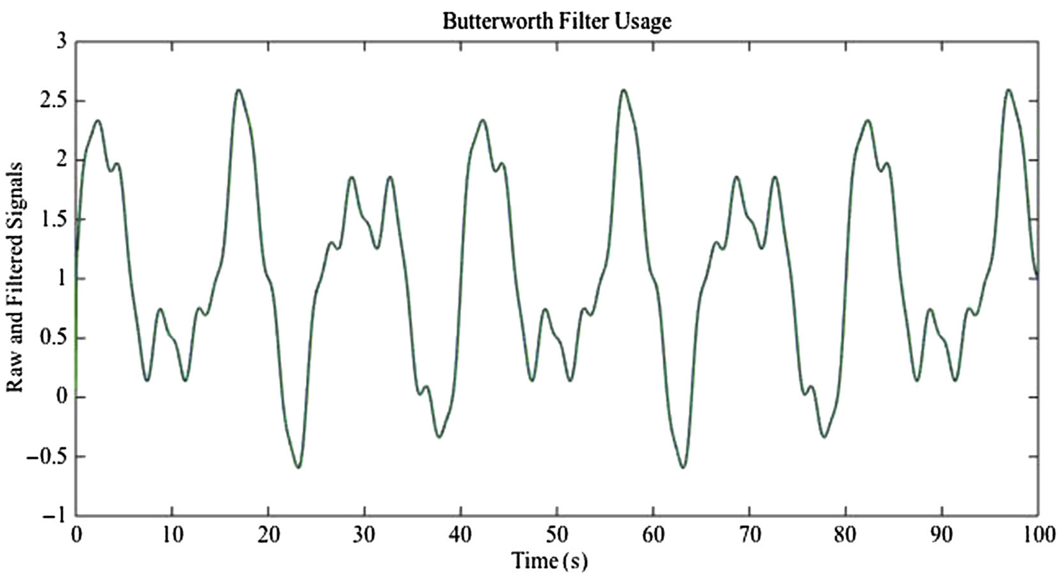

Figure 6.19
Butterworth filter response.

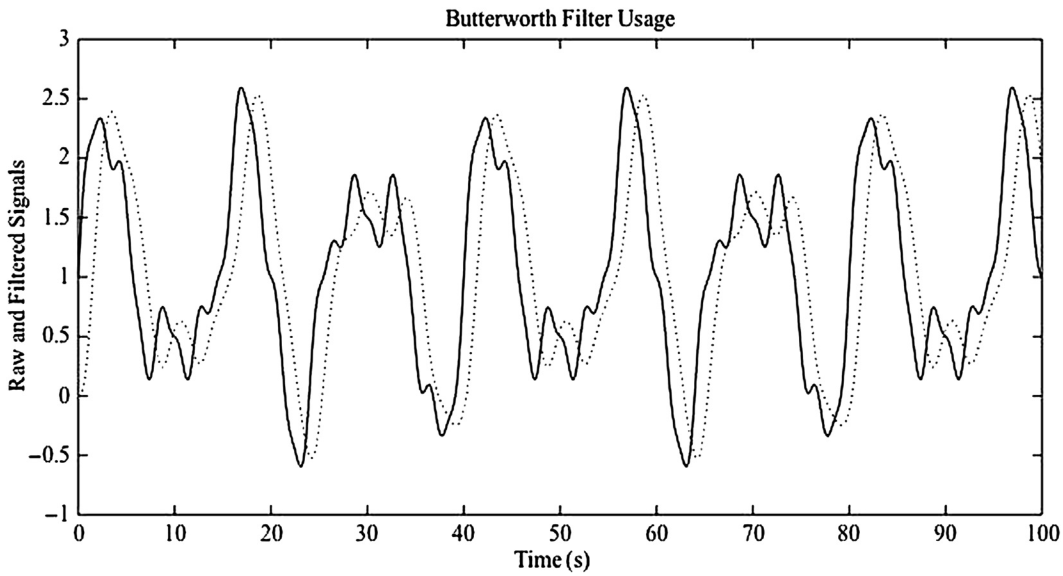

Figure 6.20
Butterworth filter with low-pass effect.

However, we can redesign the filter as follows:

```
>> [b,a]=butter(3,0.005)
b = 1.0e−05 * [ 0.0477 0.1431 0.1431 0.0477]
a = 1.0000 −2.9686 2.9377 −0.9691
```

The filter's cutoff frequency is now 0.25 Hz and the results are shown in Figure 6.20 where the dotted line shows that the filtered signal is smoother (some higher frequency components attenuated). In addition, the filtered signal is not drastically different from the original signal and does not lag behind significantly either, which is a positive aspect of this filter design.

A further redesign the filter as follows:

```
>> [b,a]=butter(3,0.001)
b= 1.0e−07 * [0.0386 0.1159 0.1159 0.0386]
a = 1.0000 −2.9937 2.9875 −0.9937
```

The filter's cutoff frequency is now 0.05 Hz and the results are shown in Figure 6.21 where the dotted line shows that the filtered signal is now missing the bulk of the information in the original signal. This means that filtering must be done judiciously (as for instance in the previous case); otherwise meaningful information may be missing from the filtered signal.

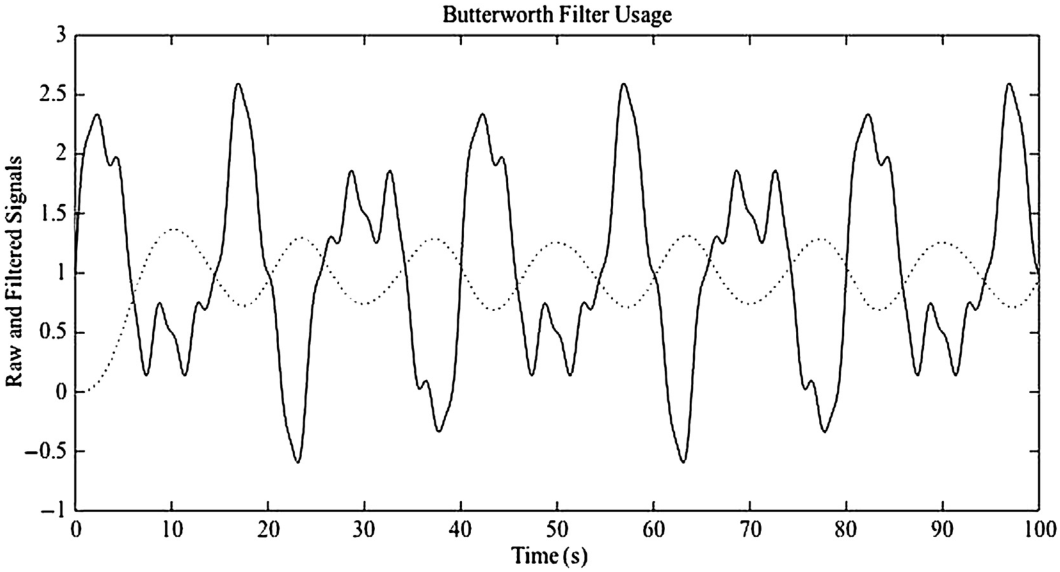

Figure 6.21
Digital filtering of compound signal.

6.11 Summary

This chapter discussed signal processing techniques commonly needed to process sensor data acquired from mechanical experiments. These basic signal processing techniques include analog filtering using passive and active components as well as digital filtering. We also pointed out the need to be mindful of aliasing, that is, presence of spurious signals if the original signal is not properly sampled. The techniques discussed in this chapter are the basic tools that are needed in processing sensor signals. Clearly both analog and digital signal processing are rich topics that the reader can benefit from consulting more exhaustive references.[3]

6.12 Exercises

6.1 A 1 kHz sine wave is sampled at frequency of 5 kHz.
   (a) What is the Nyquist frequency of the sampling system in this case? Show your work.
   (b) What is the Nyquist rate of the signal? Show your work.
   (c) Is the signal properly sampled; that is, will there be aliases in the digitally recorded signal? Explain.

[3] Proakis, John G., and Dimitris G. Manolakis. *Introduction to digital signal processing*. Prentice Hall Professional Technical Reference, 1988.

(d) What if the original analog signal included both even and odd harmonics of its primary frequency; that is, 2, 3, 5 kHz, etc. What would be the highest harmonic that will be faithfully represented in the discrete-time version of the analog signal?

(e) Following on with the previous step, what is the lowest harmonic that will be aliased and what would be resulting alias frequency?

6.2 In this chapter we proved that aliasing occurs when an analog signal is discretized. Can aliasing occur with purely analog signals? In other words, suppose we have a purely analog circuit (as opposed to a digital computer; you could call such a circuit an analog computer). Is it possible for aliasing to occur in this case? If so given an example? If not provide a proof.

6.3 What is the maximum quantization error for an A/D converter with 14 bits of resolution assuming a voltage range of ± 10 V?

6.4 Design a simple passive low-pass filter with a corner frequency of 50 Hz. You may assume that 0.01 μF capacitor is available. What would be the resistor value that best approximates this filter? With your selection of the resistor value, recalculate the corner frequency and determine the percentage difference from the original specification. In addition draw the bode plot of the filter using Matlab.

6.5 Design a simple high-pass filter with a corner frequency of 100 Hz. You may assume that 0.01 μF capacitor is available. What would be the resistor value that best approximates this filter? With your selection of the resistor value, recalculate the corner frequency and determine the percentage difference from the original specification. In addition draw the bode plot of the filter using Matlab.

6.6 Explain why the maximum value of r in the Matlab command `butter()` is 1?

6.7 Assume the sampling rate is 1000 Hz. Design a digital low-pass butterworth filter with Matlab with a cutoff frequency of 50 Hz.

6.8 Assume a sampling rate of 100 Hz. Filter the signal given by

$$x(t) = 1 + \sin(0.125\pi t) + 0.5\sin(0.25\pi t) + 0.25\sin(0.5\pi t) + .125\sin(\pi t) \tag{6.45}$$

by filters of progressively lower cutoff frequencies of 50, 5, 0.5, and 0.05 Hz.

6.9 Assume a sampling rate of 1000 Hz. Filter the signal given by

$$x(t) = 1 + \sin(100t) + \sin(400t) + \sin(500t) \tag{6.46}$$

by filters of progressively lower cutoff frequencies of 300 Hz, 30 Hz, 10 Hz, respectively.

6.13 Appendix

6.13.1 Simple Filter Solution

We start with

$$\frac{\widehat{v}_0}{\widehat{v}_i} = \frac{1}{\tau s + 1} \tag{6.47}$$

Assuming sinusoidal functions of the form $v_i = V_i(\omega t)$, or $\widehat{v}_i = \frac{\omega V_i}{s^2+\omega^2}$, we have the following partial fraction expansion,

$$\widehat{v}_0 = \frac{1}{\tau s + 1}\,\frac{\omega V_i}{s^2+\omega^2} = \frac{A}{s+1/\tau} + \frac{B}{s+j\omega} + \frac{C}{s-j\omega} \tag{6.48}$$

where

$$A = \lim_{s\to -1/\tau}(s+1/\tau)\widehat{v}_0(s) = \frac{\omega V_i}{(-1/\tau)^2+\omega^2} = \frac{\omega\tau^2 V_i}{1+\tau^2\omega^2} \tag{6.49}$$

$$B = \lim_{s\to -j\omega}(s+j\omega)\widehat{v}_0(s) = \frac{\omega V_i}{(-j\tau\omega+1)(-2j\omega)} \tag{6.50}$$

$$C = \lim_{s\to j\omega}(s-j\omega)\widehat{v}_0(s) = \frac{\omega V_i}{(j\tau\omega+1)(2j\omega)} \tag{6.51}$$

Now noting that

$$v_0(t) = A\mathrm{e}^{-t/\tau} + B\mathrm{e}^{-j\omega t} + C\mathrm{e}^{j\omega t} \tag{6.52}$$

and that the first term dies out with time, we look at steady state value of $v_0(t)$ as

$$v_0(t) = \frac{\omega V_i}{(-j\tau\omega+1)(-2j\omega)}\mathrm{e}^{-j\omega t} + \frac{\omega V_i}{(j\tau\omega+1)(2j\omega)}\mathrm{e}^{j\omega t} \tag{6.53}$$

and further into

$$v_0(t) = \frac{(j\tau\omega+1)V_i}{(\tau^2\omega^2+1)(-2j)}\mathrm{e}^{-j\omega t} + \frac{(-j\tau\omega+1)V_i}{(\tau^2\omega^2+1)(-2j)}\mathrm{e}^{j\omega t} \tag{6.54}$$

and further into

$$v_0(t) = \frac{V_i}{(\tau^2\omega^2+1)(-2j)}\left((j\tau\omega+1)\mathrm{e}^{-j\omega t} + (-j\tau\omega+1)\mathrm{e}^{j\omega t}\right) \tag{6.55}$$

and

$$v_0(t) = \frac{V_i}{(\tau^2\omega^2+1)(-2j)}\left(-j\tau\omega\left(\mathrm{e}^{j\omega t}-\mathrm{e}^{-j\omega t}\right) + \left(\mathrm{e}^{j\omega t}+\mathrm{e}^{-j\omega t}\right)\right) \tag{6.56}$$

and

$$v_0(t) = \frac{V_i}{(\tau^2\omega^2 + 1)(-2j)}(2\tau\omega \sin(\omega t) + 2\cos(\omega t)) \tag{6.57}$$

and

$$v_0(t) = \frac{V_i}{\tau^2\omega^2 + 1}(\tau\omega \sin(\omega t) + \cos(\omega t)) \tag{6.58}$$

and

$$v_0(t) = V_i(\cos(\phi)\sin(\omega t) + \sin(\phi)\cos(\omega t)) \tag{6.59}$$

where $\tan(\phi) = \tau\omega$. We thus have

$$v_0(t) = V_i \sin(\omega t + \varphi) \tag{6.60}$$

6.13.2 Common Capacitor Values

pF	pF	pF	pF	μF	μF	μF	μF	μF	μF
1.0	10	100	1000	0.01	0.1	1.0	10	100	1000
1.1	11	110	1100						
1.2	12	120	1200						
1.3	13	130	1300						
1.5	15	150	1500	0.015	0.15	1.5	15	150	1500
1.6	16	160	1600						
1.8	18	180	1800						
2.0	20	200	2000						
2.2	22	220	2200	0.022	0.22	2.2	22	220	2200
2.4	24	240	2400						
2.7	27	270	2700						
3.0	30	300	3000						
3.3	33	330	3300	0.033	0.33	3.3	33	330	3300
3.6	36	360	3600						
3.9	39	390	3900						
4.3	43	430	4300						
4.7	47	470	4700	0.047	0.47	4.7	47	470	4700
5.1	51	510	5100						
5.6	56	560	5600						
6.2	62	620	6200						
6.8	68	680	6800	0.068	0.68	6.8	68	680	6800
7.5	75	750	7500						
8.2	82	820	8200						
9.1	91	910	9100						

6.13.3 Common Resistor Values

1% Standard Values

Decade Multiples are Available from 10.0 Ω through 1.00 MΩ
(Also 1.10, 1.20, 1.30, 1.50, 1.60, 1.80, 2.00, and 2.20 MΩ)

10.0	10.2	10.5	10.7	11.0	11.3	11.5	11.8	12.1	12.4	12.7	13.0
13.3	13.7	14.0	14.3	14.7	15.0	15.4	15.8	16.2	16.5	16.9	17.4
17.8	18.2	18.7	19.1	19.6	20.0	20.5	21.0	21.5	22.1	22.6	23.2
23.7	24.3	24.9	25.5	26.1	26.7	27.4	28.0	28.7	29.4	30.1	30.9
31.6	32.4	33.2	34.0	34.8	35.7	36.5	37.4	38.3	39.2	40.2	41.2
42.2	43.2	44.2	45.3	46.4	47.5	48.7	49.9	51.1	52.3	53.6	54.9
56.2	57.6	59.0	60.4	61.9	63.4	64.9	66.5	68.1	69.8	71.5	73.2
75.0	76.8	78.7	80.6	82.5	84.5	86.6	88.7	90.9	93.1	95.3	97.6