EE 311 **January 14, 2019**

Notes – First Day

Syllabus

- Grading
- Software
  - MATLAB®
  - Goldwave
- Hardware
  - ARM Cortex M4 Nucleo board
- First three weeks are largely a review of EE 310 with an emphasis on discrete systems
- Text book – Blandford and Parr – Introduction to Digital Signal Processing.

What is a Digital Filter? From text.

A simple IIR method – based on trapezoidal integration

A simple FIR method – sampling the impulse response

Chapter 1
Introduction and Overview

A filter can be defined as a device or a process which has as its input a set of data and produces as its output a subset of the input which share common characteristics. Filters are ubiquitous and we use them in daily living without being aware that we are doing so. An automobile typically has a fuel filter, an air filter, and an oil filter. The purpose of each of these is to separate out the contaminants that might damage the engine. For such filters the parts separated out (the subset removed) are discarded. The tuning dial on a radio allows you to adjust a filter which separates a general set of frequencies received from many stations into a small subset of frequencies that you are interested in. In the case of the radio, the subset of frequencies is used to produce sound and remainder of the general set is ignored. Your body has many filters including your kidneys which separate contaminants from the blood and your small intestines which separate the useful parts of food from the waste. Your brain is the most impressive filter of all. It sorts through the huge volume of information it receives from the sensory organs and makes sense out of resulting data. Your eyes and ears in this case serve as pre-filters limiting the data set that the brain can process. Much of the filtering that your brain does is more subtle. For example, looking up a name in a phone book could be thought of as filtering. Likewise, your ability to carry on a conversation in a noisy room is also an example of the brain's ability to filter information.

In this book we will view filters in a limited context although it is still quite broad. For our purposes, filtering will be limited to the broad data set of frequency information in electrical signals. There are three reasons we are concerned with the frequency filtering of information. These are:

1) To separate a particular frequency or band of frequencies from a global set for the purpose of extracting information. For example, changing the channel selector on a television receiver adjusts a filter so that it selects a band of frequencies broadcast by a particular station.
2) The removal of noise or other interference from a given signal to make its information bearing characteristics more prominent. As an example, consider a heart rate monitor of the type a jogger might wear when she is running. Such a device employs a low pass filter to eliminate the unwanted high frequencies picked up "out of the air" by the sensors on the body.
3) To transform a signal from one form to another where its information content may be viewed differently. The Fourier transform based spectrum analyzer is a good example of this. This transform allows one to view the frequency content of a time domain signal.

The term "filter" can be readily seen to apply to items 1 and 2 above but item 3 is that of transformation. Most people prefer the more general term of *Digital Signal Processing* or DSP and they view the digital filter as a more specialized form of DSP. We will adopt this view.

Filters which handle electrical signals are broadly classified as analog or digital. Analog filters have been around since the early days of radio and their invention is generally credited to George Campbell working at Bell Labs. Analog filters can be as simple as a resistor and a capacitor.

The distinguishing characteristic of analog filters is that they operate in continuous time on signals which are also continuous in amplitude. Thus, for a signal given by $y = f(t)$, we have a

Electric Filters

Filtering is a very broad concept that is done with nearly all systems (including life forms) which take in information, sort through it, and disregard that which is of no interest. Thus our brains are filters which take in information from our five senses and keep what is useful. Mostly this process goes on at an unconscious level and we can be very unaware of what we are disregarding. Electric filters, on the other hand are a relatively new idea which can be traced to the early part of the twentieth century.

An electric filter sorts through information in the form of an electric signal and divides the information into its various frequency components. The idea of dividing a signal into its frequency components dates back to the time when the United States was founded in the latter half of the eighteenth century. Joseph Fourier published his definitive paper on the subject in 1807.

Nearly a century after Fourier published his paper showing how a continuous time signal can be thought of as an infinite sum of discrete frequencies, the world underwent a transformation. Two technological events changed the way people lived and worked together. These were the invention of the "wireless", or radio, and the telephone. These two inventions revolutionized communications at the beginning of the twentieth century in much the same way that computers and the Internet have revolutionized communications at the end of the century.

At the time of this technological revolution, George Campbell was working for Bell Labs. He had received a doctorate from Harvard University in 1901. His dissertation concerned the inductively loaded telephone line. He and Michael Pupin, a professor at Columbia University, independently developed the theory which had implications for long distance telephone communications. (Pupin was given priority in patent rights). Very few engineers at this time had the necessary mathematical background to develop or understand this theory. But Campbell went on with the aim of extending the distance for telephone reception and in 1915 he applied for a patent for the "electric filter". This filter made it possible to multiplex several calls on a single line. The patent was granted in 1917 and a paper titled "Physical Theory of the Electric Wave-Filter" published in the Bell System Technical Journal in November of 1922 contains much of the foundation material for what is currently taught in an introductory circuits class in Electrical Engineering.

From "A History of Engineering and Science in the Bell System - Electronics Technology (1925-1975), AT&T Bell Lab (Author), and F.M. Smits (Editor), 1985

value of y defined for every value of t and (in theory) y can have any real number value. Digital filters are discrete in time. Thus a digital filter might be given by $y = f(nT)$ where n is an integer and T is a constant and is referred to as the *sampling period*. In this case, y has value only at T, $2T$, $3T$, etc. In between nT and $(n+1)T$, no value of y is defined. In implementation the amplitude of *y* is likewise discrete. For example, if the filter variables are represented by 8-bit binary numbers then y could only have $2^8 = 256$ discrete values.

The process of operating on discrete time signals is very old and probably predates continuous time operations. Most people however, mark the birth of DSP as that point in 1968 when Cooley and Tukey first published their algorithm for the fast Fourier transform (FFT)[1]. The FFT made it possible to generate the frequency spectrum of a complex signal in a reasonable amount of time on a mainframe or a minicomputer.

DSP today is used for a diverse set of real-time applications such as radar and sonar, biomedical signal analysis, speech processing, telephony and other communications applications, image processing, music storage and manipulation, seismic data processing, digital TV, etc. This explosion in DSP applications over the last 50 years has been driven by the availability of high speed low cost computers and memory.

Most of the signals that humans deal with in the practical world are analog. To get a digital signal, the analog signal must first be sampled and quantized to a number which is usually in binary. The quantizer is an analog to digital converter (A/D) and the sampler is an electronic switch that can capture the value of the analog signal in an instant of time. The time between successive samples is the sampling period and the inverse of the sampling period is called the sampling frequency: $f_s = 1/T$.

The digital signal processor receives the signal which has been sampled and quantized as a stream of binary numbers – one for each sample. After receiving the sampled data, the processor must perform some series of numerical operations that typically involve multiplication, addition, and shifting. In all but trivial cases the processor also stores some samples so that its calculations involve not only the present sample but the previous input and output samples as well. After completing the required operations, the processor sends the result to the output where it typically passes through a digital to analog (D/A) converter. The D/A converter changes the binary numbers back into a signal which is discrete in time and amplitude. Reconstruction filters often follow the D/A to smooth the analog signal back into one that is continuous in time and amplitude. A critical factor in this process is the sample period since all calculations must be complete during one sample period.

The process of doing digital signal processing sounds simple enough but there are many complications along the way. The Fourier series, Fourier transform, and the *z*-transform are fundamental to the understanding of the process. These are taken up in detail in Chapters 2 and 3. The sampling process and the quantizing of the analog signal have complications of their own – many, such as aliasing, are non-intuitive. Quantization introduces errors into the process and choosing the correct sample frequency and the number of bits to use in quantization are critical design parameters which are taken up in Chapter 4.

A digital filter is represented by a difference equation of the form

[1] The fast Fourier transform is discussed in detail in Chapter 3, section 5.

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M]$$
$$+ a_1 x[n-1] + a_2 x[n-2] + \cdots a_N x[n-N]$$

where x is the input variable, y is the output variable, and N, M, and the a_i and b_i coefficients are constants. Design of a digital filter becomes one of determining the value of the coefficients. For the special case where all of the a_i coefficients are zero, the filter is called a *Finite Impulse Response* (FIR) filter. FIR filter design is the subject of Chapter 5. For the more general case, where the a_i coefficients are not zero, the filter is called an *Infinite Impulse Response* (IIR) filter. These are taken up in Chapter 6. In many modern DSP systems we interact with digital signals that have different sample frequencies. Changing between these frequencies is the subject of Chapter 7.

The first seven chapters of this text cover the fundamentals of digital signal processing. These fundamentals have been well understood since the 1970s. Chapter 8 deals with filter implementation. Implementation is the process of changing a design into a real filter that can process data. Implementation seems to be constantly changing – driven mostly by new technology. In Chapter 8 we deal with the more fundamental aspects of implementation such as quantization error, overflow, and filter architecture.

The final three chapters of the text deal with application areas. Chapter 9 looks at DSP as it is applied to audio signals. Chapter 10 presents an introduction to two-dimensional signal processing. Chapter 11 takes up wavelets which are a relatively new area in which DSP plays a major role.

The following extended example answers the question "what is a digital filter?" in more concrete terms that includes a little math and a lot of intuition.

1.1 What is a digital filter?

To answer this question we will build upon the reader's knowledge of simple analog filter concepts and show how an analog filter can be converted to a digital filter. For the analog filter we will use the RC circuit shown in Figure 1.1. This is a low pass analog filter that allows low frequency analog signals to pass and attenuates the amplitude of higher frequency analog signals. This can be easily understood on an intuitive level if you remember that a capacitor has impedance which decreases with increasing frequency. At high frequencies the capacitor begins to look more like a short circuit and the output voltage, v_o, gets vanishingly small as the frequency of the input rises.

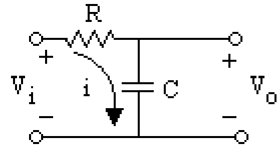

Figure 1.1
An RC circuit which serves as a low pass filter.

The analog circuit analysis
We can analyze this circuit in several ways. In the time domain we can write a differential equation to relate the output voltage to the input voltage:

$$v_o = v_c$$
$$v_i = iR + v_c$$
$$i = C\frac{dv_c}{dt} = \frac{Cdv_o}{dt}$$

and finally,

$$\frac{dv_o}{dt} + \frac{v_o}{RC} = \frac{v_i}{RC} \qquad \textbf{(1.1)}$$

To find the sinusoidal frequency response of this circuit we let the input become a complex sinusoid which may be written as

$$v_i = e^{j\omega t} \qquad \textbf{(1.2)}$$

The complex sinusoid is used as a mathematical convenience since it allows us to manipulate exponentials rather than sinusoids. Equation (1.1) becomes

$$\frac{dv_o}{dt} + \frac{v_o}{RC} = \frac{1}{RC}e^{j\omega t}$$

The solution to this differential equation is given by

$$v_o = K \cdot e^{-t/RC} + \frac{e^{j\omega t}}{1 + j\omega RC} \qquad \textbf{(1.3)}$$

The first term on the right hand side of (1.3) is the transient solution where the value of *K* is determined from initial conditions. The second term represents the steady state response to the input sinusoid. Since this is a linear system, we know that the output frequency will be the same as the input frequency but the amplitude and the phase of the output may differ from that of the input.

If we define the gain of the circuit as the output divided by the input, we can plot the magnitude and phase of the gain as a function of frequency for the steady state response. Noting that the magnitude of the input is always 1 the magnitude and phase of the circuit's gain is given by the magnitude and phase of the output.

$$\text{Magnitude of the gain} = \left|H(j\omega)\right| = \left|\frac{1}{1 + j\omega RC}\right| \qquad (1.4)$$

$$\text{Phase of the gain} = 0 - \angle(1 + j\omega RC) = -\angle(1 + j\omega RC) \qquad (1.5)$$

As expected, we see in Figure 1.2 that at low frequencies the gain of the filter approaches unity and at high frequencies the gain approaches zero.

Likewise, the response of the circuit to standard test inputs such as the unit step could be obtained by using the transient response terms of the differential equation. Figure 1.3 shows the circuit's step response.

A similar result could be obtained using Laplace transform techniques.

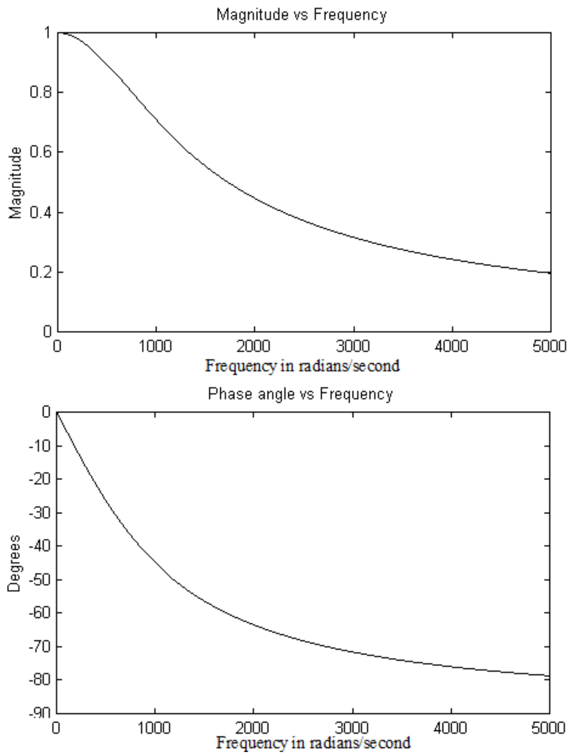

Figure 1.2
The top figure shows the magnitude vs. frequency plot for a low pass RC circuit. The bottom figure shows the phase curve.

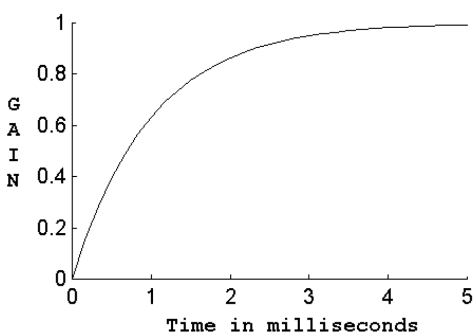

Figure 1.3
The response of the low pass RC circuit in Figure 1.0 to a unit step function. The equation for the output is $V_o = 1 - e^{-t/RC}$ for $t \geq 0$.

A digital filter replacement
To build a digital filter that could potentially be used in place of the analog filter, we need the components shown in Figure 1.4. This figure shows the analog signal V_i coming in on the left and the filtered analog signal V_o going out on the right. In between is the circuitry necessary to sample the signal, convert the samples to digital form (a series of numbers), perform the necessary numeric algorithms to do the filtering, and convert the filtered sequence back to an analog signal. Analysis of this circuitry will show that the filtered analog output approximates that of the RC circuit of Figure 1.1. The output is only an approximation because of the sampling process and because numeric algorithms in the computer introduce other errors.

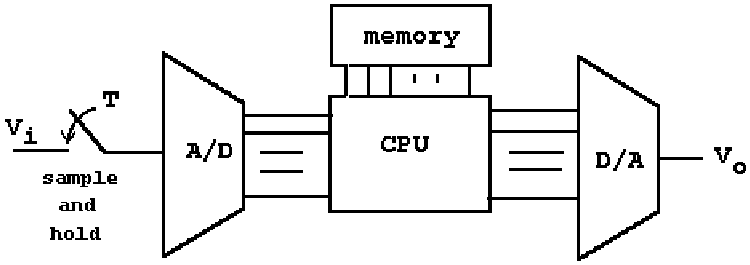

Figure 1.4
Components of a digital filter.

To better understand how the digital filter does its job, consider the components. The sampler on the left takes samples of the incoming voltage, V_i. The samples are taken at regular intervals - say every *T* seconds. The sampling frequency is therefore:

$$f_s = \frac{1}{T} .$$

Each sample taken is held for *T* seconds until the next sample comes in. During this time period the A/D converter changes the analog sample into a number - typically this is done in binary using 10 to 12 bits. The number goes into the computer where an algorithm that does the filtering is performed.

To understand how the algorithm for filtering works, we will consider two different, but very simple methods of creating filters comparable to the RC low pass filter in Figure 1.1. In the first method we will convert the differential equation which describes the RC circuit to a difference equation. A computer program will be used to implement the filter. The resulting filter will have feedback terms and will be referred to as an *Infinite Impulse Response* (IIR) filter. IIR filters are the subject of Chapter 6. For the second method we will use the impulse response of the RC circuit and create a filter which has a similar impulse response but in discrete time. This filter will have no feedback terms and will be referred to as a *Finite Impulse Response* (FIR) filter. FIR filters are the subject of Chapter 5.

A Simple IIR method

For the IIR filter we note that the differential equation is continuous in time. To implement it on a digital computer we need to convert this equation to a difference equation. There are several methods for doing this and not all of them give good results. One easy method is to convert the differential equation to an integral equation and represent the integral as discrete sums. In Figure 1.5, an arbitrary continuous function of time is plotted. If we want to find the integral of this function of time we could approximate the value by summing small trapezoids as noted in the figure.

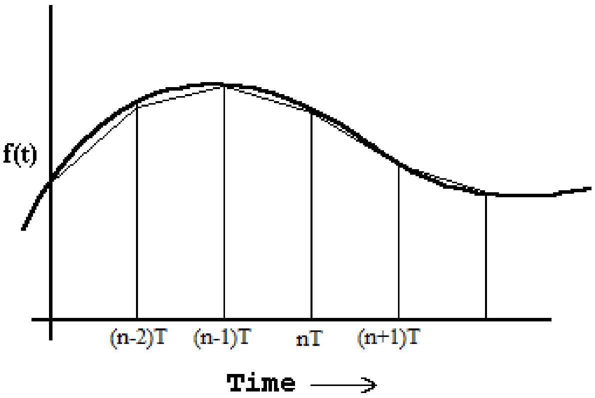

Figure 1.5
A integral to a continuous function of time can be approximated as the sum of trapezoids in discrete time.

If we take the variable *y* to represent the integral we can write:

$$y(nT) = \int_0^{nT} f(t) \cdot dt \qquad \textbf{(1.6)}$$

This equation can be broken into two separate parts and rewritten as:

$$y(nT) = \int_0^{(n-1)T} f(t) \cdot dt + \int_{(n-1)T}^{nT} f(t) \cdot dt$$

or,

$$y(nT) = y([n-1]T) + \int_{(n-1)T}^{nT} f(t) \cdot dt \qquad \textbf{(1.7)}$$

In (1.7) the integral can be approximated by the area of the trapezoid between (n-1)T and nT. This trapezoid has a width of T and an average height of [$f(nT)$+$f([n\text{-}1]T)$]/2. Equation (1.7) becomes:

$$y(nT) = y([n-1]T) + (T/2)[f(nT) + f([n-1]T)] \qquad \textbf{(1.8)}$$

The remaining task is to convert the differential equation (1.1) to an integral equation and apply (1.8) to it. To do this we write:

$$\frac{dv_o}{dt} + \frac{v_o}{RC} = \frac{v_i}{RC}$$

$$v_o + \frac{1}{RC}\int v_o \cdot dt = \frac{1}{RC}\int v_i dt$$

or

$$v_o = \frac{1}{RC}\int (v_i - v_o) \cdot dt \qquad \textbf{(1.9)}$$

To get the needed difference equations we make the following substitutions:

$$v_o(t) \Rightarrow v_o(nT)$$

$$\frac{1}{RC}\int (v_i - v_o)\cdot dt \Rightarrow v_o(n-1)T + \frac{T}{2RC}[v_i(nT) - v_o(nT) + v_i(n-1)T - v_o(n-1)T]$$

$$\Rightarrow \frac{T}{2RC}[v_i(nT) + v_i(n-1)T] - \frac{T}{2RC}v_o(nT) + v_o(n-1)T\left(1 - \frac{T}{2RC}\right)$$

Equation (1.9) becomes:

$$v_o(nT)\left(1 + \frac{T}{2RC}\right) = \frac{T}{2RC}[v_i(nT) + v_i(n-1)T] + \left(1 - \frac{T}{2RC}\right)v_o(n-1)T$$

This equation simplifies to:

$$v_o(nT) = \frac{T}{2RC+T}v_i(nT) + \frac{T}{2RC+T}v_i(n-1)T + \frac{2RC-T}{2RC+T}v_o(n-1)T \qquad \textbf{(1.10)}$$

The symbol *T* stands for the sampling period. Its choice is not arbitrary and, as we shall see in later chapters, the value of *T* is a critical design choice in the creation of any DSP system. For this example we will take *T* = 1.0 msec. With R = 1K and C = 1 μf, (1.10) becomes

$$v_o(nT) = 0.3333v_i(nT) + 0.3333v_i([n-1]T) + 0.3333v_o([n-1]T) \qquad \textbf{(1.11)}$$

Equation (1.11) is called a difference equation and expresses the output voltage v_o at time *nT* in terms of the input voltage, the previous input voltage, and the previous output voltage.

The computer in Figure 1.4 could implement this equation in a program that looks like the following:

```
Initialize Variables
DO Forever
   Call A/D(Vi)                  ;Get a sample from the A/D
   Vo = K1*Vi + K2*Vi1 + K3*Vo1  ;K1 = K2 = K3 = 0.3333
   Call D/A(Vo)                 ;Output Vo to D/A
   Vo1 = Vo             ;Reset the values of the old variables.
   Vi1 = Vi
   Wait for T seconds to pass
Loop
End
```

Note that in this program all of the statements in the loop are completed exactly once every T seconds.

Every time a new sample is taken in via the A/D converter, the computer calculates a filtered output and sends the output value to the D/A converter. The D/A converter changes this binary output back to an analog voltage. The output of the D/A will not be a smooth analog signal. Rather, it will appear in discrete steps where each step is T seconds long. If we examine the output it might look something like that shown in Figure 1.6.

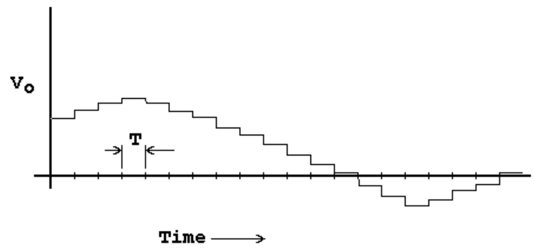

Figure 1.6
A typical stair stepped output voltage signal from the D/A converter.

The signal coming from the D/A is still discrete in time and discrete in amplitude. In a practical application this signal would be passed through a *reconstruction filter* to smooth the steps and make the signal continuous in both time and amplitude. Reconstruction filters are taken up in Chapter 4.

To find out how well our digital filter approximates the RC circuit from which it was derived, we need to do an analysis of the filter. This can be done mathematically or empirically. To do the empirical analysis we would build the filter and input a step function. We could then observe the response to get a good idea of what is called the transient response of the filter. We could also input sine waves of various frequencies and measure the amplitude of the input and output waves as well as their relative phase. We could then create a frequency vs gain curve much like that of Figure 1.2.

The mathematical analysis is a bit more complicated. Consider the step response first, since that is the easiest to calculate. We know that the filter is governed by the difference equation (1.11).

If Vi is a step function, then it has a value of 0 for all $t < 0$ and a value of 1 for all $t \geq 0$. We can therefore tabulate v_o as follows:

nT	$v_i(nT)$	$v_o([n\text{-}1]T)$	$v_o(nT)$
-1	0	0	0
0	1	0	.3333
1	1	0.3333	0.7777
2	1	0.7777	0.9258
3	1	0.9258	0.9752
4	1	0.9752	0.9916
5	1	0.9916	0.9971
6	1	0.9971	0.9989
7	1	0.9989	0.9995
8	1	0.9995	0.9997

Table 1.1
Tabulated step response.

A graph of this tabulated data shows a transient response similar to that of the RC circuit (see Figure 1.7).

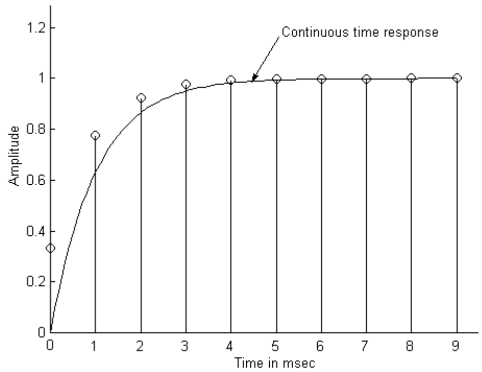

Figure 1.7
Step response of a digital filter approximating the RC circuit of Figure 1.0. The continuous time step response has been superimposed for comparison.

Example 1.1
Using trapezoidal integration as outlined above, find the difference equation and the corresponding step and impulse responses for the analog filter whose differential equation is given by

$$\frac{dv_o}{dt} + 1000v_o = \frac{dv_i}{dt}$$

Use a sample frequency of 1000 Hz.

Solution
Begin by converting the differential equation to one with integration instead of differentiation.

$$v_o + 1000\int v_o dt = v_i$$

Convert this to discrete form using (1.8).

$$v_i - v_o = 1000\int v_o dt$$

$$v_i(nT) - v_o(nT) = v_i([n-1]T) - v_o([n-1]T) + 1000T[v_o(nT) + v_o([n-1]T)]/2$$

Solve this equation for $v_o(nT)$ with $T = 0.001$.

$$v_o(nT) = K_1[v_i(nT) - v_i([n-1]T)] + K_2 v_o([n-1]T) \text{ where } K_1 = 2/3 \text{ and } K_2 = 1/3$$

Since this is a causal system all values of v_i and v_o for $t < 0$ are zero. We construct the following table to get the step and impulse response.

	Step Response					**Impulse Response**			
n	$v_i(nT)$	$v_i([n-1]T)$	$v_o(nT)$	$v_o([n-1]T)$		$v_i(nT)$	$v_i(n-1)T$	$v_o(nT)$	$v_o(n-1)T$
-1	0	0	0	0		0	0	0	0
0	1	0	0.667	0		1	0	0.667	0
1	1	1	0.222	0.667		0	1	-0.444	0.667
2	1	1	0.0741	0.222		0	0	-0.1481	-0.444
3	1	1	0.0247	0.0741		0	0	-0.494	-0.1481
4	1	1	0.0082	0.0247		0	0	-0.165	-0.494
5	1	1	0.0027	0.0082		0	0	-0.0055	-0.165

Table 1.2

Step and impulse response terms found by iteration.

The sinusoidal response is a bit more complicated. We could (if we had enough time) tabulate the response to various frequency sinusoids. We could do this by allowing v_i to take on sinusoidal values and using the difference equation to calculate v_o. We would have to be careful to allow each sinusoid to run for many cycles since we are interested in the steady state sinusoidal response and not the response to the sinusoidal start up (the transient).

For a mathematical result we can let the input become a complex, but sampled, exponential: $v_i(nT) = e^{j\omega nT}$. In this case, the difference equation (1.11) can be written as:

$$v_o(nT) = K_1 e^{j\omega nT} + K_2 e^{j\omega(n-1)T} + K_3 v_o([n-1]T)$$

where the K values are constants corresponding to the coefficients in (1.11). This equation can be rewritten as:

$$v_o(nT) - K_3 v_o([n-1]T) = (K_1 + K_2 e^{-j\omega T})e^{j\omega nT} \qquad \textbf{(1.12)}$$

Differential equations can be viewed as the limit case of a difference equation. Hence the solution of differential and difference equations is similar in concept. Equation (1.12) has both a homogeneous solution (found by setting the right side to zero) and a particular solution. The homogeneous solution is used in the transient response and for stable systems, the homogeneous solution decreases in magnitude with time. The particular solution brought about by the forcing function, produces the steady state response.

As in the case of differential equations, one method of finding the particular solution to (1.12) is to assume that the output is in a form that is similar to the input. To try this we let v_o become a complex exponential

$$v_o(nT) = Ae^{j\omega nT}.$$

In this equation A is a complex number. Substituting this value of v_o into (1.12) gives

$$Ae^{j\omega nT} - K_3 A e^{j\omega(n-1)T} = (K_1 + K_2 e^{-j\omega T})e^{j\omega nT}. \qquad \textbf{(1.13)}$$

Equation (1.13) can be solved for the amplitude factor A as follows:

$$A = \frac{(K_1 + K_2 e^{-j\omega T})e^{j\omega nT}}{(1 - K_3 e^{-j\omega T})e^{j\omega nT}} = \frac{K_1 + K_2 e^{-j\omega T}}{1 - K_3 e^{-j\omega T}}.$$

Thus the steady state solution for v_o for a complex exponential forcing function can be written as:

$$v_o(kT) = \frac{K_1 + K_2 e^{-j\omega T}}{1 - K_3 e^{-j\omega T}} \cdot e^{j\omega nT}. \qquad \textbf{(1.14)}$$

Since the input was a complex sinusoid with a magnitude of 1, we can write the gain of this filter as

$$\frac{v_o(nT)}{v_i(nT)} = \frac{K_1 + K_2 e^{-j\omega T}}{1 - K_3 e^{-j\omega T}}. \qquad \textbf{(1.15)}$$

The gain is then just a complex number whose magnitude is a function of the frequency. This magnitude is plotted in Figure 1.8. The magnitude of the continuous time RC circuit from Figure 1.1 has been superimposed so that the two can be compared.

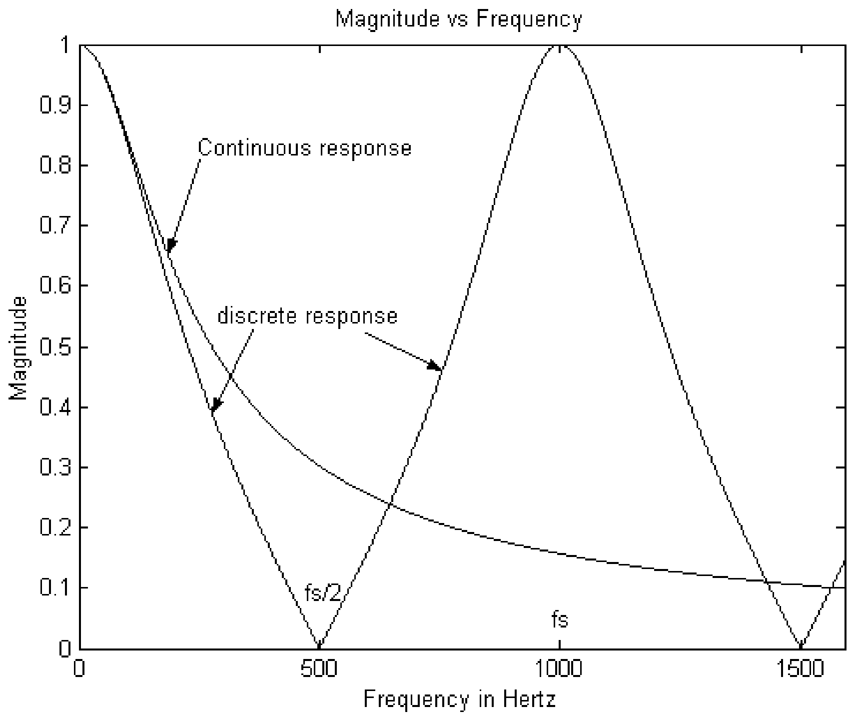

Figure 1.8
The magnitude response for the discrete filter. The continuous time response has been superimposed for comparison. The sample frequency for the discrete filter is 1000Hz.

At first glance the digital version of the filter appears to be a poor imitation of the continuous time filter. But as we shall see in Chapter 4 (on sampling), all DSP systems have a limited useful frequency range due to a phenomenon called aliasing. The useable frequency range for this particular digital filter is from 0 Hz to 500 Hz which corresponds to exactly one-half of the sample frequency. Signals which have frequencies above one-half of the sample frequency are *aliased* and produce a response that corresponds to a frequency below one-half of the sample frequency. One-half of the sample frequency is often referred to as the *fold-over frequency* since the response from 500 Hz to 1000 Hz is the mirror image of the response from 0 Hz to 500 Hz. The response of any DSP system is "re-imaged" repeatedly through all of frequency space in increments corresponding to the fold over frequency.

Example 1.2
Apply the input $v_i(nT) = e^{j\omega nT}$ to the difference equation given in Example 1.1 and plot the output frequency response. The sampling frequency is 1000 Hz.

Solution
The difference equation from Example 1.1 is:

$$v_i(nT) - v_o(nT) = v_i([n-1]T) - v_o([n-1]T) + 1000T[v_o(nT) + v_o([n-1]T)]/2$$

If we set $v_i(nT) = e^{j\omega nT}$ and solve for $v_o(nT)$ we get

$$v_o(nT) = (2/3)[v_i(nT) - v_i([n-1]T)] + (1/3)v_o([n-1]T).$$

If we assume the output is the same frequency as the input and take $v_o(nT) = Ae^{j\omega nT}$ where A is a complex number we get

$$Ae^{j\omega nT} = (2/3)(e^{j\omega nT} - e^{j\omega(n-1)T}) + (1/3)Ae^{j\omega(n-1)T} \quad \textbf{(1.16)}$$

The magnitude of $v_o(nT)$ is $|A|$. Solving (1.16) for this magnitude gives

$$A = \frac{(2/3)(1 - e^{-j\omega T})}{(1 - e^{-j\omega T}/3)}.$$

We can use MATLAB® to plot the frequency response from 0 to $f_s/2$.

```
e = exp(1);
fs = 1000;T = 1/fs;
f = 0:fs/2;
w = 2*pi*f;
jwT = j*w*T;
num = abs(2/3*(1 - e.^-jwT));
den = abs(1-e.^-jwT/3);
Amag =num./den;
figure(1);clf;
plot(f, Amag);
axis([0 fs/2 0 1.2]);
xlabel('frequency in Hz');
ylabel('Magnitude of Gain');
title('Frequency plot');
```

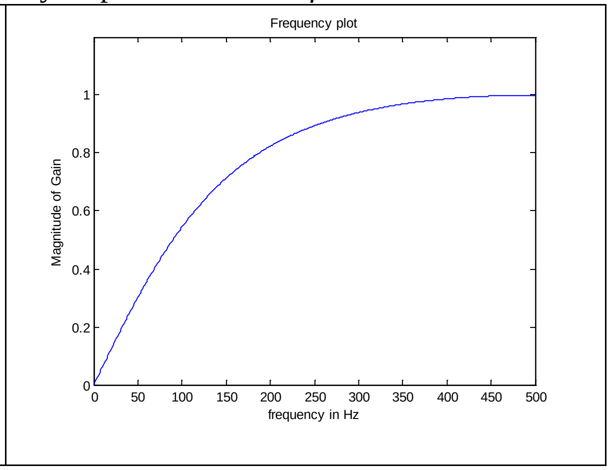

Figure 1.9
The frequency response plot for the difference equation given in Example 1.1.

We note from Figure 1.9 that the difference equation represents a high pass filter.

A Simple FIR Method

To illustrate a second method of creating a digital filter we will begin with the impulse response of the original RC circuit. From linear systems theory, we know that any two linear and time-invariant systems which have identical impulse response functions will have an identical response to any other input as well.

A block diagram for a system which can approximate the impulse response of an existing system is shown in Figure 1.10. In this figure, the blocks marked with a T are unit delays. In practical terms these could be implemented using a clocked register. The discrete signal coming in is represented as a binary number which is held in the register for one clock period before it is passed on to the next register. The set of registers in a row thus form a delay line. Putting a unit impulse into such a system at time zero will produce a response given by $h(kT) = \{b_0, b_1, b_2, b_3, b_4, b_5\}$. To create a filter using this technique, we need only sample the impulse response and set the b_i equal to the sample values.

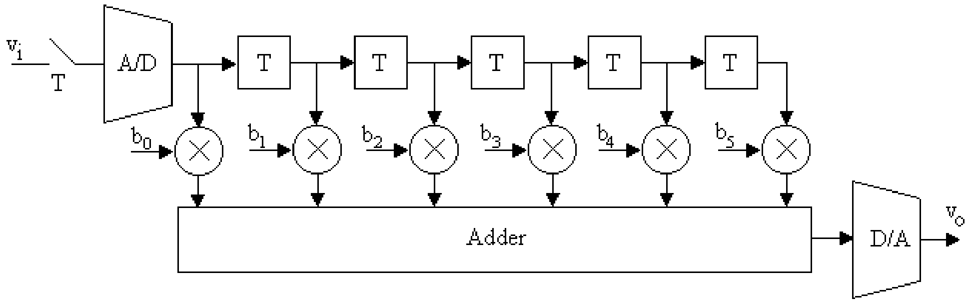

Figure 1.10

A Finite Impulse Response filter. The impulse response is $h(nT) = \{b_0, b_1, b_2, b_3, b_4, b_5\}$

The impulse response of the RC circuit of Figure 1 is given by $h(t) = \frac{1}{RC} e^{-t/RC}$. If we take R = 1K, C = 1μf, and the sample rate to 1 msec and we scale the impulse response to have a maximum value of 1.0 the sample values for the b_i are $b_0 = 1.0$, $b_1 = 0.3679$, $b_2 = 0.1353$, $b_3 = 0.0498$, $b_4 = 0.0183$, and $b_5 = 0.0067$. The difference equation for the FIR filter is

$$\begin{aligned} v_o(nT) = v_i(nT) + 0.3679 v_i([n-1]T) + 0.1353 v_i([n-2]T) + 0.0498 v_i([n-3]T) \\ + 0.0183 v_i([n-4]T) + 0.0067 v_i([n-5]T) \end{aligned} \qquad \textbf{(1.17)}$$

The frequency response function may be found by allowing $v_i(nT)$ to be a complex exponential and solving the corresponding difference equation. The magnitude of the frequency response is shown in Figure 1.11 along with the original frequency response for the RC circuit. Only that part of the response up to one-half of the sample frequency is shown. Due to sampling the frequency response beyond 500 Hz is an aliased version of that shown.

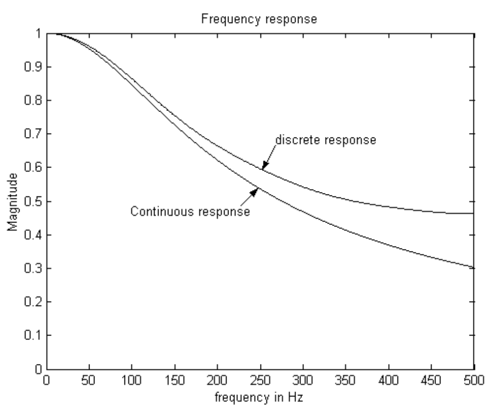

Figure 1.11

The discrete frequency response of a FIR filter using 6 samples of a continuous time RC low pass filter impulse response.

Example 1.3

A RLC circuit has an impulse response given by

$$h(t) = Ke^{-\alpha t}[\omega_d \sin(\omega_d t) - \alpha(1 - \cos \omega_d t)]$$

where $\alpha = 50$, $\omega_d = 1200$, and $K = 0.001$

Sample this impulse response at 1000 Hz and create an FIR filter from the first 30 samples. Find the frequency response of the resulting filter.

Solution

We begin by finding the first 30 terms with a sample rate of 1000 Hz.

```
fs = 1000;T = 1/fs;
f = 0:fs/2;
Len = 30;
t = 0:T:(Len-1)*T;
a = 50;wd = 1200;K = 0.001;
h = K*exp(-a*t).*(wd*sin(wd*t) - ...
               a*(1 - cos(wd*t)));
figure(1);clf;
stem(t, h);
axis([0 Len*T -2 5.2]);
xlabel('Time in seconds');
ylabel('h');
title('Sampled impulse response');
```

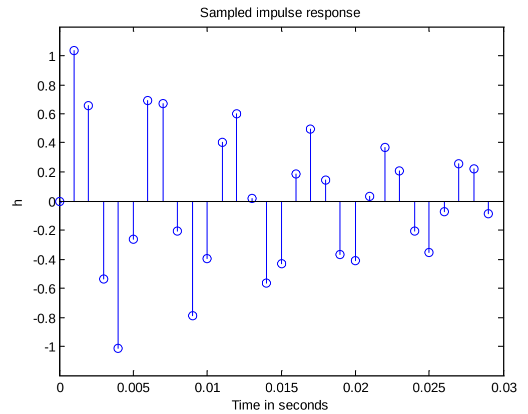

Figure 1.12

Calculating and plotting the first 30 samples of $h(t)=Ke^{-\alpha t}[\omega_d \sin(\omega_d t)-\alpha(1-\cos\omega_d t)]$

The difference equation will have the same form as (1.17).

$$v_o(nT)=h(0)v_i(nT)+h(1)v_i([n-1]T)+h(2)v_i([n-2]T)+\cdots$$
$$+h(28)v_i([n-28]T)+h(29)v_i([n-29]T) \quad \textbf{(1.18)}$$

To find the frequency response we let $v_i(nT)=e^{j\omega nT}$ and $v_o(nT)=Ae^{j\omega nT}$ where A is a complex number. Substituting these values into (1.18) gives

$$Ae^{j\omega nT}=h(0)e^{j\omega nT}+h(1)e^{j\omega(n-1)T}+h(2)e^{j\omega(n-2)T}+\cdots$$
$$+h(28)e^{j\omega(n-28)T}+h(29)e^{j\omega(n-29)T}$$

We can divide out $e^{j\omega nT}$ from both sides to get

$$A=h(0)+h(1)e^{-j\omega T}+h(2)e^{-j\omega 2T}+\cdots+h(28)e^{-j\omega 28T}+h(29)e^{-j\omega 29T}$$

This equation can be used to plot the magnitude of A versus frequency. Since the magnitude of the input is one, the magnitude of *A* is the gain for the circuit. Figure 1.13 shows the results.

```
w = 2*pi*f;
jwT = j*w*T;
mag = zeros(1, length(f));
for i = 1:Len
    mag = mag + h(i)*(exp(jwT*(1-i)));
end
mag = abs(mag);
figure(2);clf;
plot(f, mag);
```

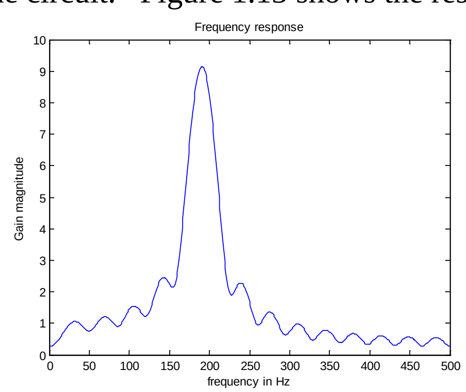

Figure 1.13

Calculating and plotting the frequency response. The frequency response shows this circuit to be a resonant circuit with a resonant frequency near 200 Hz.

The IIR and FIR filters produced in the previous examples are far from ideal and are given to illustrate the process and add some intuition as to how DSP systems work in general. Several complex design issues have been simplified in this introduction. These will be addressed in a more rigorous fashion in later chapters.

1.2 Overview of Analysis and Design

A good design requires an analysis to verify that the design works. Hence, we will take up the analysis of digital filters and then move on to design and finally to implementation.

The Analysis Process

Analysis of digital filters can be done in the time domain or the frequency domain. We are interested in both. Time domain analysis generally concerns itself with the transient analysis of the filter. This includes the impulse and step response. If a system is defined by its difference equation, we can always find its impulse or step response by iteratively evaluating the equation when the input is a unit impulse or a unit step. (In the previous section we found the step response by this method.) It is also possible to solve the difference equation in closed form. This closed form allows us to determine the value of the response at any point without first evaluating all of the previous points. To relieve the tedium of the calculations many computer programs exist which iteratively find the transient response and present it in graphical form.

The transient response of a filter is important for several reasons. It shows whether the filter is stable or not (whereas the frequency response does not always make this apparent). The transient response also can make it apparent what kind of filter we are dealing with (FIR, IIR, low pass or high pass, etc.) For FIR filters the transient response is important and central to the design process. (Indeed, for FIR filters the impulse response directly determines the filter coefficients in the implementation.)

The impulse response can be used to deduce the step response and vice-versa. Likewise, if we know the impulse response completely we can determine the transfer function (using the z-transform) and the frequency response.

The frequency response of a filter is likewise important since it is a measure of how well the filter meets the design criteria (which are normally specified in the frequency domain). Filters are classified as low pass, high pass, band pass, or band stop according to the shape of their frequency response plot.

To calculate the frequency response of a filter we will rely on the *z*-transform to make the math manageable. Transforming a difference equation from the time domain into *z* allows most of the mathematical operation to become algebraic. The math is still tedious but lends itself to computer solution. The procedure for solution is analogous to that used in the continuous-domain with the Laplace transform. A differential equation can be transformed from the time domain to the *s*-domain by way of the Laplace transform. Substitution of $j\omega$ for *s* produces a transfer function whose magnitude and phase are the frequency response function for the continuous signal. In the case of a difference equation we substitute $e^{j\omega T}$ for z. Using Euler's identity produces a complex function in ω which is the frequency response function for the filter. Thus the procedure for the digital system is analogous to that of the analog system but the math is slightly more complicated.

The Design Process

The design of digital filters is classically broken into two distinct sections - the design of FIR filters and the design of IIR filters. FIR stands for *Finite Impulse Response* and characterizes filters which have no feedback terms. That is the calculation of the next output in the difference equation depends only on the value of the input and the past inputs. IIR stands for *Infinite Impulse Response* and characterizes those filters which have feedback terms. This implies that the calculation of the next output depends not only on the input and the past inputs but also on the past outputs.

An FIR filter can be thought of as a delay line in which the input is passed through a series of registers (digital delays). Each register receives the input and passes it on serially to the next during each sample period. Figure 1.10 shows the structure for a typical FIR filter. Alternatively, we could use the computer structure shown in Figure 1.4 and implement the difference equation in software. Both methods are widely used and which method is chosen is one of the tradeoffs in the implementation process.

In Figure 1.10, each block marked with a T represents a parallel-in, parallel-out register which is clocked each sample period. Each sampled input is passed from the A/D converter through the register set "falling" out the end after n sample periods. As each sample passes through the registers it is multiplied by a constant ($b_0, b_1, \cdots, b_n$), and summed before it is passed to the D/A converter for output. A single impulse passing through such a system produces the output sequence given by:

$$h(nT) = \{b_0, b_1, b_2, ..., b_n\}$$

The coefficients of the filter structure are thus the impulse response terms. There is no feedback and the impulse response is always truncated and finite.

The difference equation for the structure in Figure 1.10 is given by:

$$v_o(nT) = b_0 v_i(nT) + b_1 v_i(n-1)T + b_2 v_i(n-2)T + \cdots + b_m v_m(n-m)T$$

The design of FIR filters thus relies on methods to extract the impulse response from a frequency specification. This can be done using the inverse Fourier transform. Since the impulse response of a practical filter is necessarily finite and causal, the digital filter response only approximates that of an ideal filter. This approximation leads to undesirable ripple and other errors. A process called "windowing" allows a designer to make tradeoffs as to where the error is located. The design process for a complete FIR filter consists of selecting a frequency specification, extracting the corresponding impulse response, and applying a window to the filter to correct certain errors.

IIR filter design is decidedly more open-ended than FIR filter design. This results in more tradeoffs and in general, a more complex process. Classically, IIR filters have been designed by starting with an existing analog filter design and mapping a transfer function in *s* to one in *z*. There are many methods of mapping *s* to *z*. Three that we will explore include the method of differences, the invariant impulse response, and the bilinear transform.

The method of differences is straight forward but does not lead to good results when frequencies approach half the sample rate. The invariant impulse response technique is widely used in digital control systems. The bilinear transform method is the most widely used method of designing IIR filters from analog filters. Empirically, it produces good results over a wide frequency range and lends itself to computer solution.

Since IIR filters have feedback terms, many different structures are possible. Figure 1.14 shows one structure for a second order IIR filter.

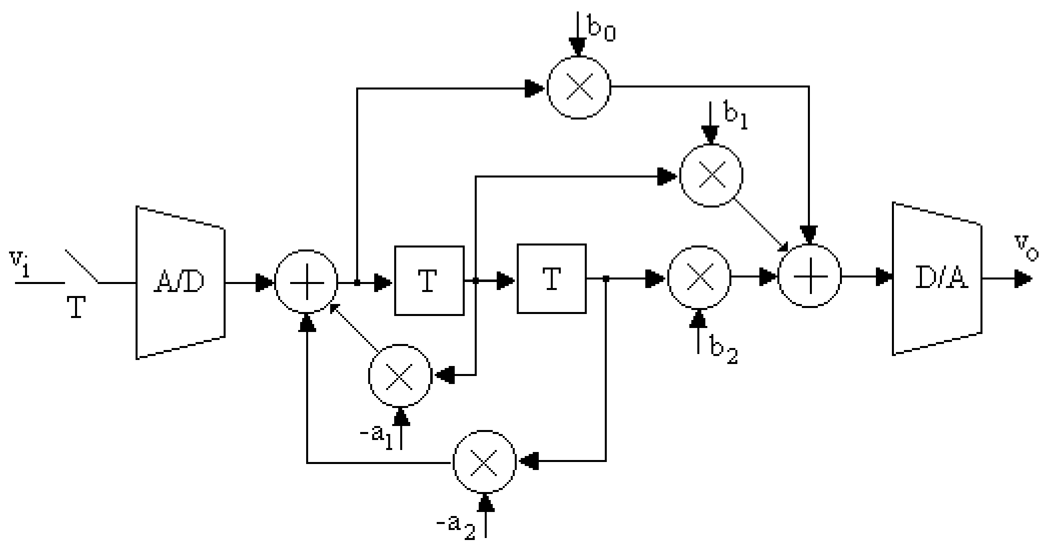

Figure 1.14
One structure for a second order IIR filter.

In comparing IIR and FIR filters it is generally true that, for a given specification, an IIR filter can be constructed to meet that specification with a lower order filter than an FIR filter. One can rightfully ask then, why FIR filters are needed. As it turns out, the process of converting *s* to *z* in the design of IIR filters alters the phase characteristics of the filter in *z* in a nonlinear fashion. In other words, the design process for IIR filters produces a faithful magnitude response but does not consider the phase response of the filter. In some applications, such as communications where phase carries information, this distortion is not tolerable. In these cases, FIR filters can be used since they can be designed to have linear phase. Indeed, FIR filters are almost always designed to meet the conditions necessary to guarantee linear phase. In cases where phase shift is unimportant, IIR filters are used.