CHAPTER 9

Frequency-Domain Signal Analysis

Frequency-domain signal analysis covers a wide variety of techniques involving the Fourier transformation of the signal. The signal's frequency domain representation is then manipulated, decomposed, segmented, classified, and interpreted. One central idea is that of a *filter*: a linear, translation-invariant system that allows one band of frequencies to appear in the output and suppresses the others. Where signal elements of interest occupy a restricted spectrum, filters invariably enter into the early processing of candidate signals. In other ways—often purely theoretical—frequency-domain analysis is important. For example, in this chapter we substantiate the methods of matched filtering and scale-space decomposition, and the Fourier transform plays a crucial role.

The main tools for frequency-domain analysis are of course the discrete signal transforms: the discrete-time Fourier transform (DTFT); its generalization, the z-transform; and especially the discrete Fourier transform (DFT). Many of the introductory applications proceed from Chapter 1 examples. There are extensions of techniques already broached in Chapters 4 and 6. Modern spectral analysis applications have a digital computer at their heart, and they rely on the either the DFT or one of its many fast versions. Some signal filtering applications use infinite impulse response (IIR) filtering methods, implemented using feedback, as discussed in Chapter 2. The DTFT is convenient for obtaining a theoretical understanding of how such filters suppress and enhance signal frequencies. We often begin with an analog filter and convert it to a discrete filter. Thus, we shall have occasion to use the continuous-domain Fourier transform. We also briefly explain how the Laplace transform, a generalization of the Fourier transform, can be used in certain analog filter constructions. The z-transform figures prominently in this conversion process.

We are generally working with complex-valued signals, but the thresholding, segmentation, and structural decomposition methods that Chapter 4 developed for time-domain signal analyis are just as useful in the frequency domain. For example, to find the spectral region where a source signal contains significant energy, we threshold the signal's magnitude or squared magnitude (power) spectrum. We know that thresholding is often improved by filtering the data, so we are inclined to filter the frequency-domain magnitudes too. This leads directly to the technique of windowing

Signal Analysis: Time, Frequency, Scale, and Structure, by Ronald L. Allen and Duncan W. Mills
ISBN: 0-471-23441-9

time-domain signal slices before Fourier transformation. The meaning of the analytical results can be quite different, of course; but as long as we understand the transform relation clearly and capture that in the application design, then the time-domain and frequency-domain procedures are remarkably similar. In some applications, the results of this analysis convey the signal content. For other tasks, an inverse transformation back to the time domain is required. In any case, the principal tools are the Fourier transform and its inverse, in both their analog and discrete formulations.

Our theoretical resources include Chapters 5, 7, and 8. This chapter introduces some further theory, appropriate to the particular applications upon which we focus. Specific applications include tone detection, speech recognition and enhancement, and chirp analysis. Some of these experiments show that the Fourier transform is precisely the tool we need to make the application work. Further reflection reveals problems in applying the Fourier transforms. This motivates a search for frequency transforms that incorporate a time-domain element: time-frequency and time-scale transforms, which are topics for the final three chapters.

Fourier-domain techniques also offer many insights into our earlier material. Scale space and random signals are considered once more, this time from the vantage point of the new frequency-domain methods. The last two sections detail the construction of special filters for frequency analysis and possible structures for their application. Later chapters will draw the link between this approach to signal processing and the notion of a multiresolution analysis of the L^2 Hilbert space.

References on Fourier transforms include Refs. 1–5. Popular signal processing texts that introduce Fourier methods and construct filters from the theory are Refs. 6–10. An older book that concludes its thorough coverage of signal theory with detailed application studies in speech and radar signal analysis is Ref. 11.

Notation. The present discussion covers both analog and discrete filters. We use the following notations for clarity:

(i) *Analog filters*: The impulse response is $h(t)$, the radial Fourier transform is $H(\Omega)$, and the Laplace transform is $H_L(s)$; in some contexts, we insert the subscript a: $h_a(t)$, $H_a(\Omega)$, and $H_{L,a}(s)$.

(ii) *Discrete filters*: The impulse response is $h(n)$, or any FORTRAN-like integer independent variable such as $h(i)$, $h(k)$, or $h(m)$; the discrete time Fourier transform is $H(\omega)$; and the discrete Fourier transform is $H(k)$.

(iii) We continue to use $j^2 = -1$.

9.1 NARROWBAND SIGNAL ANALYSIS

The most basic frequency-domain analysis task involves detecting and interpreting more or less isolated periodic components of signals. For some signals, or at least for some part of their domain, a few oscillatory components contain the bulk of the energy. It is such *narrowband* signals—sinusoids (tones) and dual-tones, mainly, but we could also allow complex-valued exponentials into this category—that we

begin our study of Fourier transform applications. We distinguish narrowband signals from wideband signals, where the energy is spread over many frequencies. A signal that contains sharp edges, for example, will generally have frequency-domain energy dispersed across a wide spectral range.

Although basic, a tone detection application leads to important practical concepts: noise removal, filtering, phase delay, group delay, and windowing. *Filters* are frequency-selective linear translation invariant systems. Filtering a signal can change the time location of frequency components. For instance, a filter might retard a sinusoidal pulse and the signal envelope itself, depending on their frequency. These time lags define the *phase* and *group delay*, respectively. Knowing them is crucial for application designs that must compensate for filter delays. Finally, we often need to analyze signals in chunks, but so doing invariably corrupts the signal spectrum. Only by looking at the signal through a well-constructed *window* can we mitigate this effect. This concept leads directly to the modern theory of the windowed Fourier transform (Chapter 10) and eventually to wavelets (Chapter 11).

Theory from earlier chapters now becomes practice. For designing filters, we employ the discrete-time Fourier transform (DTFT). For implementing filters on a computer, we use the discrete Fourier transform (DFT) or one of its fast Fourier transform (FFT) schemes. Some infinite impulse response filter implementations are particularly powerful, and we visualize their possible recursive implementation on a computer through the z-transform (Chapter 8). We definitely do not need the continuous-domain Fourier transform (FT), right? Quite wrong: We can obtain very good discrete filters by first designing an analog filter and then converting it into a discrete filter. It is perhaps a surprising fact, but this is the preferred method for constructing high-performance discrete filters. We even use the Fourier series (FS); after a bit of contemplation, we realize that the FS converts an analog $L^2[0, 2\pi]$ signal into a discrete l^2 signal—just like the inverse DTFT. Indeed, as mathematical objects, they are one and the same.

9.1.1 Single Oscillatory Component: Sinusoidal Signals

Let the real-valued signal $x(n)$ contain a single oscillatory component and perhaps some corrupting noise. We have seen such examples already in the first chapter—$x(n)$ is the Wolf sunspot count, for instance. Now, in earlier chapters, introductory applications explored the discrete Fourier transform as a tool for detecting such periodicities. The source signal's oscillatory component manifests itself as an isolated cluster of large magnitude spectral values. We can usually segment the frequency components with a simple threshold around the maximum value; this is a straightforward application of the DFT.

A tone is a time-domain region of a signal that consists of only a few sinusoidal components. Briefly, detection steps are as follows:

(i) Select signal regions that may contain tones.

(ii) For noise removal and frequency selection, processing the signal through various filters may benefit the analysis steps that follow.

(iii) Fourier transform the signal over such regions.

(iv) For each such region, examine the spectrum for large concentrations of signal energy in a few frequency coefficients.

(v) Optionally, once a possible tone has been identified through frequency-domain analysis, return to the time domain to more precisely localize the tone.

The discrete signal sometimes arises from sampling an analog signal $x_a(t)$: $x(n) = x_a(nT)$, where $T > 0$ is the sampling period. But perhaps—as in the case of sunspot estimates—the signal is naturally discrete. If we take select a window of signal values, $0 \le n < N$, we can compute the DFT over these N samples:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-2\pi jkn}{N}}. \tag{9.1}$$

In (9.1), $X(0)$ represents the direct current (DC), or constant, or zero frequency component. The signal average over the interval $[0, N-1]$ is $X(0)/N$, and it represents zero cycles per sample (hertz) in the transform. If $x(n)$ is real, then $X(k) = X(N-k)$ for all $1 \le k \le N-1$. The transform is invertible:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{2\pi jkn}{N}}. \tag{9.2}$$

Equations (9.1) and (9.2) are the analysis and synthesis equations, respectively, for the DFT. If $x_a(t) = \cos(2\pi t/NT)$ is a real-valued analog sinusoid with frequency $(NT)^{-1}$ hertz, then upon sampling it becomes $x(n) = x_a(nT) = \cos(2\pi n/N)$. We can expand $2x(n) = [\exp(2\pi jn/N) + \exp(2\pi jn(N-1)/N)]$, which is a synthesis equation (9.2) for the sinusoid. Thus, the smallest frequency represented by the transform coefficients, the frequency resolution of the DFT, is $(NT)^{-1}$. This means that energy from source signal periodicities appears in the transform coefficients in at least two different places. If N is even, then the highest frequency represented by the transform values is $1/(2T) = (NT)^{-1} \times (N/2)$ hertz. If N is odd, then the two middle coefficients share the Nyquist frequency energy.

This section explains the basic methods for applying the discrete Fourier transform (DFT) and discrete-time Fourier transform (DTFT) for detecting oscillatory components of signals. Strictly speaking, the DFT applies to periodic signals and the DTFT to aperiodic signals.

9.1.2 Application: Digital Telephony DTMF

Let us consider the problem of recognizing multiple oscillatory components in a source signal. If the source contains multiple periodic components, then the transform exhibits multiple clusters of large values. To spot significant oscillatory signal features, we might invoke the more powerful threshold selection methods covered in Chapter 4, applying them to the magnitude spectrum instead of the signal

amplitude values. Thresholding the magnitude spectrum does work, but it does not take us very far.

This is the limitation of peak finding in the magnitude spectrum: These magnitudes represent frequencies over the entire time domain of the source signal. If oscillations at different times have the same frequency, or those that occur at the same time overlap with others of different wavelengths, then this potentially crucial information for signal understanding is lost in the Fourier transformation. Many applications involve signals with localized frequency components. What complicates such applications is getting the Fourier transform—an inherently global transformation—to work for us in a time localized fashion.

This application—however humble—inspires three general approaches for identifying and localizing signal frequency components:

- Preliminary time-domain analysis, arriving at a segmentation of the signal's values, and subsequent frequency-domain analysis on the segments of promise (Section 9.1.2.2);
- An important tool—the *time-frequency map* or *plane*—which generally decomposes the signal into pieces defined by the time interval over which they occur and the frequencies over which their oscillatory components range (Section 9.1.2.3);
- Another significant tool—the *filter bank*—which directs the signal values into a parallel array of frequency selective linear, translation-invariant (LTI) systems (filters) and analyzes the outputs jointly (Section 9.1.2.4).

It will become clear that these three alternatives couple their time- and frequency-domain analyses ever more closely. Thus, in the first case, algorithms finish the time-domain segmentation and hand the results over to spectral analysis. Using the second alternative's time-frequency plane, in contrast, we decompose the signal into pieces that represent a particular time interval and a particular frequency span. The analyses within the two domains, instead of working in strict cascade, operate simultaneously, albeit through restricted time and frequency windows. Finally, in a filter bank, the signal's values are streamed into the filters in parallel, and application logic interprets the output of the filters. Since this can occur with each signal sample, the output of the intepretive logic can be associated with the particular time instant at which the frequency-domain logic makes a decision. So the filter bank, at least as we sketch it here, comprises a very intimate merging of both time- and frequency-domain signal analysis.

9.1.2.1 Dual-Tone Pulses in Noise. Let us review the discrete dual-tone multifrequency (DTMF) pulses that modern digital telephone systems use for signaling [12]. Table 9.1 shows the standard pairs.

True DTMF decoders—such as in actual use at telephone company central offices—must stop decoding DTMF pulses when there is speech on the line. One frequency-domain trait that allows an application to detect the presence of human voices

TABLE 9.1. DTMF Frequency Pairs[a]

Low (Hz) \ High (Hz):	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

[a]The letter tones are generally reserved for the telephone company's signaling, testing, and diagnostic uses.

voices is that speech contains second and third harmonics [13], which the DTMF tones by design do not [12]. For example, a vowel sound could contain significant energy at 300 Hz, 600 Hz, and 900 Hz. An upcoming speech analysis application confirms this. But note in Table 9.1 that the second harmonic of the low tone at 697 Hz (that would be approximately 1.4 kHz) lies equidistant from the high tones at 1336 Hz and 1477 Hz. Later in this chapter, we will consider mixed speech and DTMF tones and see how to discriminate between control tones and voice. For now, let us return to the basic tone detection problem.

Suppose we represent a DTMF telephony pulse by a sum of two sinusoids chosen from the above table enclosed within a Gaussian envelope. If we sample such an analog signal at $F_s = 8192$ Hz, then the sampling period is $T = 8192^{-1}$ s.

Let us briefly cover the synthesis of the dual-tone multifrequency signal used in this example.

The analog sinusoidal signals for a "5" and "9" tone are, respectively,

$$s_5(t) = \sin(2\pi t F_{5,a}) + \sin(2\pi t F_{5,b}), \tag{9.3a}$$

$$s_9(t) = \sin(2\pi t F_{9,a}) + \sin(2\pi t F_{9,b}), \tag{9.3b}$$

where $F_{5,a} = 770$, $F_{5,b} = 1336$, $F_{9,a} = 852$, and $F_{5,b} = 1477$ as in Table 9.1. We need to window these infinite duration signals with Gaussians that—effectively—die to zero outside a small time interval. We take the window width $L = 50$ ms, let $\sigma = L/2$, and use the window functions

$$g_5(t) = e^{-\frac{(t-t_5)^2}{2\sigma^2}}, \tag{9.4a}$$

$$g_9(t) = e^{-\frac{(t-t_9)^2}{2\sigma^2}}, \tag{9.4b}$$

where $t_5 = 0.125$ s and $t_9 = 0.375$ s are the centers of the "5" and "9" pulse windows, respectively. Let $x(t) = s_5(t)g_5(t) + s_9(t)g_9(t) + n(t)$, where $n(t)$ is a noise term.

The noise term $n(t)$ could be genuine noise arduously derived from a real system, such as a radioactive source or galactic background radiation. Or it could be pseudo-random noise conveniently synthesized on a digital computer.[1] In order to control the noise values for the experiments in this section, let us make noise by assuming some realistic distribution functions and using some standard algorithms for the synthesis.

A variety of methods exist for synthesizing noise. For example, an early algorithm for generating a uniform random variable [14] called the *congruential* generator is

$$x(n) = [Ax(n-1)](\text{mod } M), \tag{9.5}$$

where A is the *multiplier*, M is the *modulus*, and the iteration typically starts with a choice for $x(0)$, the *seed* value. The method produces values in the range $[0, M-1)$ and works better for large M. For uniform noise on [0, 1) divide (9.5) by M.

A better algorithm is the *linear congruential* generator:

$$x(n) = [Ax(n-1) + C](\text{mod } M), \tag{9.6}$$

where C is the *increment*. If $C = 0$, then (9.6) reduces to (9.5). The following values make the congruential method work well [15]: $A = 16{,}807$; $M = 2^{31} - 1$; and $C = 0$. For the linear congruential iteration, nice choices are $A = 8{,}121$; $M = 134{,}456$; and $C = 28{,}411$ [16].

There is a standard algorithm for generating pseudo-random normally (Gaussian) distributed sequences [17]. Let $x_1(n)$ and $x_2(n)$ be two uniformly distributed random variables on (0, 1) and define

$$y_1(n) = \cos(2\pi x_2(n))\sqrt{-2\ln(x_1(n))}, \tag{9.7a}$$

$$y_2(n) = \sin(2\pi x_2(n))\sqrt{-2\ln(x_1(n))}. \tag{9.7b}$$

Then $y_1(n)$ and $y_2(n)$ are zero-mean normally distributed random variables. References on random number generation include Refs. 18 and 19.

The chapter exercises invite readers to change signal to noise ratios and explore the impact on detection algorithms.

We begin with an analysis of the DTMF signal using the discrete Fourier transform on the entire time interval of $N = 4096$ points. Let $x(n) = x_a(nT)$ be the discretized input, where $T = 8192^{-1}$s, $x_a(t)$ is the real-world analog representation, and $n = 0, 1, ..., N-1 = 4095$. Figure 9.1 (second from top) shows the magnitude of the DFT coefficients (9.1). Even though the presence of DTMF tones in the signal is clear, we cannot be sure when the tones occurred. For example, we detect two low tones, 770 and 852 Hz, and we can see two high tones, 1336 and 1477 Hz, but this global frequency-domain representation does not reveal whether their presence

[1]"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin" (John von Neumann). And having repeated the maxim that everyone else quotes at this point, let us also confess that "Wise men make proverbs, but fools repeat them" (Samuel Palmer).

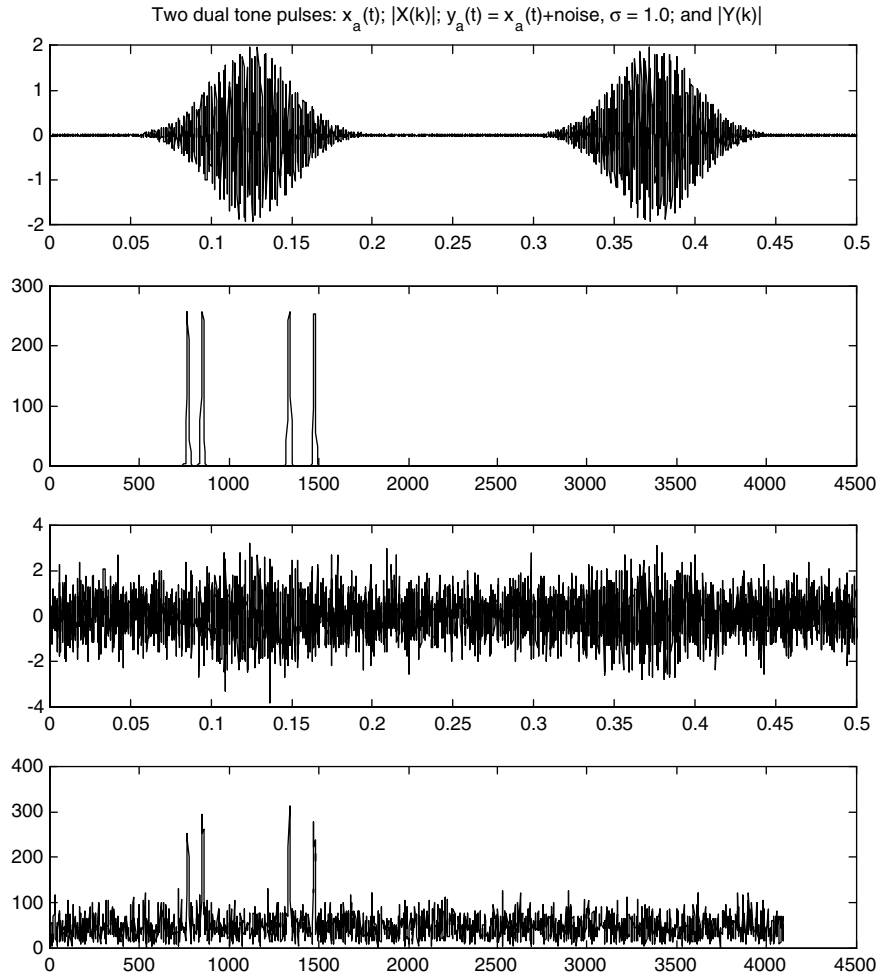

Fig. 9.1. DTMF numbers "5" and "9" tones (top). Sampling produces $x(n) = x_a(nT)$, where $T = 8192^{-1}$ s. Discrete Fourier transformation gives $|X(k)|$ (second). Adding noise of zero mean, normal distribution, and standard deviation $\sigma = 1$ effectively buries the signal (third). Yet the characteristic peaks remain in the magnitude spectrum of the noisy signal (bottom).

indicates that the time-domain signal contains a "5" pulse, a "9" pulse, a "6" pulse, an "8" pulse, or some invalid combination of tones.

The DFT efficiently detects signal periodicity. Adding a considerable amount of noise to the pure tone signals used above demonstrates this, as can be seen in the lower two panels of Figure 9.1. Should the application need to ascertain the mere existence of a periodity, obscuring noise is no problem; there is just a question of seeing the spike in $|X(k)|$. But it does become a problem when we need to find the

time-domain extent of the tone—when one tone occurs earlier than another. Indeed, high noise levels can confound as simple an application as DTMF detection.

9.1.2.2 Preliminary Time-Domain Segmentation. A straightforward approach is to preface frequency-domain interpretation with time-domain segmentation. Chapter 4's diverse thresholding methods, for example, can decide whether a piece of time-domain signal $x(n)$ merits Fourier analysis. For dual-tone detection, the appropriate steps are as follows:

(i) Segment the time-domain signal $x(n)$ into background regions and possible DTMF tone regions.
(ii) Compute the DFT of $x(n)$ on possible tone segments.
(iii) Apply the DTMF specifications and check for proper tone combinations in the candidate regions.

Time-domain signal segmentation methods are familiar from Chapter 4. If we know the background noise levels beforehand, we can assume a fixed threshold T_x. Of course, the source $x(n)$ is oscillatory, so we need to threshold against $|x(n)|$ and merge nearby regions where $|x(n)| \geq T_x$. Median filtering may help to remove narrow gaps and small splinters at the edge of high magnitude regions. If we know the probability of DTMF tones, then a parametric method such as the Chow and Kaneko algorithm [20] may be useful. However, if $x(n)$ contains other oscillatory sounds, such as speech, or the tones vary in length and temporal separation, then a nonparametric algorithm such as Otsu's [21] or Kittler and Illingworth's [22] may work better. In any case, the first step is to produce a preliminary time-domain segmentation into possible tone signal versus noise (Figure 9.2).

Nevertheless, segmentation by Otsu's method fails to provide two complete candidate pulse regions for noise levels only slightly higher than considered in Figure 9.2. It is possible to apply a split and merge procedure to fragmented regions, such as considered in the exercises. However, these repairs themselves fail for high levels of noise such as we considered in the previous section.

Since the high level of noise is the immediate source of our time-domain segmentation woes, let us try to reduce the noise. Preliminary noise removal filtering comes to mind. Thus, we can apply some frequency-selective signal processing to the input DTMF signal before attempting the partition of the source into meaning signal and background noise regions.

Let us try a moving average filter. The motivation is that the zero mean noise locally wiggles around more than the sinusoidal pulses that comprise the DTMF information. So we anticipate that filter averaging ought to cancel out the noise but leave the sinusoidal DTMF pulses largely intact.

The *moving average filter* of order $N > 0$ is $h = H\delta$, where

$$h(n) = \begin{cases} \dfrac{1}{N} & \text{if } 0 \leq n \leq N-1, \\ 0 & \text{if otherwise,} \end{cases} \tag{9.8}$$

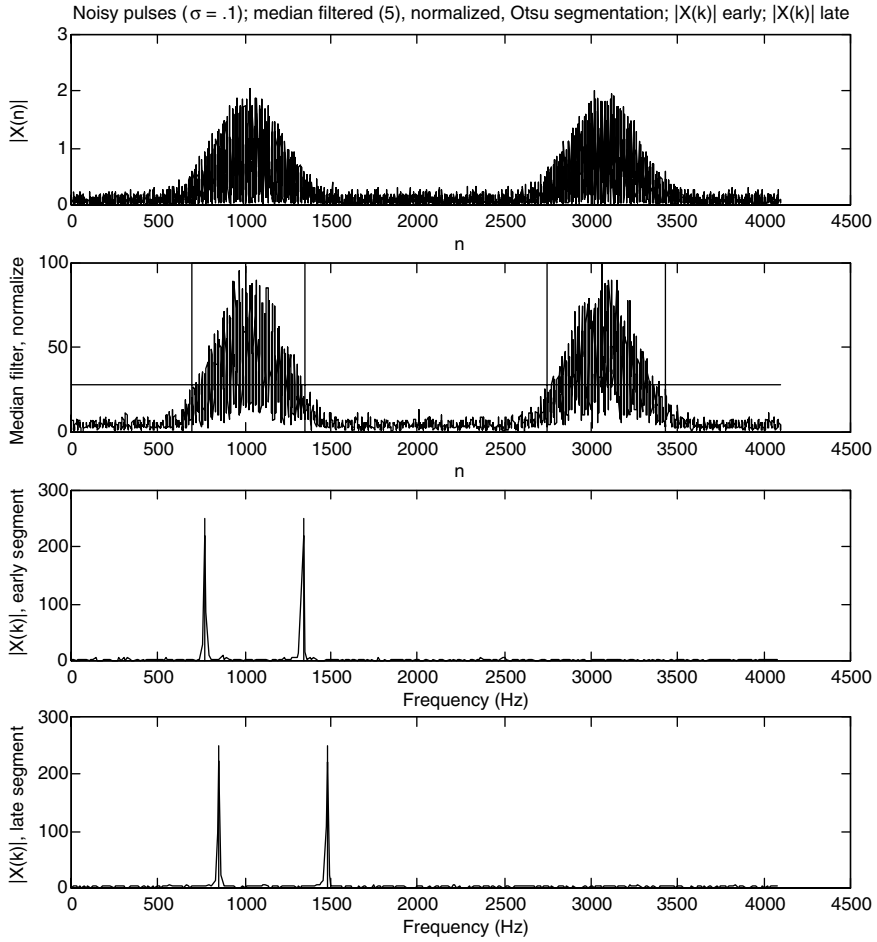

Fig. 9.2. Magnitudes of DTMF "5" and "9" tones within noise of zero mean, normal distribution, and moderate $\sigma = 0.1$ standard deviation. The second panel shows time-domain segmentation via the Otsu algorithm. Here, the magnitude spectra are median-filtered and normalized to a maximum of 100% before histogram construction and segmentation. The horizontal line is the Otsu threshold. The vertical lines are boundaries of the likely pulse regions. The lower panels show the magnitude spectra of DFTs on the two candidate regions. Note that the spikes correspond well to "5" and "9" dual-tone frequencies.

and $N > 0$ is the *order* of the filter. Let $x(n)$ be the pure DTMF signal and let us add normally distributed noise of mean $\mu_x = 0$ and standard deviation $\sigma_x = 0.8$ (Figure 9.3).

Why the time domain does not seem to aid the segmentation process is evident from examining the various frequency-domain representations. Figure 9.3 shows the spectral effect of filtering this signal with moving average filters of orders 3, 7, and 21. We can see that the smallest order is beneficial in terms of aiding a time-domain

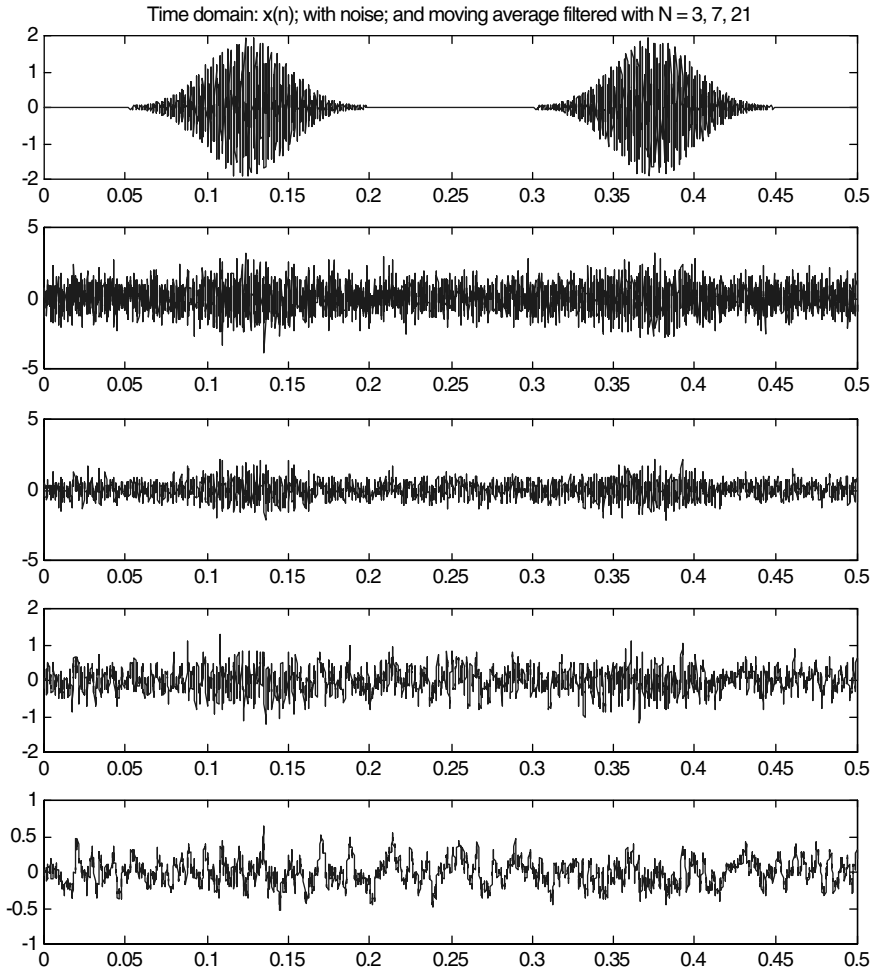

Fig. 9.3. Time-domain plots of the pure DTMF signal (top) and with Gaussian noise added, $\mu_x = 0$ and $\sigma_x = 0.8$ (second from top). The next three panels show $y = h^*x$, with H a moving average filter of order $N = 3$, $N = 7$, and $N = 21$.

segmentation, but only slightly so. The higher-order filters are—if anything—a hindrance.

We can see that the moving average filter magnitude spectrum consists of a series of slowly decaying humps (Figure 9.4). Frequencies between the humps are suppressed, and in some cases the frequency buckets that correspond to our DTMF pulses are attenuated by the filter. The filter will pass and suppress frequencies in accordance with the DFT convolution theorem: $Y(k) = H(k)X(k)$.

This explains in part why the moving average filter failed to clarify the signal for time-domain segmentation. Although it is intuitive and easy, its frequency

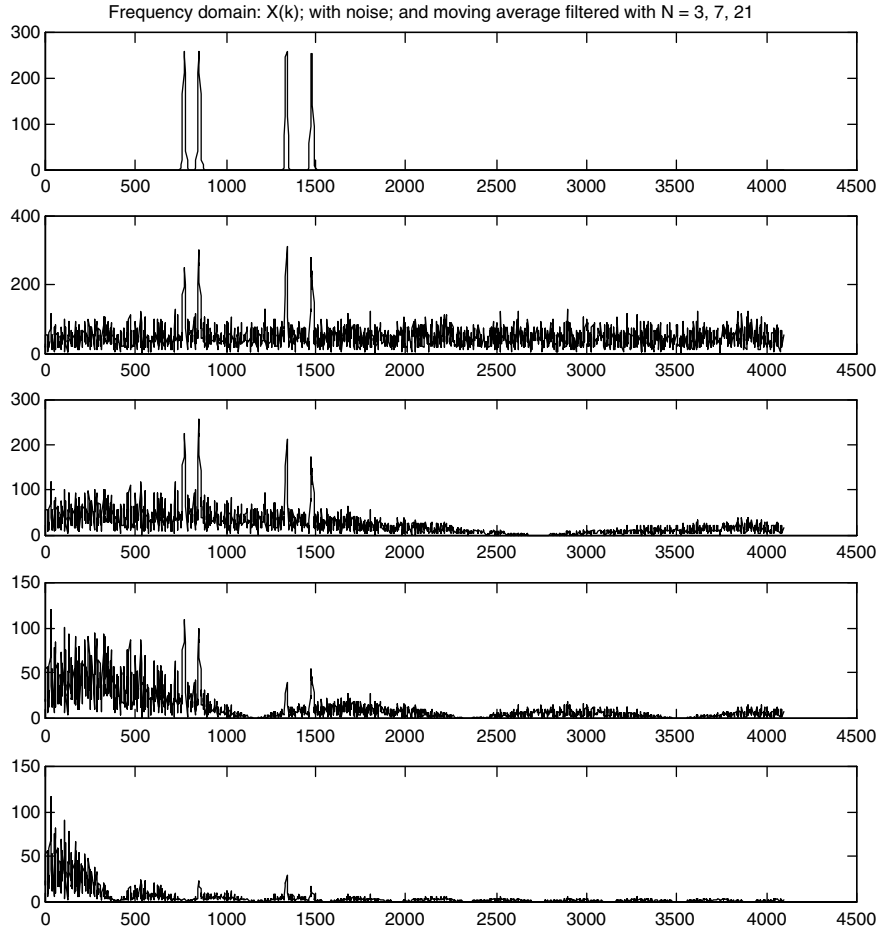

Fig. 9.4. Frequency-domain plots of the magnitude spectrum $|X(k)|$ of DTMF signal $x(n)$ (top); with Gaussian noise added, $\mu_x = 0$ and $\sigma_x = 0.8$ (b); and the final three panels are $|Y(k)| = |H(k)||X(k)|$, with H of order $N = 3$, $N = 7$, and $N = 21$.

suppression capabilities do not focus well for narrowband tones. We seem to find ourselves designing a raw signal that the moving average filter can improve. Later in this chapter, we shall discover better filters and learn how to build them in accord with the requirements of an application.

In fact, the second method, which applies the time-frequency map to the dual-tone signals, offers some help with the high noise problem.

9.1.2.3 Analysis in the Time-Frequency Plane. A time-frequency map is a two-dimensional array of signal frequencies plotted on one axis and their time location plotted on the other axis. This is a useful tool for signal interpretation problems

where the preliminary segmentation is problematic or when there is scant a priori information on frequency ranges and their expected time spans. To decompose a signal into a time-frequency plane representation, we chop up its time domain into intervals of equal size and perform a Fourier analysis on each piece.

Let us explain the new tool using the DTMF application as an example. Application necessity drives much of the design of time-frequency maps. A method appropriate for the DTMF detection problem is as follows.

(i) Select a fixed window width, say $N = 256$. This corresponds to a frequency resolution of 32 Hz at $F_s = 8192$ Hz and a time-domain width of 31 ms.

(ii) This DFT length supports efficient calculation of the transform: the Fast Fourier Transform (FFT) algorithm (Chapter 7). Since we may have quite a few time windows, Candidate segments that are too small can be padded with zeros at the end to make, say, 256 points.

(iii) We can cover longer segments with 256-point windows, overlapping them if necessary.

(iv) We have to run the FFT computation over a whole set of time-domain windows; hence it may be crucial to use a fast transform and limit overlapping.

(v) A genuine DTMF application must account for proper pulse time width (23 ms, minimum, for decoding), frequency (within 3.5% of specification), and energy ratio (DTMF tone energy must exceed that of other frequencies present by 30 dB).

(vi) Once the application checks these details, it can then invoke the logic implied by Table 9.1 and make a final tone decision.

Let us form an array of dual tone energies plotted against time window location (Figure 9.5) and thereby interpret the signal. Sixteen disjoint 256-point windows cover the signal's time domain. Over each window, we compute the FFT. For each of 16 DTMF tones, we calculate the frequency-domain signal energy in the tone frequency range, the energy outside the tone frequency range, and the ratio of the two energies in dB.

Recall that there are two formulas for expressing a gain or ratio R_{dB} between signals Y_1 and Y_2 in decibels (dB). We use either magnitude or power (which is proportional to energy, the magnitude squared):

$$R_{\text{dB}} = 20\log_{10}\left(\frac{M_1}{M_2}\right) = 10\log_{10}\left(\frac{P_1}{P_2}\right), \tag{9.9}$$

where M_i and P_i are the magnitude and power, respectively, of signal Y_i. For each time window we set $y(n) = x(n)$ restricted to the window. Let $Y(k)$ be the 256-point-FFT of $y(n)$ over the window. Then, for each DTMF tone frequency range in Figure 9.5, we take P_1 to be the sum of squared magnitudes of the transform values that lie within the tone frequency range, considering only discrete frequency values $0 \le k < 128$ that lie below the Nyquist value. We set P_2 to be the sum of squared magnitudes that remain; these represent other tones or noise. For example, for DTMF dual tone

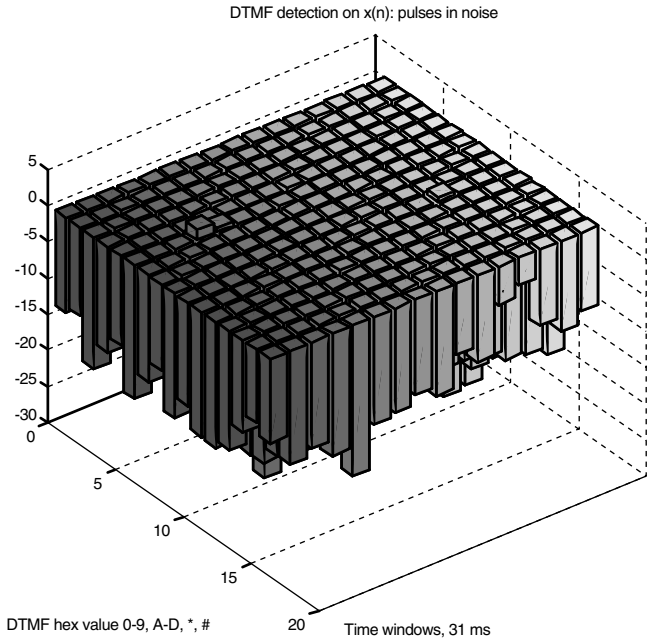

Fig. 9.5. A time-frequency array showing DTMF detection on a noisy $x(n)$. The DTMF "5" and "9" tones appear as tall blocks, representing high ratios of DTMF tone power to overall signal power (dB). Note, however, that the tones are just barely detectable, by a threshold above 0 dB.

"9," the energy lies in $Y(k)$ coefficients $26 \le k \le 27$ (which represent frequencies f (Hz) of $26 \times 32 = 832 \le f \le 864$, for the lower tone) and in $45 \le k \le 47$ (which represent frequencies $1440 \le f \le 1504 = 47 \times 32$) for the upper tone. Thus, we construct a 16×16 array of power ratios, DTMF versus non-DTMF.

Note that the joint frequency and time domain computations involve a tradeoff between frequency resolution and time resolution. When we attempt to refine the time location of a tone segment, we use a lower-order DFT, and the frequency resolution $(NT)^{-1}$ suffers. Since the sampling rate has been fixed at 8 kHz, on the other hand, improving the frequency resolution—that is, making $(NT)^{-1}$ *smaller*—requires a DFT over a larger set of signal samples and thus more imprecision in temporal location.

Let us consider the effect of noise on the time-frequency plane analysis. We have observed that preliminary time domain segmentation works well under benign noise. Heavier noise demands some additional time domain segmentation effort. Noise whose variation approaches the magnitude of the tone oscillations causes problems for segmentation, even though the Fourier analysis can still reveal the underlying tones.

The noisy signal in Figure 9.1 resists segmentation via the Otsu algorithm, for example. Without a time domain segmentation, we can try using small time domain windows and computing an array of coarse frequency resolution magnitude spectra,

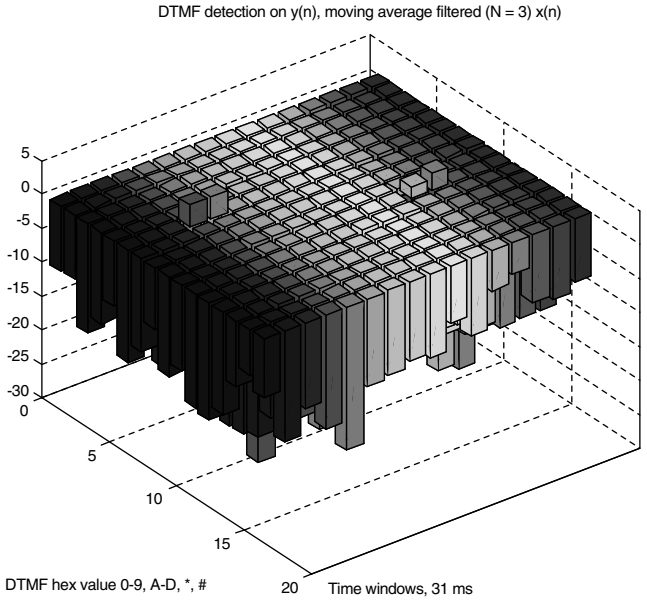

Fig. 9.6. A time-frequency array showing DTMF detection on a very noisy $x(n)$ subject to a moving average noise removal filter. The plot shows the ratio between frequency-domain DTMF power (dB) and non-DTMF power. The time location of the tones is clear, but the frequency discrimination shows little improvement.

such as in Figure 9.5. The problem is that the noisiness of the signal can obscure the tone peaks in the time-frequency array. Let us apply a moving average filter to the signal prior to time-frequency decomposition (Figure 9.6).

The moving average filter's poor performance is not so surprising. We have already empirically shown that its effectiveness is limited by our lack of control over the lobes that appear in its magnitude spectrum (Figure 9.4). We apparently require a filter that passes precisely the range of our DTMF signals, say from 600 to 1700 Hz, and stops the rest. Such a *bandpass filter* cuts down signal components whose frequencies lie outside the DTMF band.

We can construct such a filter H by specifying its frequency domain $H(k)$ as being unity over the discrete DTMF frequencies and zero otherwise. Let $f_{\mathrm{LO}} = 600$ Hz, $f_{\mathrm{HI}} = 1700$ Hz, the sampling rate $F_s = 8192$ Hz, and suppose $N = 256$ is the DFT order. The sampling interval is $T = F_s^{-1}$, so that the frequency resolution is $f_{\mathrm{res}} = 1/(N \times T)$. The Nyquist rate is $f_{\mathrm{max}} = (N/2) \times f_{\mathrm{res}} = F_s/2 = 4096$ Hz. Hence, let us define $k_{\mathrm{LO}} = (N/2) \times (f_{\mathrm{LO}}/f_{\mathrm{max}})$ and $k_{\mathrm{HI}} = (N/2) \times (f_{\mathrm{HI}}/f_{\mathrm{max}})$. An ideal bandpass filter for this Fourier order is given by

$$H(k) = \begin{cases} 1 & \text{if } k_{\mathrm{LO}} \le k \le k_{\mathrm{HI}}, \\ 1 & \text{if } N - k_{\mathrm{HI}} \le k \le N - k_{\mathrm{LO}}, \\ 0 & \text{if otherwise.} \end{cases} \tag{9.10}$$

Then we find

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)e^{\frac{2\pi jkn}{N}}. \tag{9.11}$$

This creates an N-point finite impulse response (FIR) filter (Figure 9.7). Except for the DC term $n = 0$, $h(n)$ is symmetric about $n = 128$.

We can filter the noisy $x(n)$ in either of two ways:

(i) Set up the filter as a difference equation, for example, using the Direct Form II architecture that we will cover later. This method is appropriate for on-line processing of the signal data.

(ii) Translate the filter g(n) = $h(n - 128)$ and perform the convolution $y(n) = (g*x)(n)$. This method is suitable for off-line applications, where all of the signal data is available and the noncausal filter $g(n)$ can be applied to it.

The results of the filtering are shown in the two lower panels of Figure 9.7. Note that the difference equation implementation produces a significant delay in the output

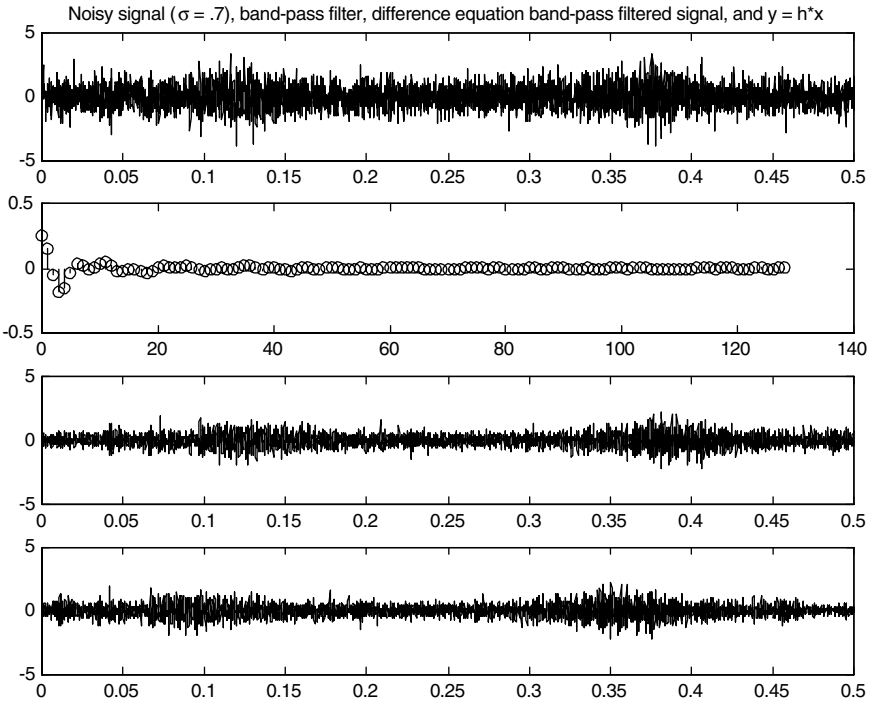

Fig. 9.7. Noisy signal $x(n)$, $\sigma = 0.7$ (top). Bandpass filter $h(n)$ for $0 \le n \le N/2$ (second from top). Alternative filtering results (lower panels).

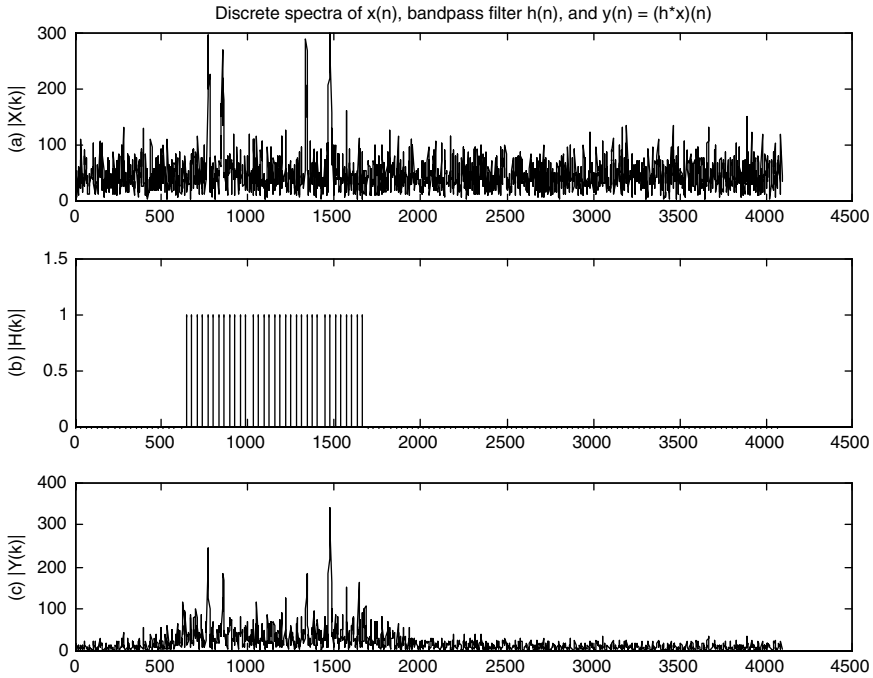

Fig. 9.8. Magnitude spectra from the bandpass filtering operation.

signal. Figure 9.8 shows the frequency-domain magnitude spectra of $X(k)$, $H(k)$, and $Y(k)$, where $y(n) = (Hx)(n)$.

An analysis of the noisy DTMF signal using a time-frequency map is shown in Figure 9.9. Note that the bandpass filter raises the peaks in the time-frequency plane, which potentially helps with detection under severe noise. The drawback is that a few false positive frequency markers appear as well. Why does the bandpass filter not do a clearly superior job compared to the simple moving average filter and analysis without prefiltering? Unfortunately, this bandpass filter is still forgiving to all noise in its pass band—that is, from 600 to 1700 Hz. For example, when the signal of interest is a DTMF "1" dual-tone (697 Hz and 1209 Hz), filtering with the above H will allow noise from 1.25 to 1.7 kHz into the output.

So bandpass filtering is a promising idea, but cleaning all DTMF tones with a single bandpass filter gives only modest results.

9.1.2.4 Filter Bank Decomposition and Analysis. A third approach to dual-tone detection employs a frequency selective filter for each tone in the DTMF ensemble. Constructing, implementing, and applying so many filters seems onerous, but the humble results in the previous two sections encourage alternatives. The appropriate frequency domain tool for this approach is called a *filter bank*. Indeed, this is the conventional approach for DTMF detection, which often calls for online implementation and real-time detection [12].

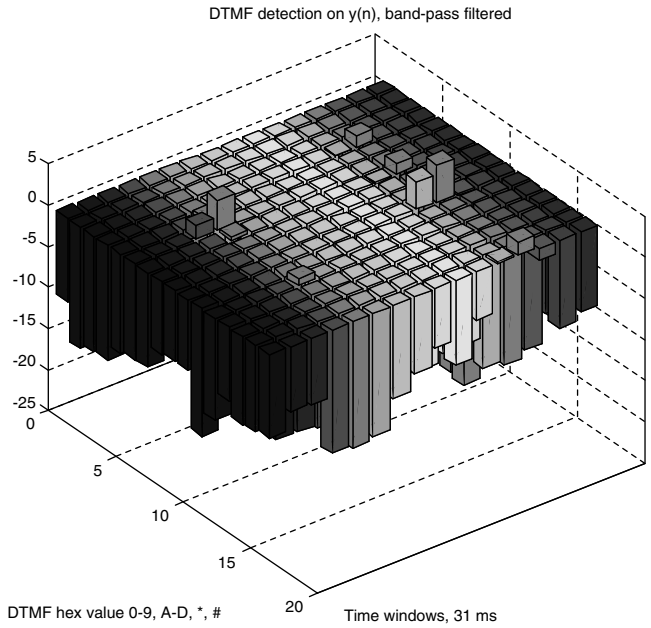

Fig. 9.9. Time-frequency map: A DTMF signal after bandpass filtering.

Filter banks have been carefully studied by signal processing researchers for applications involving compression and efficient signal transmission. More recently they have been subject to intense scrutiny because, when combined with subsampling operations, they are related to the theory of wavelets [23–25]. We will consider these ideas at the end of the chapter and follow up on them in the last two chapters especially. However, for now, our purposes are elementary.

We just want to assemble an array of filters whose parallel output might be read out to interpret a signal containing digital telephony dual-tones. Such simple filter banks are suitable for analysis applications where the input signal frequency ranges are generally known in advance, but the time at which they might occur is not known. If the several filters in the bank are implemented as causal filters, $h(n) = 0$ for $n < 0$, then the filter bank can process data as it arrives in real time.

To build a filter bank for the DTMF application, we set up bandpass filters with unit gain passbands centered about the eight (four low and four high) tones of Table 9.1. Each bandpass filter is designed exactly as in (9.10), except that the frequency range is narrowed to within 3.5% of the tone center frequencies. All filters have the same passband width, and the order of the DFT is $N = 200$ samples (Figure 9.10).

The result of filtering the input signal x(n), which contains DTMF tones "5" and "9" as well as normally distributed noise of zero mean and standard deviation $\sigma = 0.8$, is shown in Figure 9.11.

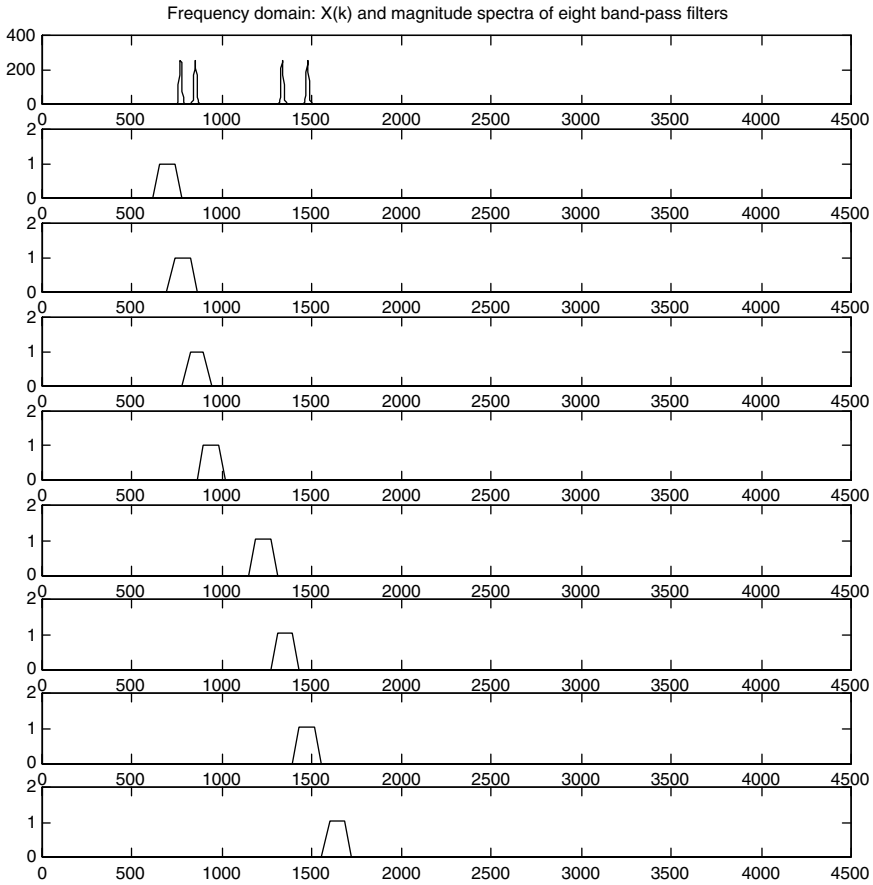

Fig. 9.10. Magnitude spectra of pure DTMF tones "5" and "9" (top) and bank of eight bandpass filters.

To complete the application, one could calculate the energy in a certain window of the last M samples. The dual-tone standard calls for 23 ms for decoding, so at the sampling rate of the example application, $M = 188$. If the energy exceeds a threshold, then the tone is detected. A valid combination of tones, one low tone and one high tone, constitutes a dual-tone detection.

The main problem with the filter bank as we have developed it is the delay imposed by the bandpass filters. The shifting of pulses after filtering must be compensated for in the later analysis stages, if there is a need to know exactly when the tones occurred. For example, do we know that the tones are delayed the same amount? If so, then the detection logic will be correct, albeit a little late, depending on the length of the filters. But if different frequencies are delayed differnet amounts, then we either need to uniformize the delay or compensate for it on a filter-by-filter basis.

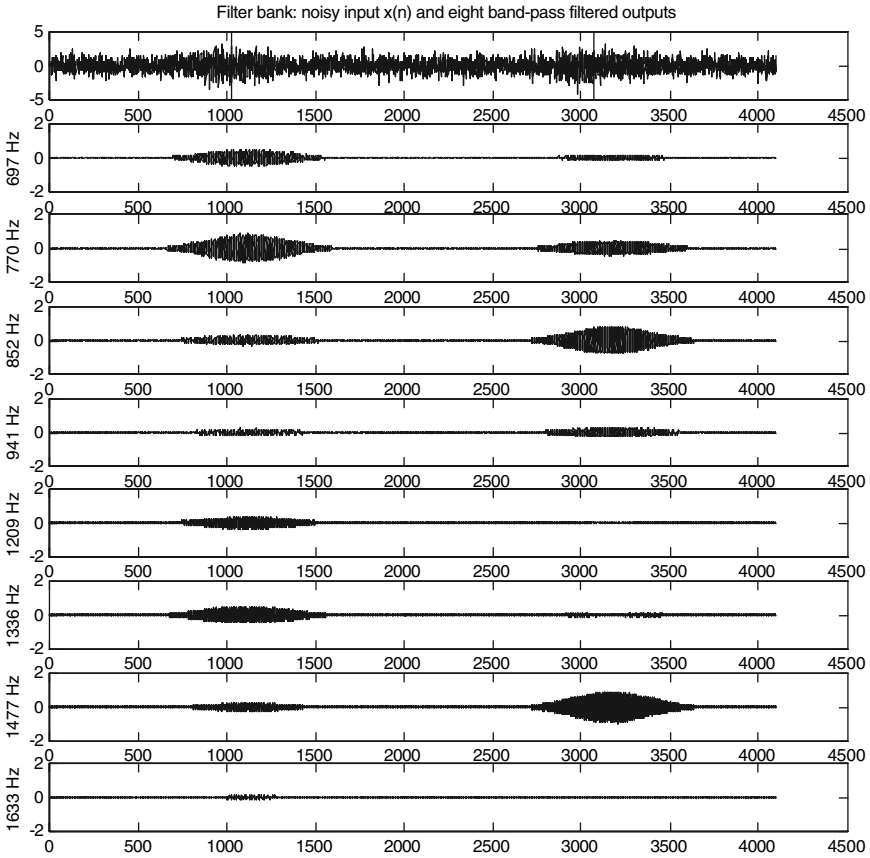

Fig. 9.11. Filter bank output, causally implemented. Note the large delay between the center of the input and output pulses. This is a consequence of the length of the filter, $N = 200$, and the causal implementation.

9.1.3 Filter Frequency Response

When input signals contain oscillatory components that are hidden within noise, the discrete Fourier transform reveals the periodicity as high magnitude spikes in the magnitude spectrum. Even when the signal is so immersed in noise that the time-domain representation refuses to betray the presence of sinusoidal components, this signal transformation is still effective. Though its power is evident for this purpose, the DFT nonetheless loses the time location of oscillations. And for this some time-domain analysis remains. But the noise obscures the time-domain location and extent of the oscillations. This can be a crucial factor in interpreting the signal. By noise removal filtering, however, we can improve visibility into the time-domain and better know the places where the periodicity hides. All of this suggests a theoretical study of the effect of filters on periodic trends in signals.

Consider an exponential signal $x(n)$ input into a linear, translation-invariant system H, producing output $y(n)$: $y = Hx$. If $\delta(n)$ is the discrete impulse and $h = H\delta$ is the *impulse response* of H, then $y(n) = (x * h)(n)$ is the convolution of $x(n)$ and $h(n)$:

$$y(n) = (x*h)(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} h(k)x(n-k). \tag{9.12}$$

Suppose $x(n) = e^{j\omega n}$ is the discrete exponential signal with radial frequency ω radians per sample. Then

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)e^{j\omega(n-k)} = e^{j\omega n}\sum_{k=-\infty}^{\infty} h(k)e^{j\omega k} = e^{j\omega n}H(\omega), \tag{9.13}$$

where $H(\omega)$ is the *frequency response* of $h(n)$. An exponential input to an LTI system produces an exponential output of the same frequency, except amplified (or attenuated) by the factor $H(\omega)$. This basic Chapter 7 result tells us that LTI systems pass exponential signals directly from input to output, multiplied by a complex constant which depends on the signal frequency.

9.1.4 Delay

We have observed empirically that noise removal filtering—and by implication, convolutional filtering in general—imposes a delay on input signals. This section explains the theory of two types of signal delay caused by filtering: *phase delay* and *group delay.*

9.1.4.1 Phase Delay. Suppose $x(n)$ is a discrete signal and $y = Hx$ is a linear, translation-invariant (LTI) discrete system. If $x(n) = \exp(j\omega n)$ is a pure, complex-valued exponential signal, then $y(n) = H(\omega)\exp(j\omega n) = H(\omega)x(n)$, where $H(\omega)$ is the discrete-time Fourier transform (DTFT) of $h(n) = (H\delta)(n)$.

Consider a sinusoidal input signal $x(n) = \cos(n\omega) = [e^{j\omega n} + e^{-j\omega n}]/2$. Then

$$\begin{aligned} y(n) &= \frac{H(\omega)e^{j\omega n}}{2} + \frac{H(-\omega)e^{-j\omega n}}{2} = \frac{H(\omega)e^{j\omega n}}{2} + \frac{\overline{H(\omega)}\,\overline{e^{j\omega n}}}{2} \\ &= 2\mathrm{Real}\left[\frac{H(\omega)e^{j\omega n}}{2}\right] = \mathrm{Real}[H(\omega)e^{j\omega n}]. \end{aligned} \tag{9.14}$$

But

$$\mathrm{Real}[H(\omega)e^{j\omega n}] = |H(\omega)|\cos[\mathrm{Arg}(H(\omega))e^{j\omega n}]. \tag{9.15}$$

If we set $\theta(\omega) = \mathrm{Arg}(H(\omega))$, then

$$y(n) = \mathrm{Real}[H(\omega)e^{j\omega n}] = |H(\omega)|\cos[\omega n + \theta(\omega)] = |H(\omega)|\cos\left[\omega\left(n + \frac{\theta(\omega)}{\omega}\right)\right]. \tag{9.16}$$

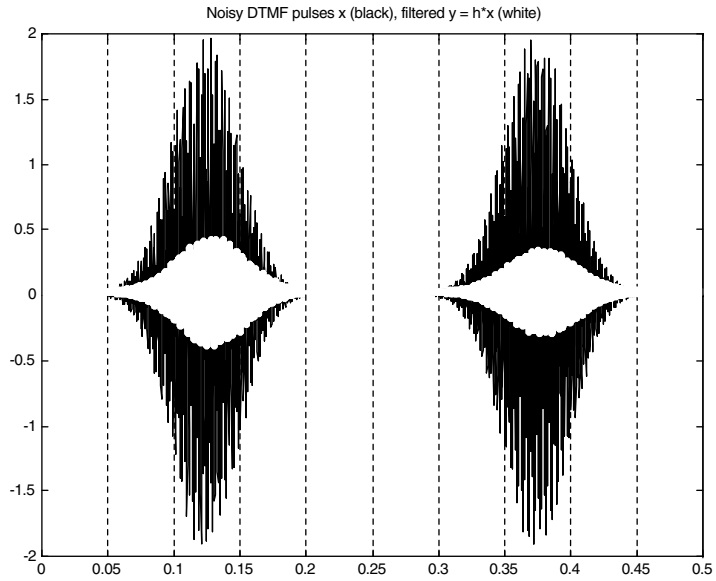

Fig. 9.12. Filter phase delay. Noisy DTMF pulses x (black), filtered $y = h*x$ (white).

So if the input $x(n)$ to H is a sinusoid, then the output $y(n)$ is a sinusoid too. Signals $x(n)$ and $y(n)$ have the same frequency, but $y(n)$ is scaled by $|H(\omega)|$ and phase shifted by $T_H = -\theta(\omega)/\omega$, which is called the *phase delay* of H [26].

If we apply a moving average filter of length $N = 101$ to the noisy DTMF pulses (), then the phase delay imposed by the filter is clearly evident (Figure 9.12).

So sinusoids too, subject to a complex scaling, pass directly through LTI systems. This helps explain the clarity with which the sinusoidal pulses of the DTMF application appear in the frequency domain. Also, we now have a tool, namely the phase delay, $T_H = -\theta(\omega)/\omega$ in (9.16) for comparing the delays induced by various filters.

9.1.4.2 Group Delay. Another type of filter delay occurs when source signals contain sinusoids of nearby frequencies that form an envelope. The superposition of the two sinusoids

$$x(n) = \cos(\omega_1 n) + \cos(\omega_2 n), \tag{9.17}$$

with $\omega_1 \approx \omega_2$, creates a long-term oscillation, called a *beat*. This holds as long as the filter H does not suppress the individual sinusoids; this means that ω_1 and ω_2 are in the *passband* of H.

By trigonometry, we can write $x(n)$ as a product of cosine functions, one of which gives the envelope, of frequency $\omega_1 - \omega_2$, and the other is a sinusoid whose frequency is the mean. Thus,

$$x(n) \;=\; 2\cos\left(n\frac{(\omega_1 - \omega_2)}{2}\right)\cos\left(n\frac{(\omega_1 + \omega_2)}{2}\right), \tag{9.18}$$

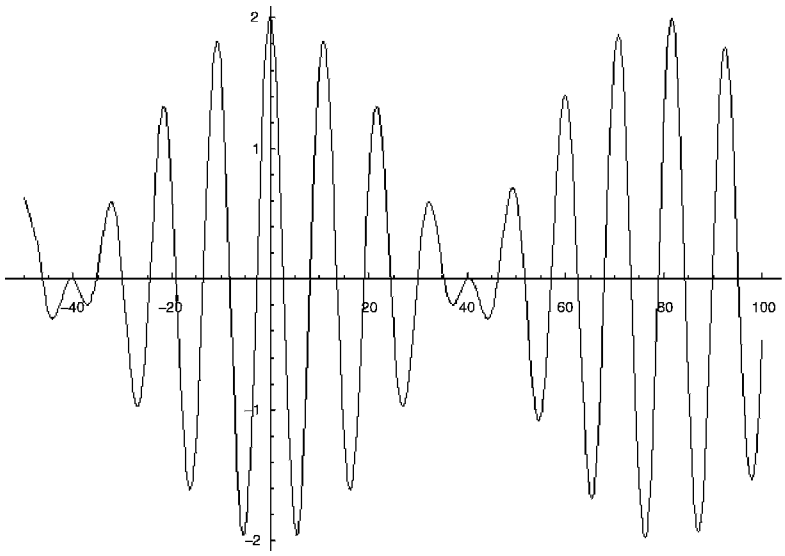

Fig. 9.13. Signal envelope formed by two sinusoids of approximately the same frequency.

which explains the amplitude modulated oscillation of Figure 9.13. Now impose a filter $y(n) = (h^*x)(n)$, with $h = H\delta$. Using (9.16) and (9.17) we have

$$y(n) = \left|H(\omega_1)\right|\cos(\omega_1 n + \theta(\omega_1)) + \left|H(\omega_2)\right|\cos(\omega_2 n + \theta(\omega_2)), \tag{9.19}$$

where $\theta(\omega) = \mathrm{Arg}(H(\omega))$. Assume $\omega_1 \approx \omega_2$ and that these lie in the passband of H, which is to say $|H(\omega_1)| \approx |H(\omega_2)| \neq 0$. Thus,

$$y(n) = \left|H(\omega_1)\right|\{\cos(\omega_1 n + \theta(\omega_1)) + \cos(\omega_2 n + \theta(\omega_2))\}. \tag{9.20}$$

From trigonometry once again,

$$y(n) = 2\left|H(\omega_1)\right|\left\{\cos\frac{\omega_1 n + \theta(\omega_1) - \omega_2 n - \theta(\omega_2)}{2} \cdot \cos\left(\frac{\omega_1 n + \theta(\omega_1) + \omega_2 n + \theta(\omega_2)}{2}\right)\right\}. \tag{9.21}$$

Rearranging the cosine arguments gives

$$y(n) = 2\left|H(\omega_1)\right|\left\{\cos\left\{\frac{\omega_1-\omega_2}{2}\left[n + \frac{\theta(\omega_1) - \theta(\omega_2)}{\omega_1 - \omega_2}\right]\right\} \cdot \cos\left(\frac{\omega_1 + \omega_2}{2}\left[n + \frac{\theta(\omega_1) + \theta(\omega_2)}{\omega_1 + \omega_2}\right]\right)\right\}, \tag{9.22}$$

where the first cosine defines the envelope of $y(n)$. This envelope is delayed by a factor $-\frac{\theta(\omega_1) - \theta(\omega_2)}{\omega_1 - \omega_2}$. As $\omega_1 \to \omega_2$, this delay factor becomes a derivative, which is called the *group delay* of the filter H: $T_G = -d\theta/d\omega$ [11].

9.1.4.3 Implications. Applications that require significant signal filtering must consider the phase and group delay inherent in the system. In our own humble DTMF example above, we noted the phase delay caused by the filter bank. In many scientific and engineering applications, delay considerations affect the actual choice of filters. We shall see later that certain types of finite impulse response (FIR) filters have *linear phase*, so that their group delay is constant. Such filters support signal processing without distortion, an important consideration in communications systems [27].

9.2 FREQUENCY AND PHASE ESTIMATION

The dual-tone multifrequency (DTMF) detection problem in the previous section required Fourier transformation of local signal slices in order to find coded tones. With many slices and many frequency bins, we built time-frequency maps. And thus, we were able to ascertain the presence of signal frequency components over the time span of the signal slice by thresholding for large-magnitude Fourier-domain values. In this section, we study the effectiveness of such techniques. Our methods will be limited and introductory, only a small part of the broad and involved theory of *spectral estimation*. In what appear to be obviously correct and quite straightforward approaches to the problem, we shall see that there are some surprising limitations.

This section introduces an important tool: *window functions*. These are special analog or discrete signals that are used to weight a local signal slice. This technique, called *windowing*, helps to suppress artifacts caused by Fourier transformation on a segment of time-domain signal values. Thus, windowing improves the estimation of local signal frequencies. The signal slice itself is called a *window* or a *region of interest*. Sometimes window functions are loosely called "windows" as well. The ideas are easy, and in context the terms are usually clear. In Chapter 10, we consider analog window functions as an instrument with which to generalize the Fourier transform. Here, we pursue signal analysis applications, computerized implementation, and our emphasis is consequently on discrete windowing.

The DTMF tutorial application did not weight the signal values before performing the discrete Fourier transform (DFT). The results were satisfactory, but we shall see later that applying a window function to the values produces a cleaner, easier to analyze time-frequency map. Moreover, we shall see that the window functions and the windowing method provide a straightforward method for designing discrete finite impulse response (FIR) filters.

In the present context, we can hardly to do justice to the vast research and engineering literature on spectral estimation [28–31].

9.2.1 Windowing

Let us experiment with a simple discrete sinusoid $x(n)$ and the task of computing its discrete Fourier transform (DFT) on a *window*—that is, over a restricted set of values. Three problematic cases emerge:

(i) Alignment of the DFT samples with the signal's spectrally significant portion;
(ii) Signal features that appear to an initial interpretation as frequency characteristics, but in fact arise from wholly different reasons—for example, the presence of an edge;
(iii) Proper sizing of the DFT for the samples.

The first two points affect one another.

9.2.1.1 Alignment and Edges. Let us consider a sinusoidal pulse and its frequency analysis on slices of varying alignments with the pulse event (Figure 9.14).

Windowing involves weighting the samples from the signal slice by window function values before computing the spectrum. We might speculate that Fourier magnitude spectra would be better represented by weighting the central values more

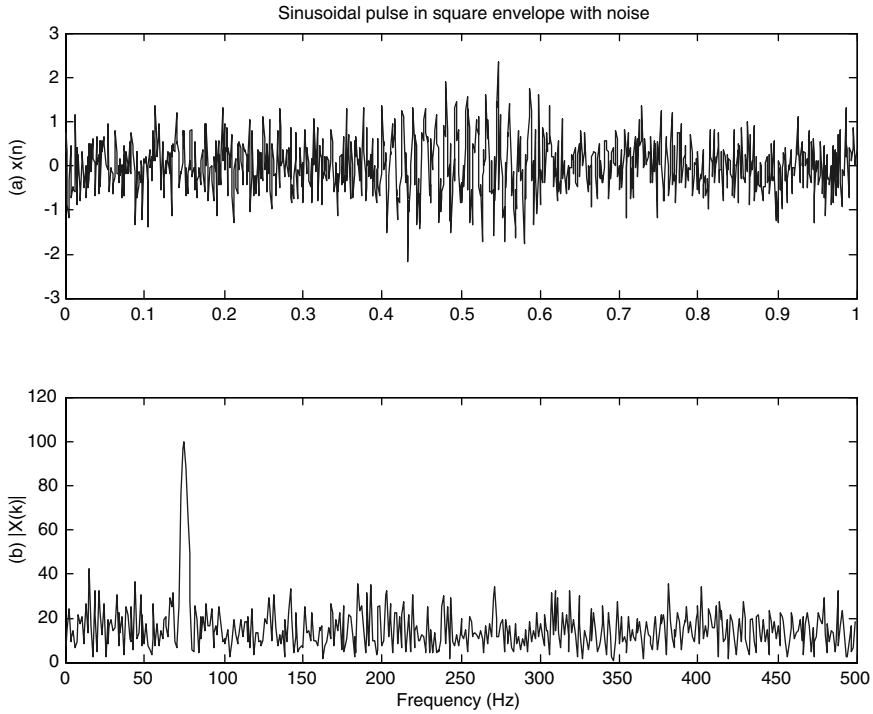

Fig. 9.14. Sinusoidal pulse in square envelope (75 Hz, 200 samples wide, T = .001 s) in moderate noise (top) and its magnitude spectrum (bottom).

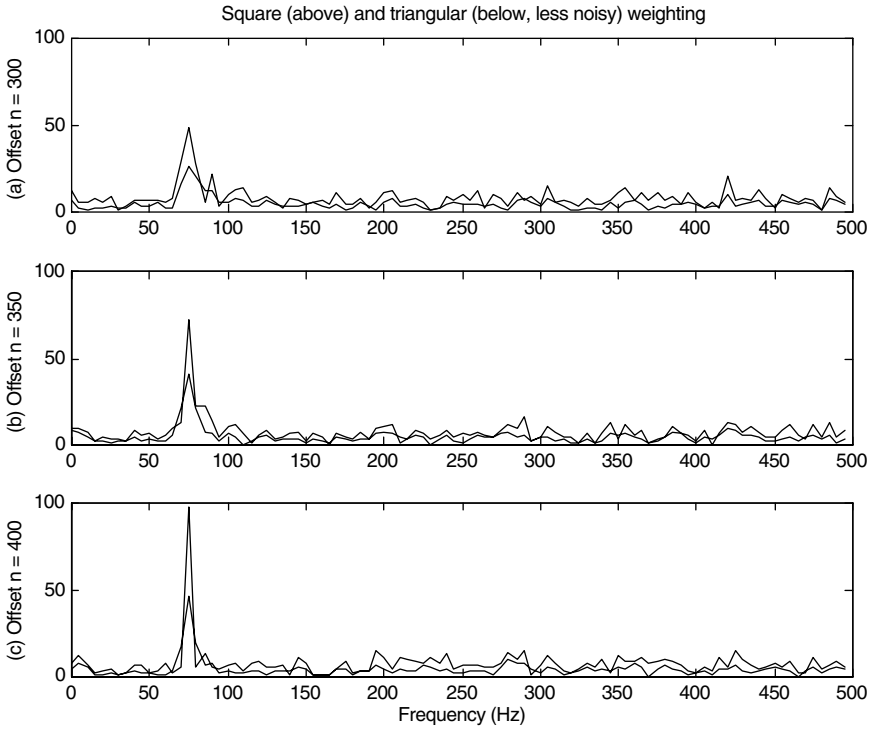

Fig. 9.15. Square pulse magnitude spectra, same DFT order ($N = 200$) at three different offsets: $n = 300$, $n = 350$, and $n = 400$ (full alignment).

than the peripheral values with a time slice from an input signal $x(n)$. The next experiment (Figure 9.15) shows the result of computing magnitude spectra for square and for triangular-weighted window functions.

The main effect of misalignment of the DFT window with the signal oscillations is a blurring of the magnitude spectrum spike. Improving the alignment—clearly—produces a more distinct spike, and invoking a weighting function (a triangular window) in this case offers only modest improvement.

In many applications, preliminary time-domain segmentation helps avoid this problem. Applications can detect signal edges early and use them to align spectral analysis windows. Sometimes edge detection can be based on signal level changes, but in other cases what constitutes an edge is a change in frequency content.

9.2.1.2 Window Size. Now we turn to another anticipated difficulty. Suppose that the signal slice for DFT computation aligns with the oscillation-bearing part of the signal, but the order of the Fourier transformation is a poor choice for the underlying frequency component. We know that a DFT of order N on data $x(n)$ sampled at $F_s = 1/T$ Hz will have frequency resolution $(NT)^{-1}$ and cover discrete frequencies $(NT)^{-1}$, $2(NT)^{-1}$, ..., $(2T)^{-1}$ as long as $x(n)$ is real-valued. Adding a pure sinusoid of

one these frequencies—say $\omega_k = k(NT)^{-1}$, for $1 \le k \le N/2$—to $x(n)$ will alter only $X(k)$ and $X(N-k)$ [32]. Superimposing onto $x(n)$ a sinusoid of frequency $\omega \neq \omega_k$, for any $1 \le k \le N/2$, will perturb all of the $X(k)$. The difference caused by adding the sinusoid diminishes in magnitude like $1/|\omega - \omega_k|$ as $|\omega - \omega_k|$ increases (exercise).

9.2.2 Windowing Methods

Windows are weighting functions that attenuate signals at their discontinuities. When we cut out a piece of a signal and use it to compute its DFT, this effectively periodicizes the signal. The problem is that the signal's early values might differ greatly from the later values in the excised portion. So the effective periodic signal has a huge discontinuity, and this creates large spectral components that are due to the time slicing rather than the trend of the original signal.

The remedy is to suppress the signal slice at its boundaries with a window function. Window functions also serve as a tool for constructing FIR filters. There are a variety of window functions [7–9, 26, 33]:

- The *rectangular* window takes raw signal values without shaping them.
- The *Bartlett*[2] or *triangular* window weights them linearly away from the center.
- The *Hann*[3] window, sometimes called a "Hanning" window, is a modified cosine weighting function.
- The *Hamming*[4] window is also a modified cosine window.
- The *Blackman*[5] window is another modified cosine window.
- The *Kaiser*[6] window uses a Bessel function for shaping the signal slice.

It seems that throughout our experimentation in Section 9.1, we employed the rectangular window. For comparison, Table 9.2 lists the window functions. Note that the window domains $|n| \le \frac{N-1}{2}$ of Table 9.2 are convenient for applications not needing causal filters, such as off-line signal analysis tasks. It is also the form that we will use for the analog windows in the next chapter. Since the windows are zero outside this interval, as linear, translation-invariant system impulse responses, the window functions are all weighted moving average filters. They remove high frequencies and preserve low frequencies when convolved with other discrete signals.

[2]After M. S. Bartlett, who used this window to estimate spectra as early as 1950.

[3]Austrian meteorologist Julius von Hann introduced this window. At some point, perhaps due to confusion with Hamming's similar window or to the use of the term "hann" for the cosine windowing technique in general (as in Ref. 32), the name "Hanning" seems to have stuck.

[4]Richard.W. Hamming (1915–1998) used this window for improved signal spectral analysis, but the American mathematician is more widely known for having invented error correcting codes (*The Bell System Technical Journal*, April 1950).

[5]After Hamming's collaborator at Bell Telephone Laboratories, Ralph B. Blackman (1904–).

[6]Introduced by J. F. Kaiser of Bell Laboratories in 1974.

TABLE 9.2. Window Functions for $N > 0$ Samples[a]

Name	Definition
Rectangular	$w(n) = \begin{cases} 1 & \text{if } \lvert n\rvert \le \frac{N-1}{2} \\ 0 & \text{otherwise} \end{cases}$
Bartlett (triangular)	$w(n) = 1 - \frac{2\lvert n\rvert}{N-1}$ if $\lvert n\rvert \le \frac{N-1}{2}$
Hann	$w(n) = \frac{1}{2}\left[1 - \cos\frac{2\pi n}{N-1}\right]$ if $\lvert n\rvert \le \frac{N-1}{2}$
Hamming	$w(n) = 0.54 - 0.46\cos\frac{2\pi n}{N-1}$ if $\lvert n\rvert \le \frac{N-1}{2}$
Blackman	$w(n) = 0.42 + 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1}$ if $\lvert n\rvert \le \frac{N-1}{2}$
Kaiser	$w(n) = \frac{I_0\left(\alpha\sqrt{1-\left(\frac{2n}{N-1}\right)^2}\right)}{I_0(\alpha)}$ if $\lvert n\rvert \le \frac{N-1}{2}$

[a]The table defines the windows as centered about $n = 0$. Outside the specified ranges, the windows are zero. It is straightforward to shift them so that they are causal [7]. The Kaiser window is defined in terms of the zeroth-order Bessel[7] function of the first kind (9.23) and a parameter α given below (9.24).

The summation

$$I_0(t) = 1 + \sum_{n=1}^{\infty}\left[\frac{1}{n!}\left(\frac{t}{2}\right)^n\right]^2 \tag{9.23}$$

defines the Bessel function. There is a standard formula [33] for the Kaiser window parameter α. To ensure a Kaiser window whose Fourier transform suppresses high-frequency components to more than $-\Delta$ dB, set

$$\alpha = \begin{cases} 0.1102(\Delta - 8.7) & \text{if } \Delta > 50, \\ 0.5842(\Delta - 21)^{0.4} + 0.07886(\Delta - 21) & \text{if } 50 \ge \Delta \ge 21, \\ 0 & \text{if } 21 > \Delta. \end{cases} \tag{9.24}$$

[7]Professor of astronomy, mathematician, and lifelong director of the Königsberg Observatory, Friedrich Wilhelm Bessel (1784–1846) devised the functions bearing his name for analyzing the motions of three bodies under mutual gravitation.

9.2.3 Power Spectrum Estimation

The Fourier transform magnitude spectrum has some important drawbacks. Our introductory digital telephony control tones application showed that—under moderate noise—measuring the contributions of frequency components by the relative magnitude of Fourier transform coefficients is effective.

To understand how this comes about, we have to consider signal noise in a mathematically tractable form.

9.2.3.1 Power Spectral Density. Let $\mathbf{x}$ be a discrete random signal; that is, $\mathbf{x} = \{x_n : n \in \mathbb{Z}\}$ is a family or *ensemble* of *random variables* (Chapter 1). This is an abstract formulation. What it means is that if a signal $x(n)$ has a random nature, then we do not know exactly what value it may take at any particular time instant $n \in \mathbb{Z}$. But we at least know that the values $x(n)$ might assume at $n = k$, for example, are given by a random variable, namely $x_k \in \mathbf{x}$. So by a random signal, we understand a signal that is random at all of its measured time instants; it is indeed an ensemble of random variables.

But that is not to say that we know nothing about the random signal $\mathbf{x}$. Associated with each random variable $r = x_n \in \mathbf{x}$ for some $n \in \mathbb{Z}$ is a *probability distribution function* F_r and a *probability density function* f_r such that $F_r(s) = P(r \leq s)$, the probability that r does not exceed $s \in \mathbb{R}$. Moreover,

$$F_r(s) = \int_{-\infty}^{s} f_r(t)\, dt, \tag{9.25}$$

which is to say that $\frac{\partial}{\partial s} F_r(s) = f_r(s)$. To the skeptically inclined individual, these are almost incredible conditions, but they do approximate naturally occurring random signals fairly well. In any case, we need them for the theoretical development.

The distribution and density functions allow us to describe random variables with averages. If $r = x_n \in \mathbf{x}$ again, then we define its *mean*

$$\mu_r = \int_{-\infty}^{\infty} t f_r(t)\, dt = E[r] \tag{9.26}$$

and *standard deviation* σ_r, the square root of the *variance*: $\sigma_r^2 = E[r^2] - \mu_r^2$.

Generalizing for two random variables, $u = x_n$ and $v = x_m$ in $\mathbf{x}$, we assume a *joint distribution function* $F_{u,v}(s, t) = P(u \leq s \text{ and } v \leq t)$ and *joint density function* $\frac{\partial^2}{\partial s \partial t} F_{u,v}(s, t) = f_{u,v}(s, t)$. If $E[uv] = E[u]E[v]$, then random variables u and v are *uncorrelated* or *linearly independent*.

Power spectrum estimation studies how signal power distributes among frequencies. For finite-energy deterministic signals $x(t)$, the power of $x(t)$ in the (unsigned) band $0 < \Omega_0 < \Omega_1 < \infty$ comes from integrating $|X(\Omega)|^2$ over $[-\Omega_1, -\Omega_0] \cup [\Omega_0, \Omega_1]$.

But for random signals, the mathematical analysis depends on a special class of signals $x(t)$ that obey the following two conditions:

(i) $E[x(t)]$ does not depend on the process variable $t \in \mathbb{R}$.
(ii) $E[x(t)x(t+\tau)]$ is a function of τ and does not depend on $t \in \mathbb{R}$.

Such signals are called *wide-sense stationary* (WSS) [34]. We define the *autocorrelation* for a WSS random signal $x(t)$ to be $r_{xx}(\tau) = E[x(t)x(t+\tau)]$. It is easy to show that $E[x(t)x(s)] = r_{xx}(t-s)$ and that $r_{xx}(\tau)$ is an even signal. A special type of WSS random signal $x(t)$ has an autocorrelation function that is an impulse: $r_{xx}(\tau) = A\delta(t)$ for some constant $A \in \mathbb{R}$. This means that signal values are completely uncorrelated with their neighbors. Such random signals are called *white noise* processes; we shall explain this colorful terminology in a moment.

In order to study the spectrum of a noisy signal, we have to limit its time-domain extent. So for $L > 0$ let us define the *localization* of random signal $x(t)$ to $[-L, L]$:

$$x_L(t) = \begin{cases} x(t) & \text{if } -L \le t \le L, \\ 0 & \text{if otherwise.} \end{cases} \tag{9.27}$$

so that

$$X_L(\Omega) = \int_{-\infty}^{\infty} x_L(t) e^{-j\Omega t} dt = \int_{-L}^{L} x(t) e^{-j\Omega t} dt. \tag{9.28}$$

The energy of x_L is $\|x_L\|_2^2 = \frac{1}{2\pi}\|X_L\|_2^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X_L(\Omega)|^2 d\Omega$ by Parseval's identity.

The approximate energy of x_L in a narrow signed frequency band, $\Delta(\Omega) = \Omega_1 - \Omega_0$, is thus $|X_L(\Omega)|^2\Delta(\Omega)$. Since frequency is the reciprocal of time, $|X_L(\Omega)|^2/(2L)$ has units of energy, which is the product of power and time, or power divided by frequency. Therefore, we may define the *power spectral density* (PSD) for $x_L(t)$ to be $\frac{|X_L(\Omega)|^2}{2L}$. This is a random variable, and its expectation is $E\left[\frac{|X_L(\Omega)|^2}{2L}\right]$. It is tempting to define the PSD of $x(t)$ as the large time window $[-L, L]$ limit of such expectations:

$$X_{\text{PSD}}(\Omega) = \lim_{L \to \infty} E\left[\frac{|X_L(\Omega)|^2}{2L}\right]. \tag{9.29}$$

But some caution is in order. We need to know that the limit (9.29) exists. A famous result shows that the desired limit operation is valid and moreover provides a way to

compute it. The Wiener[8]–Khinchin[9] theorem, says that if $x(t)$ is a real-valued, WSS random signal with autocorrelation $r_{xx}(t) \in L^1(\mathbb{R})$, then

$$X_{\text{PSD}}(\Omega) = R_{xx}(\Omega) = \int_{-\infty}^{\infty} r_{xx}(t) e^{-j\Omega t} dt \,. \tag{9.30}$$

While the exact values of a random signal are not known, it is a reasonable assumption that the autocorrelation of the signal is available. Indeed, the autocorrelation will tend to resemble a narrow pulse when local signal values $x(\mathrm{t} + \tau)$ correlate poorly with a particular $x(t)$, and it will look like a broad pulse when $x(\mathrm{t} + \tau)$ as a trend repeats $x(t)$. In any case, for τ large, $r_{xx}(\tau)$ diminishes, and we can often assume a mathematically tractable model for the autocorrelation. For example, from basic physical considerations, we can derive a model for the thermal noise across a resistor in an electric circuit. A purely theoretical example is the aforementioned white noise process. The Wiener–Khinchin theorem implies that the white noise process $r_{xx}(\tau) = A\delta(t)$ has $X_{\text{PSD}}(\Omega) = A$, for $A \in \mathbb{R}$. Thus, its frequency spectrum is flat; it contains all "colors," as it were, and is therefore "white." It turns out that white noise models the thermal noise across a resistor and that its autocorrelation scales according to the absolute temperature of the circuit elements.

Similar ideas work for discrete random signals. If $x(n) \in l^2$ is a discrete deterministic signal with DTFT $X(\omega)$, then integrating $|X(\omega)|^2$ over $[-\omega_1, -\omega_0] \cup [\omega_0, \omega_1]$ gives the power in the band $0 < \omega_0 < \omega_1 < \pi$. A *wide-sense stationary* (*WSS*) *discrete* random signal satisfies the following:

(i) $E[x(n)]$ does not depend on the process variable $n \in \mathbb{Z}$.

(ii) $E[x(n)x(n + \nu)]$ is a function of ν and does not depend on $n \in \mathbb{Z}$.

The autocorrelation for a WSS random signal $x(n)$ is $r_{xx}(\nu) = E[x(n)x(n + \nu)]$. Again, $E[x(n)x(m)] = r_{xx}(n - m)$ and $r_{xx}(\nu)$ is symmetric about $\nu = 0$. Toward analyzing the power spectrum, for $L > 0$ we define

$$x_L(n) = \begin{cases} x(n) & \text{if } -L \le n \le L, \\ 0 & \text{if otherwise.} \end{cases} \tag{9.31}$$

[8]First-generation American mathematician Norbert Wiener (1894–1964) finished the doctoral program at Harvard at age 18, concentrating on philosophy of mathematics and logic. In the tradition of Plato, the great English scholar Bertrand Russell hoped to improve Wiener's philosophical insights by having him study more mathematics. But later encounters with G. H. Hardy, D. Hilbert, and E. G. H. Landau nudged the prodigy toward mathematical analysis. After some peregrination, Wiener took a ground-floor job as a mathematics instructor at the Massachussetts Institute of Technology. He eventually arose to full Professor, contributed substantially to statistical communication and control theory, and remained at MIT for the rest of his career.

[9]Soviet mathematician Aleksandr Yakovlevich Khinchin (1894–1959) established much of the early theory of stationary stochastic processes. The author of some 150 papers, he took a mathematics professorship at Moscow State University in 1927. He was a patron of the arts and theater. Election to the Soviet Academy of Sciences (1939) recognized Khinchin's contributions to ranging from probability, number theory, information theory, statistical physics, and quantum mechanics.

The Fourier spectrum of the localized random signal is

$$X_L(\omega) = \sum_{n=-L}^{L} x(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_L(n)e^{-j\omega n}. \tag{9.32}$$

We define the PSD for $x_L(n)$ to be

$$X_{\text{PSD}}(\omega) = \lim_{L\to\infty} E\left[\frac{\left|X_L(\omega)\right|^2}{2L+1}\right]. \tag{9.33}$$

There is a discrete version of the Wiener–Khinchin theorem. If $x(n)$ is a real-valued, WSS random signal with an absolutely summable autocorrelation function $r_{xx}(n)$, then

$$X_{\text{PSD}}(\omega) = R_{xx}(\omega) = \sum_{n=-\infty}^{\infty} r_{xx}(n)e^{-j\omega n}. \tag{9.34}$$

Thus, for both analog and discrete random variables we are justified in defining the power spectral density, and it can be computed as long as the associated autocorrelation function is respectively L^1 or l^1. The exercises outline the proofs of both the analog and discrete Wiener–Khinchin theorems.

9.2.3.2 Periodogram. Now we consider approximating the power spectral density. The oldest and most straightforward approach is to compute the discrete time Fourier transform on a local time window $[-L, L]$ of sampled data points $x(n)$. Thus, we have

$$\tilde{X}_{L,\,\text{PSD}}(\omega) = \frac{1}{2L+1}\left|X_L(\omega)\right|^2 = \frac{1}{2L+1}\left|\sum_{n=-L}^{L} x(n)e^{-j\omega n}\right|^2. \tag{9.35}$$

Generally, we would take $\omega = 2\pi k/T$ for $-L \le k \le L$ and compute (9.35) on a discrete set of frequencies. After all, although we used the discrete Fourier transform magnitude spectrum in the application examples of Section 9.1, we could have equally well used the squared magnitude spectrum. Also, due to the periodicity of the discrete Fourier transforms, we could equally well shift the local window of $x(n)$ values. In practice, a window of width $2M$ is chosen to enable a fast Fourier transform computation. In any event, (9.35) is a statistical estimator for the random variable $X_{\text{PSD}}(\omega)$. The question before us is how well—for a particular frequency of interest, $-\pi < \omega \le \pi$—the estimate of $|X(\omega)|^2$ over $N = 2L + 1$ samples of noisy $x(n)$ compares to the actual power at that frequency.

Briefly, the problem with the estimated power spectrum $\tilde{X}_{\text{PSD}}(\omega)$ is twofold:

(i) As the number of samples is increased, the mean of the estimate does not approach the actual mean; it is a *biased estimator.*

(ii) As the number of samples is increased, the variance of the estimate does not approach zero; it is an *inconsistent estimator.*

Signal processing [7] and spectrum estimation [29–31] texts explain this theory. Unfortunately, the development would drag us away from our signal analysis focus. We would instead like to emphasize that the problems of the periodogram as an estimator of the PSD can be addressed by applying the window functions we developed earlier in this section along with some straightforward averaging techniques.

9.2.3.3 Periodogram Improvement. Fortunately, there are some easy ways to improve the periodogram estimate $\tilde{X}_{L,\,\mathrm{PSD}}(\omega)$ of (9.35). We cover some classic methods that use no model of the signal, its spectrum, or its autocorrelation. These *nonparametric* techniques include:

- Bartlett's method smoothes the time-domain data by breaking the interval into smaller, equally sized segments and averaging the periodograms computed for each segment [35].
- Welch's algorithm smoothes the time-domain data by breaking the interval into smaller, equally sized segments, applying a window function to each segment, and allowing the windows to overlap [36].
- Another technique, due to Blackman and Tukey [37], relies directly on the Wiener–Khinchin theorem's identification of the PSD with the Fourier transform of the autocorrelation function.

Bartlett's method divides a set of $N = K \times M$ data points of $x(n)$ into K subwindows of length M. Thus, the signal values on subwindow k are $x_k(m) = x(kM + m)$, where $0 \le k \le K - 1$ and $0 \le m \le M - 1$. For each such subwindow we set

$$\tilde{X}_{k,\,\mathrm{PSD}}(\omega) = \frac{1}{M}\left|\sum_{m=0}^{M-1} x_k(m)e^{-j\omega m}\right|^2, \tag{9.36a}$$

and then average them all to get the estimate over $[0, N - 1]$:

$$\tilde{X}_{N,\,\mathrm{PSD}}(\omega) = \frac{1}{K}\sum_{k=0}^{K-1} \tilde{X}_{k,\,\mathrm{PSD}}(\omega), \tag{9.36b}$$

The Welch algorithm improves upon the Bartlett method by

(i) Allowing the subwindows to overlap.
(ii) Applying a window function to the individual PSD estimates on the subwindows. The window function can be any of those described in Table 9.2.

The steps in the Blackman–Tukey algorithm are as follows:

(i) From a body of measured noisy signal data, the autocorrelation function for the random process is estimated.
(ii) One of the typical window functions—for example, the Hann window—is applied to the autocorrelation estimate.
(iii) The discrete Fourier transform is applied to windowed autocorrelation values.

Another class of periodogram improvement algorithms—called *parametric* methods—make a model for the noisy signal data. The idea is to assume that the signal arises from a linear system excited by white noise. The exercises cover the concept of noisy inputs to linear systems.

9.2.4 Application: Interferometry

An application that involves the precise estimation of signal frequency and phase is interferometry, which is based on the wave nature of electromagnetic radiation [38]. In interferometry, an input optical signal contains light combined from two different sources—for example, reflected from two different surfaces. If the original source of both reflecting beams is coherent (that is, the light waves are in phase with one another, such as from a laser), then the resulting interferogram will contain peaks and valleys of intensity, depending on the path distance of the component light waves. Of course, moving one reflecting surface by a wavelength amount produces the same light combination, and so the intensity only indicates relative changes in position between the two reflecting surfaces.

The technique enables us to measure minute differences in distance. Peaks in the interferogram correspond to when the peak of one sinusoidal wave matches up with the peak of the other. This is the length of the wave; and in the case of light, this value is quite small, from about 400 nm (violet) to 700 nm (red). Thus, optical interferometry is used in precision measurement and manufacture, such as semiconductor integrated circuit fabrication.

We consider a semiconductor manufacturing and control application of interferometry involving chemical mechanical planarization (CMP) of silicon wafers [39]. CMP has become an important process for ensuring the planarity of the wafer surface. A high degree of flatness eliminates flaws in later deposition steps. More importantly for modern integrated circuit manufacture, CMP is used for selectively removing thin layers of material on wafers that do not etch well in plasmas, such as copper.

Evident in the signal trace (Figure 9.16) at the top are:

(i) The long-term undulations in reflected intensity due to the combination of beams reflected from the surface and Si/SiO_2 interface;
(ii) Short-term vibrations around $f = 0.1$ Hz;
(iii) At the end of the trace, a change in process conditions causing wild instability of the sensor's measured reflectance.

Our interferometric interest is to isolate these three trends. The wavelength of the long-term oscillation, on the order of 0.0085 Hz, will be used to estimate the silicon oxide removal rate (Figure 9.17). The short-term 0.1 Hz oscillation can be removed with a notch filter in order to enhance the estimation of the removal rate. Also, with the short-term oscillation removed by a notch filter, it becomes possible to design a simple algorithm to compute the phase of the long-term reflectance oscillation and use this

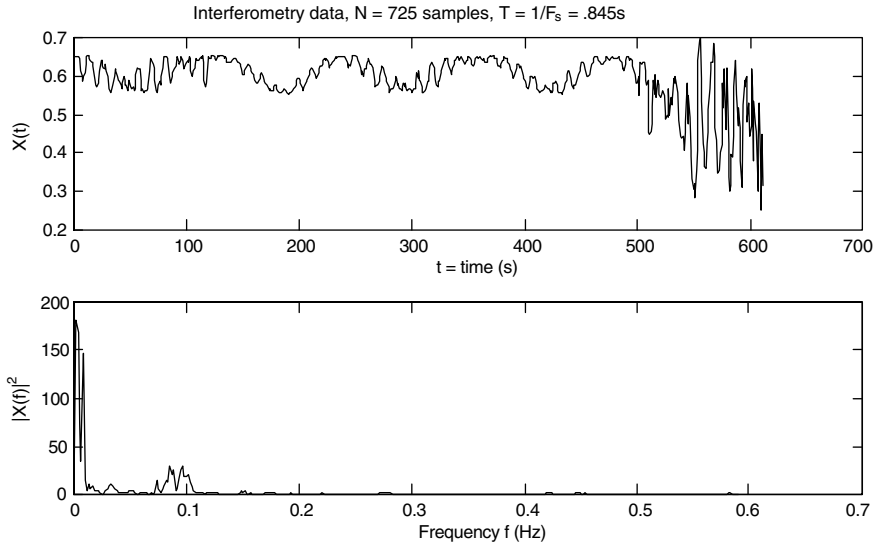

Fig. 9.16. Reflectance data from a CMP removal process on a silicon wafer with a surface silicon dioxide film. The upper panel contains the signal trace and the lower panel contains the magnitude spectrum.

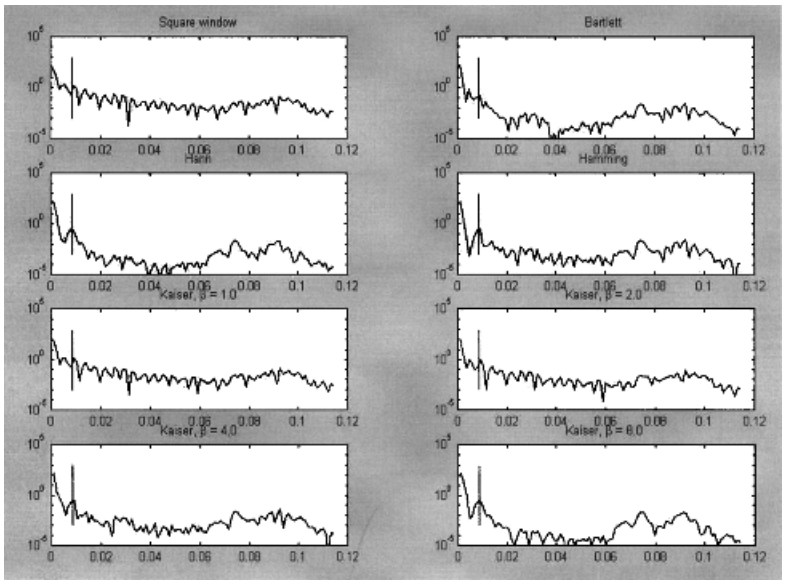

Fig. 9.17. Periodograms of several windows applied to the interferometic data from CMP processing of a silicon wafer. Top four periodograms: Square, Bartlett, Hann, and Hamming windows applied to signal before periodogram calculation. The bottom four plots show the efficacy of various Kaiser window parameters. The vertical line marks the frequency of the interference fringes.

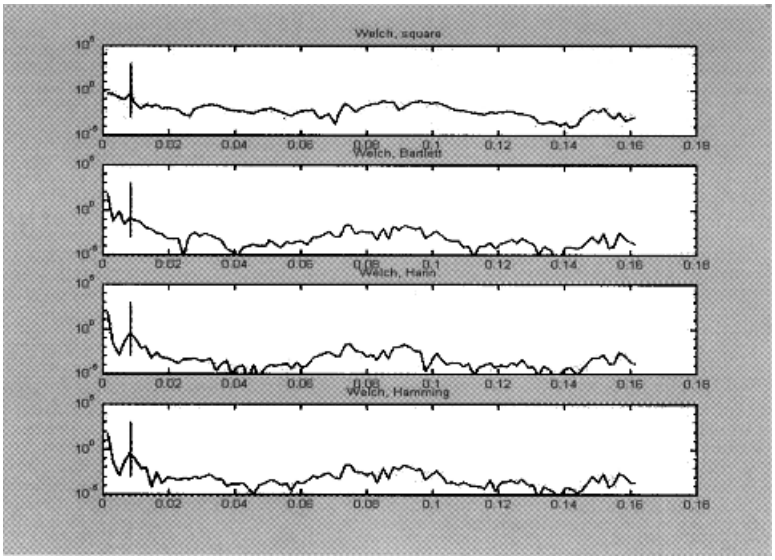

Fig. 9.18. Welch's method for periodogram computation.

as a measurement of the amount of top-level film removed. Finally, we are interested in detecting the signal instability at the end of the trace, which is an indication that the oxide has begun to disappear from the wafer, exposing the silicon substrate. This represents polish endpoint.

Let us also investigate one of the spectrogram improvement methods—in this case Welch's method with windows overlapped 50% and a variety of window functions applied before computing the local periodograms. The results are shown in Figure 9.18.

As a practical matter, some goal-directed information needs to be invoked in this example. A peak detector needs to be provided with a limited range in hertz for its search. Once the periodicity of the interferogram fringes is determined from the periodogram algorithm, it is possible to calculate the phase of individual points along the fringe trace. To do this, we would perform a discrete Fourier transform of the proper window size according to the wavelength of the long-term undulations. For the above sampling interval $T = 0.8451$ s, this would be $N = 140$ samples. The phase of points along the interference fringe trace could be computed by the complex argument of the first DFT coefficient, for example.

9.3 DISCRETE FILTER DESIGN AND IMPLEMENTATION

Discrete filters are suitable for computer implementation, and signal analysis applications depend on them directly. This section reviews some elements of filter theory, the z-transform, and ideal filters. It also covers the more practical aspects such

as (a) filter approximation and (b) the steps in designing a discrete filter, and it explains basic methods on how to implement discrete filters.

The principal tools for filter design are the discrete Fourier transform (DFT), the discrete-time Fourier transform (DTFT), and the z-transform. One important design method is to derive a discrete filter from an analog filter. So we shall also use the continuous-domain Fourier transform as well as introduce briefly the Laplace transform. The Fourier transform properties allow us to convert one type of filter into another, considerably simplifying the mechanics of filter construction. For example, we generally design a low-pass filter and then convert it into the required bandpass filter.

9.3.1 Ideal Filters

Some applications, such as in the dual-tone multifrequency application above, require fine separation in the frequency domain. Some periodicities we need to pass through for further analysis, such as the DTMF, range from 697 to 1633 Hz. Others, such as high-frequency background noise and low-frequency interference, we prefer to suppress. Offline applications can Fourier transform large time-slices of data and select spectral components according to the frequency resolution of the transform. Online applications, though, must achieve frequency selection as data enters the system, in real time, and then pass the output to interpretation algorithms. Here the filters have to be causal and efficiently implemented. Since no high-resolution Fourier transform of the input data is possible in this situation, it becomes all the more important to design filters that distinguish between nearby frequencies.

9.3.1.1 Low Pass. We can eliminate all frequencies above the range of interest to an application by processing input signals through an ideal low-pass filter H. It is easy to describe such a filter using the discrete-time Fourier transform $H(\omega)$:

$$H(\omega) = \begin{cases} 1 & \text{if } |\omega| \le \omega_c, \\ 0 & \text{if otherwise,} \end{cases} \tag{9.37}$$

where ω_c is the *cutoff frequency* of the filter. This filter's perfect cutoff is ideal for separating one frequency from another. The only caveat is that a sufficiently high sampling rate must be chosen so that ω_c is close to the frequency of a discrete component.

The inverse DTFT serves as a tool for building discrete filters from such frequency-domain descriptions. The impulse response of the ideal low-pass system (Figure 9.19) is

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega = \frac{\sin(\omega_c n)}{\pi n}. \tag{9.38}$$

Using the inverse DTFT to generate a filter impulse response from description of its ideal frequency-domain representation is elegant and straightforward, but the resulting filters are often—and most particularly in the present case—quite impractical.

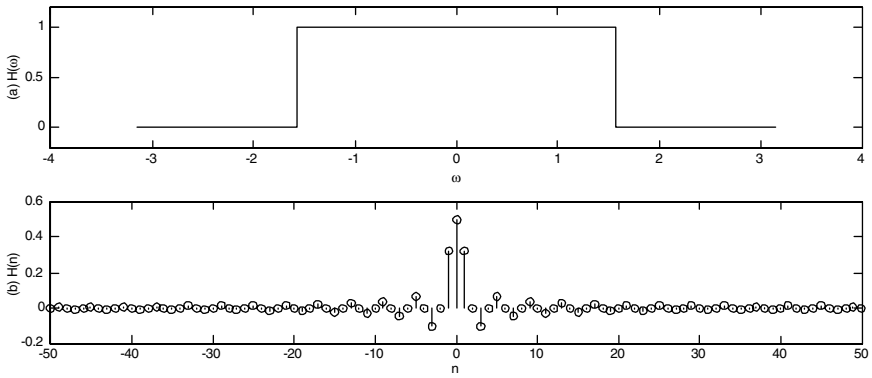

Fig. 9.19. Ideal low-pass filter in the frequency-domain (a) and truncation of impulse response in the time-domain (b).

The problems with $h(n)$ as given by (9.38) are as follows:

(i) It is an infinite impulse response (IIR) filter.
(ii) This means that it cannot be implemented by a straightforward convolution in applications.
(iii) Some IIR filters permit a recursive computation for their implementation, but in this case $h(n) \neq 0$ for arbitrarily large magnitude $n < 0$, so it is in fact unrealizable for applications.
(iv) The filter is not causal, so its realization requires future signal values and cannot work on real-time data streams.
(v) The filter is not stable, since its impulse response is not absolutely summable (Chapter 2).

But these problems are not catastrophic. We can truncate $h(n)$ so that it is supported within some reasonable interval $[-N, N]$, say. This is often satisfactory for applications where the raw signals contain high-frequency background noise. The perfect frequency-domain characteristic is lost (Figure 9.19b). We will see how to overcome these problems in a moment.

Next, however, let us look at ideal filters of other types and see how to build them out of low-pass filters.

9.3.1.2 Other Filters. Applications require filters with varying frequency-domain characteristics. For example, the DTMF application of Section 9.1 uses a low-pass filter, a simple moving averager, to suppress background noise. In the filter output, this highlights the relatively pure dual tones and improves thresholding results. But an online version of the same application would need filters that select frequency ranges and reject oscillations below and above certain bands. Here we are speaking of a bandpass filter, and we have noted that an array of them in parallel operation can pass filtered signals to the application logic for tone classification.

The mechanics of converting a lowpass filter into one of an alternative frequency-domain behavior are fortunately quite straightforward. We require only a few DTFT properties (Chapter 7).

From a low-pass filter $h(n) = (H\delta)(n)$, it is easy to make a high-pass filter $f(n) = (F\delta)(n)$. An *all-pass* filter is the discrete impulse $\delta(n)$ or one of its shifted versions $\delta(n - n_0)$. We subtract $f(n) = \delta(n) - h(n)$ for a high-pass filter $f(n) = (G\delta)(n)$.

Two elementary methods exist for building a bandpass filter F. One way to do this, starting from a low-pass filter H with cutoff ω_c, is to use the frequency shifting property of the DTFT. If we set $f(n) = e^{j\omega_L n}h(n)$, then $F(\omega) = H(\omega - \omega_L)$. F is thus a bandpass filter with lower cutoff ω_L and upper cutoff $\omega_H = \omega_L + \omega_c$. The blemish on this otherwise elegant modification of $h(n)$ is the fact that $f(n)$ becomes a complex-valued discrete signal. A second basic technique is to combine a low-pass filter H, with cutoff ω_c, and a high-pass filter G, with cutoffs ω_L and ω_H. Typically, we get G itself by subtracting $h(n)$ from an all-pass system's impulse response. If our filters satisfy $\omega_c > \omega_L$, then the system composition $F(n) = G(H(n))$ will pass precisely the frequencies that lie in the passband overlap. We know that $f = F\delta = g^*h$, the convolution of $g(n)$ and $h(n)$. So, if both $g(n)$ and $h(n)$ are real-valued, then $f(n) \in \mathbb{R}$ too. By the convolution theorem for discrete linear, translation-invariant systems, the DTFT of $f(n)$ is $F(\omega) = H(\omega)G(\omega)$. So, if H and G are ideal, then F will be a perfect bandpass filter with lower cutoff ω_L and upper cutoff ω_c.

We form band-reject or notch filters by subtracting a bandpass filter from an all-pass filter.

9.3.2 Design Using Window Functions

A useful discrete FIR filter design method uses the windowing concept from Section 9.2. This is a straightforward way to improve truncated perfect low-pass filters (Section 9.3.1.1). Again, let ω_c be the cutoff frequency so that $h(n)$ is given by

$$h(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}d\omega \Bigg]_{-N \le n \le N} = \frac{\sin(\omega_c n)}{\pi n}\Bigg]_{-N \le n \le N} \tag{9.39}$$

as shown in Figure 9.20(a).

Let $w(n)$ be a window function, for example the Hann window, of Table 9.2. Set $g(n) = h(n)w(n)$ as in Figure 9.20c. Using signal multiplication in the time domain is equivalent to convolution in the frequency domain:

$$G(\omega) = \mathcal{F}[h(n)w(n)] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\theta)W(\theta - \omega)\, d\theta. \tag{9.40}$$

Since discrete time Fourier transform of $w(n)$ has the shape of a weighted averaging function in the frequency domain, convolving it with $H(\omega)$ effectively smoothes the spectrum of $h(n)$. This blends away the Gibbs phenomenon ripples caused by the truncation of the perfect low-pass filter Figure 9.20b. The result is a magnitude spectrum almost completely devoid of the problematic ringing Figure 9.20d.

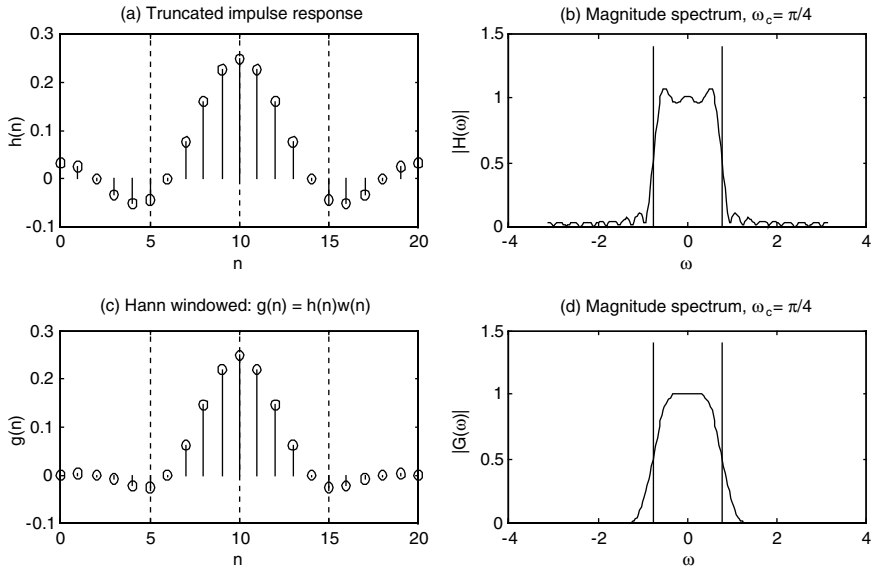

Fig. 9.20. FIR filter design using Hann window. Formerly perfect, but now truncated impulse response $h(n)$ (a). Vertical lines in the magnitude spectrum plot (b) mark the cutoff frequency. Panel (c) shows $g(n) = h(n)w(n)$, where $w(n)$ is a Hann window function having the same support as $h(n)$. The final panel (d) shows $|G(\omega)|$ with the Gibbs phenomenon ringing virtually eliminated.

9.3.3 Approximation

This section covers the approximation of ideal filters, the initial step among filter design tasks.

9.3.3.1 Design Criteria. Ideal filters are impossible to implement on a digital computer, but a filter that comes close to ideal performance is usually adequate for a signal processing and analysis application. Let us look at the magnitude spectrum of a typical discrete filter—in this case a third-order low-pass elliptic filter, which we will study in Section 9.5. A plot of the magnitude response illustrates design criteria (Figure 9.21).

Referring to Figure 9.21, we assume that the filter H has been normalized so that it has at most unit gain.

- The *passband* is the region in which the filter in which the magnitude response is near unity. For a low-pass filter, this is an interval around the origin: $\omega \in [0, \omega_p)$.
- The *stopband* is the region in which $H(\omega)$ is near zero: $\omega \in (\omega_s, \pi]$ for the low-pass filter in the figure.
- Between the passband and the stopband lies the *transition band*: $\omega \in [\omega_p, \omega_s]$.

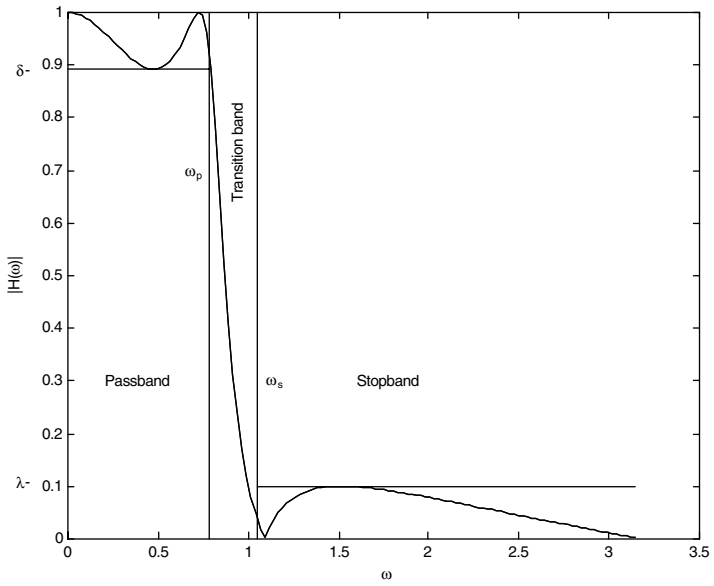

Fig. 9.21. Design characteristics of a low-pass filter (third-order low-pass elliptic), showing the positive frequencies $0 \le \omega \le \pi$.

- The *cutoff frequency* ω_c is a transition band value that is somewhat arbitrary, depending on the particular filter, but represents the frequency at which filter suppression begins. For our low-pass example, $\omega_p \le \omega_c \le \omega_s$.
- *Passband ripple* measures the variation of $|H(\omega)|$ within the passband. We typically specify a maximum value $\delta > 0$, such that $|H(\omega) - 1| \le \delta$ for $\omega \in [0, \omega_p)$. An equivalent, and sometimes convenient, mode of specification is to express tolerances in decibels (dB). Thus, $-\Delta \le 20\log_{10}|H(\omega)| \le \Delta$, for some $\Delta > 0$.
- *Stopband ripple* measures the variation of $|H(\omega)|$ within the stopband: There is a $\lambda > 0$, such that $|H(\omega)| \le \lambda$ for $\omega \in (\omega_s, \pi]$. Equivalently, in decibels: $20\log_{10}|H(\omega)| \le \Lambda$, for some $\Lambda < 0$.
- *Sharpness* indicates how narrow the transition band is. Applications that require fine frequency discrimination use filters with correspondingly sharp transition bands. One measure of sharpness is the average slope of $|H(\omega)|$ in the transition band. Another way to specify sharpness is to stipulate a derivative at a representative transition band value, such as ω_c or $(\omega_p + \omega_s)/2$.

The design criteria for other filters are similar. Thus, in a high-pass filter, the passband is at the upper end of the spectrum: $\omega_p \le \omega \le \pi$. In a bandpass filter, there are two stop bands (in the neighborhoods of 0 and π), and there is a single passband. Likewise, the notch filter has two outer passbands and a single inner stopband.

9.3.3.2 Design Steps. In signal analysis, our design steps for discrete filters are as follows:

(i) Determine the type (low-pass, high-pass, bandpass, or band-reject) of filter G appropriate to the application.
(ii) Select a low-pass filter that approximates an ideal filter's frequency-domain characteristics.
(iii) Design a suitable low-pass filter $h(n) = (H\delta)(n)$.
(iv) Convert the low-pass filter H to the application filter $g(n) = (G\delta)(n)$ using Fourier transform properties, ensuring that the approximation constraints on G are not compromised by the conversion.
(v) For an online application, maintain causality.
(vi) For IIR filters, maintain stability.
(vii) Achieve an efficient computer implementation.

It is sometimes feasible to design G directly, skipping the low-pass starter filter H and conversion. Once the filter type and approximation questions have been resolved in accord with application constraints—points (i) and (ii), above—the next step is to design a low-pass filter. The z-transform is the basic tool for working out these remaining design steps.

9.3.4 *Z*-Transform Design Techniques

We recall from the previous chapter that the *z-transform* $X(z)$ of a discrete signal $x(n)$ is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}, \tag{9.41}$$

where $z \in \mathbb{C}$. We recognize that if $z = e^{j\omega}$, then (9.41) is the discrete-time Fourier transform of $x(n)$. Evaluating the z-transform on the complex unit circle, $|z| = 1$, produces the DTFT. To avoid confusion, we may occasionally use the function notations $(\mathcal{Z}h)(z)$ or $H_z(z)$ for the z-transform and continue to use $H(\omega)$ for the DTFT. The z-transform came to signal processing from control theory in the early 1960s [40]. It is a staple topic of discrete signal processing books (for example, Refs. 7–11), and there are specialized texts covering the z-transform [41, 42].

The sum (9.41) converges absolutely on annular regions of the extended complex plane $\mathbb{C}^+ = \mathbb{C} \cup \{\infty\}$, called the *region of convergence*, ROC_X. The z-transform expression $X(z)$ for a signal is not unique; one must also specify ROC_X. Both the causal signal $x(n) = a^n u(n)$ and the anti-causal signal $y(n) = -a^n u(-n-1)$, for example, have the same z–transforms: $X(z) = Y(z) = z/(z-a)$. The difference is that $\mathrm{ROC}_X = \{z \in \mathbb{C}: |a| < |z|\}$ while $\mathrm{ROC}_Y = \{z \in \mathbb{C}: |z| < |a|\}$.

A theoretical subtlety on the region of convergence concerns square-summable signals. We know that discrete signals $h \in l^2$ have DTFTs, but it is possible

that $h \notin l^1$. In fact, the discrete Hilbert space l^2 is isomorphic to the continuous Hilbert space $L^2[-\pi, \pi]$, understanding that $x, y \in L^2[-\pi, \pi]$ are considered identical if they are equal except on a set of Lebesgue measure zero (Chapter 3). Thus, such signals $h(n)$ possess a discrete-time Fourier transform, even though ROC_H does not include the unit circle $|z| = 1$.

9.3.4.1 System Function, Stability, Causality. If H is a discrete linear, translation-invariant system, such as we might use for filtering a signal, then the z-transform $H(z)$ of its impulse response $h = H\delta$ is called the *system* or *transfer function* of H. The convolution theorem for the z-transform tells us that if $y = Hx = h*x$, then $Y(z) = H(z)X(z)$, where $X(z)$ is the z-transform of input $x(n)$, $Y(z)$ is the z-transform of output $y(n)$, and $H(z)$ is the system function.

Recall that a discrete LTI system H is *stable* if and only if the impulse response $h = H\delta$ is absolutely summable: $h \in l^1$. But if H is stable, then $h(n)$ has a DTFT $H(\omega)$. Since $H(\omega) = H_Z(e^{j\omega})$, ROC_H evidently contains the unit circle, $|z| = 1$. The converse is also true: If $H_Z(z)$ converges absolutely on a region that contains the unit circle, then H is stable (exercise).

A signal $x(n)$ is *right-sided* means that $x(n) = 0$ for $n < N \in \mathbb{Z}$. In this case, its z-transform $X(z)$ (9.41) contains at most a finite number of positive powers of z, and ROC_X is the exterior of a circle. If $N \geq 0$, then $\infty \in \mathrm{ROC}_X$, and $x(n)$ is *causal*; that is, $x(n) = 0$ for $n < 0$. Similarly, if $x(n) = 0$ for $n > N \in \mathbb{Z}$, then we say that $x(n)$ is a *left-sided* sequence. The ROC of a left-sided signal is the interior of a circle, omitting perhaps the origin $z = 0$.

An LTI system H is *causal* if its impulse response $h = H\delta$ is causal: $h(n) = 0$ for $n \leq 0$. Thus, $h(n)$ is right-sided, and ROC_X is the exterior of a circle. If H is causal and $y = Hx$, then

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)\,; \tag{9.42}$$

$y(n)$ can be computed without using future values of the input signal.

9.3.4.2 Systems Governed by Difference Equations. A wide variety of discrete systems are defined by a difference equation:

$$y(n) + \sum_{k=1}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m)\,. \tag{9.43}$$

Note that (9.43) allows us to compute a new output $y(n)$ if we know the previous N output values, the previous M input values, and the current input value. Thus, for real-time applications, signal filtering prior to frequency domain analysis is often implemented using filters governed by a difference equation. Offline applications, of course, do not worry about this detail, and they commonly employ noncausal filters.

If an LTI system H is governed by a difference equation, then its transfer function is rational. That is, $H(z) = P(z^{-1})/Q(z^{-1})$, where P and Q are complex polynomials. We can see this by taking the z-transform of both sides of (9.43):

$$Y(z) + \sum_{k=1}^{N} a_k Y(z) z^{-k} = \sum_{m=0}^{M} b_m X(z) z^{-m}. \tag{9.44}$$

Consequently,

$$Y(z)\left[1 + \sum_{k=1}^{N} a_k z^{-k}\right] = X(z) \sum_{m=0}^{M} b_m z^{-m}. \tag{9.45}$$

This becomes a rational function in z^{-1} by computing,

$$\frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\left[1 + \sum_{k=1}^{N} a_k z^{-k}\right]} = H(z). \tag{9.46}$$

Although the system function is given by the z-transform convolution theorem as a rational function in the complex variable z, it is often easier to work in terms of the variable z^{-1}, which is the z-transform of a unit delay.

9.3.4.3 Poles and Zeros Analysis. Let us continue to consider an LTI system H, defined by a difference equation. The rational system function $H(z) = Y(z)/X(z) = P(z^{-1})/Q(z^{-1})$ may also be characterized by its poles and zeros. The *poles* are the zeros of $Q(z^{-1})$, and—assuming that common factors are removed from $H(z)$—the *zeros* of $P(z^{-1})$ are those of $H(z)$ too.

To find the poles and zeros, we must factor $P(z^{-1})$ and $Q(z^{-1})$ into products of linear terms. The fundamental theorem of algebra guarantees that every complex polynomial factors into linear terms, unique except for their order. Many readers know theorem, but if we recount some complex variable theory from Chapter 1, an argument of Liouville[10] proves it fairly easily.

We may assume that some polynomial $P(z)$ has been reduced to its lowest terms and that it still has degree exceeding unity. If $P(c) = 0$ for some $c \in \mathbb{C}$, then $(z - c)$ evenly divides $P(z)$, so we must have $P(z) \neq 0$ for all $z \in \mathbb{C}$. This means that the reciprocal function $R(z) = 1/P(z)$ is defined for all $z \in \mathbb{C}$. But then $R(z)$ is everywhere differentiable, since its denominator is differentiable and has no zeros. Further as $|z|$ gets large, $|R(z)|$ gets small. So $R(z)$ is bounded and everywhere differentiable. But, by Liouville's theorem, a bounded, everywhere differentiable (analytic) function is constant (exercise). This means $R(z)$ is a constant, and we have a

[10]French mathematician Joseph Liouville (1809–1882) authored some 400 papers on number theory, integral equations, and differential geometry.

contradiction. It must be the case that we can always extract another root from a complex polynomial of degree two or more.

Now consider the case of a discrete causal LTI system H whose transfer function $H(z)$ is a rational function. Since H is causal, $h(n)$ is right-sided: $h(n) = 0$ for $n < 0$. ROC_H is the exterior of a circle. Since $H(z)$ is rational, its denominator is a complex polynomial $Q(z^{-1})$. The only values $z \in \mathbb{C}$ for which $H(z)$ does not exist are the zeros of $Q(z^{-1})$, which are the poles of $H(z)$. The number of poles is finite; it is at most the degree of $Q(z^{-1})$. Hence, there is a pole of largest modulus $|p|$, where $Q(p) = 0$. Finally, we conclude that ROC_H consists of the exterior of the circle defined by $|z| = |p|$. If all the poles of $H(z)$ are contained within the complex unit circle $|z| = 1$, then ROC_H contains the unit circle, and H is a stable system.

How a signal processing application implements a discrete filter on a digital computer depends on the support of its *impulse response*: *finite* (FIR) or *infinite* (IIR).

A simple convolution calculation suffices for an FIR filter. If $y = Hx$, and $h = H\delta$, then the filtered output $y(n)$ can be calculated from the input $x(n)$ via a convolution operation:

$$y(n) = \sum_{k=M}^{N} h(k)x(n-k)\,. \tag{9.47}$$

Finite impulse response filters are particularly easy to implement. The weighting function $h(k)$ is the impulse response of the linear, translation invariant discrete filter. The application must store at least $N - M + 1$ values for this calculation.

The system function for an FIR filter is given by

$$\frac{Y(z)}{X(z)} = \sum_{k=M}^{N} h_k z^{-k} = H(z)\,, \tag{9.48}$$

where we have written $h_k = h(k)$. Since $H(z)$ has no denominator polynomial, it is an all-zero filter. Thus, another way to characterize FIR and IIR filters is as follows:

- The system function for an FIR filter is an all-zero rational function of z^{-1}.
- The system function for an IIR filter has at least one pole.

FIR filters tolerate transient inputs. A transient in past $x(n)$ values eventually falls outside the window $[M, N]$ over which the sum is computed in (9.47), so an FIR filter's output is never permanently affected by input spikes.

FIR filters behave well. They are clearly stable; that is, a bounded output signal results from filtering a bounded input signal. If $x(n)$ represents time sampled data with a transient, then eventually the effect of the transient on the output $y = Hx$ will disappear.

This means that the delay of a sinusoid at the output is proportional to its frequency. This can be important for processing and analysis applications where it is important that tones not be scrambled by the filter. A preliminary filter for a speech recognition system, for example, should not turn a simple spoken word into a grunt followed by a squeak. Later we shall see that some FIR filters have linear phase.

Examples of FIR filters that are implemented as in (9.47) include moving average filters and weighted averaging filters. We explored moving average filters for noise reduction in the dual tone multifrequency detection application. There also we considered windowing the signal in order to obtain a time localized snapshot of its spectral characteristics.

The convolution calculation for an IIR filter is impossible to directly implement. However, a large, powerful, efficient, and therefore imporant class of IIR filters admits recursive implemenatation. Recursive filters save prior output values and combine them with current and past input values to produce a current output value. That is, these filters obey a difference equation (Chapters 2 and 8) of the form

$$y(n) + \sum_{k=1}^{N} a_k y(n-k) = \sum_{m=0}^{M} b_m x(n-m). \tag{9.49}$$

9.3.4.4 Implementation. Recursive implementation can be very efficient. In fact, certain IIR filters, defined by a difference equation, require fewer computations than equally powerful FIR filters. Moreover, these filters have almost linear phase.

To see schematically how we might implement a filter defined by a difference equation on a digital computer, let us consider the system defined by (9.49). We can see that if the blocks symbolized by z^{-1} store a value for a unit time, then the equation is implemented by the *Direct Form I* architecture [8] shown in Figure 9.22, where

$$y(n) = \sum_{m=0}^{M} b_m x(n-m) - \sum_{k=1}^{N} a_k y(n-k) = v(n) - \sum_{k=1}^{N} a_k y(n-k). \tag{9.50}$$

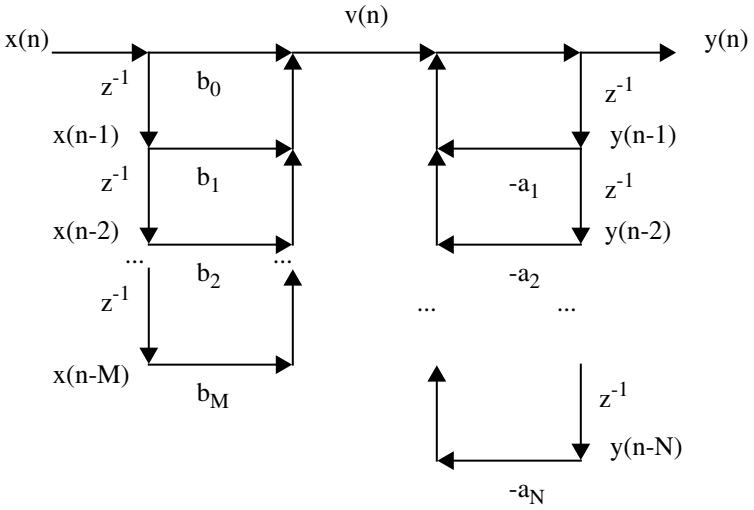

Fig. 9.22. Direct Form I implementation of a recursive system, governed by a difference equation.

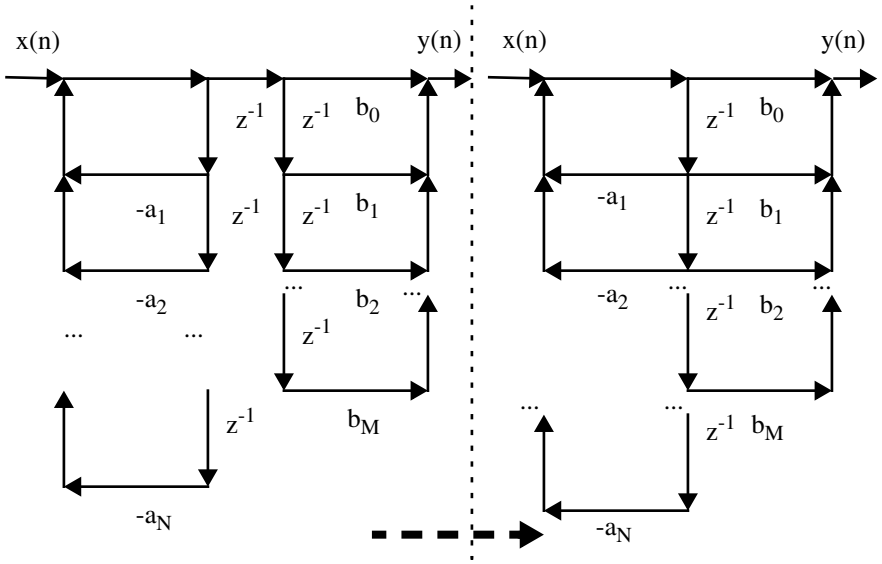

Fig. 9.23. The Direct Form II implementation of a recursive system is formed by (i) reversing the order of composition of the subsystems in the Direct Form I arrangement (as on the left) and (ii) then merging the duplicated delay operations (right).

The Direct Form I structure is seldom implemented, because a much more efficient scheme is possible. To see this, note that the lattice structures of Figure 9.22 cascades two LTI systems. One system produces $v(n)$ from input $x(n)$, and the other accepts $v(n)$ as an input and generates $y(n)$. We know from Chapter 2 that the composition of systems is characterized by the convolution of their respective impulse responses. Since convolution is commutative, so is composition. Thus, we can swap the order of the cascaded systems in Figure 9.22. This is shown on the left-hand side of Figure 9.23.

Reversing the order of these two subsystems leaves a network with two identical sets of sequential delays in the middle (Figure 9.23). The insight of the *Direct Form II* architecture is that the nodes across from one another contain the same mathematical result; the two halves can be joined, cutting the number of delay operations and economizing on memory locations. The Direct Form II, on the right-hand side of Figure 9.23, is a commonplace in computer implementations of signal processing systems.

When the difference equation coefficients (9.49) are all real, another efficient filter structure is possible. In this case, the poles and zeros of the system function $H(z)$ that have nonzero imaginary parts must consist of conjugate pairs. Then $H(z)$ is the product of terms of the form

$$G_p(z) = b_0 \frac{1 + b_{1p} z^{-1}}{1 + a_{1p} z^{-1}} \tag{9.51a}$$

and

$$K_q(z) = \frac{1 + b_{1q}z^{-1} + b_{2q}z^{-2}}{1 + a_{1q}z^{-1} + a_{2q}z^{-2}}, \tag{9.51b}$$

where all of the coefficients are real. Some coefficients may be zero to account for unequal numbers of poles and zeros. Terms of the form (9.51a) are called *first-order sections*. Terms of the form (9.51b) are called *second-order sections*. We may implement each first- or second-order section as a Direct Form I or Direct Form II network. When we serialize all of them to implement $H(z)$, we have the *cascade* architecture for a system defined by a difference equation. Strategies exist for pairing poles and zeros in the cascade architecture so as to minimize round-off error in finite precision arithmetic [11].

9.3.5 Low-Pass Filter Design

This section covers several methods for designing discrete low-pass filters. The mechanics of converting an approximate low-pass filter to one of the other types (high-pass, bandpass, or band-reject) are the same as for ideal filters. This section also covers two particularly important procedures for converting continuous domain to discrete filters: the impulse invariance technique and the bilinear transformation.

We have already constructed and applied several ad hoc discrete filters. One obvious method is to use a frequency-domain mask:

(i) Fourier transform the signal. Thus, from $x(n)$, we find $X(k)$ according to the DFT analysis equation (9.1).

(ii) Mark out all but the frequencies of interest to the application. In other words, the application determines a binary mask $H(k) \in \{0, 1\}$ and we set $Y(k) = H(k)X(k)$. Note that this is equivalent to convolving $y(n) = (h*x)(n)$, where $h(n)$ is the inverse DFT of $H(k)$.

(iii) Analyze the result $Y(k)$ by thresholding, segmentation, and classification procedures in the spirit of ordinary time-domain analysis (Chapter 4); or, alternatively, inverse transform $Y(k)$ to continue time-domain analysis on $y(n)$.

(iv) Finally, interpret the signal.

There are advantages and disadvantages to the obvious approach. It offers perfect control of the filtering, assuming that the sampling rate is high enough. Also, it allows the application designer to work in both domains: time and frequency. This could be pertinent. But one drawback is that it is only feasible for offline applications. Where decisions about the signal must be made while the data arrive, this strategy does not work very well. Even for offline applications, the method could be expensive; the discrete Fourier transform requires a lot of arithmetical operations, and if the data do not naturally come in fast transform-sized chunks, this presents another problem of data windowing and interpretation across the window boundaries.

Another method is to construct low-pass filters intuitively, using weighted averages or recursive, sample-and-hold algorithms. Although it chagrins a signal theorist, probably most of the filters used in analysis applications are of this type. Such adhoc filters are useful for salt-and-pepper noise removal, blending away transients, and preparing signals for derivative or edge finding operations. One can investigate the spectral characteristics of such filters using the discrete Fourier transform, such as we carried out in Section 9.1.

9.3.5.1 Impulse Invariance. It is possible to begin with an analog low-pass filter and convert it to a discrete filter. Often, the useful characteristics for an application are known from their analog frequency-domain description. An example is the Gaussian; it is analytically attractive, it decays quickly, and filtering with wider kernels does not create additional time domain structure, as shown by scale-space theory (Chapter 4).

The *impulse invariance* technique is as follows:

(i) From the application, a specification of the filter's continuous-domain Fourier transform $H_a(\Omega)$ is generated.

(ii) The inverse Fourier transform is applied to find $h_a(t)$, the impulse response of the analog filter.

(iii) The analog impulse respose is sampled $h(n) = h_a(nT)$, where $T > 0$ is the sampling interval.

Now, impulse invariance is simple and commonly invoked in signal analysis applications. One question, however, is whether the discrete sampling in step (iii) above undoes the frequency-domain behavior that motivated the analog filter's selection in the first place.

The sampling theorem (Chapter 7) answers this question. We can write $H(\omega)$, the discrete-time Fourier transform of $h(n)$, in terms of $H_a(\Omega)$, the analog Fourier transform of $h_a(t)$:

$$H(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(\frac{j}{T}[\omega + 2k\pi]\right). \tag{9.52}$$

We recognize the form of (9.52). The DTFT of $h(n)$ consists of a superposition of scaled (amplified by T^{-1}) versions of H_a, shifted by $2\pi/T$ and dilated by T.

Note also that if $|H_a(\Omega)| \approx 1$ for $|\Omega| \approx 0$ in (9.52), then $H(\omega)$ will provide a gain of approximately $1/T$. With the sampling rate close to 1 Hz, the effect will be negligble. However, if T is small—that is, the sampling rate is high—then the discrete filter will have a very high gain. For this reason, it is customary to set $h(n) = Th_a(nT)$ [7, 40].

9.3.5.2 Sampling Rational Analog Filters. An important discrete filter design method is to sample an analog filter whose Fourier transform is a rational

function. It turns out that the impulse invariance method applied to such an analog filter produces a discrete filter governed by a difference equation. Such analog filters are well-known from analog filter theory, and we will develop several of them later. This becomes a very powerful method for designing discrete filters with excellent design characteristics and efficient computer implementations.

Suppose that an analog low-pass filter H_a has impulse response $h_a(t)$. Let the radial Fourier transform of $h_a(t)$ be a quotient of polynomials in Ω:

$$H_a(\Omega) = \int_{-\infty}^{\infty} h_a(t) e^{-j\Omega t} dt = \frac{P(\Omega)}{Q(\Omega)}. \tag{9.53}$$

We can find $h_a(t)$ using a partial fractions expansion of $H_a(\Omega)$. This can be a messy manual computation, but it works just like partial fraction expansions for rational z-transforms. These are the steps to expand $H_a(\Omega)$ in partial fractions [10]:

(i) Normalize the fraction so that the denominator $Q(\Omega)$ has a leading coefficient of unity.

(ii) Since H_a is low-pass, as $|\Omega| \to \infty$, it must also be the case that $|H_a(\Omega)| \to 0$; the degree of $Q(\Omega)$ must exceed the degree of $P(\Omega)$:

$$H_a(\Omega) = \frac{P(\Omega)}{Q(\Omega)} = \frac{p_{M-1}\Omega^{M-1} + p_{M-2}\Omega^{M-2} + \cdots + p_0}{\Omega^M + q_{M-1}\Omega^{M-1} + \cdots + q_0}. \tag{9.54}$$

(iii) Factor the denominator $Q(\Omega)$ into its roots, ω_m, $1 \le m \le M$,

$$Q(\Omega) = \prod_{m=1}^{M} (\Omega - \Omega_m), \tag{9.55}$$

where we assume for now that the Ω_m are distinct. Possibly $Q(\Omega)$ has a form that allows us to easily derive its roots. In other cases, a computational method of root finding, such as the Traub–Jenkins algorithm may be employed [43].

(iv) Then $H_a(\Omega)$ has a partial fractions expansion of the form:

$$H_a(\Omega) = \sum_{m=1}^{M} \frac{c_m}{\Omega - \Omega_m}, \tag{9.56}$$

where c_m, $1 \le m \le M$, are constants.

(v) To find c_m, $1 \le m \le M$, note that—having assumed that the denominator's roots Ω_m are distinct—we see

$$c_m = [(\Omega - \Omega_m) H_a(\Omega)]\Big|_{\Omega = \Omega_m}. \tag{9.57}$$

The partial fraction expansion enables us to write out the impulse response $h_a(t)$ of the analog low-pass filter H_a. Indeed, the inverse radial Fourier transform of $\frac{c_m}{\Omega - \Omega_m}$ is $jc_m e^{j\Omega_m t}u(t)$, where $u(t)$ is the unit step signal. This transformation is valid so long as the imaginary part of Ω_m is positive: $\text{Imag}(\Omega_m) > 0$. By Fourier transform linearity, we have

$$h_a(t) = ju(t)\sum_{m=1}^{M} c_m e^{j\Omega_m t}. \tag{9.58}$$

Now we can discretize the filter by the impulse invariance technique, for instance. Let $T > 0$ be the sampling interval. Then,

$$h(n) = h_a(nT) = ju(n)\sum_{m=1}^{M} c_m e^{j\Omega_m nT}. \tag{9.59}$$

Taking the z-transform of (9.59) gives

$$H(z) = j\sum_{m=1}^{M} c_m \mathcal{Z}[u(n)e^{j\Omega_m nT}](z) = \sum_{m=1}^{M} \frac{jc_m}{1 - e^{jT\Omega_m}z^{-1}}. \tag{9.60}$$

The important points about this derivation are as follows:

- Note that (9.60) is already in the form of a partial fractions expansion.
- It has in fact the same partial fractions expansion coefficients as given by (9.56) except for the factor of $j \in \mathbb{C}$.
- The poles of $H(z)$ are at $\exp(jT\Omega_m)$ for $m = 1, 2, ..., M$.
- The pole at $\exp(jT\Omega_m)$ will be inside the unit circle if and only if $\text{Real}(jT\Omega_m) < 0$ and thus if and only if $\text{Imag}(\Omega_m) > 0$.
- If we consider $H(z)$ to be the z-transform of a causal filter, then $\text{Imag}(\Omega_m) > 0$ for all $m = 1, 2, ..., M$ implies that the region of convergence will be $\{z \in \mathbb{C}: |z| > a\}$, for some $1 > a > 0$, and the discrete filter H will therefore be *stable*.
- All of the partial fractions (9.60) are of the form $\frac{C}{1 - Az^{-1}} = \frac{Cz}{z - A}$, which is the z-transform of the LTI system governed by the difference equation $y(n) = Ay(n-1) + Cx(n)$.
- Finally, if the sampling rate $1/T$ differs substantially from unity, then we choose

$$H(z) = \sum_{m=1}^{M} T\frac{jc_m}{1 - e^{jT\Omega_m}z^{-1}} \tag{9.61}$$

in accord with (9.52) and the remarks thereafter.

Now let us turn to the special case of (9.55) where the denominator $Q(\Omega)$ has multiple roots. Suppose the root Ω_1 has multiplicity R and the remaining Ω_m, $2 \le m \le M$, are distinct. Then $H_a(\Omega)$ has a partial fractions expansion of the form

$$H_a(\Omega) = \sum_{r=1}^{R} \frac{c_{1,r}}{(\Omega - \Omega_1)^r} + \sum_{m=2}^{M} \frac{c_m}{\Omega - \Omega_m}. \tag{9.62}$$

where $c_{1,r}$, $1 \le r \le R$, and c_m, $2 \le m \le M$, are constants. The formula for calculating the $c_{1,r}$ is as follows:

$$c_{1,r} = \frac{\left[\dfrac{d^{R-r}}{d\Omega^{R-r}}(\Omega - \Omega_1)^R H_a(\Omega)\right]\Big|_{\Omega=\Omega_1}}{(R-r)!}. \tag{9.63}$$

If there are several multiple roots, then we follow the above procedure, inserting supplemental terms in the partial fractions expansion (9.62) and computing the coefficients with repeated derivatives (9.63).

Notice again that if the analog $H_a(\Omega)$ has at least one pole, then so will the z-transform (9.61), and the discrete filter will be IIR.

9.3.5.3 Laplace Transform Techniques. Analog filters and discrete filter designs from them are usually approached using the Laplace transform [6–11]. Readers are probably familiar with this tool from continuous-domain systems theory. The Laplace transform plays the same role in analog systems theory that the z-trasnform plays in discrete system theory. Let us briefly review how the transform is used with analog filters whose Fourier transform is a rational function. Specialized texts include Refs. 44 and 45.

The *Laplace transform* $X_L(s)$ of the analog signal $x(t)$ is defined by

$$X_L(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt, \tag{9.64}$$

where $s \in \mathbb{C}$. If H is an LTI system, then the Laplace transform $H_L(s)$ of its impulse response, $h = H\delta$, is also called the *system* or *transfer function* of H.

Note that if $x(t)$ has a Fourier transform $X(\Omega)$, then $X_L(j\Omega) = X(\Omega)$. The Fourier transform is the Laplace transform evaluated on the imaginary axis of the complex plane. If $s = \sigma + j\omega$, where $\sigma \in \mathbb{R}$, then $X_L(s)$ is the Fourier transform of $x(t)e^{-\sigma t}$. Transform convergence depends on the relation of $x(t)$ to the exponential factor $\exp(-\sigma t)$, and it does not depend on the imaginary part of $s = \sigma + j\omega$. Hence, the Laplace transform converges on vertical strips in the complex plane.

A couple of basic examples show that the Laplace transform must be associated with a *region of convergence* (ROC). If $x(t) = e^{-At}u(t)$, then $X_L(s) = 1/(s + a)$ and the $\mathrm{ROC}_X = \{s \in \mathbb{C}: \mathrm{Real}(s) > -a\}$. If $y(t) = -e^{-At}u(-t)$, then $Y_L(s) = 1/(s + a)$ and the

$\mathrm{ROC}_Y = \{s \in \mathbb{C}: \mathrm{Real}(s) < -a\}$. Recall that an analog LTI system H is stable if and only if the impulse response $h = H\delta$ is absolutely integrable: $\mathrm{h} \in L^1$. But this means $h(t)$ has a Fourier transform $H(\Omega)$. Since $H(\Omega) = H_L(j\Omega)$, ROC_H must contain the imaginary axis.

Now suppose $x(t)$ is *right-sided*: $x(t) = 0$ for $t < a \in \mathbb{R}$. If ROC_X contains the vertical line $\mathrm{Real}(s) = b \in \mathbb{R}$, then ROC_X contains $\{s \in \mathbb{C} : \mathrm{Real}(s) \geq \mathrm{Real}(b)\}$. This is fairly easy to see, because, for such $s \in \mathbb{C}$, $\exp(-\mathrm{Real}(s)t) \leq \exp(-\mathrm{Real}(b)t)$ for $t > 0$ and the transform integral (9.64) will still exist. The ROC of a right-sided signal is a right half-plane. Similarly, if $x(t)$ is *left-sided* ($x(t) = 0$ for $t > a \in \mathbb{R}$), then ROC_X is a left half-plane. Now consider the case of a causal LTI system H whose transfer function $H_L(s)$ is a rational function. Since H is causal, $h(t) = 0$ for $t < 0$. In other words, $h(t)$ is right-sided, and ROC_H is a right half-plane. Since $H_L(s)$ is rational, its denominator is a complex polynomial $Q(s)$. The only values $s \in \mathbb{C}$ for which $H_L(s)$ does not exist are the zeros of $Q(s)$, which are the rational function's *poles*. As there are only a finite number of poles of $H_L(s)$, ROC_H must be the half-plane to the right of the zero of $Q(s)$ with the largest real part.

We invert the Laplace transform using much the same methods as z-transform inversion. Rational functions $X_L(s)$ can be decomposed into a partial-fractions representation and the linearity propert applied to elementary transforms to arrive at $x(t)$. Table 9.3 lists basic Laplace transform properties.

Let us turn now to the use of the Laplace transform in designing discrete filters from rational analog filters [7]. Let the Laplace transform of $h_a(t)$ be a quotient of complex polynomials

$$H_L(s) = \int_{-\infty}^{\infty} h_a(t)e^{-st}dt = \frac{P(s)}{Q(s)} \tag{9.65}$$

TABLE 9.3. Summary of Laplace Transform Properties

Signal Expression	Laplace Transform or Property
$x(t)$	$X_L(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt$
$z(t) = ax(t) + by(t)$	$aX_L(\omega) + bY_L(\omega)$ (Linearity, $\mathrm{ROC}_X \cap \mathrm{ROC}_Y \subseteq \mathrm{ROC}_Z$)
$y(t) = x(t - a)$	$e^{-sa}X_L(s)$ (Time shift, $\mathrm{ROC}_X = \mathrm{ROC}_Y$)
$y(t) = x(t)\exp(at)$	$X_L(s - a)$ (Frequency shift, modulation, $\mathrm{ROC}_Y = \{s: s - a \in \mathrm{ROC}_X\}$)
$y(t) = x(at)$, $a \neq 0$	$\frac{1}{\lvert a\rvert}XL_L\left(\frac{s}{a}\right)$ (Scaling, dilation, $\mathrm{ROC}_Y = \{s: s/a \in \mathrm{ROC}_X\}$)
$y(t) = (x * h)(t)$	$F(s)H(s)$ (Convolution, $\mathrm{ROC}_X \cap \mathrm{ROC}_H \subseteq \mathrm{ROC}_Y$)

and let its partial fractions expansion be

$$H_L(s) = \sum_{m=1}^{M} \frac{d_m}{s - s_m}, \tag{9.66}$$

where d_m, $1 \le m \le M$, are constants, and the poles s_m are distinct. Laplace transform linearity and inversion of the summands in (9.66) implies

$$h_a(t) = u(t) \sum_{m=1}^{M} d_m e^{s_m t}. \tag{9.67}$$

Impulse invariance applies exactly as above. We set

$$h(n) = h_a(nT) = u(n) \sum_{m=1}^{M} d_m e^{s_m nT}, \tag{9.68}$$

and the z-transform of $h(n)$ is

$$H(z) = \sum_{m=1}^{M} d_m Z[u(n) e^{d_m nT}](z) = \sum_{m=1}^{M} \frac{d_m}{1 - e^{T s_m} z^{-1}}. \tag{9.69}$$

Notice that if $d_m = jc_m$ and $s_m = j\Omega_m$, then the two expressions for $H(z)$, (9.69) and (9.60), are identical. The conditions for stability and causality are similar too. From a Fourier transform perspective, we need the poles of the rational function $H_a(\Omega) = \frac{P(\Omega)}{Q(\Omega)}$ to have positive imaginary parts. From the Laplace transform standpoint, however, we require the poles of $H_L(s) = \frac{P(s)}{Q(s)}$ to have negative real parts. Of course, if Ω_0 is a pole of $H_a(\Omega)$, then $j\Omega_0$ is a pole of $H_L(s)$.

9.3.5.4 Bilinear Transformation. The bilinear transformation obtains a discrete filter from the frequency domain representation of an analog filter by directly mapping the analog frequency values to discrete frequency values. What sort of operation performs such a mapping? Note that analog frequencies can be arbitrarily large, $-\infty < \Omega < +\infty$, whereas discrete frequencies are limited to a 2π-wide interval: $-\pi < \omega \le \pi$. So we seek a function that maps the real line to the circumference of a circle. The arctangent, $\tan^{-1}$, maps $\mathbb{R}$ to the interval $(-\pi/2, \pi/2)$. Let T be the sampling interval. Then the following relation maps continuous to discrete frequency values:

$$\omega = 2\tan^{-1}\left(\frac{\Omega T}{2}\right). \tag{9.70}$$

Observe that as $\Omega \to \pm\infty$, the maximum analog frequency values, then $\omega \to \pm\pi$, respectively, the maximum discrete frequency values.

Suppose that $H_a(\Omega)$ is an analog lowpass filter with cutoff frequency Ω_c. Then the bilinear transformation (9.70) allows us to define a discrete low-pass filter as follows:

$$H(\omega) = H_a\left(\frac{2}{T}\tan\left(\frac{\omega}{2}\right)\right). \tag{9.71}$$

The cutoff frequency for $H(\omega)$ is $\omega_c = 2\tan^{-1}\left(\frac{\Omega_c T}{2}\right)$.

How does the scaling factor T come to appear in (9.70)? One practical reason is that for small frequencies, T controls the rate of change of ω with respect to Ω: $d\omega/d\Omega \approx T$. Some authors (e.g., Ref. 46) set $\omega = \frac{2}{T}\tan^{-1}\left(\frac{\Omega T}{2}\right)$ to ensure that for low frequencies $\omega \approx \Omega$. Another way of justifying the frequency mapping (9.70) is to consider the relation between poles of rational analog filters and the discrete filters obtained from them. If $s = s_m$ is a pole of the Laplace transform $H_L(s)$ of a filter (9.66), then corresponding to it is a pole $z = \exp(s_m T)$ of the z-transform $H(z)$ (9.69). This suggests a mapping from the Laplace s-plane to the z-plane: $z = e^{sT}$. Thus,

$$z = e^{sT} = \frac{e^{s(T/2)}}{e^{-s(T/2)}} \approx \frac{1 + s\frac{T}{2}}{1 - s\frac{T}{2}} = \frac{2 + sT}{2 - sT}, \tag{9.72}$$

where we approximate the quotient on the right in (9.72) using the first two Taylor series terms for the exponential function. This implies

$$s \approx \frac{2}{T}\left(\frac{z-1}{z+1}\right), \tag{9.73}$$

which relates the Laplace transform variable to the z-transform variable. To relate continuous and discrete frequency responses, we use $z = e^{j\omega}$ and $s = j\Omega$ in (9.73), treating it as an equality. After a little algebra, (9.70) results (exercise).

9.3.6 Frequency Transformations

There are convenient mappings of the independent variable of a filter's system function that convert a low-pass filter to a high-pass, bandpass, bandstop, or even another low-pass filter [7, 11, 33, 47].

9.3.6.1 Analog. Consider an analog low-pass filter with Laplace transform $H_L(s)$ and cutoff frequency $\Omega_c = 1$. The transformations are as follows:

(i) Let $\phi(s) = s/\Omega_H$. Then $H_L(\phi(s))$ is a low-pass filter with cutoff frequency Ω_H.
(ii) Let $\phi(s) = \Omega_H/s$. Then $H_L(\phi(s))$ is a high-pass filter with cutoff frequency Ω_H.

(iii) Let $\phi(s) = \dfrac{s^2 + \Omega_L\Omega_H}{s(\Omega_H - \Omega_L)}$. Then $H_L(\phi(s))$ is a bandpass filter with lower cutoff frequency Ω_L and upper cutoff frequency Ω_H.

(iv) Let $\phi(s) = \dfrac{s(\Omega_H - \Omega_L)}{s^2 + \Omega_L\Omega_H}$. Then $H_L(\phi(s))$ is a bandstop filter with lower cutoff frequency Ω_L and upper cutoff frequency Ω_H.

9.3.6.2 Discrete. Consider a discrete low-pass filter with z-transform $H(z)$ and cutoff frequency ω_c. The transformations are as follows:

(i) Let $\phi(z^{-1}) = \dfrac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$. If we set $\alpha = \sin\left(\dfrac{\omega_c - \omega_H}{2}\right) / \sin\left(\dfrac{\omega_c + \omega_H}{2}\right)$, then $H(\phi(z))$ is a low-pass filter with cutoff frequency ω_H.

(ii) Let $\phi(z^{-1}) = -\dfrac{z^{-1} + \alpha}{1 + \alpha z^{-1}}$. If $\alpha = -\cos\left(\dfrac{\omega_c + \omega_H}{2}\right) / \cos\left(\dfrac{\omega_c - \omega_H}{2}\right)$, then $H(\phi(z))$ is a high-pass filter with cutoff frequency ω_H.

(iii) Let $\phi(z^{-1}) = -\dfrac{z^{-2} - 2\alpha\beta z^{-1} + \gamma}{\gamma z^{-2} - 2\alpha\beta z^{-1} + 1}$. If $\alpha = \cos\left(\dfrac{\omega_H + \omega_L}{2}\right) / \cos\left(\dfrac{\omega_H - \omega_L}{2}\right)$,

$\kappa = \cot\left(\dfrac{\omega_H - \omega_L}{2}\right)\tan\left(\dfrac{\omega_c}{2}\right)$, $\beta = \kappa/(\kappa + 1)$, and $\gamma = (\kappa - 1)/(\kappa + 1)$,

then H(ϕ(z)) is a bandpass filter with upper cutoff frequency ω_H and lower cutoff frequency ω_L.

(iv) Let $\phi(z^{-1}) = \dfrac{z^{-2} - 2\alpha\beta z^{-1} + \gamma}{\gamma z^{-2} - 2\alpha\beta z^{-1} + 1}$. If $\alpha = \cos\left(\dfrac{\omega_H + \omega_L}{2}\right) / \cos\left(\dfrac{\omega_H - \omega_L}{2}\right)$,

$\kappa = \tan\left(\dfrac{\omega_H - \omega_L}{2}\right)\tan\left(\dfrac{\omega_c}{2}\right)$, $\beta = 1/(\kappa + 1)$, and $\gamma = (1 - \kappa)/(\kappa + 1)$,

then $H(\phi(z))$ is a bandstop filter with upper cutoff frequency ω_H and lower cutoff frequency ω_L.

Example (Low-Pass to High-Pass). Suppose that $\omega_c = \pi/4$ and $\omega_H = 3\pi/4$ in (ii) above. Then $\alpha = -\dfrac{\cos(\pi/2)}{\cos(-\pi/4)} = 0$ and $H(\phi(z)) = H(-z)$.

9.3.7 Linear Phase

It is possible to construct causal finite impulse response filters with *linear phase*. Where the analysis steps in an application depend on linear phase during signal

processing steps, this can be factor in favor of using finite impulse response (FIR) filters. Note that it is possible to have infinite impulse response (IIR) filters with linear phase. For example, if $h = H\delta$ and $h(n)$ is symmetric about $n = 0$, then H will have zero phase. However, we are interested in filters that are practically implementable, and therefore we require right-sided impuse responses: $h(n) = 0$ for $n < N$ for some N.

9.3.7.1 FIR Characterization. Let H be a linear, translation invariant system, let $h(n)$ be its impulse response, and let $H(\omega)$ be its discrete time Fourier transform. If $H(\omega) = e^{j\phi(\omega)}H_R(\omega)$, with $\phi(\omega)$ a linear function of ω, and $H_R(\omega) \in \mathbb{R}$, then we say H has *linear phase*.

The theoretical result we are going to prove is as follows. If H is a discrete causal FIR filter with impulse response $h = H\delta$ such that $h(n) = 0$ for $n > N - 1$, then H has linear phase if and only if for some $c \in \mathbb{C}$, with $|c| = 1$, $h(n) = ch^*(N - 1 - n)$, where h^* is the complex conjugate of h.

To begin, let us assume that H is causal, FIR, and has linear phase. Let $H(\omega) = e^{j\phi(\omega)}H_R(\omega)$, with $\phi(\omega) = a + b\omega$, for some $a, b \in \mathbb{R}$, and $H_R(\omega) \in \mathbb{R}$. Assume that Support$(h) = [0, N - 1]$ with $N > 0$, so that $h(0) \neq 0$ and $h(N - 1) \neq 0$. Let $A_H(\omega)$ be the amplitude function for $H(\omega)$ the DTFT of $h(n)$: $H(\omega) = e^{j\phi(\omega)}A_H(\omega)$. Then

$$H(\omega) = \sum_{-\infty}^{\infty} h(n)e^{-j\omega n} = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}, \tag{9.74}$$

$$H(\omega) = e^{j[2a + 2b\omega - \phi(\omega)]}A_H(\omega) = e^{2ja}e^{2jb\omega}e^{-j\phi(\omega)}A_H(\omega) = e^{2ja}e^{2jb\omega}H^*(\omega). \tag{9.75}$$

Note that $H(\omega)$ and its complex conjugate $H^*(\omega)$ are both 2π–periodic; we must have $2b = K \in \mathbb{Z}$ on the right-hand side of (9.75). Let $c = e^{2ja}$ and $g(n) = h^*(-n)$, so that $G(\omega) = H^*(\omega)$. Then the discrete signal $s(n) = cg(n + K)$ has DTFT $S(\omega) = ce^{jK\omega}G(\omega) = ce^{jK\omega}H^*(\omega) = H(\omega)$. Because the DTFT is invertible, we must have $h(n) = s(n) = cg(n + K) = ch^*(-K - n)$. We know that for $n < 0$, $h(n) = h^*(n) = 0$. Also, if $n > -K$, we have $ch^*(-K - n) = h(n) = 0$. Thus, $-K = N$–1 because that is the upper limit of the support of $h(n)$, and so $h(n) = \text{ch}^*(N - 1 - n)$, as claimed.

Conversely, suppose Support$(h) = [0, N - 1]$ and $h(n) = ch^*(N - 1 - n)$ for some $c \in \mathbb{C}$ with $|c| = 1$. Applying the DTFT properties gives $H(\omega) = ce^{-j(N-1)\omega}H^*(\omega)$. Let $c = e^{j\theta}$. If $H(\omega) = e^{j\phi(\omega)}A_H(\omega)$, then $H^*(\omega) = e^{-j\phi(\omega)}A_H(\omega)$. Putting these together, we have

$$e^{j\phi(\omega)} = e^{j\theta}e^{-j\phi(\omega)}e^{-j(N-1)\omega}, \tag{9.76}$$

and thus for some $K \in \mathbb{Z}$,

$$\phi(\omega) = \frac{(1 - N)\omega}{2} + \frac{\theta}{2} + \pi K. \tag{9.77}$$

Clearly, $\phi(\omega)$ is a linear function of ω, and we are done.

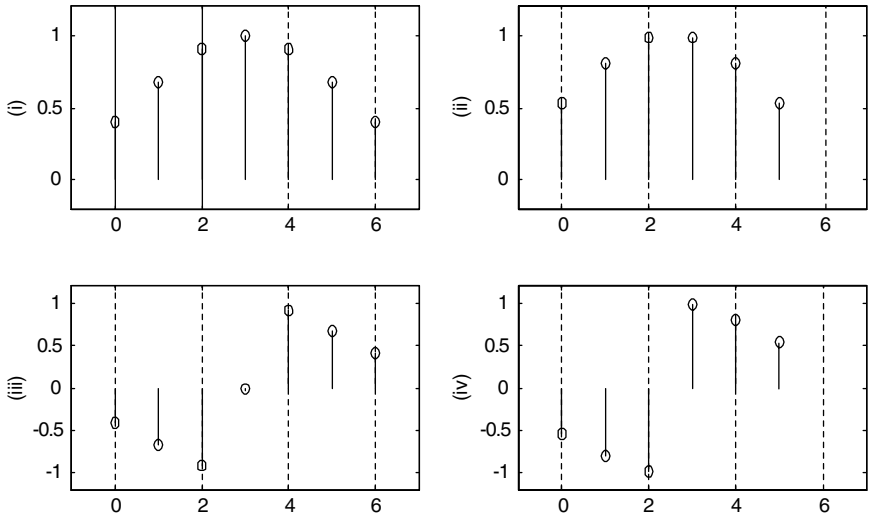

Fig. 9.24. Four classes of linear phase FIR filters.

9.3.7.2 Linear Phase Filter Classes. For real-valued filters $h = H\delta$, where $h(n)$ has support $[0, N-1]$, with $N > 0$, there are four categories of linear phase filters. We showed that a finite impulse response filter H has linear phase if and only if for some $c \in \mathbb{C}$, $|c| = 1$, $h(n) = ch^*(N-1-n)$. If $h(n) \in \mathbb{R}$, then $c = 1, -1$.

We can thus put the filter into one of four classes (Figure 9.24):

(i) $c = 1$ and $N = 2M + 1$ is odd.
(ii) $c = 1$ and $N = 2M$ is even.
(iii) $c = -1$ and $N = 2M + 1$ is odd.
(iv) $c = -1$ and $N = 2M$ is even.

9.3.7.3 Examples. We review two cases where the need for linear phase motivates the specific use of FIR filters: electroencephalogram (EEG) interpretation and seismogram intrpretation.

The first chapter explained the multichannel EEG [48], a biomedical signal that is often employed in studying brain functions and diagnosing injuries and illnesses. Electrodes attached to the scalp record the minute voltages produced by the interactions of large numbers of neurons. The signals are often quite noisy, and successive averaging is often employed to improve the quality of EEG traces. In studying auditory potentials—EEG traces of the part of the brain that is involved in the front-end processing of auditory nerve impulses—linear filtering has been investigated in order to improve upon successive averaging, the efficacy of which diminishes after a large number of sampling epochs. Frequencies above 2 kHz are removed by lowpass filtering, since they cannot be due to neuronal changes, which take place on the order

of 1 ms. The remaining noise is usually at low frequencies, DC to about 150 Hz [49]. The most popular filtering methods for such an application use IIR filters derived from difference equations (Section 9.3.4.2), since they are efficient and generally have better sharpness than FIR filters. But the nonlinear phase response of the causal IIR filters distorts the EEG trace, making linear phase FIR filtering preferable [49].

In seismic processing, the signals are generally quite noisy, composed of many frequency components, which are filtered by their propagation through the earth. These different sinusoidal pulses arrive at the sensing unit—the seismometer—at different times [50], a kind of phase delay. Perhaps the most basic task of earthquake seismology is to estimate the arrival time of an event, so that different seismometer stations can compare seismograms and locate the epicenter. Against the background noise of minor earth movements, impacts from construction equipment, and vehicle traffic vibrations, the seismic station determines the edge of a significant transient. At what time this transient occurs for a seismic station depends on the group delay of Mother Earth acting as a filter. Thus, for automated seismogram interpretation, a signal processing filter that introduces a nonlinear phase delay into the system might distort the signal and cause an error in pinpointing the onset time of a seismic shock. In order to facilitate the comparison of arrival times among different stations with different equipment, it is essential that their diverse noise removal filters not introduce any frequency dependent delays at all. The filtering requirement is even more stringent; successful analysis of the seismogram usually demands *zero phase* filtering [51].

9.4 WIDEBAND SIGNAL ANALYSIS

This section considers signals that contain diverse spectral components. These signals include chirps, which consist of rising or falling tones; transient signals, such as seismic pulses; signals with sharp edges; and irregularly shaped signals, such as image object boundaries. Our earlier methods of periodicity detection are successful only with much simpler waveforms.

9.4.1 Chirp Detection

A chirp is a signal segment where the frequency rises or falls over time. Strictly speaking, of course, a chirp is not a narrowband signal. But locally, at least, the signal energy is contained in a narrow spectral range. If this is indeed the case, then the task of chirp analysis becomes based upon a series of pure tone detection problems where the detected tones regulary rise or fall. This section considers the case where locally, at least, a signal contains mainly one frequency component, but that the frequency itself is changing over time.

9.4.1.1 Synthetic Chirps. Section 6.5 presented the theory of signal modulation, which is the theoretical foundation of chirp signal analysis. Let us first consider

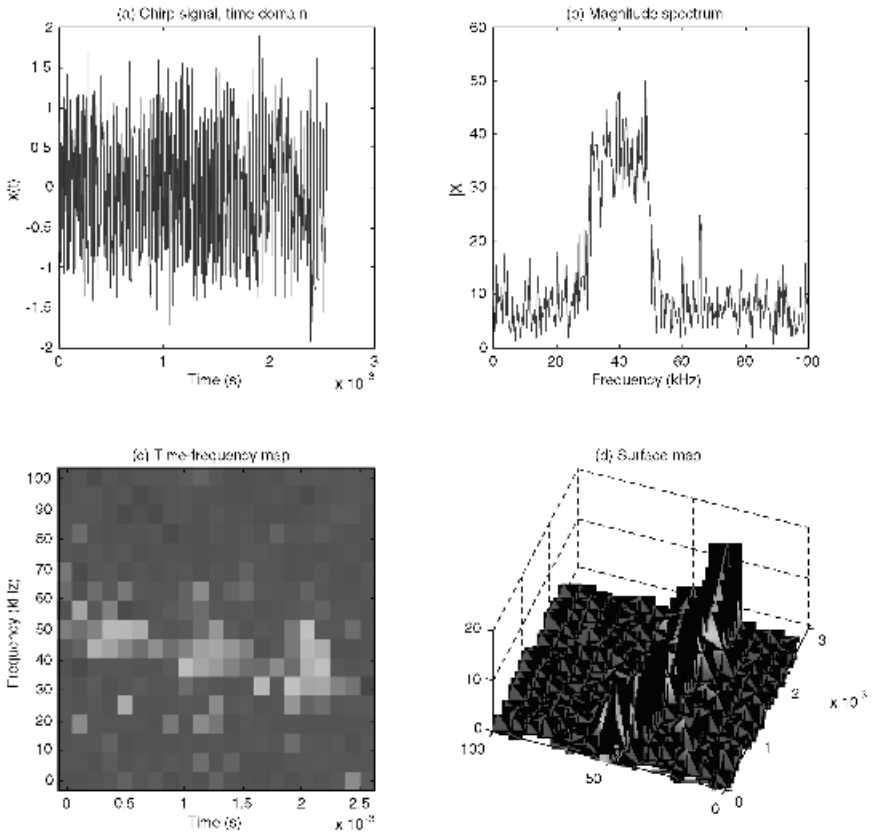

Fig. 9.25. Synthetic chirp with noise added (a). Panel (b) shows the magnitude spectrum. A time-frequency map (c) shows how the frequencies change over time. Surface maps (d) are useful aids for visualizing signal structure.

chirp signals for which the change in frequency over time is fairly simple—linear, for instance. Thus, we have $\omega(t) = t\omega_1 + \omega_0$ as shown in Figure 9.25.

From its time-domain plot Figure 9.25a, it is hard to understand the signal. The magnitude spectrum Figure 9.25b shows that a concentrated range of tones is present, in the range from 30 to 50 kHz, amidst some moderate background noise. But the time evolution of these frequencies and therefore the structure of the signal-itself remains unclear. We compute a sequence of 32-point discrete Fourier transforms over the time span of the signal. Overlapping the windows by eight samples helps smooth the time-frequency representation. The resulting map, shown in Figure 9.25c, reveals a linear chirp beginning at about 50 kHz and decreasing in frequency in a linear fashion down to about 30 kHz.

As long as its frequency content is, over a certain time interval, basically a tone, a chirp signal can be analyzed using tone detection techiques and a state machine.

Local discrete Fourier transforms carry out the frequency analysis, and some straightforward intepretive logic carry out the chirp discrimination. We can surmise the following steps:

(i) An order N for the number of samples for discrete Fourier transformation (DFT) is selected, depending on the sampling rate of the source signal and the range of frequencies expected in the chirp signal.
(ii) A DFT window overlap is selected.
(iii) DFT computations are conducted on the source signal within overlapping windows as chosen in (i) and (ii).
(iv) Where a relatively pure tone in the expected frequency range is found using local Fourier analysis, the state machine enters a tone detected state.
(v) Step (iv) is carried out again, and if there is a tone in the acceptable spectral range, then the tone is checked for purity, and the machine enters a state of increasing or decreasing tone frequency.
(vi) Continued local frequency analysis extends the time-domain support of the tone, breaks out of the tone detected state based on an invalid frequency response, or decides that the tone has the proper quality and range to continue the chirp defined in the current machine state.
(vii) This process continues until the chirp ends or the input signal is exhausted.

The main difficulties with this analysis is that it requires—for the most part—offline data analysis. That is, the DFT windows are applied around a time center value in a noncausal fashion. This could be an expensive operation, and for real-time processing, it may be impossible. One alternative might be to employ a bank of causal filters and seek significant outputs from the banks tuned to increasing or decreasing frequency bands. To achieve this, however, we need to devise filtering methods that are causal and sufficently efficient for online implementation.

9.4.1.2 Biological Signals: Bat Echolocation Chirp.

Now let us study the echolocation chirp recorded from a large brown bat (*Eptesicus fuscus*).[11] The sampling period is $T = 7$ μs, and there are $N = 400$ samples in the data set. The time-domain signal oscillates and rises in amplitude, but the plotted values evince few other clues as to its structure (Figure 9.26a). The magnitude spectrum explains a little more. There are frequencies between 20 kHz and 50 kHz, centered more or less strongly around a spectral peak at some 35 kHz. The spectrum appears to be bimodal (Figure 9.26b). From a Fourier domain perspective, we cannot tell whether the modes are frequency components that appear at certain times, one after the other, or whether they substantially overlap and the bimodality is an artifact of relatively weaker middle frequencies.

[11]This data set is available from the signal processing information base (SPIB): `http://spib.rice.edu/spib.html`. The authors wish to thank Curtis Condon, Ken White, and Al Feng of the Beckman Center at the University of Illinois for the bat data and for permission to use it in this book.

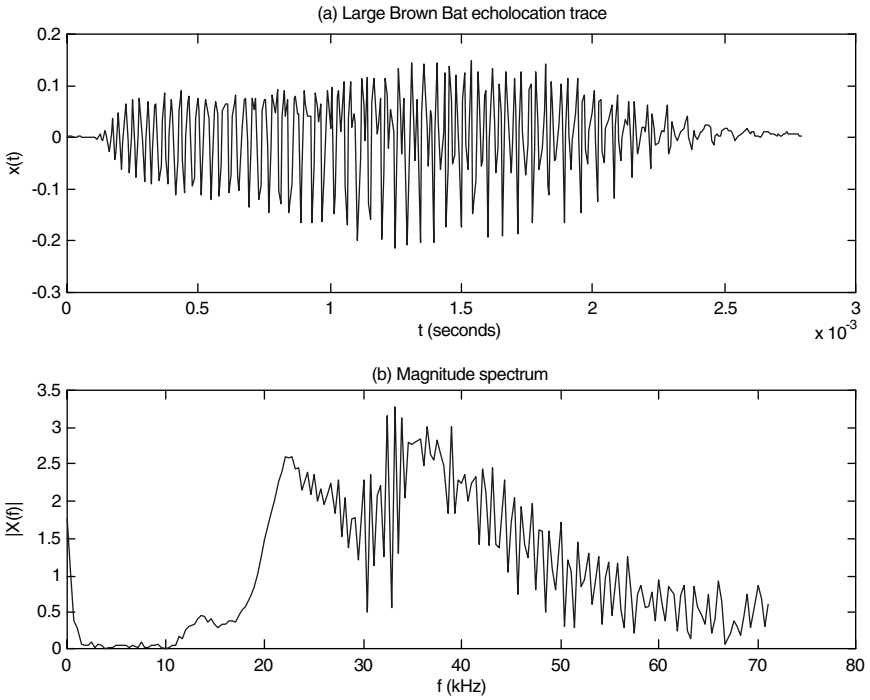

Fig. 9.26. Large brown bat echolocation pulse time-domian (a) and frequency-domain (b).

Our knowledge of the echolocation pulse's content changes dramatically when we develop the time-frequency map (Figure 9.27).

We use a boxcar window of width $N = 128$ to generate the local spectral values, overlapping successvie windows by $M = 120$ samples. This reveals three descending chirps (Figure 9.27a) and shows that the time-frequency plot is at least tri-modal. The bar-shaped artifacts most visible in the lower frequencies appear to correlate with window alignment. We try a Hann window function of length N and overlap M to improve the local frequency estimates, as shown in Figure 9.27b. This reduces the time-frequency artifacts, as one might expect. However, Hann windowing has the added benefit of resolving the time-frequency mode of highest initial frequency into two chirps; the echolocation pulse in fact contains four modes.

9.4.2 Speech Analysis

Let us now consider some low-level speech signal analysis problems. There is, to be sure, a large research literature on natural language processing from initial filtering methods, detection of utterances in noise, phoneme recognition, word recognition, contextual analysis, and artificial intelligence techniques for computerized speech understanding [13, 52–54].

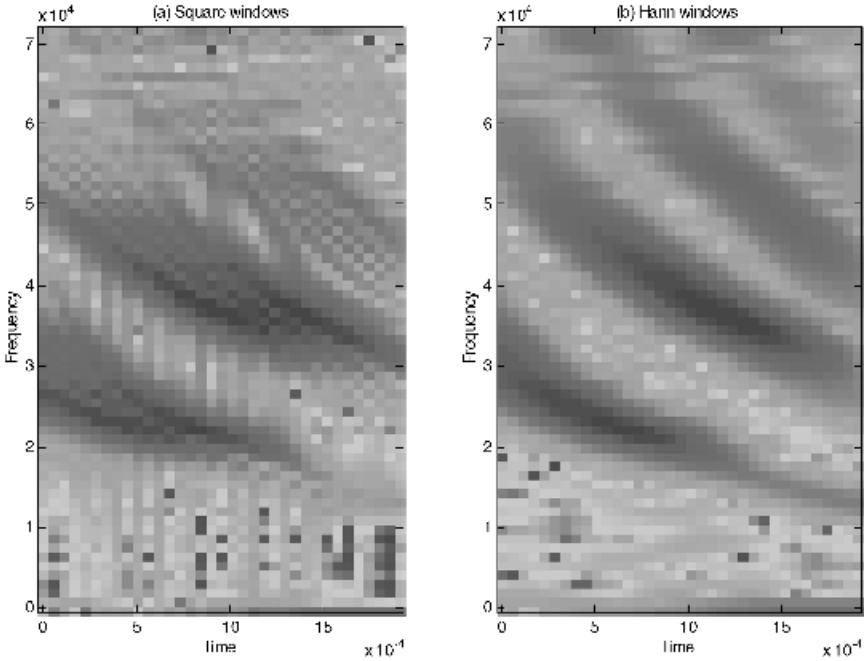

Fig. 9.27. The time-frequency map of the large brown bat echolocation pulse evidently has three components (a), each a descending chirp. Instead of a 128-point square window, panel (b) employs a 128-point Hann window to generate the local frequency information.

9.4.2.1 Formant Detection. *Formants* are relatively high-energy, tone-like components within a spoken word. They appear in the speech spectrum as isolated peaks. Consider, for example, a digitized voice fragment, consisting of a single word (Figure 9.28).

It is possible to discover formants using peak detection in the Fourier magnitude spectrum. The vowel phoneme /a/ in Figure 9.28c exhibits three strong peaks at approximately 300 Hz, 600 Hz, and 900 Hz. Such harmonics are characteristic of sounds produced in a tube, with a source at one end and open at the other. This crudely models the vocal tract, with the vocal cords at one end and the open mouth at the other. The vowel phoneme /i/ in in Figure 9.28e shows three resonant components as well as significant energy in many higher frequencies. Parts of the speech signal that do not contain high-energy tones, such as the /k/ in Figure 9.28b, cannot have formant structures.

Another formant detection task is to identify a *pitch* or *fundamental harmonic frequency* among the significant tones. This is the frequency of vibration of the vocal cords. In real speech recognition systems, this frequency must be identified and tracked, as it varies with the utterance as well as with the gender and emotional

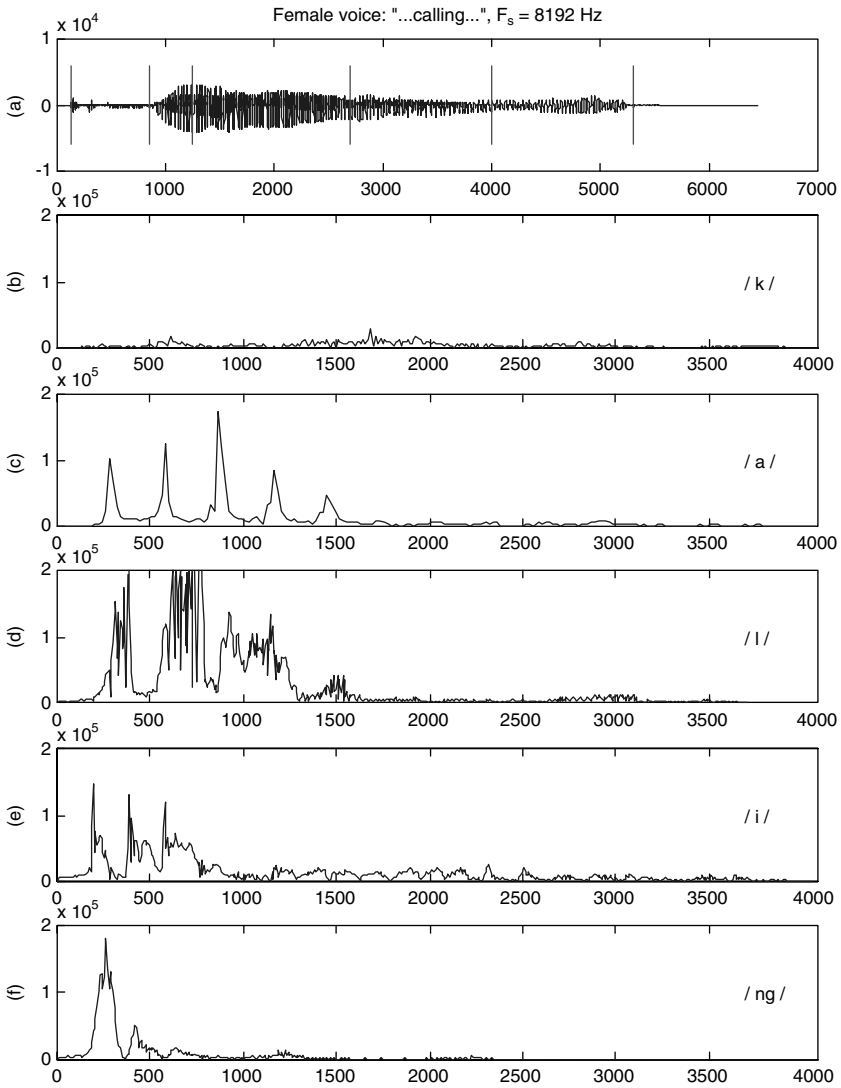

Fig. 9.28. Digitized voice (a) of a woman: "...calling...." Vertical lines mark the approximate locations of five phonemes. Panels (b)–(f) show the magnitude spectra for phonemes /k/, /a/, /l/, /i/, and /ng/.

state of the speaker. Some goal-directed information applies here. Pitch ranges from about 60 Hz to 300 Hz in adult males and up to 600 Hz in adult females.

A third formant detection task is to compare the center frequencies of significant tones. If such peaks represent formants, they must be integral multiples of the pitch frequency. Thus, in addition to ordinary peak finding, a formant detection

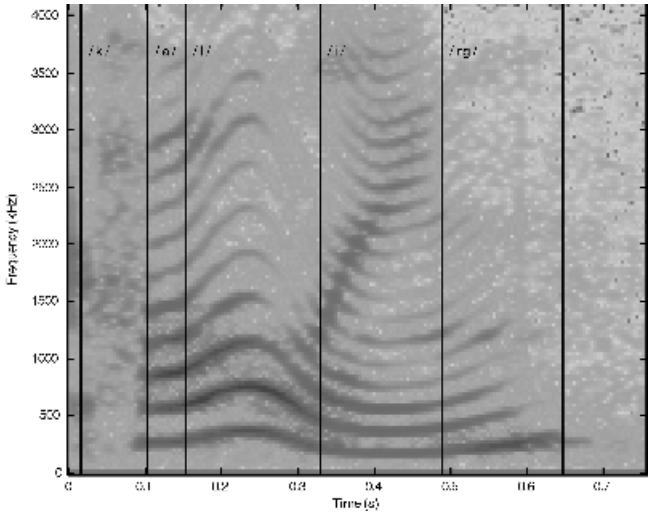

Fig. 9.29. Mixed-domain analysis of "calling" speech fragment. Using $N = 256$ point Hann windows for local frequency estimations centered $K = 32$ samples apart along the time axis produces the above time-frequency map. In many cases, voiced versus unvoiced speech segments can be found using the time-frequency map.

algorithm must include an assessment of the relative energy contributed by resonant frequencies.

9.4.2.2 Voice and Unvoiced Speech Determination. Speech sounds can be divided into *voiced* and *unvoiced* sounds, according to whether the vocal cords vibrate or do not vibrate, respectively (Figure 9.29). Unvoiced sounds split roughly into two categories: fricatives, such as /s/ or /f/, and aspirates, such as /k/.

9.4.2.3 Endpoint Detection. An important early speech analysis step involves automatic discrimination between background noise and speech signals. This segmentation procedure is important for automated compilation of speech databases and for detecting word boundaries in speech recognition systems. A fairly old but reliable method uses local energy and zero crossing rate parameters to isolate speech [55]. From a background noise sample, upper- and lower-energy parameters E_u and E_l, respectively, and a zero crossing threshold Z_c are determined. The algorithm refines the boundaries of a speech fragment in three stages as follows:

(i) The initial energy-based segmentation, say speech exists within $[M_u, N_u]$, is given by where local signal energy exceeds E_u.

(ii) The refined energy-based segmentation widens this interval to $[M_l, N_l]$ by searching outside $[M_u, N_u]$ for the points where energy diminishes to E_l.

(iii) Finally, a measure of local signal frequency—the zero crossing rate—is checked outside of $[M_l, N_l]$. If the rate exceeds Z_c three or more times in the 250-ms intervals on either side of $[M_l, N_l]$, then the speech fragment boundaries grow again to $[M_z, N_z]$, where $[M_u, N_u] \subseteq [M_l, N_l] \subseteq [M_z, N_z]$.

Noise versus speech discrimination problems continue to attract experimental researchers [56, 57].

9.4.3 Problematic Examples

Let us mention a couple of signal analysis problem domains where Fourier transform-based interpretation techniques begin to break down.

Seismic signals contain oscillations, but these oscillations are of unusually short time domain and are interspersed with transient artifacts that often thwart analysis. The Fourier transform is adequate for long-term periodic signal trends, but its efficiency as a signal descriptor diminishes with the duration of the oscillations. Geophysicists resorted to the short-time or windowed Fourier transform, which we cover in the next chapter, with some success. This theory constructs a time- and frequency-domain transform using the Fourier transform and the windowing techniques of Section 9.2. However, transient components are still problematic. Finally, efforts to get around the difficulties of windowed Fourier methods let to the development of the wavelet transform, which is the subject of Chapter 11.

Another problem area for Fourier methods is shape analysis. It seems that object shapes, however they might be described mathematically, are comprised of parts. One-dimensional methods, such as we develop here, can be applied to object boundaries, and we attempt such applications in the last chapter. There are Fourier transform-based approaches:

(i) An early method, called Fourier descriptors, approximates image object boundaries with varying numbers of Fourier series components.

(ii) Another method, the Fourier–Mellin transform, incorporates a temporal parameter into the transform.

These strategies work, but when the bounded object resolves into separate parts, the overall analytic techniques collapse. As a result, automated object recognition systems tend to retreat into structural methods of pattern recognition and cordon off, perhaps giving up on frequency-domain interpretation.

9.5 ANALOG FILTERS

Although our present goal is discrete time signal analysis and the frequency selective systems that support it, we have ample reasons for developing a respectable theory of analog filtering.

(i) While signal analysis is computerized interpretation of signals and therefore assumes a digital implementation, it relies upon discrete theory.

(ii) However, our ultimate source signals come from analog world.

(iii) We need to filter incoming analog so as to remove high frequencies that would otherwise cause aliasing during sampling, according to the Nyquist criterion.

(iv) We may be able to implement cheaper analog filters.

(v) A lot of discrete filter designs are based on analog filters.

(vi) We have shown that it is easy to derive discrete filters from analog versions—and especially if the analog filters have a rational transfer function.

(vii) Thus, if we develop a sound theory of analog filters, then we shall have a correspondingly sound theory of discrete filters.

(viii) Analog filters are historically prior.

(ix) The analog theory of continuous theory involves continuous sinusoidal signals, and hence the analog Fourier transform is a convenient theoretical tool. Anyway, this is consonant with our analog-first treatment of the frequency transforms.

Just as the contiuous Fourier transform is the natural setting for studying signal frequency, so the class of continuous domain, or analog, filters constitute the right beginning place for our study of frequency-domain signal analysis. Signal analysis usually takes place on a digital computer, but the frequency-selective algorithms that operate on digital signal representations often derive from continuous-domain filters. That is one reason for studying analog filters. But even before signals are digitized and fed to the computer, the signal digitization must often be filtered by analog means so that aliasing (Chapter 7) is minimized. This section presents some basic analog filters, introduces their characterisitic descriptors, and outlines the mechanics of converting one filter into another.

Conventional analog filter designs begin by examining the frequency-domain behavior $X(\omega)$ or the s-domain behavior $X_L(s)$ of analog signals $x(t)$. System theory texts covering analog filter theory and the Laplace transform include Refs. 6, 58, and 59. Texts that concentrate on the subsequent conversion to discrete time processing are [7–11, 26].

Classical electronics studies networks of electrical components—resistors, capacitors, and inductors—which implement the analog filtering operation in hardware. Circuit design texts cover the electronic circuit designs [60–63]. More modern electrical and computer engineering texts cover the design of hardware for digital filtering [64]. For signal analysis using digital computers, we need digital filters that selectively enhance and suppress signal frequency components of interest to the application. We derive the digital filters from their analog equivalents using some classic methods.

9.5.1 Introduction

A *filter* is a frequency-selective linear, translation-invariant system. Analog filtering takes place by virtue of the convolution property: $Y(\Omega) = H(\Omega)X(\Omega)$. So, if $|H(\Omega)|$ is small for values of Ω where it is desirable to suppress frequencies in the input $x(t)$

and $|H(\Omega)|$ is near unity where it is desirable to preserve frequencies in $x(t)$, then convolution with $h(t)$, $y(t) = (h^*x)(t)$ performs the requisite frequency selection operation. To describe system filtering we specify the magnitude spectrum, $|H(\Omega)|$, or, equivalently, $|H(\Omega)|^2$. We are mainly interested in real-valued filters: $h(t) \in \mathbb{R}$. Since $H(-\Omega) = H^*(\Omega)$ in this case and since $|H(\Omega)| = |H^*(\Omega)|$, the filters herein have both positive and negative frequency components. Thus, the magnitude spectra are symmetric about the frequency-domain origin, $\Omega = 0$.

This section explains how to construct bandpass and high-pass filters from low-pass filters. The discussion begins with the Gaussian; its Fourier transform is also Gaussian, so it is a natural choice for a low-pass filter. Filter constructions depend on the Fourier transform properties from Chapter 5.

The analog convolution operation is once again denoted by the * operator: $y = x^*h$. We define

$$y(t) = (x^*h)(t) = \int_{-\infty}^{\infty} x(s)h(t-s)\, ds\,. \tag{9.78}$$

Section 6.4.1 introduced ideal analog filter types. One filter missing there is the notch or band-reject filter. It is a like a reverse bandpass, and it suppresses rather than preserves a range of frequencies.

9.5.2 Basic Low-Pass Filters

Since we can move easily from low-pass filters to any of the other three basic types—high-pass, bandpass, or band-reject—let us consider some examples.

9.5.2.1 Perfect. The ideal low-pass filter completely removes all frequencies higher than some given frequency and preserves the rest without amplifying or attenuating them. This is the familiar analog *moving average* or *boxcar* filter.

9.5.2.2 Gaussian. The Gaussian

$$g_{\mu,\sigma}(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} = g(t) \tag{9.79}$$

with mean μ and standard deviation σ has Fourier transform

$$G(\Omega) = \int_{-\infty}^{\infty} g(t)e^{-j\Omega t}dt = \exp\left(-\left[\frac{\sigma^2\Omega^2}{2} + j\Omega\mu\right]\right). \tag{9.80}$$

Its magnitude spectrum is also a Gaussian, centered at $\Omega = 0$. Thus, $g(t)$ is the impulse response of an analog low-pass filter. If $x(t)$ is an analog signal and $y = Gx = (g * x)(t)$ is the convolution with the Gaussian (9.79), then $Y(\Omega) = X(\Omega)G(\Omega)$, where $G(\Omega) = (\mathcal{F}g)(\Omega)$ is given by (9.80). Gaussians decay rapidly, faster than the inverse of any

polynomial. Thus, the Fourier transform of the system response $Y(\Omega) = (\mathcal{F}y)(\Omega)$ will contain the $x(t)$ frequency components near $\Omega = 0$, but they will be suppressed by the product with the Gaussian $G(\Omega)$. This idea is basic to all low-pass filtering.

Now we also gain a deeper understanding of scale space analysis, introduced in Chapter 4. Recall that we showed that smoothing signals by ever broader Gaussian kernels had the unique property that no structure was created in the process. We understand signal structure to be determined by the concavity regions within the signal. Now (9.80) shows that the wider kernels are actually low-pass filters with smaller passbands. That is, the wider kernels progressively remove the high-frequency components, leaving relatively lower frequency undulations, and—more importantly—not creating additional changes in signal curvature.

A particularly important area of signal analysis is the detection, classification, and recognition of signal features that vary according to the size of their features—according to their *scale*, for instance.

9.5.2.3 Rational Functions. We can find other examples based on rational functions. For example, suppose $H(\Omega) = (1 + \Omega^2)^{-1}$. The inverse Fourier transform is

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\Omega)e^{j\Omega t}d\Omega = \frac{e^{-|t|}}{2}, \tag{9.81}$$

which is easy to see from the forward radial transform $\mathcal{F}(\exp(-a|t|)) = 2a/(a + \Omega^2)$. Evidently, convolution with $h(t)$ performs a weighted averaging on input signal data. In signal analysis applications, the pulse $h(t)$ is often called a Lorentzian, and it is used to find peaks, valleys, and transients in general by the method of template matching.

9.5.2.4 Better Low-Pass Filters. It turns out that very good filters can be built by pursuing the idea of rational functions introduced in the previous example. The important features of such filters are as follows:

- They have superior cutoffs—sharper and more like perfect low-pass filters.
- Upon discrete conversion, they will be causal.
- They are stable.
- They have efficient implementations, relying on difference equations.

For signal analysis applications involving real-time data, such as in speech recognition or industrial control applications, causality is important. Of course, many signal analysis applications do not involve real-time data; that is, they are offline applications, and so causal systems are less critical. Nonetheless, very good filters can be developed for data whose discrete values are only known for the present and past.

9.5.3 Butterworth

Butterworth[12] *filters* have maximally flat pass and stop bands. Thus, the filter designer that values, above all else, reduced pass- and stop-band ripple inclines toward this filter.

9.5.3.1 Conditions for Optimally Flat Filters. We describe these filters via their frequency-domain representation. Their Fourier transforms are based on rational functions—quotients of continuous domain polynomials. The Butterworth filter specializes the rational function by looking at the Taylor series representations of its numerator and denominator.

Indeed, we have already constructed lowpass filters out of rational functions. The idea is to look at analog filters $h = H\delta$, such that their Fourier transform power spectra $|H(\Omega)|^2$ are rational functions $B(\Omega)/A(\Omega)$, where $A(\Omega)$ and $B(\Omega)$ are polynomials. We impose conditions on the rational functions so that we achieve our design criteria: passband performance, stopband performance, cutoff frequency, allowable ripple, and required transition band sharpness. Significantly, for signal analysis on digital computers, when the power spectrum of an analog filter is a rational function, then it can be used to derive a discrete filter.

A simple but useful fact is that if $h(t) \in \mathbb{R}$, then the squared magnitude spectrum $|H(\Omega)|^2$ is an even function of Ω. To see this, note that Fourier transform symmetry properties imply $H(-\Omega) = H^*(\Omega)$. So $|H(\Omega)|^2 = H(\Omega)H^*(\Omega) = H(\Omega)H(-\Omega)$. But then $|H(-\Omega)|^2 = H(-\Omega)H^*(-\Omega) = H(-\Omega)H(\Omega)$ too.

We thus consider $P(\Omega) = |H(\Omega)|^2 = B(\Omega)/A(\Omega)$, such that $P(\Omega)$ is symmetric about $\Omega = 0$. This means $A(\Omega)$ and $B(\Omega)$ are polynomials in Ω^2 (exercise). Thus,

$$A(\Omega) = a_0 + a_2\Omega^2 + a_4\Omega^4 \ldots + a_{2N}\Omega^{2N}, \tag{9.82a}$$

$$B(\Omega) = b_0 + b_2\Omega^2 + b_4\Omega^4 \ldots + b_{2M}\Omega^{2M}, \tag{9.82b}$$

with $a_0 = b_0$. We may assume that $a_0 = b_0 = 1$. A low-pass filter implies $|H(\Omega)| \to 0$ as $|\Omega| \to \infty$, so $N > M$. For the filter stopband to be maximally flat as $\Omega \to \infty$, the maximal number of numerator terms in (9.82b) should be zero. Thus, $b_2 = b_4 = \cdots = b_{2M} = 0$, and we see

$$P(\Omega) = |H(\Omega)|^2 = \frac{1}{A(\Omega)} = \frac{1}{1 + a_2\Omega^2 + a_4\Omega^4 \ldots + a_{2N}\Omega^{2N}}. \tag{9.83}$$

Butterworth criteria also require the filter's passband to be maximally flat at $\Omega = 0$, which entails $a_2 = a_4 = \cdots = a_{2n-2} = 0$. We define the cutoff frequency Ω_c of the

[12]After S. Butterworth, a British engineer who first analyzed this response profile ["On the theory of filter amplifiers," *Wireless Engineer*, vol. 7, pp. 536–554, 1930]. V. D. Landon later described this same filter as *maximally flat* [Cascade amplifiers with maximal flatness, *RCA Review*, vol. 5, pp. 347–362, 1941].

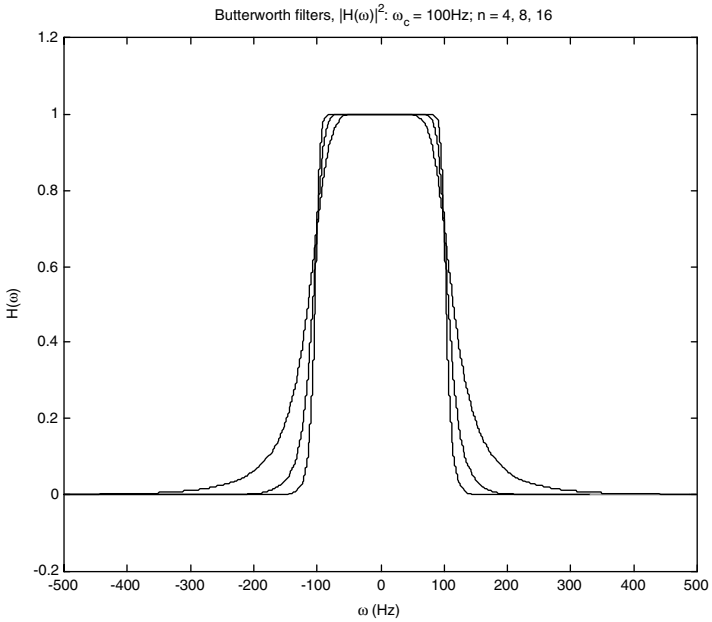

Fig. 9.30. Analog Butterworth filters for a few orders.

Butterworth filter by $(\Omega_c)^{-2n} = a_{2n}$. Thus, the *Butterworth filter* of order $N > 0$ is defined by its Fourier transform $H(\omega)$ (Figure 9.30):

$$H(\Omega) = \sqrt{\frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}}}; \tag{9.84}$$

an important next step in practical implementation is to decide which square roots to choose for $H(\Omega)$ in (9.84). Note that $H(\Omega_c) = 2^{-1/2} \approx 0.707$, for any filter order.

It is possible to invert the Butterworth filter Fourier transform $H(\Omega)$, truncate the impulse response $h(t)$, and then sample the result. As we have observed, this would induce some errors due to aliasing into the final filtering result. The filter designer might reduce these errors by preserving a large number of discrete samples. But this necessitates a time-consuming convolution operation. It turns out, however, that because a Butterworth filter has a rational Fourier transform, an efficient discrete implementation is possible using difference equations.

Now let us turn to the approximation problem for the Butterworth filter. The procedure differs depending on how the designer performs conversion from the analog to discrete filter form: impulse invariance or bilinear transformation.

9.5.3.2 Butterworth Approximation: Impulse Invariance. Given bounds on how far the filter's magnitude spectrum can stray from the ideal passband and

stopband, the Butterworth filter approximation finds two parameters: the filter order $N > 0$ and the radial cutoff frequency Ω_c in (9.84). Suppose that we require a discrete low-pass Butterworth filter with unit DC gain: $|H(\omega)| = 1$ for $\omega = 0$, where $H(\omega)$ is the DTMF of the filter impulse response $h(n)$. We assume that the sampling frequency is high enough so that aliasing of $H_a(\Omega)$, the filter's (radial) Fourier transform, is not an issue. *This allows us to use the analog filter's magnitude response in the design approximation*.

Suppose that the passband is $|\Omega| < \Omega_p$ and the application requires that $|H_a(\Omega)|$ is within $\delta > 0$ of unity. Suppose the stopband begins at $\Omega_s > \Omega_p$, and we need $|\mathrm{H}_a(\Omega)|$ to be within $\lambda > 0$ of zero. The Butterworth magnitude response is monotone; it suffices to consider the dual constraints:

$$\left|H(\Omega_p)\right| = \sqrt{\frac{1}{1+\left(\frac{\Omega_p}{\Omega_c}\right)^{2N}}} \geq 1 - \delta \approx 1, \tag{9.85a}$$

$$\left|H(\Omega_s)\right| = \sqrt{\frac{1}{1+\left(\frac{\Omega_s}{\Omega_c}\right)^{2N}}} \leq \lambda \approx 0. \tag{9.85b}$$

Filter designers often prefer approximation by differences in decibels. Thus, we say the passband magnitude (dB) is greater than a small negative value (Δ) and the stopband magnitude (dB) is less than a large negative value (Λ):

$$0 > 10\log_{10}\left|H(\Omega_p)\right|^2 \geq \Delta, \tag{9.86a}$$

$$10\log_{10}\left|H(\Omega_s)\right|^2 \leq \Lambda < 0. \tag{9.86b}$$

The above constraints reduce to

$$1 + \left[\frac{\Omega_p}{\Omega_c}\right]^{2N} \leq R_\Delta = 10^{-\frac{\Delta}{10}} \approx 1, \tag{9.87a}$$

$$1 + \left[\frac{\Omega_s}{\Omega_c}\right]^{2N} \geq R_\Lambda = 10^{-\frac{\Lambda}{10}} >> 1, \tag{9.87b}$$

but (9.85a) and (9.85b) give similar relations too. Let $R = \dfrac{\log(R_\Delta - 1)}{\log(R_\Lambda - 1)}$ so that solving for Ω_c gives

$$\log\Omega_c = \frac{R\log\Omega_s - \log\Omega_p}{R-1}. \tag{9.88}$$

At this point, the filter designer makes choices. We can use (9.87a) to find the Butterworth filter order

$$\nu = \frac{\log(R_\Delta - 1)}{2\log\left(\frac{\Omega_p}{\Omega_c}\right)}, \tag{9.89}$$

which will generally not be a whole number. Designers usually round the order upward, taking the Butterworth order N to be the *integral ceiling* of ν in (9.89)—the integer greater than ν, or $Ceil(\nu)$. Using equality in either (9.87a) or (9.87b), the cutoff frequency must be recomputed. For instance, setting

$$\Omega_{c,N} = \frac{\Omega_p}{\sqrt[2N]{R_\Delta - 1}} \tag{9.90}$$

establishes a new cutoff frequency based on the upwardly rounded order and the passband constraint. The resulting low-pass Butterworth filter satisfies the passband condition determined by $\Omega_{c,N}$ and improves upon the stopband condition. This is a good approach for low sampling rates, when the aliasing caused by filter conversion using impulse invariance is a concern [7].

Alternatively, tight application timing contraints might force the designer to round down and choose a smaller N, the *floor* of ν in (9.89). Perhaps $N = Floor(\nu) \approx \nu$, or deviation from the passband and stopband specifications is acceptable. The designer might also opt for a cutoff frequency that favors an exactly met stopband. Assuming that the filter order is rounded upward, the filter then meets a more stringent passband specification. The cost of this option is that it does not counteract the aliasing arising from impulse invariance. The exercises explore these design alternatives.

We determine the ω_p and ω_s (radians/sample) from the sampling rate $F_s = 1/T$ and the application's specified analog passband and stopband frequencies (Hz).

Example (DTMF Passband and Stopband Calculations). Suppose we require a lowpass filter that removes high-frequency noise for the dual-tone multifrequency application of Section 9.1. Let the digital sampling rate be $F_s = 8192$ Hz. Then discrete frequency $\omega = \pm\pi$ corresponds to the Nyquist frequency of $F_s/2 = 4096$ Hz. DTMF tones range up to 1633 Hz. Thus, the lowpass filter bands could be specified by setting $\omega_p = (1633/4096) \times \pi \approx 0.3989\pi$, and, depending upon the desired filter sharpness, $\omega_s = (1800/4096) \times \pi \approx 0.4395\pi$.

Example (Butterworth impulse invariance approximation). We require a low-pass filter with a passband within 1 dB of unity up to $\omega_p = \pi/4$ and at least 5 dB below unity beyond $\omega_s = \pi/3$. Assume the sampling interval is $T = 1$. Thus, we require $\Delta = -1$, and $\Lambda = -5$. The approximation steps above give $R_\Delta = 1.2589$, $R_\Lambda = 3.1623$, and $R = -1.7522$. The exact cutoff frequency for this example is $\Omega_c = 0.9433$, but we elect to round $\nu = 3.6888$ upward to $N = 4$. Revising the cutoff frequency, we find $\Omega_{c,N} = 0.9299$.

9.5.3.3 Poles and Zeros Analysis. Let us continue the design of the Butterworth low-pass filter of order $N > 0$ by factoring the denominator of the squared magnitude response. Indeed, many analog filters satisfy

$$P_H(\Omega) = |H(\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}} = H(\Omega)H(-\Omega)\,. \tag{9.91}$$

$P_H(\Omega)$ has $2N$ poles in the complex plane. Let the roots of the denominator in (9.91) be $\Omega_1, \Omega_2, ..., \Omega_{2N}$. In the case of the Butterworth squared magnitude, the roots lie on a circle of radius Ω_c in the complex plane. They are in fact the order-$2N$ roots of unity scaled by the cutoff frequency Ω_c. All we have to do to find $H(\Omega)$ that satisfies (9.91) is to select one pole from each pair $\{\Omega_i, -\Omega_i\} \subset \{\Omega_1, \Omega_2, ..., \Omega_{2N}\}$. But a judicious root selection allows us to construct a causal discrete filter governed by a difference equation and therefore having an efficient computer implementation. We can obtain a discrete difference equation from an analog filter that has a rational Fourier transform $H(\Omega)$ only if its poles have positive imaginary parts (Section 9.2.5.5). (Equivalently, if we are working with the Laplace transform, because $H(\Omega) = H_L(s)\Big|_{s = j\Omega}$, this means that the poles of $H_L(s)$ must have negative real parts.) With causality and difference equation implementation in mind, we retain those roots, $\Omega_1, \Omega_2, ..., \Omega_N$, such that

$$H(\Omega) = \frac{1}{(\Omega - \Omega_1)(\Omega - \Omega_2)\ldots(\Omega - \Omega_N)}\,. \tag{9.92}$$

with $\text{Imag}(\Omega_i) > 0$.

Let the partial fractions expansion of (9.92) be

$$H(\Omega) = \sum_{i=1}^{N} \frac{c_i}{\Omega - \Omega_i}\,, \tag{9.93}$$

where c_i, $1 \le i \le N$, are constants. The results of Section 9.2.5.5 give the corresponding discrete filter's transfer function:

$$H(z) = \sum_{i=1}^{N} T\frac{jc_i}{1 - e^{jT\Omega_i} z^{-1}}\,, \tag{9.94}$$

where T is the sampling interval. We generally compute ω_p and ω_s using the sampling interval, as explained in the previous section. Then, we derive the order N and the cutoff frequency Ω_c from the Butterworth approximation. The poles and zeros analysis gives a Butterworth filter of order N with cutoff frequency Ω_c, no matter what sampling interval we choose; it is conventient to assume $T = 1$. It is usually necessary to scale the coefficients of the discrete filter's impulse response for unit DC gain. Let us see how this works by following through on the previous example.

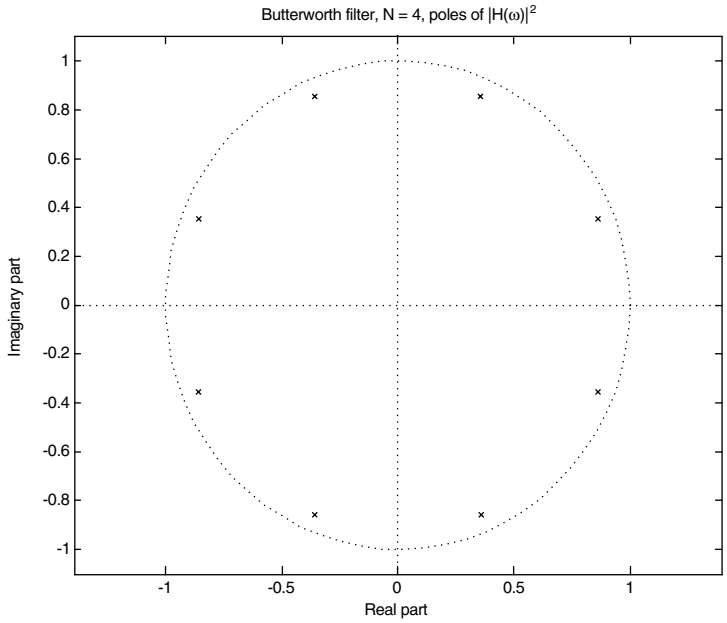

Fig. 9.31. Butterworth filter, $N = 4$, eight poles of $|H(\Omega)|^2$.

Example (Butterworth impulse invariance design, $N = 4$). Suppose we require a lowpass filter with order $N = 4$ and cutoff frequency $\Omega_c = 0.9299$, as in the previous example. Figure 9.31 shows the eight roots of the Butterworth squared magnitude function denominator. To find $H(\Omega)$, we select poles having a positive imaginary part. This selection corresponds to the poles of $H_L(s)$ whose real parts are negative, because if Ω_0 is a pole of $H(\Omega)$, then $H_L(j\Omega_0)$ is a pole of $H_L(s)$ (Table 9.4).

TABLE 9.4. Pole Selection in Butterworth Filter Design, $N = 4$, $\Omega_c = 0.9299$

Poles of $\lvert H(\Omega)\rvert^2$	Fourier-Selected Poles	Poles of $\lvert H_L(s)\rvert^2$	Laplace-Selected Poles
$-0.8591 + 0.3559j$ $-0.8591 - 0.3559j$	$-0.8591 + 0.3559j$	$-0.3559 - 0.8591j$ $0.3559 - 0.8591j$	$-0.3559 - 0.8591j$
$-0.3559 + 0.8591j$ $-0.3559 - 0.8591j$	$-0.3559 + 0.8591j$	$-0.8591 - 0.3559j$ $0.8591 - 0.3559j$	$-0.8591 - 0.3559j$
$0.3559 + 0.8591j$ $0.3559 - 0.8591j$	$0.3559 + 0.8591j$	$-0.8591 + 0.3559j$ $0.8591 + 0.3559j$	$-0.8591 + 0.3559j$
$0.8591 + 0.3559j$ $0.8591 - 0.3559j$	$0.8591 + 0.3559j$	$-0.3559 + 0.8591j$ $0.3559 + 0.8591j$	$-0.3559 + 0.8591j$

Thus, performing the partial fractions calculations for $H(z)$ gives

$$H(z) = T\left[\frac{j(-0.2379 + 0.5745j)}{1 - e^{jT(-0.8591+0.3559j)}z^{-1}} + \frac{j(1.3869 - 0.5745j)}{1 - e^{jT(-0.3559+0.8591j)}z^{-1}} + \frac{j(-1.3869 - 0.5745j)}{1 - e^{jT(0.3559+0.8591j)}z^{-1}} + \frac{j(0.2379 + 0.5745j)}{1 - e^{jT(0.8591+0.3559j)}z^{-1}}\right] \tag{9.95}$$

where T is the sampling interval. Writing $H(z)$ as a quotient of polynomials in z^{-1}, we see that (9.46) becomes

$$\frac{\sum_{m=0}^{M} b_m z^{-m}}{\left[1 + \sum_{k=1}^{N} a_k z^{-k}\right]} = \frac{0.0873z^{-1} + 0.1837z^{-2} + 0.0260z^{-3}}{1 + (-1.7091)z^{-1} + 1.3967z^{-2} + (-0.5538)z^{-3} + 0.0880z^{-4}}. \tag{9.96}$$

Now, finally, we can implement (9.95) with a cascade architecture and (9.96) with a Direct Form II. We can compute the impulse response $h(n)$ as in Figure 9.32 by feeding a discrete impulse through either of the difference equation architectures or by using (9.68).

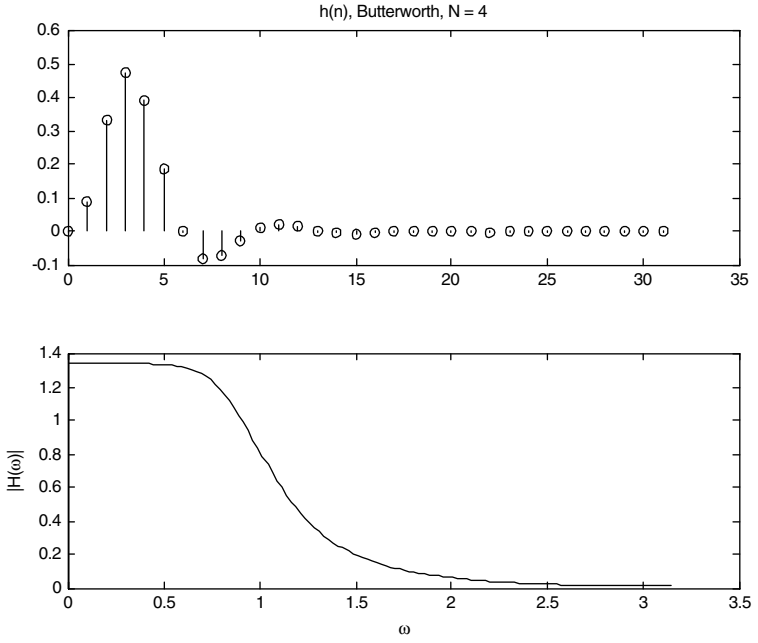

Fig. 9.32. Impulse response $h(n)$ of Butterworth low-pass filter, $N = 4$, $\Omega_c = 0.9299$ (top), and magnitude response (bottom) $|H(\omega)|$. Note that $H(0) \approx 1.3386$, so for unit DC gain, we scale $h(n)$ by $(1.3386)^{-1}$.

9.5.3.4 Butterworth Approximation: Bilinear Transformation. This section considers the Butterworth low-pass filter approximation using the bilinear transformation. The crucial difference in finding the filter order N and the cutoff frequency Ω_c is the frequency mapping $H(\omega) = H_a\left(\frac{2}{T}\tan\left(\frac{\omega}{2}\right)\right)$. Here, $H(\omega)$ is the DTMF of the desired discrete low-pass filter, $T > 0$ is the sampling interval, and $H_a(\Omega)$ is an analog Butterworth low-pass filter whose order N and cutoff frequency Ω_c remain to be found.

Let us suppose specifications on the passband and stopband similar to those with which we began the impulse invariance approximation. Thus,

$$0 > 10\log_{10}\left|H(\omega_p)\right|^2 = 10\log_{10}\left|H_a\left(\frac{2}{T}\tan\left(\frac{\omega_p}{2}\right)\right)\right|^2 \geq \Delta, \tag{9.97a}$$

$$10\log_{10}\left|H(\omega_s)\right|^2 = 10\log_{10}\left|H_a\left(\frac{2}{T}\tan\left(\frac{\omega_s}{2}\right)\right)\right|^2 \leq \Lambda < 0\,, \tag{9.97b}$$

where ω_p is the discrete passband frequency, $\omega_s > \omega_p$ is the stopband frequency, and $\Lambda < \Delta < 0$. Applying the Butterworth filter condition as before, we calculate

$$\left[\frac{2}{T\Omega_c}\tan\left(\frac{\omega_p}{2}\right)\right]^{2N} \leq 10^{-\frac{\Delta}{10}} - 1 = R_\Delta - 1, \tag{9.98a}$$

$$\left[\frac{2}{T\Omega_c}\tan\left(\frac{\omega_s}{2}\right)\right]^{2N} \geq 10^{-\frac{\Lambda}{10}} - 1 = R_\Lambda - 1. \tag{9.98b}$$

We treat these as equalities and solve for the filter order:

$$\nu = \frac{\log\left(\frac{R_\Delta - 1}{R_\Lambda - 1}\right)}{2\log\left(\frac{\tan(\omega_p/2)}{\tan(\omega_s/2)}\right)} \tag{9.99}$$

where, in general, ν is not an integer. Typically, we round ν upward: $N = \text{Ceil}(\nu)$. We can use equality in either (9.98a) or (9.98b) to find Ω_c. An advantage of bilinear transformation over impulse invariance is that there is no stopband aliasing. Thus, computing Ω_c in terms of ω_s gives

$$\Omega_c = \frac{2\tan(\omega_s/2)}{T}(R_\Lambda - 1)^{-\frac{1}{2N}}. \tag{9.100}$$

Example (Butterworth Bilinear Transformation Approximation). Suppose we require the same low-pass filter: $\omega_p = \pi/4$, $\omega_s = \pi/3$, $\Delta = -1$, and $\Lambda = -5$. Again, $R_\Delta = 1.2589$, $R_\Lambda = 3.1623$, but we calculate $\nu = 3.1957$ and choose $N = 4$. The exact analog cutoff frequency with $T = 1$ is $\Omega_c = 1.048589$, corresponding to a discrete filter cutoff of $\omega_c = 0.965788$. Note that as ν nears its integral floor—in this case $\nu \approx 3$—it might be feasible to round to the *lower* integral order. This could be useful when computing time is a consideration and some cushion exists for passband and stopband specifications.

Having derived the filter order N and the analog cutoff frequency Ω_c, the rest of the bilinear filter design steps are the same as for impulse invariance.

Example (Butterworth Bilinear Transformation Design, $N = 4$). For the low-pass filter with order $N = 4$ and cutoff frequency $\omega_c = 0.9658$, as in the previous example, the analog cutoff frequency is $\Omega_c = 1.0486$, since $\Omega_c = \frac{2}{T}\tan\left(\frac{\omega_c}{2}\right)$. Of the eight poles of

$$\left|H_a(\Omega)\right|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} = H_a(\Omega)_a H(-\Omega), \tag{9.101}$$

we select those having a positive imaginary part for $H_a(\Omega)$. The poles of the Laplace transform $H_{L,a}(s)$ will thus have negative real parts (Table 9.5).

Thus, using the chosen poles (Table 9.5) the rational Laplace transform is

$$H(s) = \frac{1}{s^4 + 2.7401s^3 + 3.7541s^2 + 3.0128s + 1.2090}. \tag{9.102}$$

Substituting the bilinear transform relation $s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$ with $T = 1$ into (9.102) gives the discrete transfer function,

$$H(z) = \frac{0.0166z^4 + 0.0665z^3 + 0.0997z^2 + 0.0665z + 0.0166}{z^4 - 1.5116z^3 + 1.2169z^2 - 0.4549z^2 + 0.0711}. \tag{9.103}$$

TABLE 9.5. Pole Selection in Butterworth Filter Design Using Bilinear Transformation, $N = 4$, $\Omega_c = 0.9658$.

$\lvert H_a(\Omega)\rvert^2$ Poles	Fourier Poles Selected	$\lvert H_{L,a}(s)\rvert^2$ Poles	Laplace Poles Selected
$-0.9688 + 0.4013j$	$-0.9688 + 0.4013j$	$-0.4013 - 0.9688j$	$-0.4013 - 0.9688j$
$-0.9688 - 0.4013j$		$0.4013 - 0.9688j$	
$-0.4013 + 0.9688j$	$-0.4013 + 0.9688j$	$-0.9688 - 0.4013j$	$-0.9688 - 0.4013j$
$-0.4013 - 0.9688j$		$0.9688 - 0.4013j$	
$0.4013 + 0.9688j$	$0.4013 + 0.9688j$	$-0.9688 + 0.4013j$	$-0.9688 + 0.4013j$
$0.4013 - 0.9688j$		$0.9688 + 0.4013j$	
$0.9688 + 0.4013j$	$0.9688 + 0.4013j$	$-0.4013 + 0.9688j$	$-0.4013 + 0.9688j$
$0.9688 - 0.4013j$		$0.4013 + 0.9688j$	

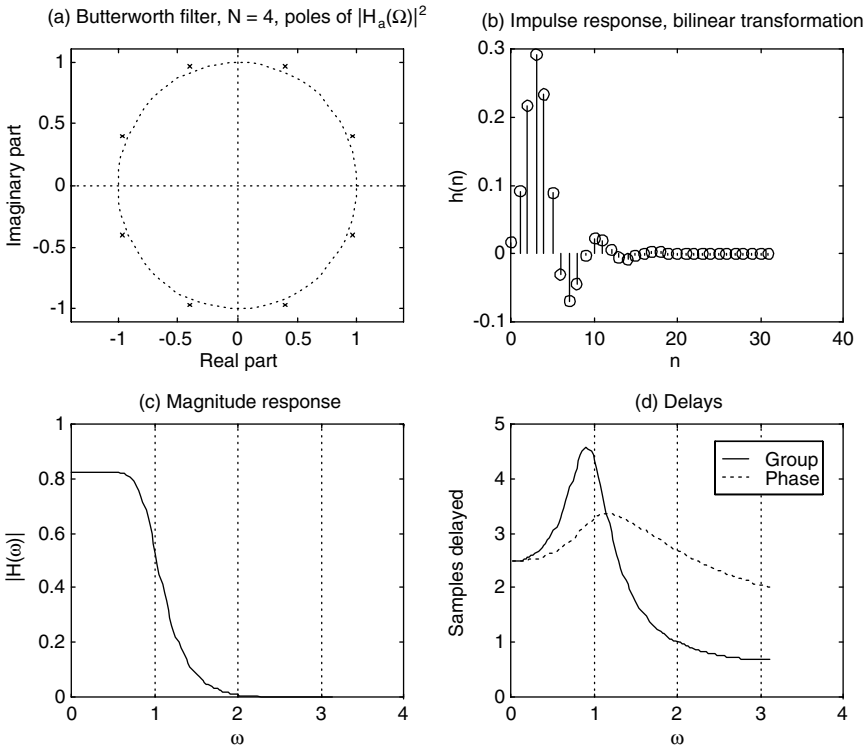

Fig. 9.33. Butterworth filter design, bilinear transformation. Pole locations for analog squared magnitude response (a). Impulse response $h(n)$ of Butterworth low-pass filter, $N = 4$, $\omega_c = 0.9658$ (b) and (c) magnitude response $|H(\omega)|$. Note that $H(0) \approx 0.8271$, so for unit DC gain, we scale $h(n)$ by $(0.8271)^{-1} \approx 1.2090$. Panel (d) shows the phase and group delay for this filter.

The partial fraction expansion of $H(z) = B(z)/A(z)$ is

$$\frac{B(z)}{A(z)} = \frac{-0.3048 + 0.1885j}{1 - (0.4326 + 0.5780j)z^{-1}} + \frac{-0.3048 - 0.1885j}{1 - (0.4326 - 0.5780j)z^{-1}} + \frac{0.1962 - 1.2480j}{1 - (0.3232 + 0.1789j)z^{-1}} + \frac{0.1962 + 1.2480j}{1 - (0.3232 - 0.1789j)z^{-1}} + 0.2337 \tag{9.104}$$

The filter has a difference equation implementation (9.49), from which the impulse response follows (Figure 9.33b).

Note that it is possible to skip the approximation step, instead stipulating a discrete cutoff frequency ω_c and guessing a filter order $N > 0$. After computing the associated analog cutoff Ω_c, the above poles and zeros analysis follows. This step produces an impulse response $h(n)$, from which one can derive a Fourier magnitude response $|H(\omega)|$. Should the passband or stopband not meet the anticipated

constraints, the filter order is incremented and trial-and-error goes on. This is a good use case for a computerized filter design package.

Consider the group and phase delays of the above Butterworth low-pass filter. If $H(\omega) = e^{j\phi(\omega)}H_R(\omega)$ is a filter's DTFT, where $H_R(\omega) \in \mathbb{R}$, and $\phi(\omega)$ is its phase response, then its group delay is $-d\phi(\omega)/d\omega$. Since FIR filters enjoy linear phase (Section 9.3.6), their group delay is constant. Butterworth filters are IIR, so the group delay varies. In fact, Figure 9.33d illustrates that this system's group delay can change as much as four samples over the discrete frequency domain.

9.5.4 Chebyshev

Suppose a signal analysis application needs a low-pass filter with an especially sharp cutoff frequency, but tolerates some passband ripple. The Butterworth condition began by hypothesizing flatness in both the pass- and stopbands, so we need to relax one of these constraints. The first type of *Chebyshev*[13] filter approximation achieves a sharper transition than the Butterworth, but it does so at the cost of allowing passband ripple. On the other hand, the stopband is flat, and the designer can easily reduce the ripple to any positive value.

9.5.4.1 Chebyshev Polynomials and Equiripple Conditions. For the filter stopband to be maximally flat as $\Omega \to \infty$, any (rational) analog low-pass filter will have a squared Fourier magnitude response with a constant numerator (9.83). If we hope to improve upon the Butterworth filter's sharpness, we have to relax the Butterworth contraints. By the logic of the argument for maximally flat filters in Section 9.5.3.1, we must allow more a_{2k} to be nonzero. Let $\varepsilon^2 T(\Omega) = A(\Omega) - 1$ be the nonconstant part of the denominator in (9.83), where $\varepsilon > 0$ is a parameter controlling the passband ripple height. A low-pass filter with unity gain requires a denominator near unity when $\Omega = 0$, so let us stipulate that $|T(\Omega)| \leq 1$ for $|\Omega| \leq 1$. Then a suitable ε can always make $\varepsilon^2 T(\Omega)$ small for Ω near zero. Since $T(\Omega)$ is still a polynomial, its magnitude will get arbitrarily large as $|\Omega| \to \infty$, and so away from the origin, $P(\Omega) \to 0$. Are there such polynomials?

Indeed approximation theory provides us with precisely such polynomials. We set $T(\Omega) = T_N(\Omega)$, where $T_N(\Omega)$ is the *Chebyshev polynomial* of order $N \geq 0$ [43, 65]. These are defined recursively as follows: $T_0(\Omega) = 1$, $T_1(\Omega) = \Omega$, and

$$T_{N+1}(\Omega) = 2\Omega T_N(\Omega) - T_{N-1}(\Omega). \tag{9.105}$$

The Chebyshev polynomials have nice properties, which are explored in the exercises and exploited by our filter designs. It can be shown that

$$T_N(\Omega) = \cos(N\cos^{-1}(\Omega)), \tag{9.106a}$$

[13]In addition to his work on orthogonal functions, Russian mathematician Pafnuty Lvovich Chebyshev (1821–1894) proved Bertrand's conjecture: For $n > 3$, there is at least one prime number between n and $2n - 2$.

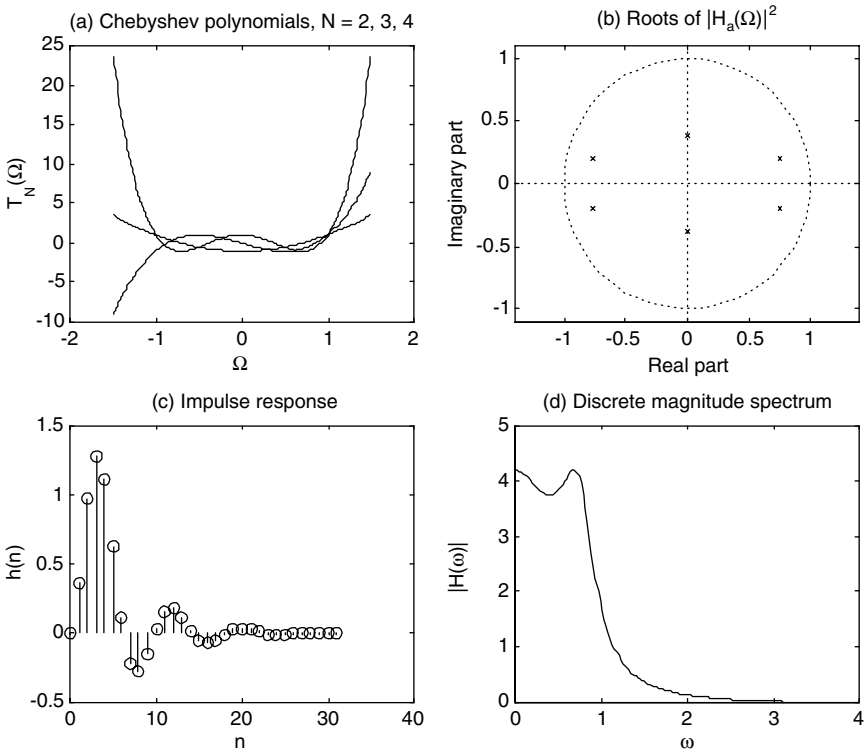

Fig. 9.34. Chebyshev polynomials for a few orders (a). Roots of the Fourier squared magnitude response for a third-order low-pass filter (b). Corresponding discrete impulse response (c) and magnitude spectrum (d).

so that indeed $|T_N(\Omega)| \leq 1$ for $|\Omega| \leq 1$. Furthermore, the polynomials are orthogonal on $[-1, 1]$. Chebyshev polynomials are also given for $|\Omega| > 1$ by the relation

$$T_N(\Omega) \;=\; \cosh(N \cosh^{-1}(\Omega)) \,, \tag{9.106b}$$

as shown in Ref. 65 (see Figure 9.34).

Importantly for the filter approximation problem:

- If $1 \leq N$ and $\Omega \in [-1, 1]$, then $T_N(\Omega)$ oscillates between -1 and $+1$.
- $T_N(\Omega)$ always achieves the minimum of -1 and the maximum of $+1$ on $[-1, 1]$ (for this reason it is called the *equal ripple approximation*).
- $T_N(1) = 1$ for all N.
- For $|\Omega| > 1$, $T_N(\Omega)$ is strictly increasing or strictly decreasing.

Thus, the *Chebyshev squared magnitude response* is defined by

$$\left|H_a(\Omega)\right|^2 \;=\; \frac{1}{A(\Omega)} \;=\; \frac{1}{1 + \varepsilon^2 T_N^2(\Omega/\Omega_c)} \,. \tag{9.107}$$

9.5.4.2 Impulse Invariance Approximation. For the Chebyshev low-pass filter approximation using the impulse invariance transformation, we are given a discrete passband frequency ω_p, a stopband frequency ω_s, an allowable passband ripple, and a required stopband attenuation value. We seek the analog cutoff frequency Ω_c, the filter order N, and the ripple parameter ε (9.107).

The Chebyshev polynomials' properties simplify the approximation. $|T_N(\Omega/\Omega_c)|$ is strictly increasing for $\Omega > \Omega_c$, so we can set $\Omega_c = \omega_p$. If the passband constraint is $10\log_{10}\left|H(\omega_p)\right|^2 \geq \Delta$, for some $\Delta < 0$ as in (9.86a), then the maximum departure from unity will occur for some $-\Omega_c \leq \Omega \leq \Omega_c$. Thus, we need $\varepsilon \geq \sqrt{10^{-\Delta/10} - 1}$. Here our design assumes that *ripple* is defined by the passband peak-to-valley difference. But some treatments assume that the ripple is half of this value—how far the passband magnitude strays from its mean [26]. So readers should be aware of the differences this assumption can make in the final design specifications. Finally, suppose the stopband specification (9.86b) is $10\log_{10}\left|H(\omega_s)\right|^2 \leq \Lambda$, for some $\Lambda < 0$, and the sampling rate is sufficiently high so that in the stopband, $|\omega| > \omega_s$,

$$\left|H(\omega)\right|^2 \approx \left|H_a(\Omega)\right|^2 = \frac{1}{1 + [\varepsilon T_N(\Omega/\Omega_c)]^2}. \tag{9.108}$$

Thus, we seek $N > 0$ such that

$$\left|H_a(\omega_s)\right|^2 = 10\log_{10}[1 + [\varepsilon T_N(\omega_s/\Omega_c)]^2]^{-1} \leq \Lambda. \tag{9.109}$$

We solve (9.109) as an equality for N,

$$N = \frac{\cosh^{-1}\left(\frac{\sqrt{10^{-\Lambda/10} - 1}}{\varepsilon}\right)}{\cosh\left(\frac{\omega_s}{\Omega_c}\right)}, \tag{9.110}$$

and round upward to the nearest integer.

Example (Chebyshev Impulse Invariance Approximation). Suppose we try the Chebyshev approximation on a filter with the same specifications as in the Butterworth impulse invariance design (Section 9.5.3.2). We need a passband within 1 dB of unity for $\omega < \omega_p = \pi/4$. This means $\Delta = -1$, $\Omega_c = \pi/4$, and $\varepsilon = 0.5088$. We need a stopband that is 5 dB or more below unity for $\omega > \omega_s = \pi/3$. We have $\Lambda = -5$, and (9.110) gives $N = 2.1662$, which we round up to $N = 3$. Thus, the Chebyshev approximation gives a third-order IIR filter, whereas the Butterworth approximation needed a fourth-order system—a benefit from allowing passband ripple.

Example (Chebyshev Impulse Invariance design, $N = 3$). Let us continue the previous example: $N = 3$, $\Omega_c = \pi/4$, and $\varepsilon = 0.5088$. The poles of $|H_a(\Omega)|^2$ are $-0.7587 \pm 0.1941j$, $0.7587 \pm 0.1941j$, and $\pm 0.3881j$. They lie on an ellipse in the complex plane [26] as shown in (Figure 9.34b). To find $H_a(\Omega)$, we select the three poles with positive imaginary parts. (Equivalently, for the Laplace transform-based filter derivation, these are $-0.1941 \pm 0.7587j$ and -0.3881.) Figure 9.34c shows the resulting impulse response. Figure 9.34d shows the discrete magnitude spectrum.

9.5.4.3 Bilinear Approximation. In a bilinear transformation, $\Omega_c = 2\tan\left(\frac{\omega_p}{2}\right)$ gives the cutoff frequency. The frequency mapping does not alter the passband ripple, so we can calculate ε just as with impulse invariance. For the filter order, the stopband condition says

$$10\log_{10}\left[\left(1 + \left[\varepsilon T_N\left(\frac{2}{\Omega_c}\tan\left(\frac{\omega_s}{2}\right)\right)\right]^2\right)^{-1}\right] \leq \Lambda\,, \tag{9.111}$$

which implies

$$N \geq \frac{\cosh^{-1}\left(\frac{\sqrt{10^{-\Lambda/10} - 1}}{\varepsilon}\right)}{\cosh^{-1}\left(\frac{2}{\Omega_c}\tan\left(\frac{\omega_s}{2}\right)\right)}\,. \tag{9.112}$$

Example (Chebyshev Bilinear Approximation). Let us turn to the bilinear transformation method for the same filter design problem as above: $\omega_p = \pi/4$, $\omega_s = \pi/3$, $\Delta = -1$, $\Lambda = -5$, with sampling interval $T = 1$. Thus, $\Omega_c = 2\tan\left(\frac{\omega_p}{2}\right) = 0.8284$, using the bilinear frequency mapping. The ripple factor is $\varepsilon = 0.5088$. Solving (9.112) as an equality gives $N \approx 2.0018$, but we opt for the integral ceiling, setting $N = 3$.

Note that for the present design criteria, the Chebyshev filter comes very close to reducing the required filter order to $N = 2$. In fact, unless the application parameters are unusually rigid, this is an attractive possibility. The lesson is twofold:

- For the same filter order the Chebyshev filter has faster (sharper) rolloff than the equivalent Butterworth filter.
- It is possible to achieve the same rolloff as the equivalent Butterworth filter using a Chebyshev filter with a smaller order.

For a quicker rolloff, the Chebyshev suffers some passband ripple. Its transient response is also worse than the Butterworth, as shown below (Section 9.5.4.5).

Example (Chebyshev Bilinear Design, $N = 3$). Let us continue the above approximation: $N = 3$, $\varepsilon = 0.5088$, and $\Omega_c = 0.8284$. The poles of $|H_a(\Omega)|^2$ are $-0.8003 \pm 0.2047j$, $0.8003 \pm 0.2047j$, and $\pm 0.4094j$ (Figure 9.35a). The Laplace transform poles of choice are therefore $-0.2047 \pm 0.8003j$ and -0.4094. The analog system function is

$$H_{L,a}(s) = \frac{1}{s^3 + 0.8188s^2 + 0.8499s + 0.2793}. \tag{9.113}$$

Inserting the bilinear map $s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$ with $T = 1$ into (9.113) gives

$$H(z) = \frac{0.0754z^3 + 0.2263z^2 + 0.2263z + 0.0754}{z^3 - 1.8664z^2 + 1.4986z - 0.4637}. \tag{9.114}$$

$H(z)$ has partial fractions expansion

$$\frac{B(z)}{A(z)} = \frac{-0.6458 - 0.1170\text{j}}{1 - (0.6031 + 0.5819\text{j})z^{-1}} + \frac{-0.6458 + 0.1170\text{j}}{1 - (0.6031 - 0.5819\text{j})z^{-1}} + \frac{1.5297}{1 - (0.6602)z^{-1}} - 0.1627. \tag{9.115}$$

Figure 9.35 shows the impulse (b) and magnitude responses (c).

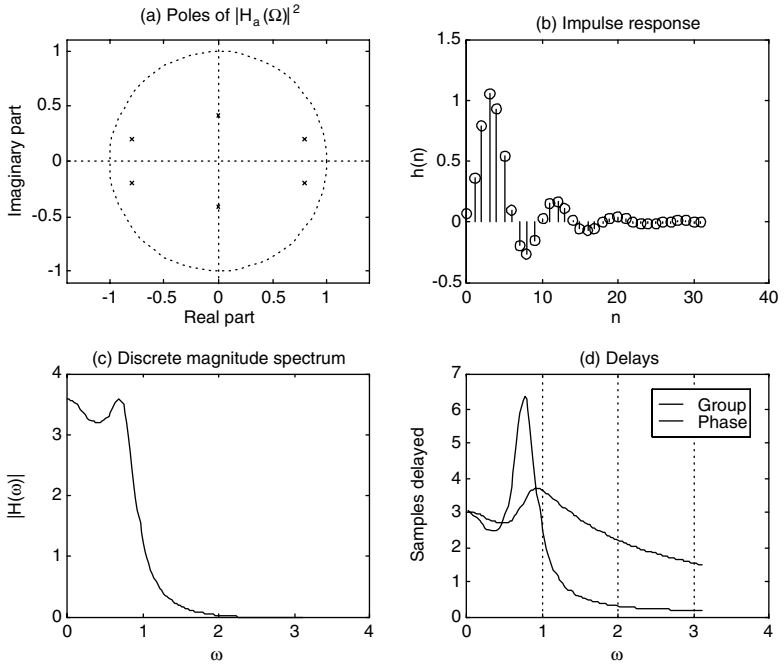

Fig. 9.35. Chebyshev low-pass filter, bilinear transformation. Pole locations (a), unnormalized discrete impulse response (b), magnitude response (c) and phase and group delay (d).

9.5.4.4 Phase and Group Delay of IIR Filters. Let us compare the group delay of Butterworth and Chebyshev filters. Figure 9.35d shows that the group delay of the Chebyshev filter can be as much as six samples for $\omega \in [0, \pi]$—worse than the equivalent Butterworth design Figure 9.33d. This is one thing the Chebyshev filter gives up in order to improve its rolloff performance.

9.5.4.5 Application: Transient Response Comparison. The Chebyshev filter's better rolloff, compared to the maximally flat filter, also costs it some transient response performance. To see this, let us consider a basic transient filtering application. Observe first that the Chebyshev filter exhibits a sharper rolloff Figure 9.36a.

We apply the Butterworth and Chebyshev low-pass filters ($N = 3$) developed in previous sections to the transient Figure 9.36b. The Chebyshev filter produces a longer delay and the ringing induced by the step edge persists for a longer time interval than with the Butterworth. On the other hand, the pulse's later sloped edge provokes only a little bad behavior from the Chebyshev filter Figure 9.36c.

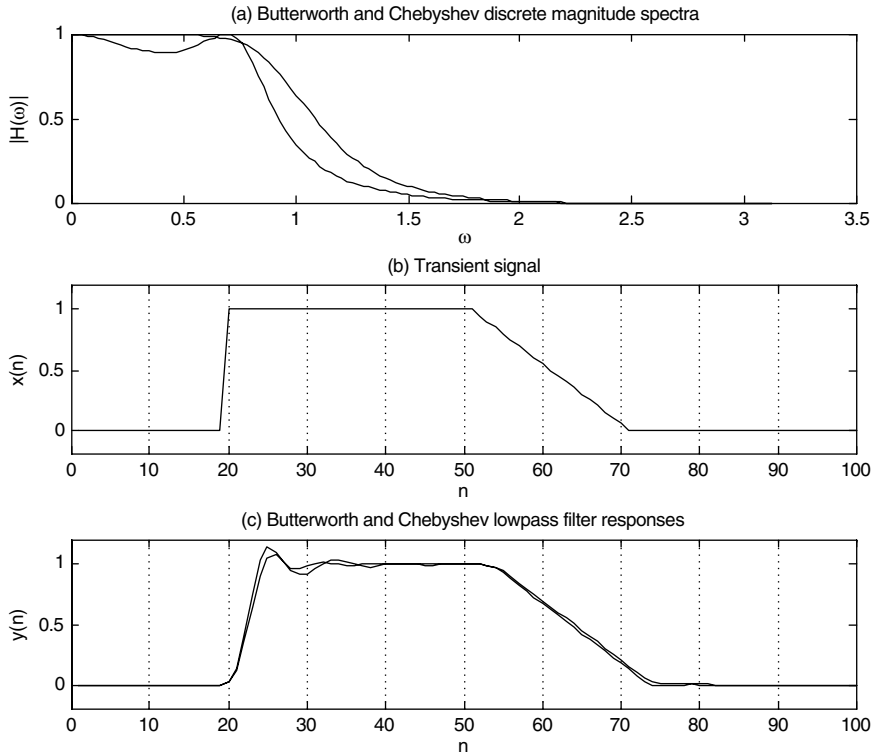

Fig. 9.36. Butterworth and Chebyshev comparison. Magnitude spectra (a), transient signal (b), and Butterworth and Chebyshev responses (c).

The following points should be taken into consideration when choosing the Chebyshev over the Butterworth filter for a signal analysis application:

- For less computational overhead with the same frequency discrimination performance, prefer the Chebyshev.
- If the frequencies in the filter passband will be further characterized by their relative strength, then the ripple in the Chebyshev becomes a detriment.
- If the application does not further analyze passband frequencies and is concerned with their mere presence or absence (such as in the DTMF application in Section 9.1) then the Chebyshev should be better.
- This is moreover the case when the application needs to segment relatively close bands of spectral information and sharp rolloff becomes a priority.
- Finally, if the time location of edges is important, and the application needs a crisp response from a crisp input edge to satisfactorily identify and locate the transition, then the Butterworth filter is superior.

9.5.5 Inverse Chebyshev

The inverse Chebyshev filter provides a flat passband and an equiripple stopband. This filter is also called the *Chebyshev Type II* filter.

9.5.5.1 Stopband Equiripple Conditions. The specification of the filter's squared magnitude response is based on the Chebyshev filter of Section 9.5.4 [26, 65]. The steps (Figure 9.37a) are as follows.

(i) We begin with the Chebyshev squared magnitude response function

$$\left|H_a(\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\Omega/\Omega_c)} = P(\Omega)\,, \tag{9.116}$$

where $T_N(\Omega)$ is the order-N Chebyshev polynomial.

(ii) Subtract this response from unity, $1 - P(\Omega)$, to form the squared magnitude response of a high-pass filter having a flat passband and equiripple in the stopband.

(iii) Reverse the frequency axis to find $Q(\Omega) = 1 - P(\Omega^{-1})$:

$$Q(\Omega) = 1 - \frac{1}{1+\varepsilon^2 T_N^2(\Omega_c/\Omega)} = \frac{\varepsilon^2 T_N^2(\Omega_c/\Omega)}{1+\varepsilon^2 T_N^2(\Omega_c/\Omega)}\,, \tag{9.117}$$

which is the squared magnitude response of a low-pass filter. It has a maximally flat passband and puts the Chebyshev equiripple characteristic in the stopband.

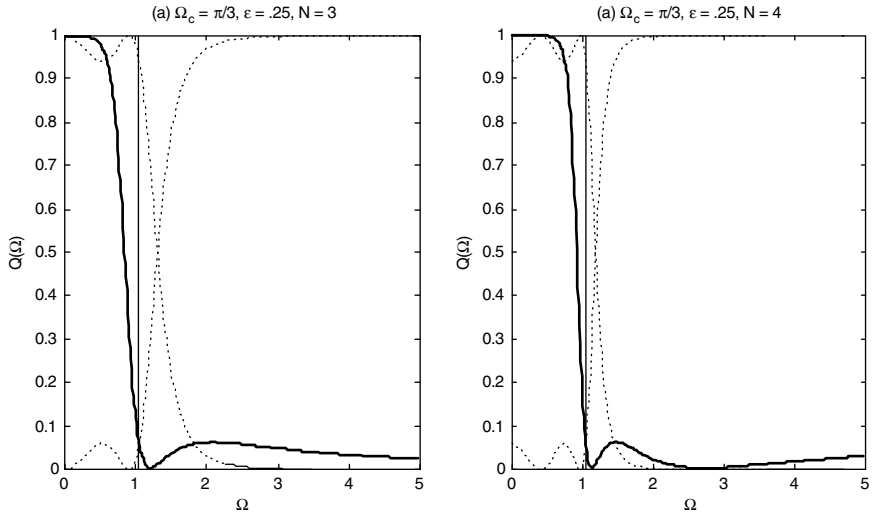

Fig. 9.37. Conversion from a Chebyshev to a third-order Inverse Chebyshev squared magnitude response function (a). The dotted traces are intermediate steps. The vertical line is Ω_c from (9.117) and represents the stopband frequency. For comparison, panel (b) shows an $N = 4$ squared magnitude response.

Note that $Q(\Omega)$ is a rational function in Ω^{-1}, but we can write it equivalently as a rational function of Ω. The poles of $Q(\Omega)$ are the reciprocals of the poles of $P(\Omega)$ (exercise).

9.5.5.2 Impulse Invariance Approximation. Consider the impulse invariance approximation for the inverse Chebyshev low-pass filter. Suppose the discrete passband frequency is ω_p, the stopband frequency is ω_s, the passband is within $\Delta < 0$ (dB) of unity, and the stopband ripple (dB) does not exceed $\Lambda < \Delta < 0$. We need the analog stopband frequency Ω_c, the filter order N, and the ripple parameter ε (9.117).

As remarked above, the parameter Ω_c in (9.117) specifies the analog stopband. So, although for an ordinary Chebyshev filter we took $\Omega_c = \omega_p$, now we set $\Omega_c = \omega_s$. The stopband condition is $10\log_{10}Q(\Omega) \le \Lambda$ for $\Omega \ge \Omega_c$. But as long as $\Omega \ge \Omega_c$ we have $\Omega_c/\Omega \le 1$ and so $1 + \varepsilon^2 T_N^2(\Omega_c/\Omega) \le 1 + \varepsilon^2$. Hence, we can determine the stopband ripple factor ε from the stopband condition with $\Omega = \Omega_c$: $10\log_{10}Q(\Omega_c) \le \Lambda$. Using the elementary Chebyshev polynomial property, $T_N(1) = 1$ for all $N \ge 0$, this reduces to

$$\varepsilon \le \sqrt{\frac{10^{\Lambda/10}}{1 - 10^{\Lambda/10}}}. \tag{9.118}$$

Observe that the ripple parameter does not depend on the filter order. Typically, we solve (9.118) as an equality to obtain ε.

The passband condition gives the analog filter order. If the discrete sampling rate is high enough, this means that for $\Omega < \omega_p < \omega_s = \Omega_c$ we can assume $10\log_{10}Q(\Omega) \geq \Delta$. for $\Omega < \omega_p$. $Q(\Omega)$ is strictly increasing as $\Omega \to 0$. Thus, we know that the passband condition applied to $\Omega = \omega_p$ is a worst case. The usual algebra boils this down to

$$\cosh\left[N\cosh^{-1}\left(\frac{\Omega_c}{\omega_p}\right)\right] = T_N\left(\frac{\Omega_c}{\omega_p}\right) \geq \frac{1}{\varepsilon}\sqrt{\frac{10^{\Delta/10}}{1-10^{\Delta/10}}}. \tag{9.119}$$

The familiar steps of taking (9.119) as an equality, solving for N, and rounding upward give the filter order.

Example (Inverse Chebyshev, Impulse Invariance, $N = 3$). We require a low-pass filter with a passband within 1 dB of unity up to $\omega_p = \pi/4$ and at least 5 dB below unity beyond $\omega_s = \pi/3$. Again, we see $\Delta = -1$ and $\Lambda = -5$ and assume T = 1. As above, we set $\Omega_c = \omega_s = \pi/3$. From (9.118) we find $\varepsilon = 0.680055$. Solving (9.119) as an equality produces $N = 2.1662$. This value we round up to $N = 3$. Since $T_3^2(\theta) = 16\theta^6 - 24\theta^4 + 9\theta^2$, it must be the case that

$$Q(\Omega) = \frac{16{\Omega_c}^6 - 24{\Omega_c}^4\Omega^2 + 9{\Omega_c}^2\Omega^4}{16{\Omega_c}^6 - 24{\Omega_c}^4\Omega^2 + 9{\Omega_c}^2\Omega^4 + \dfrac{\Omega^6}{\varepsilon^2}}, \tag{9.120}$$

where we have expressed $Q(\Omega)$ in positive powers of Ω (Figure 9.38a).

9.5.5.3 Poles and Zeros Analysis. Unlike the others we have already considered, this filter has finite zeros in the extended complex plane. For $H_a(\Omega)$ we select the poles of $Q(\Omega)$ which have positive imaginary parts. In general, among the zeros of a squared magnitude response function, we select one each of the numerator's conjugate roots.

Example (Inverse Chebyshev, Impulse Invariance, $N = 3$). Let us continue with the poles and zeros analysis of the previous example, for which the squared magnitude response is given by (9.117). The poles of $Q(\Omega)$ are $\pm 2.5848j$, -1.0581 ± 0.2477j, and 1.0581 ± 0.2477j. The selected Laplace transform poles are thus -2.5848, $-0.2477 - 1.0581j$, and $-0.2477 + 1.0581j$. The zeros of $Q(\Omega)$ are $-1.1885 \pm 0.1276j$ and 1.1885 ± 0.1276j. They all have the same magnitude, so we can choose Laplace transform zeros to be $-0.1276 - 1.1885$j, and $-0.1276 + 1.1885$j. The poles and zeros plot for $Q(\Omega)$ is shown in Figure 9.38b.

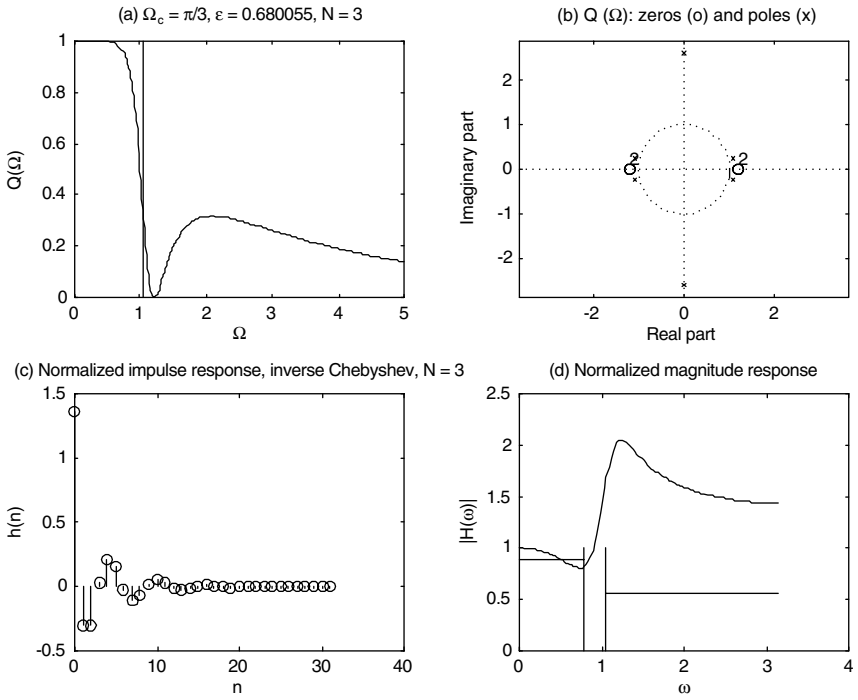

Fig. 9.38. The squared magnitude response $Q(\Omega)$ for an $N = 3$ analog filter approximated using the impulse invariance method (a). Associated poles and zeros (b), extracted discrete impulse response (c), and magnitude response (d). In panel (d), vertical lines mark the discrete passband and stopband, $\pi/4$ and $\pi/3$, respectively. The horizontal lines are the level criteria associated with the Δ and Λ parameters.

As a quotient of polynomials in z^{-1}, the discrete system function $H(z)$ is thus given by

$$\frac{\sum_{m=0}^{M} b_m z^{-m}}{\left[1 + \sum_{k=1}^{N} a_k z^{-k}\right]} = \frac{1 - 0.7624z^{-1} + 0.6826z^{-2}}{1 + (-0.8412)z^{-1} + 0.6671z^{-2} + (-0.0460)z^{-3}}. \qquad (9.121)$$

Figure 9.38(c) shows the impulse response. Note that it consists of an impulse and some low magnitude correction terms.

The magnitude response Figure 9.38d shows the effect of the rather loose stopband ripple constraint for this example. The filter sharpness is adequate, as can be seen from the bounds in the figure. However, there are high frequencies present in the analog filter, because of the allowed stopband ripple. Because of impulse invariance sampling, aliasing occurs and the derived discrete magnitude response does

not satisfy the required stopband criterion. It might appear that increasing the sampling rate should reduce the aliasing. But unfortunately this causes the cutoff frequency of the analog filter to increase as well [7]. The best choice is to adjust the design parameters so as to meet the discrete filter's stopband criterion. The filter order may be increased, the ripple parameter may be reduced, or the bilinear approximation may be worthwhile.

9.5.5.4 Bilinear Approximation. Let us consider the bilinear approximation for the inverse Chebyshev low-pass filter. Suppose the discrete passband frequency is ω_p, the stopband frequency is ω_s, the passband is within $\Delta < 0$ (dB) of unity, and the stopband ripple (dB) does not exceed $\Lambda < \Delta < 0$. We need the analog stopband frequency Ω_c, the filter order N, and the ripple parameter ε (9.117) in order to specify the analog filter.

We know from our study of the inverse Chebyshev squared magnitude resonse that the Ω_c parameter governs the analog stopband frequency. Thus, $\Omega_c = \frac{2}{T}\tan\left(\frac{\omega_s}{2}\right)$ using the bilinear transformation. The passband and stopband conditions on the desired discrete filter are

$$0 > 10\log_{10}|H(\omega)|^2 = 10\log_{10}\left|Q\left(\frac{2}{T}\tan\left(\frac{\omega}{2}\right)\right)\right|^2 \geq \Delta\,, \tag{9.122a}$$

for $\omega < \omega_p$, and

$$10\log_{10}|H(\omega)|^2 = 10\log_{10}\left|Q\left(\frac{2}{T}\tan\left(\frac{\omega}{2}\right)\right)\right|^2 \leq \Lambda < 0\,, \tag{9.122b}$$

for $\omega > \omega_s$, where $Q(\Omega)$ is the analog squared magnitude response (9.117). By Chebyshev polynomial properties, $Q(\Omega)$ achieves its stopband maximum at the analog stopband value $\Omega = \Omega_c$. From (9.122b) we have $\varepsilon \leq \sqrt{\frac{10^{\Lambda/10}}{1-10^{\Lambda/10}}}$, as in the impulse invariance approximation (9.118). We assume an equality to compute the ripple parameter. In the passband constraint (9.122a) we use $\Omega = \Omega_p = \frac{2}{T}\tan\left(\frac{\omega_p}{2}\right)$ as a worst case. This entails

$$T_N(\Omega_c/\Omega_p) = \cosh^{-1}[N\cosh(\Omega_c/\Omega_p)] \geq \frac{1}{\varepsilon}\sqrt{\frac{10^{\Lambda/10}}{1-10^{\Lambda/10}}}\,. \tag{9.123}$$

Changing this relation to an equality, solving for N, and rounding to the integral ceiling gives the filter order.

Example (Inverse Chebyshev, Bilinear Approximation, $N = 3$). For a low-pass filter with a passband within 1 dB of unity for $\omega < \omega_p = \pi/4$ and at least 5 dB of attenuation for $\omega > \omega_s = \pi/3$, we again have $\Delta = -1$, $\Lambda = -5$. Let $T = 1$ be the sample

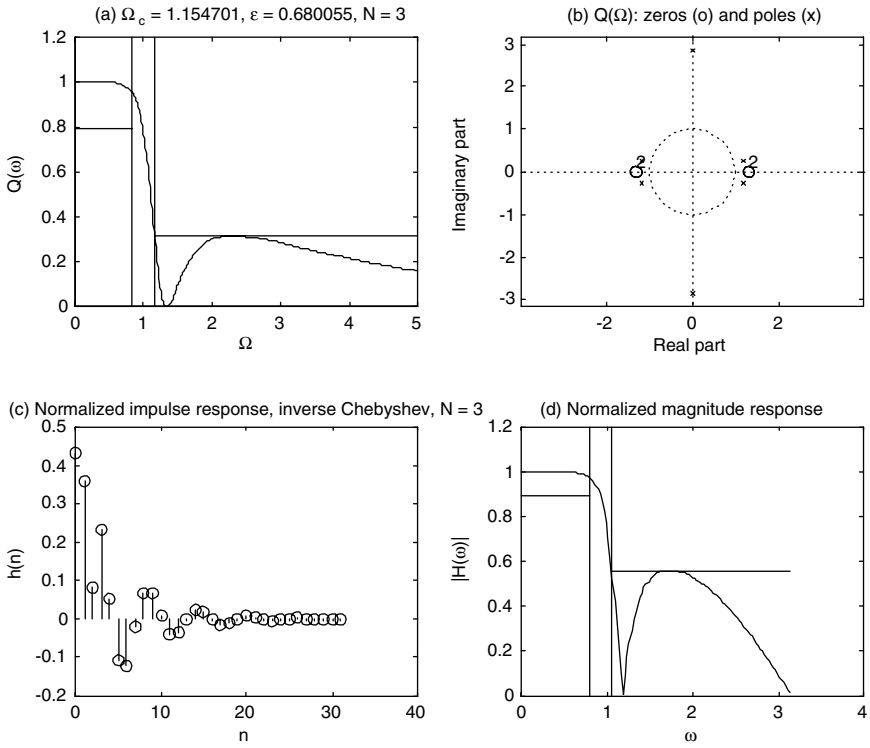

Fig. 9.39. The squared magnitude response $Q(\Omega)$ for an $N = 3$ analog filter approximated using the bilinear transformation method (a). Panel (b) shows the associated poles and zeros. Note that the zeros are second order. In (c), after normalization, is the impulse response. Finally, there is the normalized magnitude response (d). In panel (d), vertical lines mark the discrete passband and stopband, Ω_p and Ω_c, respectively. The horizontal lines are the level criteria associated with the Δ and Λ parameters.

distance. We find $\Omega_c = 2\tan(\omega_s/2) \approx 1.154701$ and $\Omega_p \approx 2\tan(\omega_p/2) \approx 0.8284$. Again, from (9.118) we get $\varepsilon = 0.680055$. Solving (9.123) as an equality produces $N = 2.0018$, and although we are breathtakingly close to a second order filter, we prudently round up to $N = 3$. The three estimates, Ω_c, ε, and N, give the squared magnitude response $Q(\Omega)$ (Figure 9.39a).

9.5.5.5 Poles and Zeros Analysis. Among the poles of $Q(\Omega)$, we select those having positive imaginary parts to form $H_a(\Omega)$. After bilinear transformation, this filter too has finite zeros in the extended complex plane.

Example (Inverse Chebyshev, Bilinear Transformation, $N = 3$). Let us wrap up the previous example. In the squared magnitude response (9.117). The poles of $Q(\Omega)$ are $\pm 2.8657j$, $-1.1817 \pm 0.2550j$, and $1.1817 \pm 0.2550j$. The good Laplace transform poles are thus -2.8657, $-0.2550 - 1.1817j$, and $-0.2550 + 1.1817j$. The

zeros of $Q(\Omega)$ are ± 1.3333 and each has order two. The Laplace transform zeros must be $\pm 1.3333j$. Figure 9.39b shows the poles and zeros plot for $Q(\Omega)$. Thus,

$$H_{L,\,a}(s) \;=\; \frac{s^2 + 1.7778}{s^3 + 3.3757s^2 + 2.9228s + 4.1881}\,. \tag{9.124}$$

Applying the bilinear map $s \;=\; \frac{2}{T}[(z-1)/(z+1)]$ with $T = 1$ to (9.124) gives

$$H(z) \;=\; \frac{0.1832z^3 + 0.0423z^2 + 0.0423z + 0.1832}{z^3 - 0.6054z^2 + 0.5459z + 0.1219}\,. \tag{9.125}$$

The z-transform $H(z)$ has partial fractions expansion

$$\begin{aligned} H(z) \;=\; & \frac{-0.0821 - 0.0082j}{1 - (0.3917 + 0.7293j)z^{-1}} + \frac{-0.0821 + 0.0082j}{1 - (0.3917 - 0.7293j)z^{-1}} \\ & + \frac{-1.1551}{1 - (-0.1779)z^{-1}} + 1.5026 \;=\; \frac{B(z)}{A(z)}. \end{aligned} \tag{9.126}$$

Feeding a discrete impulse $\delta(n)$ through the difference equation implementation for (9.126) gives the unnormalized impulse response. This we scale (Figure 9.39c) by a factor of $(0.4245)^{-1}$ so that the magnitude response has unit DC value, as shown in Figure 9.39d.

9.5.6 Elliptic Filters

The fourth common transfer function—the *elliptical*, or *Cauer*[14] (1958), filter—has ripple in both the passband and stopband, nonlinear phase response, and the fastest rolloff from passband to stopband for a given IIR filter order [26, 66, 67].

The squared magnitude response function for the elliptic or Cauer filter[14] is

$$\left|H_a(\Omega)\right|^2 \;=\; \frac{1}{1 + \varepsilon^2 R^2(\Omega)} \;=\; \frac{B(\Omega)}{A(\Omega)}, \tag{9.127}$$

where $R(\Omega) = U(\Omega)/V(\Omega)$ is a rational function, $\varepsilon > 0$ is a parameter, $R(0) = 0$, and the Degree(U) > Degree(V).

The rational function approximations (9.127) for the elliptic filter response are derived from the analytic properties of the Jacobi elliptic functions, which are encountered in the study of nonlinear oscillations. This oscillatory behavior gives rise to the passband and stopband ripple associated with the elliptic filter transfer function. Under certain conditions, the elliptic functions are qualitatively similar to

[14]German circuit theorist Wilhelm Cauer (1900–1945) invented and patented elliptic filters in the mid-1930s. While on his way to his office to get some papers, Cauer was arrested and executed by troops taking control of Berlin at the end of World War II. A short note on Cauer's life and accomplishments is given by A. Fettweis, Fifty years since Wilhelm Cauer's death, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, vol. 42, no. 4, pp. 193–194, April 1995.

the trigonometric sine and cosine. But they have advanced general features which, when properly manipulated, give rise to better rolloff characteristics—for a given filter order—than the Chebyshev filter. Since the reader may have only minimal exposure to the elliptic functions, we will describe the analytical background for Cauer's elliptic filter prescription prior to developing the rational function approximations implied by (9.127).

9.5.6.1 Elliptic Functions and Integrals. Just as the Chebyshev filter response had an analytic description in terms of the circular functions (e.g., (9.106a)), the general elliptic filter response,

$$\left|H_a(\Omega)\right|^2 = \frac{1}{1+\varepsilon^2 R_N^{\ 2}(\Omega)}, \tag{9.128}$$

can be described in terms of a class of analytic functions known as the *Jacobi elliptic sine*, designated $sn(z, m)$, where the parameter $0 \le m < 1$ is the modulus, and $z \equiv u + jv$ is the argument, which may be complex-valued. The Cauer design calls for a function of the form

$$R_N(\Omega) = sn(f \cdot z + c, m), \tag{9.129}$$

where the argument consists of a factor f; a constant additve offset c—both of which can be specified to give the desired filter response; and a variable z that is the inverse of a Jacobi elliptic sine,

$$z = sn^{-1}(\Omega/\Omega_c, m). \tag{9.130}$$

The *Jacobi*[15] *elliptic sine* of modulus m is defined by

$$sn(z, m) = \sin(\phi(z, m)). \tag{9.131}$$

The argument u is described by the *elliptic integral of the first kind*:

$$u(\phi, m) = \int_0^{\phi} \frac{1}{\sqrt{(1 - m^2 \sin^2\theta)}}\, d\theta, \tag{9.132}$$

where the modulus is restricted to the interval $0 \le m < 1$. The function $\phi(u, m)$ the inverse of the $u(\phi, m)$, for fixed modulus m, and when we refer to a specific value we denote ϕ as the amplitude of the elliptic integral [33]. For the special case of amplitude $\phi = \pi/2$, the elliptic integral of the first kind is a function only of the modulus and reduces to the *complete elliptic integral of the first kind* (Figure 9.40),

$$K(m) = \int_0^{\pi/2} \frac{1}{\sqrt{(1 - m^2 \sin^2\theta)}}\, d\theta. \tag{9.133}$$

[15]Carl Gustav Jacobi (1804–1851), along with Gauss and Legendre, contributed to the early theory.

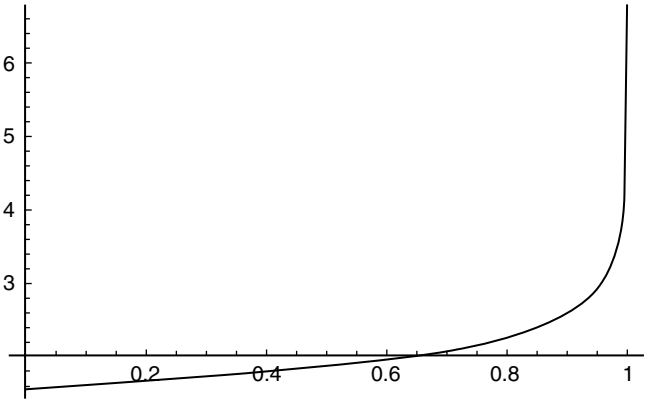

Fig. 9.40. The complete elliptic integral as a function of the modulus m. It is real-valued, but becomes singular as $m \to \infty$.

The complement to this integral is defined

$$K'(m) = \int_0^{\pi/2} \frac{1}{\sqrt{(1 - m_1{}^2 \sin^2\theta)}}\, d\theta, \tag{9.134}$$

where $m_1 = 1 - m$. (We follow the usual convention and denote the complement by a prime, but emphasize that it has nothing to do with differentiation.) From (9.134) it is obvious that

$$K(m_1) = K'(m) \tag{9.135}$$

and we will use these interchangeably.

In the real-valued interval $x \in [0, \infty]$, the Jacobi elliptic sine is qualitatively similar to a sine wave: It is real-valued, restricted in amplitude to the interval $[-1, 1]$, and exhibits oscillations which qualitatively resemble a pure sinusoid, as illustrated in Figure 9.41. As this illustration suggests, $K(m)$ is one-fourth of a full period of the elliptic sine. For this reason it is also known as the *real quarter period*.

The Jacobi elliptic sine exhibits a richness that surpasses the simpler pure sinusoid. The most important new property is double periodicity,

$$sn(z + r \cdot 4K + s \cdot 4K', m) = sn(z, m), \tag{9.136}$$

where r and s are arbitrary integers. In Figure 9.42 and Figure 9.43 we illustrate the elliptic sine along other important intervals of the complex plane.

The validity of the Cauer's construction of the elliptic filter response is dependent upon the value of the Jacobi elliptic sine at several strategic points in the complex plane. For convenience, these points are listed in table Table 9.6. This table applies to both even- and odd-order elliptic filters.

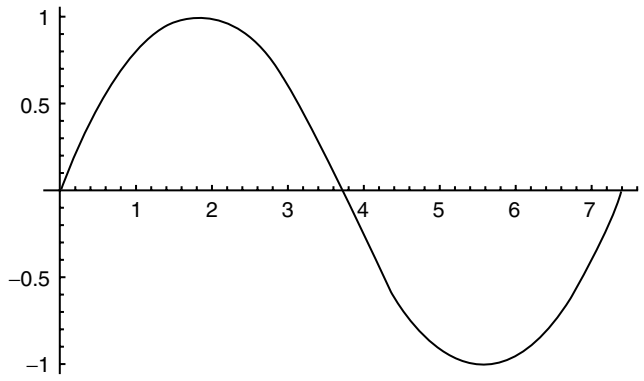

Fig. 9.41. The Jacobi elliptic sine $sn(x, m)$ on the interval $x \in [0, 4K]$. The case $m = 0.5$ is shown.

The foregoing illustrations were selected because they will aid in the understanding of Cauer's design of the elliptic filter response. Note that along these chosen intervals, $sn(z; m)$ is real-valued, although the argument itself may acquire nonzero real and imaginary parts. Although excursions from the intervals selected here may result in generally complex values for $sn(z; m)$, Cauer's design conveniently limits us to these intervals in which the elliptic sine remains real-valued. With suitable manipulation of the free constants f and c in (9.129), we can ensure continuity of the response at the transition points between the pass- and stopbands. The third column in Table 9.6 gives the points in the frequency plane at which the conditions in the first two columns are applied. These issues are considered in the next section.

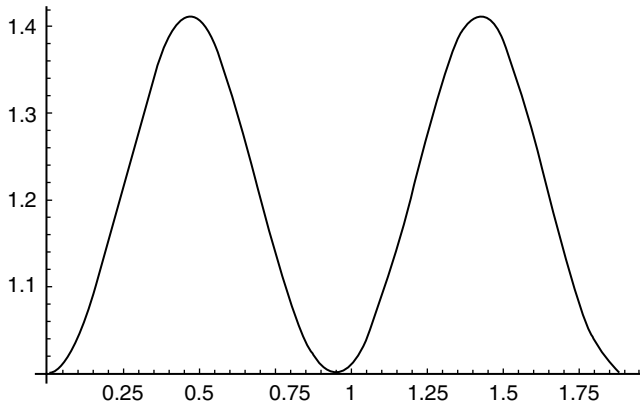

Fig. 9.42. The Jacobi elliptic sine $sn(K + jy, m)$ on the interval $y \in [0, 4K]$. The case $m = 0.5$ is shown.

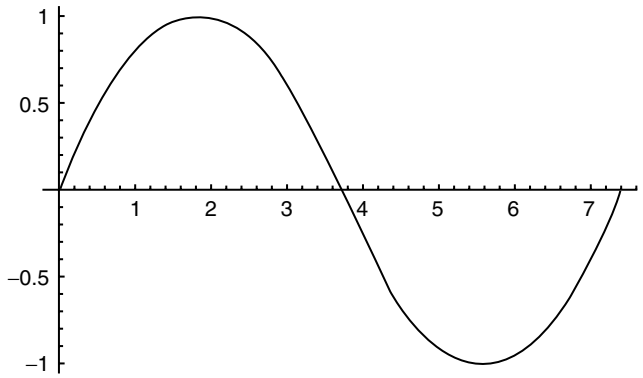

Fig. 9.43. The Jacobi elliptic sine $sn(x + 4jK', m)$ on the interval $x \in [0, 4K]$, shown for $m = 0.5$.

9.5.6.2 Elliptic Filter Response. For the elliptic low-pass filter approximation using the impulse invariance transformation, we are given a discrete passband frequency Ω_p, a stopband frequency Ω_s, an allowable passband ripple, and a required stopband attenuation value. We seek the filter order N and the ripple parameter ε. As in the case of the Chebyshev response, it is not possible to specify a specific set of parameters $\{\Omega_p, \Omega_s, N, \varepsilon\}$ that are identically satisfied by a low-pass elliptic filter response, but an allowable upper bound on the passband ripple and a lower bound on the stopband attenuation can be achieved, with both pass- and stopbands exhibiting a flat response.

The design of an elliptic filter response involves the specification of parameters m, p, and integer N such that these specific acceptable bounds can be achieved. At the same time, the filter response must be continuous at $\Omega = \Omega_p$ and $\Omega = \Omega_s$ while also satisfying acceptable bounds at $\Omega = 0$ and in the limit $\Omega \to \infty$. Continuity can be achieved by proper specification of the constants f and c (see (9.129)), while the proper bounds arise naturally from the behavior of the elliptic sine. The following relations are imposed by the design process and hold for arbitrary positive integer N. First,

$$m = \Omega_p^2 / \Omega_s^2 \tag{9.137}$$

TABLE 9.6. The Argument z and the associated value of the Jacobi Elliptic Sine as the Frequency Ω Traverses the Passband, Transition Band, and stopband of a Cauer Elliptic Filter[a]

z	$sn(z;m)$	Ω
0	0	0
$\beta K(m))$	1	Ω_p
$\beta K(m) \pm j\gamma K'(m)$	$m^{-1/2}$	Ω_s
$\alpha K(m) \pm j\gamma K'(m)$	∞	∞

[a]The integer α is even, the integers β and γ are odd.

which ensures that $m \in [0, 1]$, as required by the elliptic sine. Second, we define

$$f \equiv N\frac{K(p)}{K(m)} = \frac{K(p_1)}{K(m_1)}. \tag{9.138}$$

The second equality in (9.138) is imposed (it is not an identity) and gives the *flatness condition*,

$$N = \frac{K(m)}{K(p)}\frac{K(p_1)}{K(m_1)}. \tag{9.139}$$

Since parameter m is specified by the filter rolloff in (9.137), and the order N of the filter is typically specified in advance, relation (9.139) amounts to a default specification of the unknown parameter p. We note that (9.139) must be satisfied to give identically flat pass- and stopband ripple. In practice, deviations from an integer lead to good flatness provided they are small, and we will find it necessary to finesse the pass- and stopband ripple levels a_1 and a_2 to achieve something close to (9.139).

The value of remaining unknown, namely the additive offset c, depends on whether the filter is of odd or even order. We will now consider these cases in turn. For *odd* filter order N, the design process will result in a filter response having the following general characteristics $\left|H(\Omega/\Omega_p)\right|^2$ at selected critical frequencies:

$$\left|H(0)\right|^2 = 1, \tag{9.140}$$

$$\left|H(1)\right|^2 = \frac{1}{1+a_1^2}, \tag{9.141}$$

$$\left|H(\Omega_s/\Omega_p)\right|^2 = \frac{1}{1+a_2^2}, \tag{9.142}$$

$$\left|H(\infty)\right|^2 \to 0. \tag{9.143}$$

For filters of an *even* order N, the DC value of the response will be

$$\left|H(0)\right|^2 = \frac{1}{1+a_1^2} \tag{9.144}$$

but the other three points are the same as given in (9.141)–(9.143). In an actual design problem, the real-valued parameters a_1 and a_2 will specified according to the desired acceptable ripple and will be adjusted so as to leave the ripple within specified bounds while also providing a flat response in the pass- and stopbands.

Remark. Readers consulting further references on elliptic filter design may encounter alternative design procedures which result in a filter response with asymptotic

behavior which deviates from that specified in (9.140) and (9.143), especially when consulting prepackaged tables or routines. Unless otherwise noted, we confine ourselves to the limits defined here.

Case of N Odd. For an elliptic filter of odd order we stipulate

$$c = 0. \tag{9.145}$$

Consider the conditions at the stopband $\Omega = \Omega_s$, as laid out in Table 9.6. Expressing (9.29), we have

$$R_N(\Omega_s/\Omega_p) = sn\left(N\frac{K(p)}{K(m)} \cdot (K(m) + jK(m_1)), m\right), \tag{9.146}$$

where we have also used expression (9.135). After straightforward algebra this reduces to

$$R_N(\Omega_s/\Omega_p) = sn(NK(p) + jK(p_1), p) = 1/\sqrt{p}. \tag{9.147}$$

Similarly, the passband edge at $\Omega = \Omega_p$ leads to

$$R_N(1) = sn(NK(p), p) = 1. \tag{9.148}$$

Substitution of (9.147) and (9.148) into the expression for the filter characteristics leads to the relations,

$$\left|H(\Omega_s/\Omega_p)\right|^2 = \frac{1}{1 + \left(\frac{\varepsilon}{p}\right)} = \frac{1}{1 + a_2^2} \tag{9.149}$$

and

$$\left|H(1)\right|^2 = \frac{1}{1 + \varepsilon} = \frac{1}{1 + a_1^2}. \tag{9.150}$$

Combining these lead to expressions for ε and the parameter p in terms of the pass- and stopband ripple:

$$\varepsilon = a_1^2, \tag{9.151}$$

$$p = \frac{a_1^2}{a_2^2}. \tag{9.152}$$

Remark. In the design of an elliptic characteristic for specified Ω_p and Ω_s, the essential relations are (9.139), (9.151), and (9.152). A successful design will involve juggling of a_1 and a_2 (within acceptable bounds) such that p from (9.152) will lead to close agreement with the flatness criterion (9.139). This can be done graphically utilizing packaged math routines, as we will do in the following example.

Example (Cauer Elliptic Filter, $N = 3$). Consider a specification calling for the design of an $N = 3$ elliptic filter with $\Omega_p = 0.9$, $\Omega_s = 1.39$, a maximum passband ripple of −2.0 dB, and a stopband ripple not exceeding −25.0 dB. As a first pass, one can barely make the ripple tolerances by setting $a_1 = 0.763$ and $a_2 = 17.9$, which lead to

$$10 \cdot \log[1/(1 + a_1^2)] = -1.99253, \tag{9.153}$$

$$10 \cdot \log[1/(1 + a_2^2)] = -25.0706. \tag{9.154}$$

However, the "integer" is given by

$$N = \frac{K(p_1)}{K(m_1)} \cdot \frac{K(m)}{K(p)} = 2.68387, \tag{9.155}$$

The effect of the deviation from the ideal value of 3 is to cause a departure from flatness, which is particularly notable in the stopband, as illustrated in Figure 9.44.

The problem can be alleviated by reducing the passband ripple such that $a_1 = 0.445$ and

$$N = \frac{K(p_1)}{K(m_1)} \cdot \frac{K(m)}{K(p)} = 3.00271, \tag{9.156}$$

which is close to the ideal target of 3. The reduction in passband ripple required to achieve this exceeds 1 dB, since

$$10 \cdot \log[1/(1 + a_1^2)] = -0.784659. \tag{9.157}$$

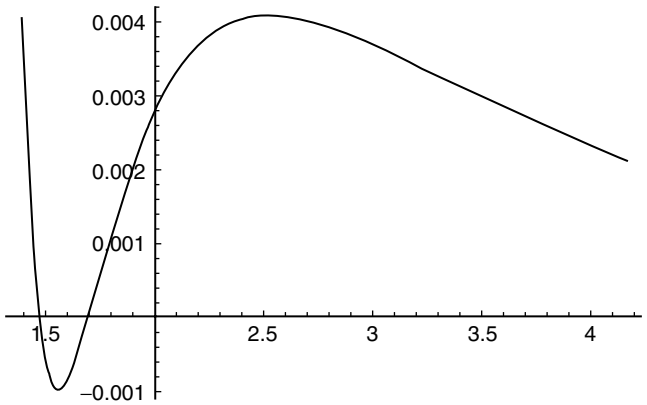

Fig. 9.44. Detail of the stopband response when the integer flatness condition is not met. Note that the characteristic becomes negative in a small region of the spectrum.

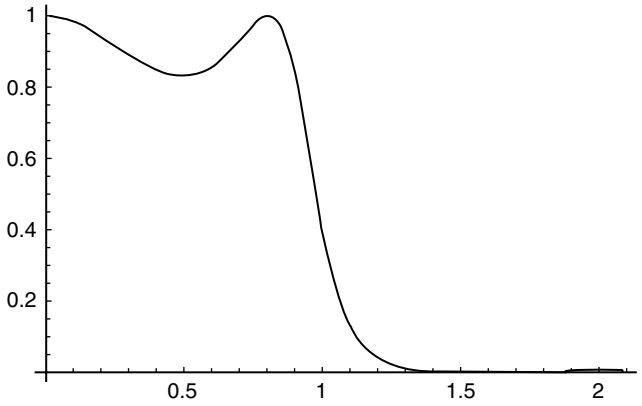

Fig. 9.45. The $N = 3$ elliptic filter response after the parameters have been adjusted to obtain pass- and stopband flatness.

However, the result is the desired flat elliptic filter response representing (9.128), as shown in Figure 9.45.

Remark. In practice, deviations from the integer ideal N have the greatest effect on the stopband response. This includes the flatness, as noted, but even small deviations can induce a small imaginary part to $H(\Omega)$. When setting up a plot, it is useful to specify the real part to eliminate the small but unwanted imaginary component.

Case of *N* Even. For an elliptic filter of even order we apply a nonzero offset to the argument,

$$c = K(p). \tag{9.158}$$

At the passband edge $\Omega = \Omega_p$ it is easy to show, since $N + 1$ is an odd number,

$$R_N(1) = sn((N+1)K(p), p) = 1. \tag{9.159}$$

Likewise, at the stopband edge,

$$R_N(\Omega_s/\Omega_p) = sn((N+1)K(p) + iK(p_1), p) = 1/\sqrt{p}. \tag{9.160}$$

The effect of the nonzero offset (9.158) is to give edge conditions (9.159) and (9.160) identical to their counterparts in the odd-order case. The offset will have the effect of changing the elliptic filter response at zero frequency, but otherwise the even order characteristic resembles that of the odd-order case. These points are illustrated in the following example.

Example (Cauer Elliptic Filter, $N = 4$). Consider the specifications laid out in the example for $N = 3$, but suppose we require faster rolloff by specifying $\Omega_s = 1.12$. By increasing the order of the filter to 4, and setting $a_1 = 0.39$ and $a_2 = 23.9$, we obtain

$$10 \cdot \log[1/(1 + a_1^2)] = -0.614902, \tag{9.161}$$

$$10 \cdot \log[1/(1 + a_2^2)] = -27.5756, \tag{9.162}$$

and

$$N = \frac{K(p_1)}{K(m_1)} \cdot \frac{K(m)}{K(p)} = 4.01417, \tag{9.163}$$

which leads to a nominally flat response shown in Figure 9.46. Note that by increasing the order of the filter we have achieved the desired faster rolloff and brought the stopband ripple under the specification by more than 1.5 dB.

Remark. Note that the even-order elliptic response has the low-frequency limit,

$$|H(0)|^2 = \frac{1}{1 + a_1^2} = 0.86798. \tag{9.164}$$

This outcome is the legacy of the nonzero offset specified in (9.158): At $\Omega = 0$, the argument of the Jacobi elliptic sine is no longer zero, as it was in the odd-order construction.

9.5.7 Application: Optimal Filters

Finally, let us consider the problem of designing a filter that conditions a signal so that later processing preserves just the desired features of the input. To be more precise, suppose that an analog signal $x(t)$ contains an original trend $s(t)$ and an

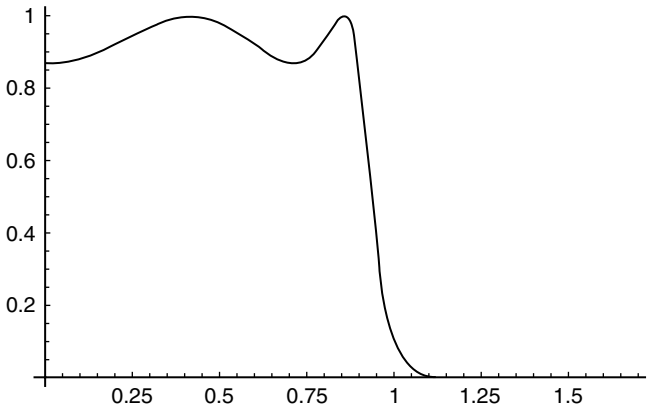

Fig. 9.46. The $N = 4$ elliptic filter response. The result is flat in the pass- and stopbands.

unknown corrupting noise component $n(t)$: $x(t) = s(t) + n(t)$. The signal $x(t)$ passes through a linear, translation-invariant system H, $y(t) = (Hx)(t)$. Assume all signals are real-valued. We seek the best H with real impulse response $h(t)$ so that $(x*h)(t)$ is optimally close to $s(t)$.

Frequency-domain methods provide a solution: the *optimal* or *Wiener*[16] filter [68]. Arguing informally, let us use the L^2 norm as a measure of how close $y(t)$ is to $s(t)$. For an optimal noise removal filter, then, we seek to minimize

$$\|y - s\|^2 = \frac{1}{2\pi}\|Y - S\|^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |S(\omega) - S(\omega)H(\omega) - H(\omega)N(\omega)|^2 \, d\omega, \quad (9.165)$$

where we have used Parseval's result and the convolution theorem for the radial Fourier transform (Chapter 5). The integrand on the right-hand side of (9.165) is

$$\begin{aligned} &S(\omega)\overline{S(\omega)}[1 - H(\omega)]\overline{[1 - H(\omega)]} - \overline{H(\omega)N(\omega)}[1 - H(\omega)]S(\omega) \\ &\quad - H(\omega)N(\omega)\overline{[1 - H(\omega)]S(\omega)} + N(\omega)\overline{N(\omega)}H(\omega)\overline{H(\omega)}. \end{aligned} \quad (9.166)$$

The noise $n(t)$ is random, so it is uncorrelated with $s(t)$, and integrating products involving their respective Fourier transforms gives zero:

$$\|y - s\|^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} (|S(\omega)|^2|1 - H(\omega)|^2 - |H(\omega)N(\omega)|^2) \, d\omega, \quad (9.167)$$

To find the minimum of $\|y - s\|^2$, we must minimize the integral in (9.167). Thus, we must minimize its integrand, and—arguing informally—the criterion for this is that the function $f(H) = |S|^2|1 - H|^2 - |HN|^2$ has zero derivative. Taking the derivative with respect to H gives $\frac{\partial f}{\partial H} = H[|S|^2 + |N|^2] - |S|^2$. Setting $\frac{\partial f}{\partial H} = 0$ and solving gives

$$H = \frac{|S|^2}{|S|^2 + |N|^2}, \quad (9.168)$$

which is the Fourier transform of the optimal or Wiener filter for removing noise from the signal $x(t)$.

9.6 SPECIALIZED FREQUENCY-DOMAIN TECHNIQUES

This section introduces and applies some signal analysis methods arising from Fourier transform theory.

[16]Although it had been developed in 1942, Wiener's optimal filter was made public only in 1949, when the first edition of Ref. 68 was published. The theory had remained classified during World War II, because of its application to radar.

9.6.1 Chirp-*z* Transform Application

In many applications, the frequencies of interest within candidate signals are known in advance of their processing. Such a priori information can simplify the design of the analysis system. For instance, instead of computing a broad range of spectral values using the discrete Fourier transform, the engineer may elect to compute only a small portion of the spectrum, namely that part that might contain useful signal information. One way to focus in on a spectral interval without computing large numbers of useless coefficients is the use the *chirp-z transform* (CZT), introduced in the previous chapter (Section 8.3.1).

Let us first recall the basic ideas of the CZT. The CZT computes z-transform on a spiral contour in the complex plane [11]. It is determined by two parameters: A and W—the spiral starting point and arc step, respectively. Via an example, we shall see how to apply it to zoom in on DFT frequency components. As in Chapter 8, we take the notation of Rabiner and Gold [11]. Suppose that $A = A_0\exp(2\pi j\theta_0)$; $W = W_0\exp(2\pi j\phi_0)$; M, N are positive natural numbers; $x(n) = 0$ outside $[0, N-1]$; and $z_k = AW^{-k}$ for $0 \le k < M$. The chirp z-transform of $x(n)$ with respect to A and W is

$$X_{A,\,W}(k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} = \sum_{n=0}^{N-1} x(n) A^{-n} W^{nk}. \tag{9.169}$$

The exercises of Chapter 8 explained that the CZT reduces to the DFT of order N when $A = 1$, $M = N$, and $W = \exp^{(-2\pi j/N)}$.

Let us return to the speech fragment considered earlier, "calling" in a female voice. The fundamental frequency range of the /a/ phoneme for a woman is from about $F_{lo} = 100$ Hz to $F_{hi} = 400$ Hz. We would like to design a CZT detector for this spectral range and apply it to the digitized speech sample. From a manual segmentation of the speech sample in question (Figure 9.28), we know that the /a/ phoneme occurs from samples $n = 800$ to $n = 1200$. It also degrades off into the /l/ sound immediately following.

To set up the CZT for this formant detection application, we set the sliding disjoint windows to consist of $N = 400$ samples. The sampling frequency $F_s = 8192$ Hz. Also,

$$A = \exp\left[\frac{2\pi j F_{lo}}{F_s}\right], \tag{9.170a}$$

$$W = \exp\left[-2\pi j\left(\frac{F_{hi} - F_{lo}}{NF_s}\right)\right]. \tag{9.170b}$$

Applying the algorithms of Chapter 8, we find the first six detection windows (Figure 9.47).

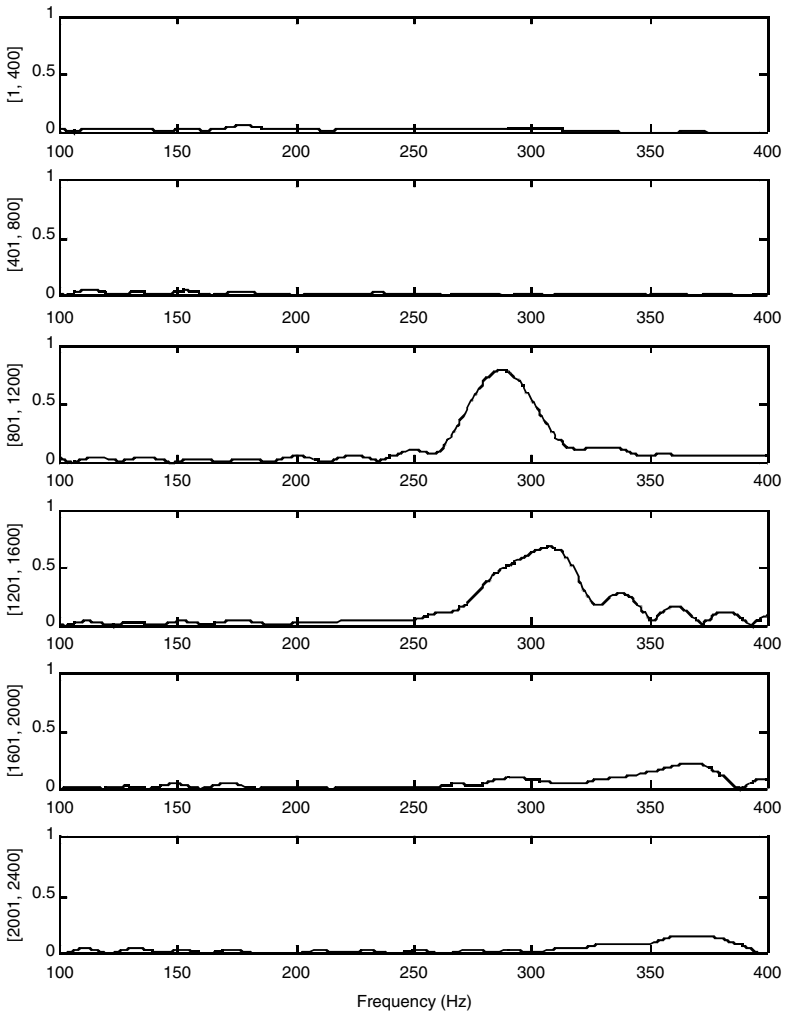

Fig. 9.47. Fundamental frequency detector for the /a/ phoneme based on the chirp-z transform. A lone peak rises in the third window, which concurs with a manual segmentation of the speech sample.

9.6.2 Hilbert Transform

The Hilbert transform[17] is a powerful technique that can be used to:

(i) Find the envelope of a signal.

(ii) Find the instantaneous phase of a signal.

[17]David Hilbert originated the idea in his papers on integral equations, reprinted in the book *Grundzuge einer allgemeinen Theorie der linearen Integralgleichungen*, Leipzig and Berlin: Teubner, 1912.

(iii) Find the instantaneous frequency of a signal.
(iv) Suppress one of the sidebands in order to create a single sideband (SSB) modulation of a signal.

The Hilbert transform has rich theory and many interesting properties [7, 69, 70]. We shall skim the theoretical material and show how the transform works in another speech analysis application.

9.6.2.1 Definition and Properties. There are analog and discrete Hilbert transforms. The *analog Hilbert transform* of a signal $x(t)$ is defined to be

$$x_H(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(s)}{t-s}\, ds = (\mathcal{H}x)(t)\,. \tag{9.171}$$

The integral, due to the singularity of its integrand at $t = s$, must be interpreted in a special way in order to make sense. The standard way to define the integral is by the Cauchy principal value [71]:

$$PV \int_{-\infty}^{\infty} \frac{x(s)}{t-s}\, ds = \lim_{\varepsilon \to 0^+} \left[\int_{\varepsilon}^{\infty} \frac{x(s)}{t-s}\, ds + \int_{-\infty}^{-\varepsilon} \frac{x(s)}{t-s}\, ds \right], \tag{9.172}$$

which is valid as long as the limit of the sum of the two partial integrals exists. The principal value is written with a PV before the integral sign to signify that a special, augmented form of the Lebesgue integral is supposed. Note that the individual limits of the integrals inside the square brackets of (9.172) may not exist. It is in general not permissable to move the limit operation inside the brackets when using the principal value of the integral.

Example (Square Pulse). Consider the signal $x(t) = u(t + 1) - u(t - 1)$. The function $h(t) = t^{-1}$ defies integration on $[-1, 1]$, because of the singularity at the origin. But using the Cauchy principal value, we can still compute $x_H(0)$:

$$x_H(0) = PV \int_{-\infty}^{\infty} \frac{x(s)}{0-s}\, ds = -\lim_{\varepsilon \to 0^+} \left[\int_{\varepsilon}^{1} \frac{1}{s}\, ds + \int_{-1}^{-\varepsilon} \frac{1}{s}\, ds \right] = 0\,. \tag{9.173}$$

Consequently, we can interpret the transform integral as a special kind of convolution. Let us investigate how the Hilbert transform system affects an analog signal. Let $h(t) = (\pi t)^{-1}$. Then $x_H(t) = (x^* h)(t)$. The generalized Fourier transform of $h(t)$ is

$$H(\Omega) = \int_{-\infty}^{\infty} \frac{1}{\pi t} e^{-j\Omega t} dt = -j\,\mathrm{sgn}(\Omega) = \begin{cases} -j & \text{for } \Omega > 0, \\ 0 & \text{for } \Omega = 0, \\ j & \text{for } \Omega < 0, \end{cases} \tag{9.174}$$

The *duality principle* of the Fourier transform explains (9.174). If $x(t)$ has radial Fourier transform $y(\Omega)$, then $y(t)$ will have Fourier transform $2\pi x(-\Omega)$. From Chapter 6, we know that the Fourier transform of $(j/2)\text{sgn}(t)$ is Ω^{-1}. Hence by duality, t^{-1} transforms to $(j/2)(2\pi)\text{sgn}(-\Omega)$, and (9.174) follows.

The Hilbert transform system $x(t) \rightarrow x_H(t)$ is also called a *90° phase shift* or *quadrature filter* [50]. To see why, we look at the system's frequency-domain effect. The Fourier transform of $x_H(t)$ is $X_H(\Omega) = X(\Omega)H(\Omega) = -jX(\Omega)\text{sgn}(\Omega)$. This operation multiplies positive spectral components by the factor $-j = \exp(-j\pi/2)$ and negative spectral components by $j = \exp(j\pi/2)$. These correspond to phase shifts of $-\pi/2$ and $\pi/2$, respectively. Thus, the Hilbert transform converts sines to cosines and vice versa. Let us examine at these basic transformations.

Example (Sinusoids). Consider $x(t) = \cos(\Omega_0 t)$ and $y(t) = \sin(\Omega_0 t)$. The generalized Fourier transforms of $x(t)$ and $y(t)$ are $X(\Omega) = \pi[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)]$ and $Y(\Omega) = (\pi/j)[\delta(\Omega - \Omega_0) - \delta(\Omega + \Omega_0)]$. Note that $\delta(\Omega - \Omega_0)$ is a positive frequency impulse, whereas $\delta(\Omega + \Omega_0)$ lives across the origin, in $\Omega < 0$ land. The Fourier transform of $x_H(t)$ is $X_H(\Omega) = F(x_H)(\Omega) = -j\pi\text{sgn}(\Omega)[\delta(\Omega - \Omega_0) + \delta(\Omega + \Omega_0)]$. But this is $(\pi/j)[\delta(\Omega - \Omega_0)] - (\pi/j)[\delta(\Omega + \Omega_0)] = Y(\Omega)$. Evidently, $x_H(t) = y(t)$. As an exercise, we leave the other relation $y_H(t) = -x(t)$ to the reader.

Typically, then, we compute the Hilbert transform of a signal $x(t)$ by examining the frequency domain product $X(\Omega)H(\Omega)$. This is usually much simpler than evaluating the Cauchy principal value integral (9.172), although the results can be counter-intuitive. The generalized Fourier transform of the signal $x(t) = 1$ is the Dirac $2\pi\delta(\Omega)$, for instance. Multiplication by $-j\text{sgn}(\Omega)$ therefore gives zero.

We summarize analog Hilbert transform properties in Table 9.7 and leave the derivations as exercises. Note that if $X(0) = 0$, then the inverse transform is $\mathcal{H}^{-1} = -\mathcal{H}$. Also, many algebraic properties of the Fourier transform carry through to the Hilbert transform.

9.6.2.2 Discretization. Moving toward computer applications, let us now consider how to define a discrete Hilbert transform. Again, the frequency-domain behavior is the key; we seek a discrete 90° phase shift system. Such a system would turn each cosine component $\cos(\omega n)$ in a signal $x(n)$ into a $\sin(\omega n)$ term and each $\sin(\omega n)$ into a $-\cos(\omega n)$.

First, we consider the case of aperiodic discrete signals $x(n)$. The appropriate 90° phase shift system should have a frequency response $H(\omega)$ given by

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} = -j\,\text{sgn}(\omega) = \begin{cases} -j & \text{for } \omega > 0, \\ 0 & \text{for } \omega = 0, \\ j & \text{for } \omega < 0. \end{cases} \tag{9.175}$$

TABLE 9.7. Some Analog Hilbert Transform Properties

Signal Expression	Hilbert Transform or Property
$x(t)$	$x_H(t) = \frac{1}{\pi} PV \int_{-\infty}^{\infty} \frac{x(s)}{t-s}\, ds = (\mathcal{H}x)(t)$ (Analysis equation)
$x_H(t)$	$(\mathcal{H}x_H)(t) = -x$ (Inverse, synthesis equation)
$ax(t) + by(t)$	$ax_H(t) + by_H(t)$ (Linearity)
dx/dt	dx_H/dt (Derivative)
$\langle x, x_H \rangle = \int_{-\infty}^{\infty} x(t)\overline{x_H(t)}\, dt = 0 \;\; x \in L^2(\mathbb{R})$	Orthogonality
$\|x\|_2 = \|x_H\|_2 \;\; x \in L^2(\mathbb{R})$	Energy conservation

The inverse DTFT computation gives

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega = \frac{j}{2\pi}\int_{-\pi}^{0} e^{j\omega n}d\omega + \frac{-j}{2\pi}\int_{0}^{\pi} e^{j\omega n}d\omega = \begin{cases} 0 & \text{if } n \text{ is even,} \\ \dfrac{2}{n\pi} & \text{if } n \text{ is odd.} \end{cases} \tag{9.176}$$

So the above discrete Hilbert transform system is neither causal nor FIR. Now let us consider a discrete $x(n)$ with period $N > 0$.

Let $X(k)$ be the DFT of a real-valued signal $x(n)$ defined on $[0, N-1]$. So corresponding to each positive discrete frequency $k \in [1, N/2)$ there is a negative frequency $N - k \in (N/2, N-1]$. The DFT coefficients $X(0)$ and $X(N/2)$—corresponding the DC and Nyquist frequency values—are both real. Mimicking the analog Hilbert transform, let us therefore define the system function of the discrete Hilbert transform to be

$$H(k) = \begin{cases} 0 & \text{if } k = 0, \\ -j & \text{if } 1 \le k < \dfrac{N}{2}, \\ 0 & \text{if } k = \mathrm{N}/2, \\ j & \text{if } \dfrac{N}{2} < k \le N-1. \end{cases} \tag{9.177}$$

We claim this works. For if $x(n) = A\cos(2\pi k_0 n/N)$, then its representation in terms of the inverse DFT is $x(n) = (A/2)\exp[2\pi j k_0 n/N] + (A/2)\exp[2\pi j(N - k_0)n/N]$. That is,

$X(k) = (A/2)\delta(k - k_0) + (A/2)\delta(k - (N - k_0))$. Multiplying by $H(k)$ gives $X(k)H(k) = (-jA/2)\delta(k - k_0) + (jA/2)\delta(k - (N - k_0))$. Applying the inverse DFT, this becomes $y(n) = (-jA/2)\exp[2\pi jk_0n/N] + (jA/2)\exp[2\pi j(N - k_0)n/N] = A\sin(2\pi k_0n/N)$. So $y(n) = x_H(n)$, as claimed. Similarly, discrete Hilbert transformation of $A\sin(2\pi k_0n/N)$ gives $-A\cos(2\pi k_0n/N)$.

Note that the Hilbert transform $x_H(n)$ of a discrete signal $x(n)$ on $[0, N - 1]$ loses the energy of both the DC term and the Nyquist frequency term. To find the impulse response of the discrete Hilbert transform system, we calculate the inverse DFT of (9.177) to get $h(n)$. This allows us to implement discrete Hilbert transforms on a digital computer. However, the value of Hilbert transform applications revolves around the related concept of the analytic signal, which the next section covers.

9.6.2.3 Analytic Signal. Given an analog signal $x(t)$ and its Hilbert transform $x_H(t)$, the associated *analytic signal* [72, 73] is

$$x_A(t) = x(t) + jx_H(t)\,. \tag{9.178a}$$

Although replacing a real-valued with a complex-valued signal may make things seem needlessly complicated, it does allow us to define the following related—and quite valuable—concepts. The *signal envelope* is

$$\left|x_A(t)\right| = \sqrt{x^2(t) + x_H^2(t)}\,. \tag{9.178b}$$

Thus, we can write the analytic signal as

$$x_A(t) = \left|x_A(t)\right|e^{j\phi(t)}, \tag{9.178c}$$

where the *instantaneous phase* $\phi(t)$ is

$$\phi(t) = \tan^{-1}\left[\frac{x_H(t)}{x(t)}\right]. \tag{9.178d}$$

In the first chapter, we argued that the derivative of the phase with respect to time is a reasonable way to define the *instantaneous radial frequency*. Hence, we set

$$\omega(t) = \frac{d}{dt}\phi(t)\,. \tag{9.178e}$$

We may also define discrete versions of these notions. Notice that the signal envelope for a sinusoid is precisely its amplitude. Thus, the definition of signal envelope (9.178b) gives us a definition that applies to aperiodic signals, but reduces to what we should expect for the case of sinusoids. The imaginary part of the analytic signal (9.178a) fills in the gaps, as it were, left in the signal by its fine scale oscillations (Figure 9.48).

In many ways, the analytic signal is more important than the Hilbert transform itself. It is possible to show that the analytic signal satisfies the Cauchy–Riemann

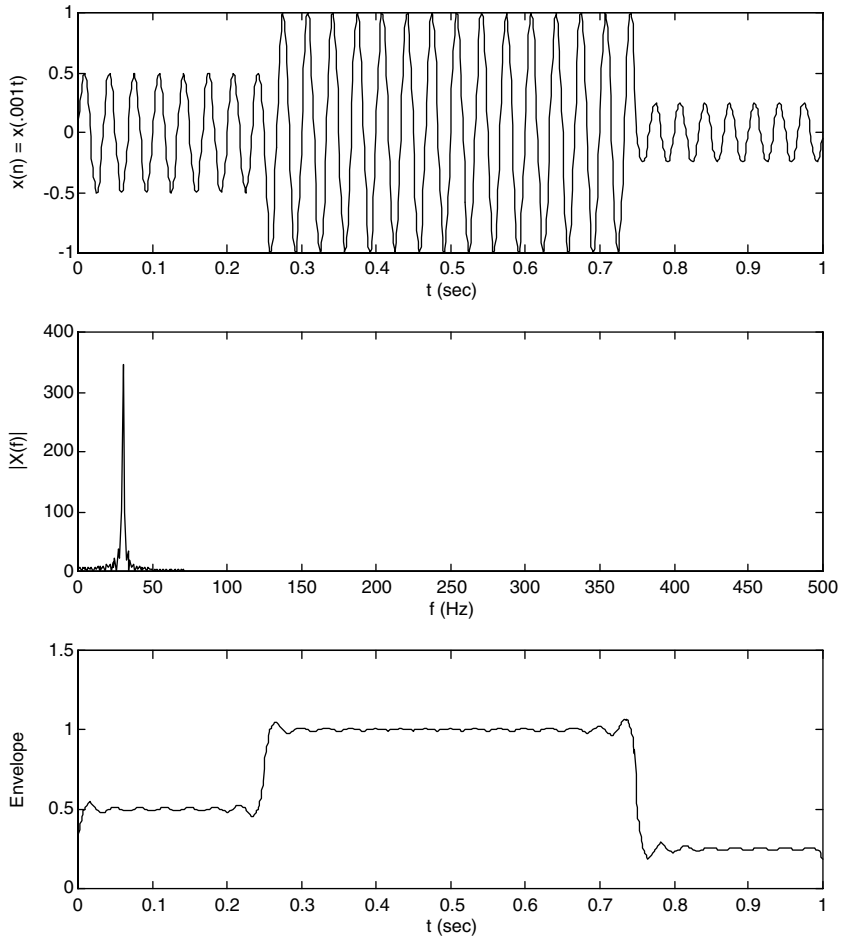

Fig. 9.48. Signal $x(t)$ contains a unit amplitude 30-Hz sinusoid that has been attenuated in the first and last 250 ms of its domain (top). Magnitude spectrum (middle panel). Signal envelope (bottom).

equations , so that it can be extended to an analytic function of a complex variable $x(z)$ [71, 74]. The next section contains an example of envelope computation on a speech signal.

9.6.2.4 Application: Envelope Detection. An important early task in speech analysis is to segment the input signal into regions containing utterances and those holding only background noise. The utterance portions can be further broken up into separate words, although this is by no means a simple task. One tool in either segmentation procedure is the signal envelope. Here, as an example, we compute the envelope of the speech signal considered earlier, namely, the "calling" clip.

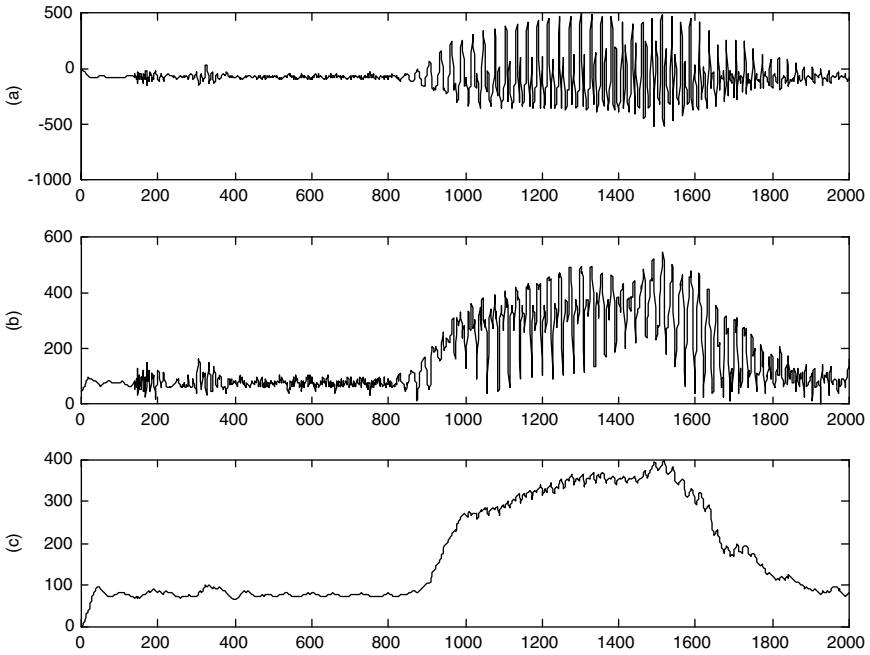

Fig. 9.49. Speech signal (a), its envelope (b), and its filtered envelope (c).

Working on the first 2000 speech samples (Figure 9.49a), we can compute the Hilbert transform, analytic signal, and envelope (Figure 9.49b). The problem is that the envelope remains rather jagged and therefore problematic for segmentation purposes.

One way to improve the envelope is to apply low-pass filter. In Figure 9.49c we created a third order Chebyshev Type II low-pass filter with stopband attenuation of 20 dB. The speech sampling rate is 8 kHz for this example, so the Nyquist rate is 4 kHz, and the cutoff frequency for the low-pass filter is 200 Hz.

In general, envelope detection problems can be improved with such smoothing filters. Even simple sample-and-hold filters with $H(z) = z/(z - a)$ may prove adequate for envelope amelioration.

Signal analysis applications requiring instantaneous phase or frequency computations—such as interferometry, for instance—may demand more refined filtering. A typical strategy is to use filters based on fitting methods, such as the Savitzky–Golay filters to the raw signal envelope before computing the phase and its derivative.

9.6.3 Perfect Reconstruction Filter Banks

In this chapter's first section we considered simple signal analysis problems using an array of filters selective of different frequency ranges. By examining the energy outputs of the separate filters, the frequency content according to time location of signals could be ascertained. This section investigates filter banks more deeply, and,

in particular, takes up the problem of reconstructing the original signal from its separately filtered versions.

Why should this matter? There are two basic reasons:

- If a signal can be broken down into separate components and perfectly (or approximately) reconstructed, then this provides a basis for an efficient signal transmission and compression technology.
- There is also the possibility of constructing signal libraries for detection and interpretation purposes that support a coarse-to-fine recognition methodology but provide a compact library of signal prototypes.

One more involved reason is that a perfect reconstruction filter bank is closely related to a type of time-scale transform, the orthogonal wavelet transformation, which we shall cover in Chapter 11.

9.6.3.1 Laplacian Pyramid. An early and innovative approach combining signal scale and frequency-domain analysis is the Laplacian pyramid decomposition. Constructing hierarchical image decompositions was employed by Ref. 75 in their development of the Laplacian pyramid. They approached the problem of managing the sheer volume of information in a pixel image by making two points: First, the gray-scale pixel values are highly correlated in natural scenes; second, it is possible to decompose the original image into both a coarse representation which contains the gross features of the image and a difference image which contains sufficient information to reconstruct the original image from the coarse representation. Their objective was to remove the correlations that typically exist between neighboring pixels in natural scenes. This is a primary goal of image compression.

Burt and Adelson used a discrete filter, which in certain instances closely resembles a Gaussian, to derive the coarse images. The filtered representations are subsampled at twice the unit distance of the previous image to obtain new levels in the pyramid. The authors call this the Gaussian pyramid. This process for one-dimensional signals, passes the original signal at resolution level 0, $f(n)$, given by the digitizer, to the first coarser level of the Gaussian pyramid. The filter coefficients, $w(n)$, are chosen to have an approximately Gaussian shape by Burt and Adelson, although the technical conditions the authors impose on the $w(n)$ allow some quite different filters to arise [75].

To extract a difference signal from two successive layers of the Gaussian pyramid, Burt and Adelson began by inserting zeros between the values of the coarse pyramid level. This is necessary because the coarser level contains pixels whose unit of size is twice that of the finer level. The addition of zero elements causes extra high frequency components to be added to the signal when this up-sampling operation is performed. This requires a second smoothing operation. The new smoothed signal, the values of which are now taken at unit intervals, can be subtracted from the original signal. Figure 9.50 illustrates these operations on the signals. The coarse images are obtained by Gaussian-like filtering and the difference images are

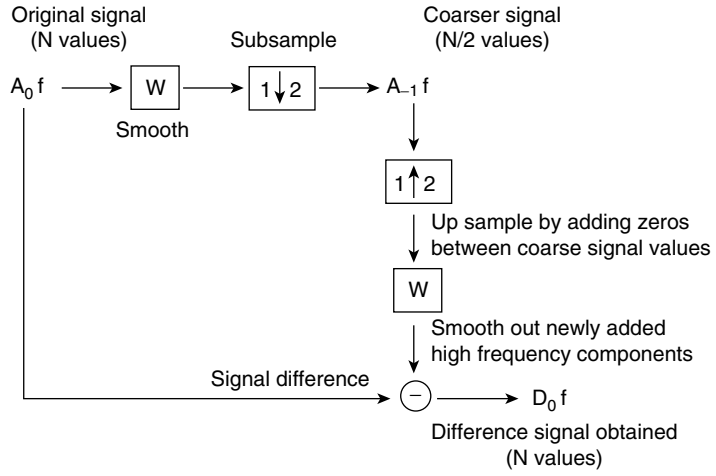

Fig. 9.50. Signal processing diagram of the Laplacian pyramid decomposition.

obtained by subtracting such filtered representations. Since the difference of two Gaussians so closely resembles the Laplacian of a Gaussian operation studied by Marr [77], Burt and Adelson called their construction the Laplacian pyramid.

The signal operations of Figure 9.50 may be repeated. Successive difference signals are produced together with a final coarse, or approximate, signal. The Laplacian pyramid, then, consists of $D_0, D_{-1}, D_{-2}, ..., D_{-J}, A_{-J}$. As is evident from the simple decomposition procedure shown in Figure 9.50, the finer resolution layers of the pyramid may be recovered from the appropriate difference and coarse signals.

When a one-dimensional signal with N samples is hierarchically analyzed with a Laplacian pyramid, the number of coefficients required increases to approximately $2N$. This is evident from the diagram (Figure 9.50).

The Laplacian pyramid provides a scale-based signal recognition strategy. Notice that quasi-Gaussian filters of identical shape applied at differing scales and basic arithmetic operations between the levels of the pyramid to decompose and

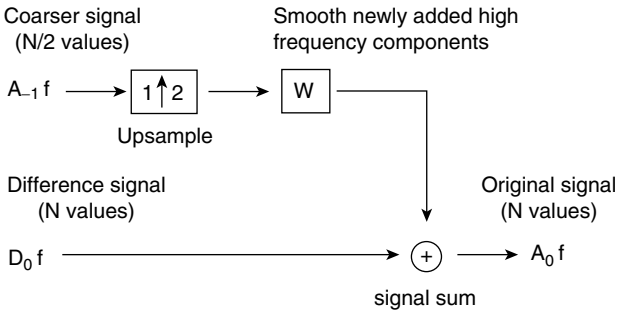

Fig. 9.51. Laplacian pyramid reconstruction of signal.

reconstruct the original image. The computational strategies of the Laplacian pyramid have been implemented in a series of special-purpose vision machines [76].

9.6.3.2 Exact Reconstruction Filter Banks. There is a type of pyramid decomposition that allows perfect signal reconstruction but does not increase the amount of data required for storing the decomposition components. Of course, this is an ideal feature for signal compression applications. But it can also be applied for progressive signal transmission as well as for coarse-to-fine recognition applications. Here we introduce the theory. We shall have occasion to refer back to it when we cover wavelet transforms in the last two chapters.

Consider a real-valued signal $a_0(n)$. The decomposition and reconstruction scheme resembles the Laplacian pyramid's signal flow (Figure 9.52). For the present and the sequel (Chapter 11, in particular) it is useful to set up the following notations. Given the filter impulse responses $h(n)$ and $g(n)$, we set $\overline{h}(n) = h(-n)$ and $\overline{g}(n) = g(-n)$ to be their reflections in time. Observe that $\overline{H}(\omega) = H^*(\omega)$.

Also, we have $a_1(n) = (a_0 * \overline{h})(2n)$, which is the original signal convolved with $h(-n)$ and subsampled. Similarly, we have $d_1(n) = (a_0 * \overline{g})(2n)$. Typically, we shall select $h(n)$ to be a low-pass filter and $g(n)$ to be a high-pass filter. In terms of the classic paper [78], $a_1(n)$ and $d_1(n)$ are the first-level approximate and detail signals, respectively, in the decomposition of source signal $a_0(n)$. Furthermore, we obtain $a_1'(n)$ by upsampling $a_1(n)$ and then $b_1'(n)$ by filtering with $h'(n)$. Similarly, $c_1'(n)$ comes from filtering $d_1'(n)$ with $g'(n)$, where $d_1'(n)$ is an upsampled version of $d_1(n)$. Finally, $a_0'(n) = (a_1' * h')(n) + (d_1' * g')(n)$. We seek conditions that will guarantee $a_0(n) = a_0'(n)$.

Let us note some properties of the upsampling and downsampling operations. First, let $x(n)$ be a discrete signal and $y(n) = x(2n)$. Then

$$Y(2\omega) = \sum_{n=-\infty}^{\infty} x(2n)e^{-2jn\omega} = \frac{X(\omega) + X(\omega + \pi)}{2}. \tag{9.179}$$

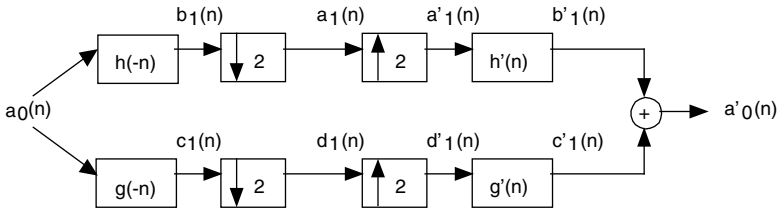

Fig. 9.52. Decomposition and reconstruction signal paths. With the proper choice of filters, the original signal can be recovered and yet its decomposition does not increase the amount of data to be stored.

The second term on the right in (9.179) is called the *folding* or *aliasing* term. This is what wrecks the desired exact reconstruction. Next, if $y(n)$ comes from $x(n)$ by upsampling and inserting zeros,

$$y(n) = \begin{cases} x(m) & \text{if } n = 2m, \\ 0 & \text{if otherwise.} \end{cases} \tag{9.180}$$

then $Y(\omega) = X(2\omega)$. We leave the straightforward proofs as exercises.
The next result is due to Ref. 79.

Theorem. Let $h(n)$ and $g(n)$ be real-valued impulse responses for discrete filters. Then the decomposition and reconstruction filter bank (Figure 9.52) performs perfect reconstruction if and only if

(i) $H^*(\omega + \pi)H'(\omega) + G^*(\omega + \pi)G'(\omega) = 0$ and
(ii) $H^*(\omega)H'(\omega) + G^*(\omega)G'(\omega) = 2$.

Proof: Referring to Figure 9.52, let us calculate the frequency response of the reconstructed signal $a_0'(n)$. We know that $B_1(\omega) = A_0(\omega)\overline{H}(\omega)$ and $C_1(\omega) = A_0(\omega)\overline{G}(\omega)$. Thus,

$$\begin{aligned} 2A_1(2\omega) &= B_1(\omega) + B_1(\omega + \pi) = A_0(\omega)\overline{H}(\omega) + A_0(\omega + \pi)\overline{H}(\omega + \pi) \\ &= A_0(\omega)H^*(\omega) + A_0(\omega + \pi)H^*(\omega + \pi) \end{aligned} \tag{9.181a}$$

and similarly,

$$2D_1(2\omega) = A_0(\omega)G^*(\omega) + A_0(\omega + \pi)G^*(\omega + \pi)\,. \tag{9.181b}$$

On the reconstruction side of the diagram, we have

$$A_0'(\omega) = B_1'(\omega) + C_1'(\omega) = A_1(2\omega)H'(\omega) + D_1(2\omega)G'(\omega)\,. \tag{9.182}$$

Substituting (9.181a) and (9.181b) into (9.182) and simplifying leads to

$$\begin{aligned} A_0'(\omega) = {} & \frac{1}{2}A_0(\omega)[H^*(\omega)H'(\omega) + G^*(\omega)G'(\omega)] \\ & + \frac{1}{2}A_0(\omega + \pi)[H^*(\omega + \pi)H'(\omega) + G^*(\omega + \pi)G'\omega]. \end{aligned} \tag{9.183}$$

Inspecting (9.183), we see that the only way that $A_0'(\omega)$ can equal $A_0(\omega)$ is if the nonaliased term is doubled, and the aliased term is zero. These are precisely the conditions (i) and (ii) of the theorem's statement. ■

The theorem gives a necessary and sufficient condition on the reconstruction filters $h'(n)$ and $g'(n)$ so that the decomposition scheme provides exact reconstruction.

Theorem. Let $h(n)$ and $g(n)$ be as in the previous theorem. Then the decomposition and reconstruction scheme (Figure 9.52) performs perfect reconstruction only if

$$H^*(\omega)H'(\omega) + H^*(\omega+\pi)H'(\omega+\pi) = 2\,. \tag{9.184}$$

Proof: After taking complex conjugates, we may write the result of the previous theorem

(iii) $H(\omega+\pi)H'^*(\omega) + G(\omega+\pi)G'^*(\omega) = 0$ and
(iv) $H(\omega)H'^*(\omega) + G(\omega)G'^*(\omega) = 2$.

Let us rewrite these relations in matrix form as follows:

$$\begin{bmatrix} H(\omega) & G(\omega) \\ H(\omega+\pi) & G(\omega+\pi) \end{bmatrix} \begin{bmatrix} H'^*(\omega) \\ G'^*(\omega) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}. \tag{9.185}$$

If the determinant is nonzero, (9.185) can be solved,

$$\begin{bmatrix} H'^*(\omega) \\ G'^*(\omega) \end{bmatrix} = \frac{2}{\Delta(\omega)} \begin{bmatrix} G(\omega+\pi) \\ -H(\omega+\pi) \end{bmatrix}, \tag{9.186}$$

where $\Delta(\omega) = H(\omega)G(\omega+\pi) - G(\omega)H(\omega+\pi)$. Note too that $\Delta(\omega+\pi) = -\Delta(\omega)$. From (9.186) we get

$$H'^*(\omega+\pi) = \frac{2G(\omega+2\pi)}{\Delta(\omega+\pi)} = \frac{-2G(\omega)}{\Delta(\omega)}, \tag{9.187}$$

implying

$$G(\omega)G'^*(\omega) = \frac{\Delta(\omega)H'^*(\omega+\pi)}{-2}\frac{(-2)H(\omega+\pi)}{\Delta(\omega)} = H'^*(\omega+\pi)H(\omega+\pi). \tag{9.188}$$

Using the complex conjugate of (9.188) with (ii) of the previous theorem gives (9.184) above. ■

Corollary. Under the theorem's assumptions,

$$G^*(\omega)G'(\omega) + G^*(\omega+\pi)G'(\omega+\pi) = 2\,. \tag{9.189}$$

Proof: Exercise. ■

Definition (Quadrature Mirror Filters). If the decomposition filter is the same as the reconstruction filter $h(n) = h'(n)$, then it is called a *quadrature mirror filter* (QMF) or *conjugate mirror filter* (CMF).

Corollary. If $h(n)$ is QMF, then

$$|H(\omega)|^2 + |H(\omega+\pi)|^2 = 2\,. \tag{9.190}$$

Proof: From (9.184). ■

9.7 SUMMARY

This chapter explored a variety of frequency domain signal analysis applications, developed several related tools, explained how to construct and implement filters for frequency selective input processing, and studied the theoretical issues that arose in the course of experimentation.

We can divide analysis applications according to whether the source signals are basically narrowband or wideband:

(i) Narrowband signals are interpretable—at least locally—via pulse detection in the Fourier magnitude spectrum.

(ii) Wideband signals, on the other hand, demand more complicated procedures, multicomponent, and shape-based techniques.

For narrow band signals we developed three basic approaches:

(i) Time-domain segmentation, based on the histogram for example, prior to frequency-domain analysis.

(ii) Local windowing using a short (say $N = 256$) DFT in many locations, thus forming a time-frequency map.

(iii) The filter bank.

Either of the first two methods work well under moderate noise. Under heavy noise, or faced with real-time processing and interpretation constraints, a bank of carefully crafted filters, operating in parallel and equipped with decision logic at their output, can provide satisfactory results.

We significantly extended filter bank theory at the end of the chapter. We showed that subsampling combined with low- and high-pass filtering can be used to decompose signals for analysis. Moreover, under certain conditions on the filters, the filter bank supports an exact reconstruction algorithm using upsampling and filtering. We will revisit these ideas in Chapters 11 and 12, showing that there is a link between filter banks and the theory of orthogonal wavelets—a time-scale signal analysis technique.

The concepts of phase and group delay arose in frequency-domain signal analysis applications where discrete filters were necessary. Applications that filter incoming signals for noise removal, frequency selection, or signal shaping and then analyze the output must take into account the delay characteristics of the filter. We observed the phase delay of sinusoidal tones in the DTMF filter bank application, for example. Other applications involve group delay, such as speech analysis and edge detection.

Many applications require that the filters provide linear, or nearly linear, phase. In communication systems, for example, the information is carried on the envelope and the carrier has a constant frequency. Thus, nonlinear phase delay could well stretch and compress the frequencies in the signal so that the distortions render the signal unintelligible. Important aspects of seismic signal processing are to properly

measure the time delays between different signal components. Thus, linear phase is crucial when filtering; and for this reason, finite impulse response (FIR) filters, which we proved to have linear phase, are preferred.

REFERENCES

1. J. S. Walker, *Fourier Analysis*, New York: Oxford University Press, 1988.
2. G. B. Folland, *Fourier Analysis and Its Applications*, Pacific Grove, CA: Wadsworth and Brooks/Cole, 1992.
3. R. E. Edwards, *Fourier Series: A Modern Introduction*, vol. I, New York: Holt, Rinehart and Winston, 1967.
4. T. W. Körner, *Fourier Analysis*, Cambridge: Cambridge University Press, 1988.
5. D. C. Champeney, *A Handbook of Fourier Theorems*, Cambridge: Cambridge University Press, 1987.
6. H. Baher, *Analog and Digital Signal Processing*, New York: Wiley, 1990.
7. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
8. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 2nd ed., New York: Macmillan, 1992.
9. R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Reading, MA: Addison-Wesley, 1987.
10. A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, 2nd ed., Upper Saddle River, NJ: Prentice-Hall, 1997.
11. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
12. P. Mock, Add DTMF generation and decoding to DSP-µP designs, *Electronic Design News*, March 21, 1985; also in K.-S. Lin, ed., *Digital Signal Processing Applications with the TMS320 Family*, vol. 1, Dallas: Texas Instruments, 1989.
13. J. L. Flanagan, *Speech Analysis, Synthesis, and Perception*, 2nd ed., New York: Springer-Verlag, 1972.
14. D. H. Lehmer, Mathematical methods in large-scale computing units, *Proceeding of the 2nd Symposium on Large-Scale Digital Calculating Machines*, Harvard University Press, pp. 141–146, 1951.
15. S. K. Park and K. W. Miller, Random number generators: Good ones are hard to find, *Communications of the ACM*, vol. 31, no. 10, pp. 1192–1201, 1988.
16. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge: Cambridge University Press, 1988.
17. G. E. Box and M. E. Muller, A note on the generation of random normal deviates, *Annals of Mathematical Statistics*, vol. 29, pp. 610–611, 1958.
18. D. E. Knuth, *Seminumerical Algorithms*, 2nd ed., Reading, MA: Addison-Wesley, 1981.
19. L. Devroye, *Non-Uniform Random Variate Generation*, New York: Springer-Verlag, 1986.
20. C. K. Chow and T. Kaneko, Automatic boundary detection of the left ventricle from cineangiograms, *Computers and Biomedical Research*, vol. 5, pp. 388–410, 1972.

21. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-9, no. 1, pp. 62–66, January 1979.
22. J. Kittler and J. Illingworth, On threshold selection using clustering criteria, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 652–655, 1985.
23. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
24. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subband, and Wavelets*, Boston: Academic, 1992.
25. M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Upper Saddle River, NJ: Prentice-Hall, 1995.
26. T. W. Parks and C. S. Burrus, *Dignal Filter Design*, New York: Wiley, 1987.
27. L. W. Couch II, *Digital and Analog Communication Systems*, 4th ed., Upper Saddle River, NJ: Prentice-Hall, 1993.
28. R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra*, New York: Dover, 1958.
29. S. M. Kay, *Modern Spectral Estimation: Theory and Application*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
30. L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*, Reading, MA: Addison-Wesley, 1991.
31. D. G. Childers, ed., *Modern Spectral Analysis*, New York: IEEE Press, 1978.
32. C. Bingham, M. D. Godfrey, and J. W. Tukey, Modern techniques of power spectrum estimation, *IEEE Transactions on Audio and Electroacoustics*, vol. AU-15, pp. 56–66, June 1967. Also in Refs. 31, pp. 6–16.
33. A. Antoniou, *Digital Filters: Analysis and Design*, New York, McGraw-Hill, 1979.
34. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed., New York: McGraw-Hill, 1984.
35. M. S. Bartlett, Smoothing periodograms from time series with continuous spectra, *Nature* (*London*), vol. 161, pp. 686–687, 1948.
36. P. D. Welch, The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, *IEEE Transactions on Audio and Electroacoustics*, vol. AU-15, no. 2, pp. 70–73, 1967. Also in Refs. 31, p. 17–20.
37. R. B. Blackman and J. W. Tukey, *Measurement of Power Spectra from the Point of View of Communications Engineering*, New York: Dover, 1959.
38. D. Malacara, *Optical Shop Testing*, New York: Wiley-Interscience, 1992.
39. J. M. Steigerwald, S. P. Murarka, and R. J. Gutmann, *Chemical Mechanical Planarization of Microelectronic Materials*, New York: Wiley, 1997
40. J. F. Kaiser, Design methods for sampled data filters, *Proceedings of the 1st Allerton Conference on Circuit and System Theory*, pp. 221–236, 1963. Also in L. R. Rabiner and C. M. Rader, eds., *Dignal Signal Processing*, New York: IEEE Press, 1972.
41. E. I. Jury, *Theory and Applications of the Z-Transform Method*, Malabar, FL: Krieger, 1982.
42. R. Vich, *Z Transform Theory and Applications*, Boston: D. Reidel, 1987.
43. A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis*, New York: McGraw-Hill, 1978.

44. R. Beals, *Advanced Mathematical Analysis*, New York: Springer-Verlag, 1983.
45. E. D. Rainville, *The Laplace Transform: An Introduction*, New York: Macmillan, 1963.
46. F. J. Taylor, *Digital Filter Design Handbook*, New York: Marcel Dekker, 1983.
47. A. G. Constantinides, Spectral transformations for digital filters, *Proceedings of the IEE*, vol. 117, no. 8, pp. 1585–1590, August 1970.
48. M. Akay, *Biomedical Signal Processing*, San Diego, CA: Academic Press, 1994.
49. M. Granzow, H. Riedel, and B. Kollmeier, Linear phase filtering of early auditory evoked potentials, in A. Schick and M. Klatte, eds., *Seventh Oldenburg Symposium on Psychological Acoustics*, pp. 39–48, 1997.
50. J. F. Claerbout, *Fundamentals of Geophysical Data Processing with Applications to Petroleum Prospecting*, New York: McGraw-Hill, 1976.
51. F. Scherbaum, *Of Poles and Zeros: Fundamentals of Digital Seismology*, Dordrecht, The Netherlands: Kluwer, 1996.
52. T. Parsons, *Voice and Speech Processing*, New York: McGraw-Hill, 1987.
53. F. J. Owens, *Signal Processing of Speech*, New York: McGraw-Hill, 1993.
54. L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
55. L. R. Rabiner and M. R. Sambur, An algorithm for determining the endpoints of isolated utterances, *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.
56. L. Lamel et al., An improved endpoint detector for isolated word recognition, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 777–785, Aug. 1981.
57. J. Junqua et al., A robust algorithm for word boundary detection in the presence of noise, *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 406–412, July 1994.
58. R. E. Ziemer, W. H. Tranter, and D. R. Fannin, *Signals and Systems: Continuous and Discrete*, 3rd ed., New York: Macmillan, 1993.
59. L. B. Jackson, Signals, *Systems, and Transforms*, Reading, MA: Addison-Wesley, 1991.
60. E. A. Guillemin, *Synthesis of Passive Networks*, New York: Wiley, 1957.
61. J. E. Storer, *Passive Network Synthesis*, New York: McGraw-Hill, 1957.
62. M. E. Van Valkenburg, *Analog Filter Design*, New York: Holt, Rinehart and Winston, 1982.
63. L. Weinberg, *Network Analysis and Synthesis*, Huntington, NY: Kreiger, 1975.
64. V. K. Madiseti, *VLSI Digital Signal Processors*, Piscataway, NJ: IEEE Press, 1995.
65. R. W. Daniels, *Approximation Methods for Electronic Filter Design*, New York: McGraw-Hill, 1974.
66. W. Cauer, *Synthesis of Linear Communication Networks*, vols. I and II, New York: McGraw-Hill, 1958.
67. D. A. Calahan, *Modern Network Synthesis*, vol. 1, New York: Hayden, 1964.
68. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, New York: Technology Press of MIT and Wiley, 1957.
69. R. N. Bracewell, *The Fourier Transform and Its Applications*, New York: McGraw-Hill, 1978.
70. S. L. Hahn, *Hilbert Transforms in Signal Processing*, Norwood, MA: Artech House, 1996.
71. L. V. Ahlfors, *Complex Analysis*, 2nd ed., New York: McGraw-Hill, 1966.

72. S. L. Marple, Computing the discrete-time "analytic signal" via FFT, *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2600–2603, September 1999.

73. D. Gabor, Theory of communication, *Journal of the IEE*, vol. 93, pp. 429–457, 1946.

74. L. Cohen, *Time-Frequency Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1995.

75. P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, *IEEE Transactions on Communication*, vol. 31, no. 4, pp. 532–540, April 1983.

76. P. J. Burt, Smart sensing within a pyramid vision machine, *Proceedings of the IEEE*, vol. 76, no, 8, pp. 1006–1015, August 1988.

77. D. Marr, *Vision*, New York: W. H. Freeman, 1982.

78. S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, July 1989.

79. M. Vetterli, Filter banks allowing perfect reconstruction, *Signal Processing*, vol. 10, no. 3, pp. 219–244, April 1986.

PROBLEMS

1. Suppose an analog signal is sampled $x(n) = xa(nT)$, where T is the sampling period. A discrete Fourier transform of order N follows: $X(k) = (\mathcal{F}x)(k)$. Find the frequency resolution, the Nyquist frequency, and the highest frequency represented by the DFT coefficients:

(a) $N = 80$, $T = 0.02$;

(b) $N = 10$, sampling frequency $F = 1$ kHz;

(c) $N = 21$, $T = 1$;

(d) $N = 256$, sampling frequency = 8192 Hz.

2. Which of the following finite impulse response (FIR) filters have linear phase? zero phase?

(a) $x(n) = u(n + 2) - u(n - 2)$, where $u(n)$ is the discrete unit step signal.

(b) $y(n) = u(n + 2) - u(n - 3)$.

(c) $v(n) = (-1)^n x(n)$.

(d) $w(n) = (-1)^n y(n)$.

(e) Can an IIR filter have linear phase?

3. Provide sketches of all four types of linear phase filters H with $h(n) \in \mathbb{R}$ [26].

4. Let $r_{xx}(\tau) = E[x(t)x(t + \tau)]$ be the autocorrelation for a wide-sense stationary (WSS) analog random signal $x(t)$. Prove:

(a) $E[x(t)x(s)] = r_{xx}(t - s)$.

(b) $r_{xx}(\tau) = r_{xx}(-\tau)$.

(c) If $x = Ay$ for some constant A, find $r_{yy}(\tau)$.

(d) State and the corresponding properties for the autocorrelation $r_{ss}(\kappa) = E[s(n)s(n + \kappa)]$ of a discrete WSS random signal $s(n)$.

5. Suppose data from a noisy signal is collected for two seconds at a sampling rate of 8 kHz.

(a) If a periodogram is calculated for the entire data set, what is its frequency resolution? What is the Nyquist frequency?

(b) Suppose that Bartlett's method is tried for the purpose of improving the periodogram. The data is partitioned into sixteen disjoint windows. Now what is the frequency resolution? What is the Nyquist frequency?

(c) It is decided to try Welch's method using windows of a larger size than in part (b), but to overlap them by fifty percent. How many windows are needed to cut the frequency resolution in half? Sketch your window layout.

6. Let H be a discrete LTI system, let $H(z)$ be its system function, and let ROC_H be the region of convergence of $H(z)$.

(a) Show that if H is *stable* (if the input $x(n)$ is bounded, then the output $y(n) = (Hx)(n)$ is also bounded), then this implies that ROC_H contains the unit circle $|z| = 1$ of the complex plane.

(b) Suppose that $\{z \in \mathbb{C}: |z| = 1\} \subset \mathrm{ROC}_H$. Show that H is stable.

(c) Given: an example of a discrete signal $h(n)$ that has a discrete-time Fourier transform $H(\omega)$, but ROC_H does not include the unit circle. Under this circumstance, can the system $y = Hx = h*x$ be causal, anti-causal, or both?

7. Let H be a discrete LTI system, $h(n)$ its impulse response, $H(z)$ its transfer function, and ROC_H the region of convergence of $H(z)$.

(a) Show that if H is stable, then ROC_H contains the unit circle $|z| = 1$.

(b) Show the converse: If $\mathrm{ROC}_H \supset \{z \in \mathbb{C}: |z| = 1\}$, then H is stable.

8. Let $x(n)$ be a discrete signal, $X(z)$ its z-transform, and ROC_X the region of convergence.

(a) Show that $x(n)$ is right-sided implies that ROC_X is the exterior of a circle in the complex plane.

(b) More particularly, if $x(n)$ is causal, show that $\infty \in \mathrm{ROC}_X$.

(c) If $x(n)$ is left-sided, show that ROC_X is the interior of a circle and may or may not include 0.

(d) Give a condition on a left-sided signal $x(n)$ so that $0 \in \mathrm{ROC}_X$.

(e) Give a characterization in terms of ROC_H for causal, stable LTI systems $y = Hx$.

9. Let $x(t)$ have Fourier transform $X(\Omega)$ and Laplace transform $X_L(s)$.

(a) Show that $X_L(j\Omega) = X(\Omega)$, for $\Omega \in \mathbb{R}$.

(b) Let $s = \Sigma + j\Omega$, where $\Sigma \in \mathbb{R}$. Show that $X_L(s)$ is the Fourier transform of $x(t)e^{-\Sigma t}$.

(c) Show that Laplace transform convergence does not depend on the imaginary part of $s = \Sigma + j\Omega$.

(d) Conclude that $X_L(s)$ converges on vertical strips in the complex plane.

10. Let $x(t) = \exp(-a|t|)$ be the Lorentzian function, where $a > 0$.

(a) Show that $X(\Omega) = 2a/(a + \Omega^2)$.

(b) Sketch $x(t)$ for $a = 1, 2, 3$.

(c) Explain how convolution with $x(t)$ performs a weighted averaging of input signal data.

11. Suppose we map the Laplace s-plane to the z-plane via $z = e^{sT}$, where we shall assume equality in (9.72).

(a) Show that this implies $s = \frac{2}{T}\left(\frac{z-1}{z+1}\right)$, the bilinear relation between the Laplace and z-transformations.

(b) Let $z = e^{j\omega}$ and $s = j\Omega$ in (a) and derive the bilinear mapping (9.70).

12. Derive the Laplace transform properties given in Table 9.3.

13. Let $T_N(\Omega)$ be the Chebyshev polynomial of order $N \geq 0$.

(a) Show that $T_N(1) = 1$ for all N.

(b) Show $T_N(\Omega)$ is even if N is odd, and $T_N(\Omega)$ is odd if N is even.

(c) Show all the zeros of $T_N(\Omega)$ lie on the open interval $(-1, 1)$.

(d) Show that $|T_N(\Omega)| \leq 1$ on $[-1, 1]$.

14. Let $P(\Omega)$ and $Q(\Omega)$ be the squared magnitude response for the Chebyshev and inverse Chebyshev low-pass filters, respectively.

(a) Show that the poles of $Q(\Omega)$ are the reciprocals of the poles of $P(\Omega)$.

(b) Find $P(\Omega)$ for IIR filters of order 2 and 3.

(c) Find $Q(\Omega)$ for IIR filters of order 2 and 3.

(d) Under what conditions is $Q(\Omega)$ an all-pass filter? Explain.

(e) Find the zeros of $Q(\Omega)$.

(f) Show that the zeros of $Q(\Omega)$ do not depend on the stopband ripple function.

15. Compute the Hilbert transform $x_H(t)$ for the following signals:

(a) $x(t) = \cos(2\pi t)$.

(b) $x(t) = \sin(-3\pi t)$.

(c) $x(t) = \cos(5\pi t) + 2\sin(2\pi t)$.

(d) $x(t) = \delta(t)$, the Dirac delta.

(e) $x(t) = C_0$, a constant signal.

(f) $x(t) = u(t)$, the unit step signal.

(g) $x(t) = u(t + 1) - u(t - 1)$.

16. Let $x(t) = \cos(\Omega_0 t)$ and $y(t) = \sin(\Omega_0 t)$.

(a) Show that $y_H(t) = -x(t)$, where $y_H(t)$ is the Hilbert transform of $y(t)$.

(b) Compute the analytic signal $x_A(t)$.

(c) Compute the signal envelope for $x(t)$.

(d) Compute the instantaneous phase $\phi(t)$.

(e) Compute the instantaneous frequency $\omega(t)$.

17. Show that the Hilbert transform is linear: If $x(t)$ and $y(t)$ are analog signals and $z(t) = Ax(t) + By(t)$, then $z_H(t) = Ax_H(t) + By_H(t)$.

18. Let $x(n)$ be discrete with period $N > 0$. Show that the discrete Hilbert transformation of $A\sin(2\pi k_0 n/N)$ is $-A\cos(2\pi k_0 n/N)$.

19. Suppose the discrete filter H has impulse responses $h(n)$. Suppose $\overline{h}(n) = h(-n)$ and let $\overline{H}(\omega)$ be its DTFT.

(a) Show that $\overline{H}(\omega) = H^*(\omega)$, the complex conjugate of $H(\omega)$.

(b) Let $y(n) = x(2n)$. Show that $Y(2\omega) = [X(\omega) + X(\omega + \pi)]/2$.

(c) Let $y(n) = x(n/2)$. Show that $Y(\omega) = X(2\omega)$.

20. Let $h(n)$ and $g(n)$ be discrete filters that provide an exact reconstruction scheme as in Figure 9.52. Show that $G^*(\omega)G'(\omega) + G^*(\omega + \pi)G'(\omega + \pi) = 2$.

Advanced problems and projects:

21. This problem outlines a proof of the Wiener–Khinchin theorem for discrete random signals. Assume the notation from Section 9.2.3.1.

(a) First show that

$$\sum_{n=-L}^{L}\sum_{m=-L}^{L} x(n)x(m)e^{-j\omega(n-m)} = \left|X_L(\omega)\right|^2, \tag{9.191}$$

where $X_L(\omega)$ is the local discrete time Fourier transform of real-valued WSS random signal $x(n)$, and we assume that the autocorrelation function $r_{xx}(\nu)$ is absolutely summable: $r_{xx}(\nu) \in l^1$.

(b) Apply the expectation operator to both sides of (9.191) to get

$$\begin{aligned} E[\left|X_L(\omega)\right|^2] &= \sum_{n=-L}^{L}\sum_{m=-L}^{L} E[x(n)x(m)]e^{-j\omega(n-m)} \\ &= \sum_{n=-L}^{L}\sum_{m=-L}^{L} r_{xx}(n-m)e^{-j\omega(n-m)}. \end{aligned} \tag{9.192}$$

(c) Divide by $2L + 1$ and take limits on both sides of (9.192). From the absolute summability of $r_{xx}(\nu)$, argue that the equal limits of the double summation can be replaced with independent limits to get

$$\begin{aligned} \lim_{L\to\infty}\frac{1}{2L+1}E[\left|X_L(\omega)\right|^2] &= \lim_{L\to\infty}\frac{1}{2L+1}\sum_{n=-L}^{L}\sum_{m=-L}^{L} r_{xx}(n-m)e^{-j\omega(n-m)} \\ &= \lim_{L\to\infty}\frac{1}{2L+1}\sum_{n=-L}^{L}\lim_{K\to\infty}\sum_{m=-K}^{K} r_{xx}(n-m)e^{-j\omega(n-m)}. \end{aligned} \tag{9.193}$$

(d) Conclude as follows:

$$X_{\text{PSD}}(\omega) = \lim_{L\to\infty}\frac{1}{2L+1}\sum_{n=-L}^{L} R_{xx}(\omega)$$
$$= \lim_{L\to\infty}\frac{2L+1}{2L+1}R_{xx}(\omega) = R_{xx}(\omega). \tag{9.194}$$

22. This problem outlines a proof of the Wiener–Khinchin theorem (9.30) for WSS analog random signals $x(t)$. Assume the notation from Section 9.2.3.1.

(a) By interchanging the order of integration, show that

$$E\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_L(t)x_L(s)e^{-j\omega(t-s)}ds\,dt\right] = E[\left|X_L(\omega)\right|^2], \tag{9.195}$$

where $X_L(\omega)$ is the local radial Fourier transform of $x(t)$.

(b) Use the results of probability theory: $E[ax + by] = aE[x] + bE[y]$ and $r_{xx}(t - s) = E[x(s)x(t)]$. Show that the expectation operation may be moved inside the integral in (9.195) and therefore that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[x_L(t)x_L(s)e^{-j\omega(t-s)}]\,ds\,dt = \int_{-L}^{L}\int_{-L}^{L} r_{xx}(t-s)e^{-j\omega(t-s)}ds\,dt\,. \tag{9.196}$$

(c) Let $u = t - s$ for a change of integration variable in (9.196) and show that the iterated integral becomes $\int_{-2L}^{2L} r_{xx}(u)e^{-j\omega u}[2L - |u|]\,du$.

(d) Put the above expressions together and take limits to get

$$\lim_{L\to\infty}\frac{1}{2L}E[\left|X_L(\omega)\right|^2] = \lim_{L\to\infty}\int_{-2L}^{2L} r_{xx}(u)e^{-j\omega u}\left[\frac{2L-|u|}{2L}\right]du\,. \tag{9.197}$$

(e) Show that the integrand on the right-hand side of (9.197) is bounded by $|r_{xx}(u)|$ so that Lebesgue's dominated convergence theorem (Chapter 3) applies.

(f) Since the the limit and integration operations can be interchanged, show

$$X_{\text{PSD}}(\omega) = \int_{-\infty}^{\infty} r_{xx}(u)e^{-j\omega u}\left\{\lim_{L\to\infty}\left[\frac{2L-|u|}{2L}\right]\right\}du = R_{xx}(\omega)\,, \tag{9.198}$$

where $R_{xx}(\omega)$ is the radial Fourier transform of $r_{xx}(t)$.

23. Consider a linear system $y = Hx$, where the WSS random signal $x(t)$ is the input and $y(t)$ is the corresponding output. Let $r_{xx}(\tau) = E[x(t)x(t + \tau)]$ and $r_{yy}(\tau) = E[y(t)y(t + \tau)]$.

(a) Show that the cross correlation $r_{xy}(\tau) = r_{xx}(\tau) * \overline{h(-\tau)}$.

(b) Show that $r_{yy}(\tau) = r_{xy}(\tau) * h(\tau)$.

(c) Show that $Y_{\text{PSD}}(\Omega) = X_{\text{PSD}}(\Omega)|H(\Omega)|^2$, where $H(\Omega)$ is the radial Fourier transform of $h(t)$.

24. Suppose that real-valued discrete signal $x(n)$ is sampled at $F_s = 1/T$ Hz. We select a window of $N > 0$ values and compute the DFT of order N. Thus, the DFT $X(k)$ has frequency resolution $(\text{NT})^{-1}$ and its coefficients represent discrete frequencies $(NT)^{-1}$, $2(NT)^{-1}$, ..., $(2T)^{-1}$.

(a) Show that adding a pure sinusoid of one these frequencies—say $\omega_k = k\text{F}_s/N$, for $1 \le k \le N/2$—to $x(n)$ will alter only $X(k)$ and $X(N-k)$.

(b) Verify experimentally using that $y(n) = x(n) + \cos(2\pi nk/N)$ a sinusoid of frequency $\omega \neq \omega_k$, for any $1 \le k \le N/2$, will perturb all of the $X(k)$.

(c) Show that the difference caused by adding the sinusoid diminishes in magnitude like $1/|\omega - \omega_k|$ as $|\omega - \omega_k|$ increases [32].

25. This problem motivates use of the sampling interval T in the bilinear transformation (9.70). The derivation closely follows [33]. Consider the analog integrator system with impulse response

$$h(t) = \begin{cases} 0 & \text{if } t < 0, \\ 1 & \text{if } t \ge 0. \end{cases} \tag{9.199}$$

(a) Let $y = Hx$, so that $y(t) = (h*x)(t)$ and show that for $0 < a < b$,

$$y(b) - y(a) = \int_a^b x(t)\, dt\,. \tag{9.200}$$

(b) Argue that as $a \to b$,

$$y(b) - y(a) \approx \frac{b-a}{2}[x(a) + x(b)]\,. \tag{9.201}$$

(c) Let $a = nT - T$ and $b = nT$ so that (9.201) becomes a discrete integrator and show that

$$Y(z) - z^{-1}Y(z) = \frac{T}{2}[z^{-1}X(z) + X(z)]\,. \tag{9.202}$$

(d) Show that the discrete integrator has z-transform

$$H(z) = \frac{T}{2}\frac{(z+1)}{(z-1)}\,. \tag{9.203}$$

(e) Show that the Laplace transform of the analog integrator is $H(s) = s^{-1}$.

(f) Argue that an analog system defined by a difference equation (9.65) can be implemented using adders, amplifiers, and integrators.

(g) Replace every analog element by its corresponding discrete element and conclude that the discrete transfer function corresponding to (9.65) is given by

$$s = \frac{2}{T}\frac{(z-1)}{(z+1)}. \tag{9.204}$$

26. Prove Liouville's theorem, which we used to justify the fundamental theorem of algebra. Let $|f(z)| \leq B$ on $\mathbb{C}$ be everywhere differentiable, or *analytic*.

(a) Show that the Cauchy Residue Theorem (Chapter 1) implies

$$f(z) = \frac{1}{2\pi j}\oint_C \frac{f(s)}{s-z}\, ds, \tag{9.205}$$

where C is a large circular contour centered about $z \in \mathbb{C}$.

(b) Show that

$$f'(z) = \frac{1}{2\pi j}\oint_C \frac{f(s)}{(s-z)^2}\, ds. \tag{9.206}$$

(c) If C has radius $R > 0$, show that

$$|f'(z)| \leq \frac{B}{R}. \tag{9.207}$$

(d) Since the radius of C may be arbitrarily large, conclude that $f'(z) = 0$ and $f(z) \equiv$ constant.

27. Suppose a rational function $P(\Omega) = B(\Omega)/A(\Omega)$ is even: $P(\Omega) = P(-\Omega)$ for all $\Omega \in$ R. Show that $A(\Omega)$ and $B(\Omega)$ have no terms of odd degree.

28. Consider the analog Lorentzian signal $h(t)$ with rational (radial) Fourier transform. $H(\Omega) = (1 + \Omega^2)^{-1}$.

(a) Show that $h(t)$ is given by

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\Omega)e^{j\Omega t}d\Omega = \frac{e^{-|t|}}{2}. \tag{9.208}$$

(b) Using synthetic signals, evaluate the performace of $h(t)$ as a peak, valley, and transient detector.

(c) Evaluate the signal analysis performance of $h(t)$ for detecting the same signal features in the presence of synthetic additive noise.

(d) Identify suitable real data sets and report on the Lorentzian's capabilities for detecting these same features.

29. Develop software for a Butterworth filter design tool.

(a) Given a discrete cutoff frequency ω_c and filter order $N > 0$, compute the analog cutoff Ω_c.

(b) Find the above poles of the analog squared Fourier magnitude response function.

(c) Select poles in order to ensure a stable filter.

(d) Derive the discrete impulse response $h(n)$.

(e) Compute the discrete magnitude response $|H(\omega)|$.

(f) Compute the margin by which the filter H passes or fails the passband and stopband constraints.

(g) Provide an approximation feature that computes the filter order given ω_c as well as passband and stopband criteria.