ELECTRONOTES 197

NEWSLETTER OF THE

MUSICAL ENGINEERING GROUP

1016 Hanshaw Rd., Ithaca, NY 14850

Volume 20, No. 197 April 2001

GROUP ANNOUNCEMENTS

CONTENTS OF EN#197

In this issue, having finished the series on Analog Signal Processing (ASP), we can begin the presentation on Digital Signal Processing (DSP). As promised, however, we will be having a small "Corner" on ASP, doing some problems and amusements.

Analog Signal Processing Corner

The Strange Root Locus of Negative Gain VCVS

-by Bernie Hutchins

The negative-gain voltage-controlled-voltage-source (NG-VCVS) realization of an active filter was discussed in Analog Signal Processing in Sections 5-1 and 7-5. A typical circuit is repeated here in Fig. 1 for convenience. One notable finding about this choice of configuration was that it had excellent passive sensitivity properties, and terrible active sensitivity properties. Both of these we attributed to the large negative gain of the amplifier used: the large amount of

(Continues on Page 34)

Basic Elements of Digital Signal Processing

Introduction to the Series

What are the "Basic Elements of Digital Signal Processing?" In some sense, they are the fundamental concepts that absolutely must be included in the syllabus of any introductory course in Digital Signal Processing (DSP). I have taught DSP at Cornell for over 20 years and have been fortunate to work with many excellent educators, including Tom Parks. Tom has always maintained that these fundamental topics are: Sampling, Fourier Transform, and Filtering. I think we all agree on this, and at the same time, we find that there are many additional topics that we would love to include either as part of the introduction or in follow-up courses Indeed, in any DSP course, we expect to have more material in mind than we can possibly present. This is preferable to the alternative of course.

A second notion with regard to "Basic Elements" relates to a need to strengthen the foundations presented in an introductory course We seldom do this. Once we have seen sampling, DFT, FFT, and digital filter design, we tend to move on to new things. A remarkable thing about DSP is that nearly every idea can be understood in more than one way - often in a half dozen ways. Most of the usual notions can be twisted and extended in interesting ways. Sampling can be done with unequal spacing. A Discrete-Time Fourier Transform (DTFT) can be made equivalent to a Fourier Series. Filter design techniques can be modified for different phase specifications. These ideas may often be considered to be intermediate or advanced level extensions of the most fundamental ideas.

In DSP, electrical engineering, and likely in most fields, there is often little room in formal course offerings for "intermediate" level work. Somehow, no one really wants to offer (or take) a course with "intermediate" in the title. "Introductory" is always an acceptable course descriptive. Subsequently many students may well move on to something entirely different, being satisfied (or bored, overwhelmed, etc.) after just this introduction. Otherwise we prefer to be considered advanced instead of intermediate. Of course, a semester-based academic year (as opposed to trimester) often suggests two levels and not three.

The material presented here is intermediate in level. It always reviews the basics, but then goes further and deeper than usual. In the "Basic Elements of DSP," we will be covering fundamental concepts, but going somewhat deeper into them. We need to know the fundamentals, of course, but it is very helpful to understand them very, very well. We hope to develop a very strong understanding - one more typical of persons who have worked in the DSP area many years, and who have spend many enjoyable hours discussing the magic of DSP.

Because we have just finished up a series on analog filtering, it makes sense to us to begin our DSP Elements with Digital Filtering. These we will follow with Sampling, and with Fourier Transform. Also in the works in a segment on Finite Wordlength Effects. And there will likely be more. Another possibility is multirate filtering, although much of this we would prefer to consider as "intermediate sampling." We are open to additional suggestions.

Basic Elements of Digital Signal Processing

Filter Element - Part 1

-by Bernie Hutchins

1. INTRODUCTION

The design of digital filters is a major element of Digital Signal Processing (DSP). The filter design art is well-developed, and numerous formulas leading to simple calculations are available. Further, computer-aided design packages are ubiquitous for both simple and complicated design procedures. Special purpose devices (for example, Hilbert transformers) are included among the solved problems. Accordingly, we find that the issues of design are well understood, that attractive design options for given applications are generally available, and as hardware continues to get faster and cheaper, there is little excuse not to strive for a design offering the very best performance.

Filter design almost always begins with an engineering notion of some task to be accomplished: an overall product or study, and one or more specific requirements for filtering. It needs to be pointed out however that more often than not, the original specification is nowhere near as precise as those found in textbook problems. A textbook might specify a filter to be designed as being lowpass, of a certain order, with a certain allowable passband error, a precise cutoff frequency, and with certain minimum stopband rejection. A more likely scenario in the real world would be that a certain job to be accomplished is roughly outlined (e.g., we want to isolate the bandwidth of speech) and we have certain resources (you can't make the filter so big that it uses all the instruction cycles available during a clock interval.) The design engineer must often explore a wide range of issues (often involving such things as physics and perceptual psychology) and then try for a reasonable specification, while remaining as flexible as possible.

Often the most general outline of the job to be accomplished is in terms of a frequency-domain description. This is what we mean when we say that a filter is to be low-pass, high-pass, bandpass, or notch. These terms describe the most basic shape of the magnitude of the frequency response. (Even an all-pass filter, with a flat magnitude response, is likely a frequency-domain description, giving phase as a function of frequency.) Accordingly it is common to find digital filter design methods that begin with some idealized notion of such a filter. Commonly such methods attempt to control the error, the difference between the desired frequency response and the actual frequency response, with regard to a calculable criterion.

Yet we cannot ignore the time-domain performance. For example, we often design linear-phase filters knowing that they will have a constant time delay for all frequencies, a generally attractive performance feature. Or perhaps on the other hand we need to give up linear phase if we must achieve a much shorter delay. Neither is it unusual to find the filter itself specified (at least originally) in the time domain. The "Impulse Invariance" design of Infinite Impulse Response (IIR) filters is a prime example. Yet time-domain descriptions are also found in Finite Impulse Response (FIR) filters. For example we may seek a filter that does time-domain interpolation of a sequence. The "moving-average" filter that will be mentioned shortly below is a second FIR example.

It should also be recognized that neither the design computation time nor the effort of the designer (e.g., even time spent learning a new design technique) are important issues. This is because the results of the design are largely a set of filter coefficients (path multipliers), and the overall filter, once finished and made part of a product, is constant and infinitely repeatable with no further design effort or calculation time. That is, we are interested in the numerical efficiency of the filter as it calculates a new output sample based on previous input samples (by multiplying by filter coefficients) but the coefficients themselves are just fixed numbers. Thus, for example, if we had a filter with 25 coefficients, the running efficiency for 25 coefficients obtained from a simple formula would be the same as for a somewhat different set of 25 coefficients obtained from a more complicated design program, often with superior performance.

1a. THE TOOLS

One fundamental relationship we need in filter design is the relationship between a filters "frequency response" and its "impulse response." This derives as a special form of the z-Transform:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)\, z^{-n} \qquad \textbf{[z-Transform]} \qquad (1)$$

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-jn\omega} \qquad \textbf{[Discrete Time Fourier Transform]} \qquad (2)$$

where H(z) is the filter's "transfer function," $H(e^{j\omega})$ is the Discrete Time Fourier Transform (DTFT) or "frequency response," and h(n) is the impulse response. Accordingly $H(e^{j\omega})$ is H(z) evaluated at $z=e^{j\omega}$, the unit circle in the z-plane. The "normalized frequency" ω runs over a convenient interval of length 2π, where the sampling frequency is considered to be 2π. In the event that a designer prefers to work with non-normalized frequencies, we can use a different from of equation (2):

$$H(e^{j\Omega T}) = \sum_{n=-\infty}^{\infty} h(n)\, e^{-jn\Omega T} \qquad \textbf{[Frequency Response} \text{ (Physical Frequencies)}\textbf{]} \quad (3)$$

where T is the sampling interval and Ω is frequency in the physical units of radians-per-second The sampling frequency in Hertz is of course $f_s=1/T$ and in terms of Hertz, $f = \Omega/2\pi$.

It needs to be stressed that in specifying a digital filter, the normalized view makes good sense (often, normalized views lead to confusion). This is because it is the ratio of the various frequencies of interest, relative to the particular sampling frequency, that determines the filter parameters. Thus a particular filter designed for a cutoff frequency of 1 kHz with a 10 kHz sampling frequency will have the same coefficients as a filter designed for a cutoff of 2 kHz with a 20 kHz sampling frequency (or, for that matter, a filter designed for a cutoff of π/5 = 2π/10 with a 2π sampling frequency).

1b. A CHOICE OF TIME OR FREQUENCY DOMAINS

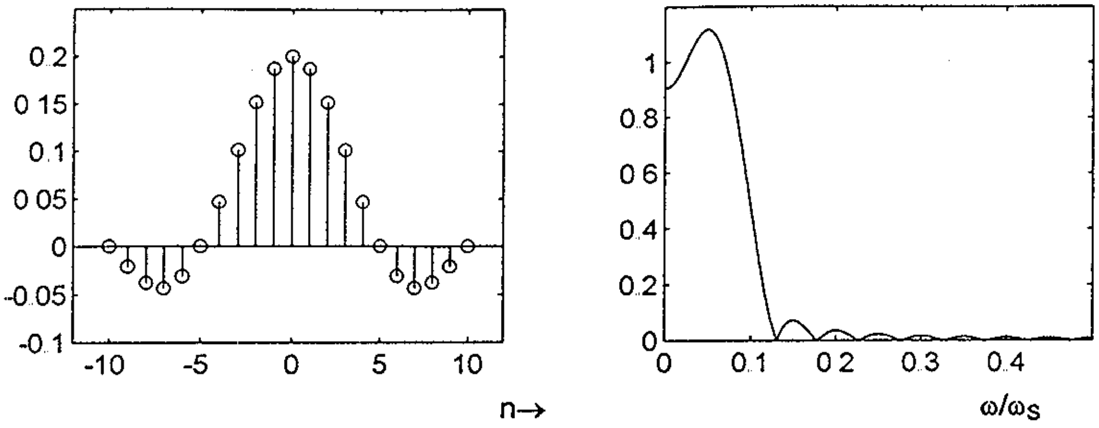
Filter design can be done in the frequency domain or in the time domain. Often we start in the frequency domain because this is usually how the filter is specified for a particular application. For example, based on equation (2) we might have a notion of $H(e^{j\omega})$, or at least of the magnitude $|H(e^{j\omega})|$, and this would allow us to consider how using the inverse DTFT might lead to h(n):

$$h(n) = (1/2\pi) \int_{-\pi}^{\pi} H(e^{j\omega})\, e^{jn\omega}\, d\omega \qquad \textbf{[Inverse DTFT]} \qquad (4)$$

For example, we might think of needing a low-pass filter with $H(e^{j\omega}) = 1$ for $|\omega| < \pi/5$, otherwise 0 on $-\pi$ to $+\pi$. Directly integrating equation (4) leads to:

$$h(n) = (1/5)\,[\,\sin(n\pi/5)/(n\pi/5)\,] \qquad -\infty < n < +\infty \qquad (5)$$

[Here we will generally write out sin(x)/x instead of using sinc, as the definition of sinc is not always consistent among texts. Sometimes it is sin(πx)/πx.] This is not a bad filter, but it does illustrate certain limitations. First, there is the issue of causality - the impulse response begins before n=0. Secondly, there is the infinite length, which we can probably handle by finding values of |n| beyond which the impulse response is negligible. Taken together, we can truncate the impulse response of equation (5) and then shift it to the right, which is equivalent to adding a non-zero linear phase, and arrive at a realizable filter. (An example of a length 21 filter following this procedure is shown in Fig. 1.) But, the largest "limitation" is that we really have to look at additional options, and not just at this simple formula.



Impulse Response

Magnitude Response

Fig. 1 A length 21 digital low-pass based on inverse DTFT. The cutoff was set to $\pi/5$ (thus 1/10 the sampling frequency). The filter's impulse response is non-causal.

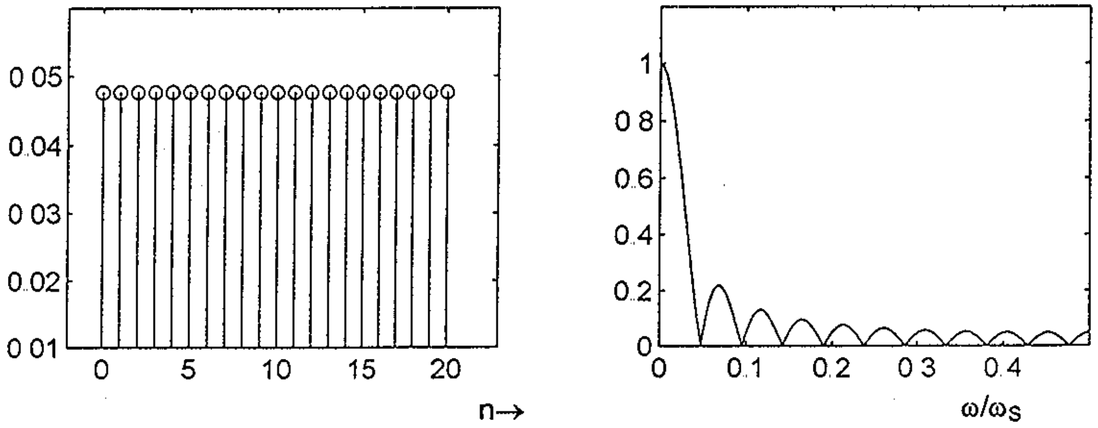
In a time domain approach, we might choose $h(n) = 1/N$ for $n=0$ to $n=N-1$, reasoning that such a "moving average" has a low-pass nature to it. Here we have chosen a finite length, causal time sequence for the impulse response, thus avoiding the problems above. The frequency response is available from equation (2) as:

$$H(e^{j\omega}) = (1/N)\,[\,1 + e^{-j\omega} + e^{-2j\omega} + \dots\ e^{-(N-1)j\omega}\,]$$

$$= (1/N)\, e^{-j(N-1)\omega/2}\ [\,\sin(N\omega/2) / \sin(\omega/2)\,] \qquad (6)$$

This is a simple, finite length, causal, low-pass digital filter. It is just not a very good one. Fig. 2 shows a length-21 example of this nature.

[Here, and in much of what we do here, we will be using examples that are low-pass. This choice is mostly a matter of tradition, mixed with the idea that probably low-pass filters are the most commonly needed filters in practice. We could usually use other types just as easily, and converting one type of filter to another is likely fairly easy. In cases where there are specific difficulties associated with other types, we will try to point this fact out.]



Impulse Response

Magnitude Response

Fig. 2 A 21-tap "moving average" filter is a time-domain based low-pass. It is well defined in the time domain, but is not a very good low-pass as seen in the frequency domain. Note however that it does return a magnitude near to 1 for very low frequencies, and it does reject, completely, frequencies that are integer multiples of $f_S/21$.

2. FREQUENCY DOMAIN BASED DESIGNS - BASICS

2A. INTRODUCTION TO FREQUENCY-DOMAIN BASED PROCEDURES

As mentioned above, we often look at design specifications and design procedures in the frequency domain. Yet the design itself is not finished until we can give actual numbers for the filter's transfer function (essentially frequency domain), or equivalently, its impulse response (time domain). Another way to state this is to say that we need the "coefficients" for the network that can be used to actually realize the filter.

Digital filtering is done with three basic devices: delays, multipliers, and summers. In terms of actual DSP program code, we look for instructions to move data from one location to another (thus forming a delay), for multiply instructions, and for add (or subtract) instructions. Now we can do digital filtering of data in memory. But we do still need the network (the interconnection of delays and summers), and the path multiplier coefficients. And of course, we need to design the filter (determine the coefficients).

Digital filters are often categorized as IIR (Infinite Impulse Response) or FIR (Finite Impulse Response). This terminology refers to an impulse response that is finite or infinite in duration. The amplitude of the impulse response should always be finite. A number of factors (pole-zero cancellation, finite word size for coefficients) can generate exceptions, but IIR filters are generally found to contain poles (as well as zeros) and feedback paths (as well as feedforward paths), while FIR filters have only zeros and feedforward paths. Because of the poles, IIR filters will usually give a far superior performance to FIR for a given order (number of delays). [See Fig. 27.] Also because of the poles, IIR filters can be potentially unstable, and phase properties of IIR filters are often more of a worry. IIR filter designs almost always involve design data derived from continuous-time filters. (Indeed, all continuous-time filters are IIR.) FIR filters on the other hand are usually designed starting with discrete-time mathematics.

Here in Section 2 we will be looking at the basic design methods for digital filters starting with a frequency domain specification. This will be basically a review of textbook methods, but will also establish some essential procedures we will need later. In Section 3, we will generalize these methods and look at some others as well. Section 4 will then get around to time-domain specifications.

In Section 1, the use of the inverse DTFT, equation (4), was suggested as an example of a frequency domain based design method. This will be developed through a series of improvements into what is the same as a least squared error criterion. [The initial result had problems with infinite length and non-causality - in fact the original result from equation (5) is IIR.] We will begin to develop these ideas here.

A second familiar means of getting back and fourth between the time and frequency domains is the DFT (Discrete Fourier Transform):

$$H(k) = \sum_{n=0}^{N-1} h(n)\, e^{-j(2\pi/N)nk} \qquad \textbf{[Discrete Fourier Transform]} \qquad (7)$$

$$h(n) = (1/N) \sum_{k=0}^{N-1} H(k)\, e^{j(2\pi/N)nk} \qquad \textbf{[Inverse DFT]} \qquad (8)$$

Less there be any residual misunderstanding, the DFT and DTFT are not the same thing. The DFT corresponds to a sampling of the DTFT, and is almost always computed by the FFT (Fast Fourier Transform).

In fact the sampled nature of the DFT suggests using equation (8), the inverse DFT, as a means of obtaining an impulse response h(n) (the filter coefficients) based on samples H(k) of a desired frequency response. While we might well expect to do worse by working from just samples, it does seem to take care of the finite length worries. This method, called appropriately "frequency sampling," will be studied in Section 2c.

There are only two methods of IIR filter design that are common. In fact, the so-called Bilinear-z transform method is ubiquitous. In Section 2d it will be presented in its frequency-domain form. It will also be seen in Section 4 with regard to integrator realization, and in a Section 5 special topic of placing stopband zeros.

2b. THE INVERSE DTFT - A START ON LEAST SQUARED ERROR

We can often easily sketch exactly what type of frequency response we would prefer to end up with (often an ideal filter) if we could have our wishes. When we do select such an idealized response, we might expect that at some point thereafter in the design procedure, some choice of a practical nature would force us to abandon the ideal response. But we would still hope for an acceptable approximation. In the event that we can write down a mathematical function for the desired response $H(e^{j\omega})$, we can hope to integrate equation (4) and then see if the resulting h(n) is practical. This we did in equation (5), for example, where it was only necessary to be able to integrate an exponential. If we were unable to integrate, or perhaps do not even have a mathematical expression for $H(e^{j\omega})$ to put inside the integral, we would likely resort to frequency sampling with the inverse DFT [equation (8)], which would amount to a numerical integration.

2b-1 Linear Phase and Causality

In the result given in equation (5), we found that h(n) had a sinc shape, centered about zero. The sinc shape was helpful, as it suggests that for large n, |h(n)| is small, and perhaps can be ignored. In cases like this where h(n) is clustered in this way, we expect that it can be somehow truncated to finite length with limited penalty. But even as we recognize the problem with truncation, we still have the causality problem: h(n) is significantly non-zero for negative n. Intuitively we might expect to be able to just truncate between limits $|n| \leq M$ to get a length 2M+1 impulse response, and then shift the entire response to the right to make it causal. That is, we are willing to settle for some delay, and in many applications, a relatively small delay is of no consequence. For example, in a CD player, we push the play button and expect to hear music that was performed and recorded days, or even years ago. So a delay on the order of a fraction of a second isn't going to matter in this regard. As long as music starts coming out after not much more than a second or two, we are satisfied with the product.

We need to examine the consequence (as seen in the frequency domain) of this delay, which we achieved by brashly shifting the time-domain impulse response to the right. Another way to look at this is to ask what changes we would need to make to $H(e^{j\omega})$ to have this delay come out naturally from equation (5). When we specified $H(e^{j\omega})$ we set it to exactly 1 for the passband, 0 for the stopband. In general, $H(e^{j\omega})$ is a complex number, represented by a real part and an imaginary part, or equivalently as a magnitude and an angle. Electrical engineers call this angle "phase," of course. What we are looking for in specifying a frequency response is thus:

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\phi(\omega)} \tag{9}$$

where $\phi(\omega)$ is the phase as a function of frequency. In this view, when we used $H(e^{j\omega})=1$, we were specifying a magnitude of 1, and a phase of zero.

Suppose we choose $\phi(\omega)$ to be a "linear phase," a linear function of frequency such as $\phi(\omega)=-\tau\omega$ where τ is some constant. We recognize τ as being a time, for dimensional reasons (or at least as a delay of whatever time index we choose, if ω is assumed dimensionless). Now if we assume a passband response of $H(e^{j\omega}) = 1 \cdot e^{-j\omega\tau}$, with stopband still zero (phase thus irrelevant), equations (4) leads to:

$$\begin{aligned} h(n) &= (1/2\pi) \int_{-\pi}^{\pi} H(e^{j\omega})e^{jn\omega}\, d\omega \\ &= (1/2\pi) \int_{-\pi/5}^{\pi/5} e^{-j\omega\tau}\, e^{jn\omega}\, d\omega \\ &= (1/5) \sin[(n-\tau)\pi/5] \,/\, [(n-\tau)\pi/5] \end{aligned} \tag{10}$$

This is identical to equation (5) except for a slide to the right by τ. For example, if $\tau=10$, then the center of the sinc is at $(n-\tau)=0$, or at $n=\tau=10$. Fig. 3 shows the impulse response for both cases, and the magnitude response for a length 51 filter chosen in various ways.

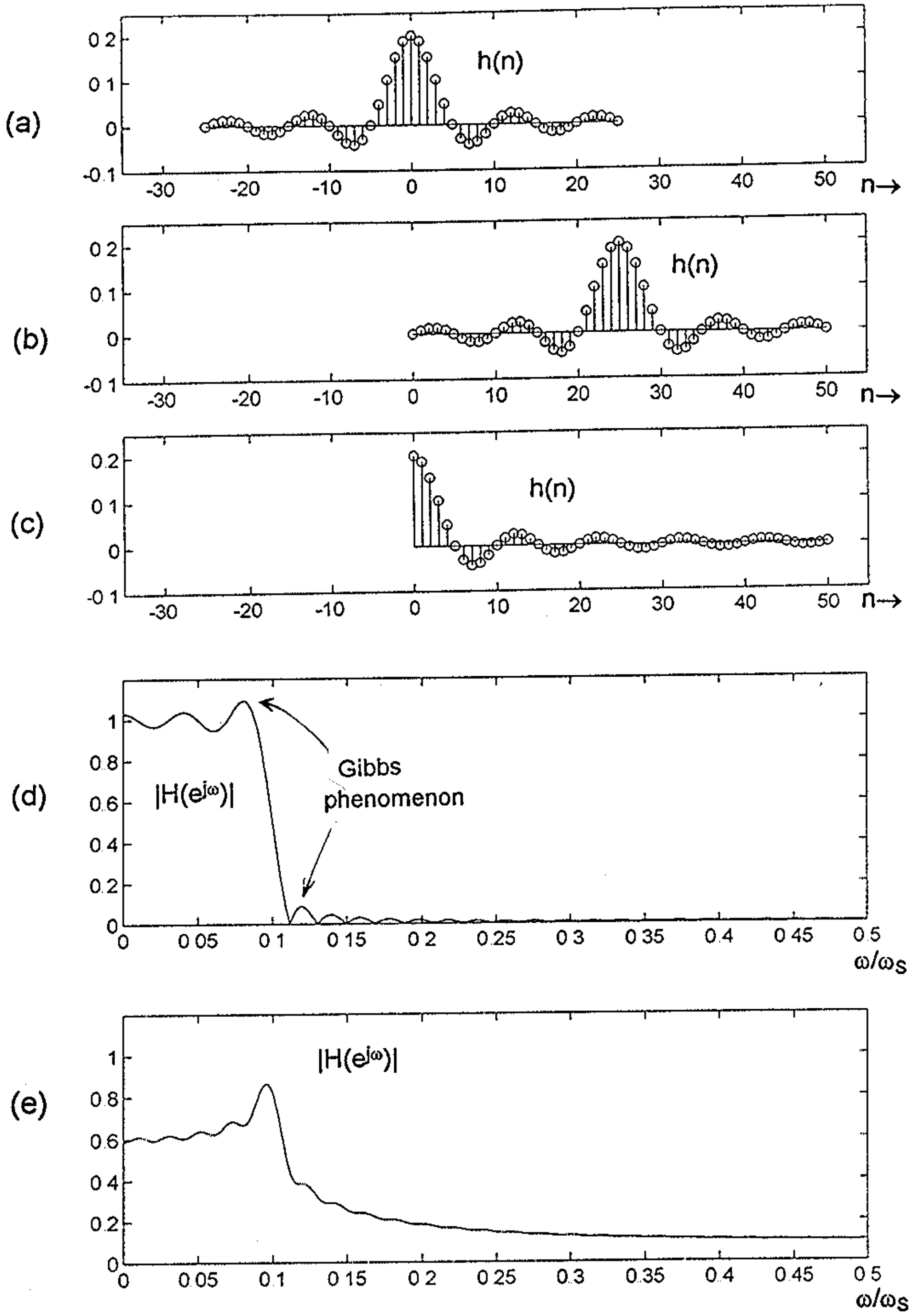


Fig. 3 Here we have three different ways of choosing 51 taps from among the infinite set of taps returned by the inverse DTFT. In (a) we see a zero-phase, non-causal choice. In (b) we have a linear phase (delay of 25) choice, which amounts to just shifting (a) 25 places to the right. The choice in (c) is a causal, zero-phase choice. In (d) we see the magnitude response corresponding to either (a) or (b), while in (e) we see the magnitude response corresponding to (c). The response in (d) is typical of truncation, showing a Gibbs phenomenon ripple. The response in (e) is far from the desired specification. What it amounts to is that we do not want to discard large taps when we can make some other choice.

Here putting the linear phase into the desired response has achieved the shift which we previously added ad hoc after the zero-phase calculation. The distinction is perhaps not worth the effort here, but not putting the phase in right in the first place is a bad habit, and other procedures (frequency sampling - as we shall see) are not as forgiving.

While we can enjoy a certain satisfaction in understanding the problem of using linear phase, as part of the filter specification, to shift the most significant filter taps to the right side (favoring the causal side), the truncation problem remains. There is first the issue of truncating in a way that maintains the linear phase that was built into the specification, and there is also of course the issue of just what truncation in itself perhaps does to the magnitude response. The issue of maintaining linear phase is easy to study and is a matter of retaining symmetry of the impulse response - a very natural way of truncating. (Intuitively one correctly expects that a reasonable way of minimizing errors is to keep the largest impulse response values. Notice the large errors in the response of Fig. 3e where large impulse response values were discarded.)

For example, we can truncate to five symmetric terms for a zero-phase case, and then plug back into the DTFT, equation (2) to get the post-truncation frequency response:

$$H(e^{j\omega}) = h_2 e^{2j\omega} + h_1 e^{j\omega} + h_0 + h_1 e^{-j\omega} + h_2 e^{-2j\omega}$$

$$= 2h_2 \cos(2\omega) + 2h_1 \cos(\omega) + h_0 \tag{11}$$

which remains zero phase. Adding a delay of 2 allows us to make a causal filter which has frequency response:

$$H(e^{j\omega}) = h_2 + h_1 e^{-j\omega} + h_0 e^{-2j\omega} + h_1 e^{-3j\omega} + h_2 e^{-4j\omega}$$

$$= e^{-2j\omega} [\, 2h_2 \cos(2\omega) + 2h_1 \cos(\omega) + h_0 \,] \tag{12}$$

where we have been able to take a common exponential factor, representing exactly the linear phase, outside the zero-phase result. Another interesting case would be a length six filter:

$$H(e^{j\omega}) = h_2 + h_1 e^{-j\omega} + h_0 e^{-2j\omega} + h_0 e^{-3j\omega} + h_1 e^{-4j\omega} + h_2 e^{-5j\omega}$$

$$= 2\, e^{-(5/2)j\omega}[h_2 \cos(5\omega/2) + h_1 \cos(3\omega/2) + h_0 \cos(\omega/2)] \tag{13}$$

which is interesting in that the delay is 5/2, not an integer, and the sum of cosines represents a series of odd multiples of the argument $\omega/2$. We note that the delay (τ) in the linear phase is exactly the center of the impulse response. If there are N terms, than the delay is (N-1)/2.

We note that the expansions of equations (11), (12), and (13), which grow out of the DTFT, equation (2), make it obvious that the series expansion represented by the DTFT is mathematically equivalent to the Fourier Series, except we have interchanged the roles of time and frequency. In as much as we are really still doing a Fourier Series, we expect the effects on the magnitude response that are the result of truncation to be the same as those for the Fourier Series. In particular: the "Gibbs phenomenon."

2b-2 Truncation and the Gibbs Phenomenon

Gibbs phenomenon is a popular topic in math courses, but is fairly difficult to understand. In this regard, signal processing engineers are in a very favorable position to treat it is almost obvious, because we understand the ideas involving the interchange of convolution and multiplication operations in Fourier transform domains. It is clear that truncation of an impulse response is equivalent to multiplying the impulse response by a discrete-time rectangular window. Suppose we have an impulse response whose infinite duration version is an ideal low-pass (its DTFT is a rectangle in frequency). In the frequency domain we need to convolve this with the DTFT of a rectangular window. Fortunately, we have already calculated this - we just need to remove the (1/N) from equation (6). It's the so-called Dirichlet function.

Fig 4 shows the zero-phase version of the DTFT of a rectangular window of length 21 (purely real, so we can show the function itself, and not just the magnitude seen in Fig. 2). (Actually we are only looking at one cycle of this, since it is periodic) This we need to convolve with the block edge of the ideal low-pass. In a case where the window is fairly long, the wiggles of the Dirichlet function are close together relative to the width of the ideal low-pass Accordingly we look at the convolution as the Dirichlet function passes through the edge of the low-pass.

When the Dirichlet function's center is far from the edge, there is little overlap (Fig. 4a). As it moves slightly, the convolution integral wavers up and down about zero slightly as the sidelobes of the Dirichlet function that are inside the low-pass add and subtract. As the Dirichlet function invades the low-pass edge further, the rippling increases (Fig 4b). At some point, the main lobe of the Dirichlet enters the low-pass, and we get a rapid, dominating departure from zero. This is the transition region from the stopband to the passband, and it will be at its maximum just when the main lobe is completely inside (Fig. 4c). Now, as the Dirichlet continues inside, the contribution of the first sidelobe is negative, and subtracts from the peak (Fig. 4d). (Note that there was a similar peak at the edge of the stopband, but this was difficult to describe on the way in.) Now, the height of the ripple at the transition edge is directly related to the difference in the areas of the main lobe, and the first sidelobe. This first sidelobe of the Dirichlet function, as the length of the window increases, is never smaller than the first sidelobe of the corresponding sinc function, which is 0.2122 times the peak. Note also that it the first sidelobe is only half as wide as the main lobe. We thus can estimate the minimum height of the ripple on the transition edge by assuming that the lobes of the Dirichlet have the same shape, so the areas are proportional to the heights times the bases. Thus we estimate the area of the main lobe as 1×2 and the area of the first sidelobe as 0.2122×1, for a ratio of 10.6%.

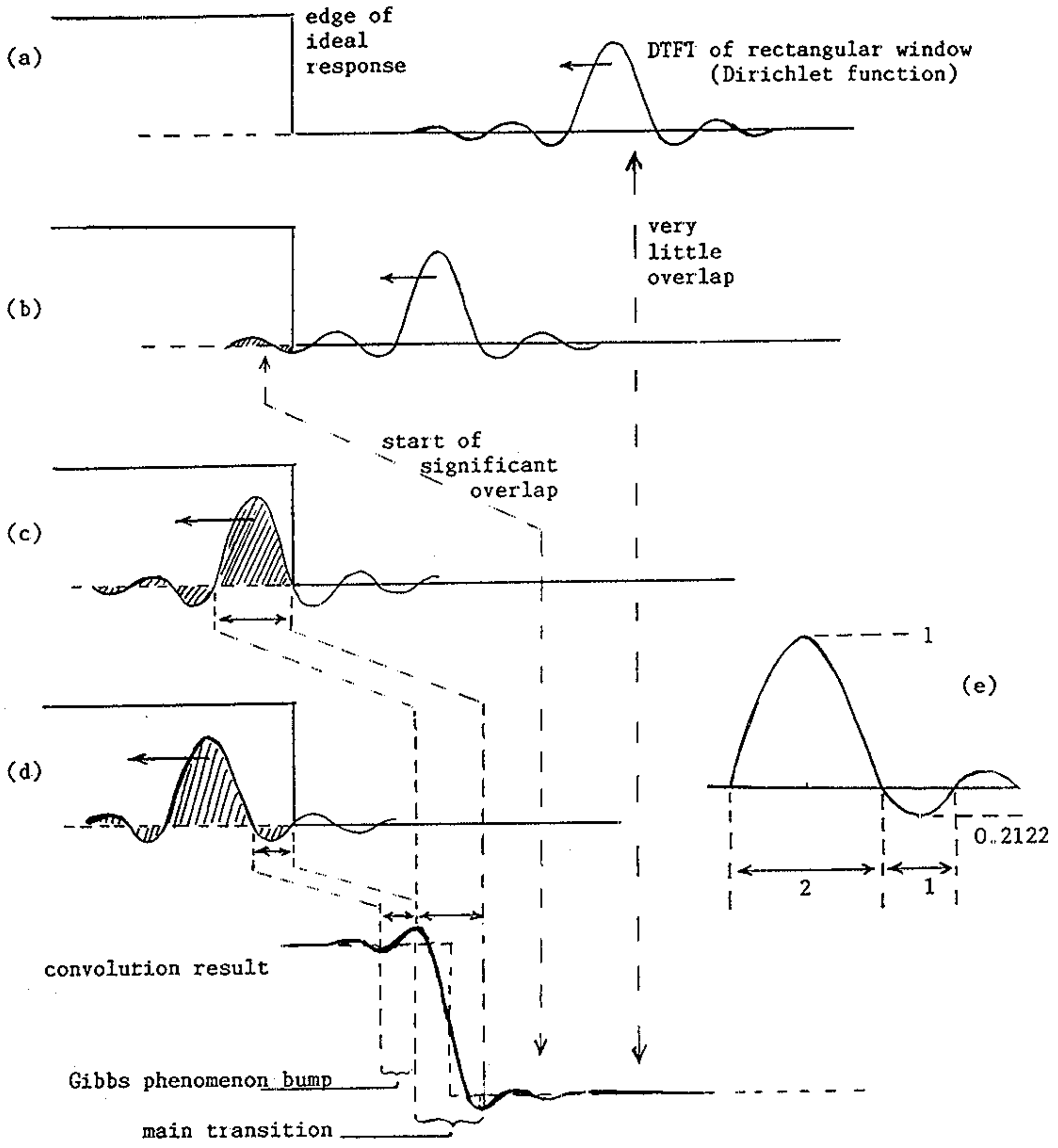


Fig. 4 Truncating an impulse response is the same as multiplying the impulse response by a rectangular window, in the time domain. Thus we convolve, in the frequency domain, a Dirichlet function (periodic sinc) with the ideal response, assumed to be a sharp transition. Here we assume that the Dirichlet function is narrow compared to the ideal response. This means that the window is comfortably long. In the (a) stage of the convolution the Dirichlet function is just approaching, causing minor ripples in the response. At (b) the ripples are getting larger near the edge. At (c), the main lobe has entered the block edge of the ideal frequency response, causing the large transition to the passband. At (d), the first sidelobe has entered, bringing the response back down, completing the major peak at the edge. In (e) we roughly compare the main lobe and first sidelobe of the Dirichlet function.

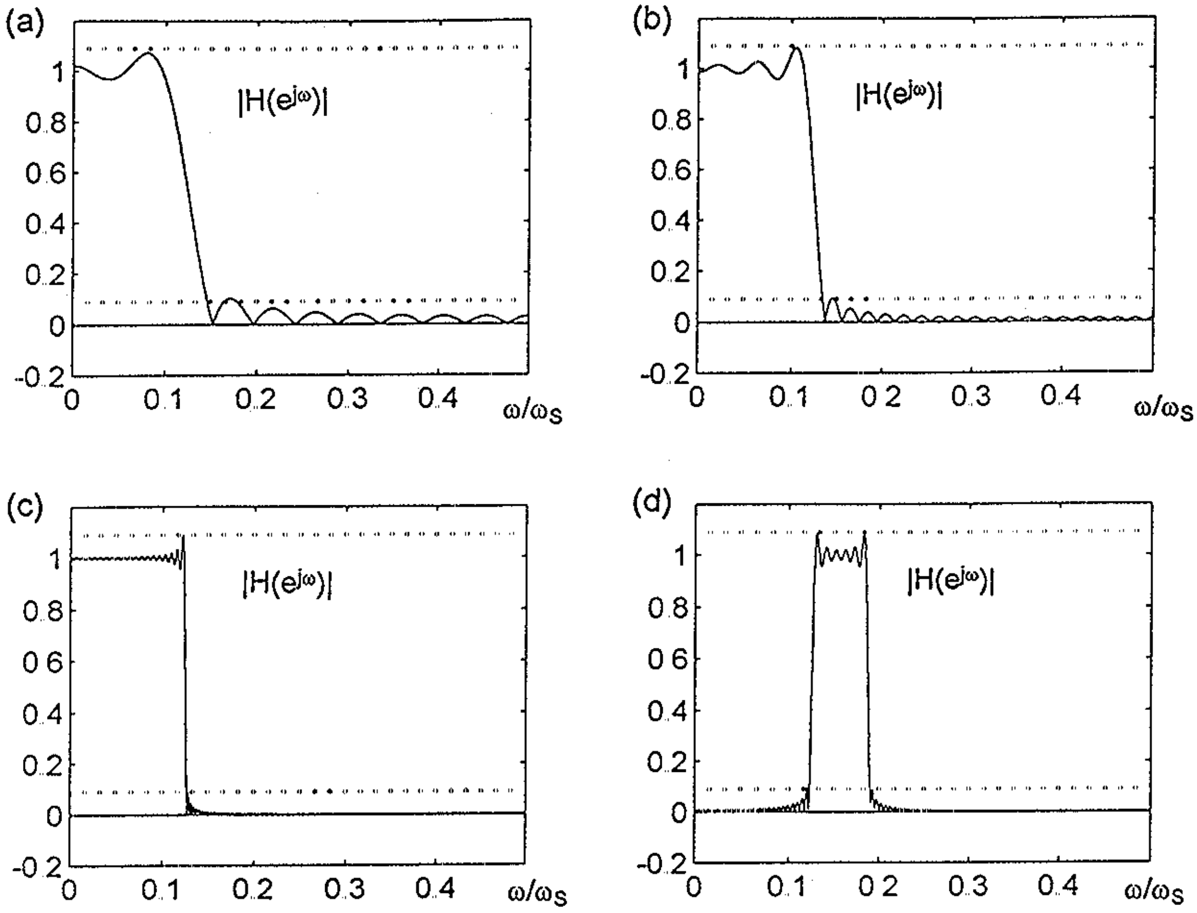


Fig. 5 Four examples of Gibbs phenomenon due to truncation. In (a), (b), and (c) we have low-pass filters with a cutoff at 1/8 (0.125) of the sampling frequency. In (a) the length is 21, in (b) it is 51, and in (c) it is 311. In (d) we have a bandpass with a passband from 2/16 to 3/16 of the sampling frequency, with a length of 201. This bandpass is formed easily be subtracting a low-pass with cutoff 2/16 from a low-pass with cutoff 3/16. The DTFT is linear so we can see this easily in the frequency domain, and realize it with a subtraction of impulse responses in the time domain. All these shows Gibbs phenomenon - the dotted lines correspond to magnitude levels of 0.09 and 1.09. Note the Gibbs phenomenon peaking on both edges of the bandpass. The short length (a) is not quite typical - we are seeing an effect of a significant leading tail of the transform of the rectangular window already leaving the negative side even as the main lobe is just entering. That is, the transform in this case is too "fat" for the picture of Fig. 4 to be completely valid.

This tells us two things, the first of which is the good news that we finally can explain Gibbs phenomenon. The bad news is that it has followed us into the digital filter design. What it says is that no matter how long we make our filter, we are going to have at least the 10% peaks on the edges (see Fig. 5 for example). Admittedly, these get more and more narrow as the length increases, but they don't keep getting smaller. We will need to look at ways of controlling the Gibbs phenomenon, and these are of course the same as for the Fourier Series itself for the similar cases. One way will be to give up the perfectly sharp transition of the filter for something more gradual. The second way will be to taper the taps at the ends so that truncation is gradual rather than abrupt. That is, we use something more gradual than a rectangular window. In particular, we will be willing to choose a window with a wider main lobe (resulting in a wider transition region) as long as sidelobes can be made much smaller.

2b-3 Modifying the Desired Response

Gibbs phenomenon is associated with two things: the Fourier series of a function with sharp discontinuities (thus requiring an infinite number of series terms), and the truncation of the Fourier series to a finite number of terms. In the cases of our filter designs, where the DTFT which gives us the frequency response is analogous to the Fourier series (with the roles of time and frequency reversed); it is the ideally sharp filter cutoffs, along with truncation of the impulse response that generates Gibbs phenomenon. In this section, we will attempt to control Gibbs phenomenon by relaxing the demand for an infinitely sharp cutoff region [1]. Instead we will look for a more gradual transition region (of some form) between the passband and the stopband. [In Section 2b-4, we will look at the alternative approach of modifying the sharp cutoff of the impulse response using a technique called "windowing." When we get to the point where this inverse DTFT approach is seen to be a special case of least squared error (Section 3a), we can defeat Gibbs phenomenon by specifying zero weight on the error in a transition band.]

Our first attempt at a gradual transition is shown in Fig. 6, where we show a linear transition from 1 at $\pi/2$ to 0 at $9\pi/16$. This transition we select instead of a perfectly sharp step from 1 to 0 at $17\pi/32$, which is the exact center of the sloped transition. While it is possible to just write the equation for the sloped region, and integrate this along with the rectangle, using equation (4), the inverse DTFT, this is tedious. Instead we will take a well-known shortcut of considering a trapezoid to be the convolution of two unequal width rectangles. In fact, the trapezoid of Fig. 6, $H(e^{j\omega})$, is the convolution of a rectangular of width $17\pi/16$ (cutoff $17\pi/32$), with a rectangle of width $\pi/16$ (cutoff = $\pi/32$). Both of these rectangles are themselves low-pass filters, and their impulse responses are obtained directly from equation (10). It is convenient to set τ to 0, so we will be able to compare zero phase filters. Thus we will be obtaining impulse responses as:

$$h(n) = (\omega_c/\pi)\sin(n\omega_c)/(n\omega_c) \qquad (14)$$

Convolving the two frequency responses is just a matter of multiplying, point by point, the two impulse responses From Fig 6, we see that there is a considerable reduction in Gibbs phenomenon through the use of this linear transition (the filter with a sharp transition at 17π/32 is shown for comparison). But the linear transition region is not a particularly natural filter shape. In fact, the FIR digital filter, as the sum of sinusoidal waveforms, does not support either a discontinuous derivative, or a straight-line segment.

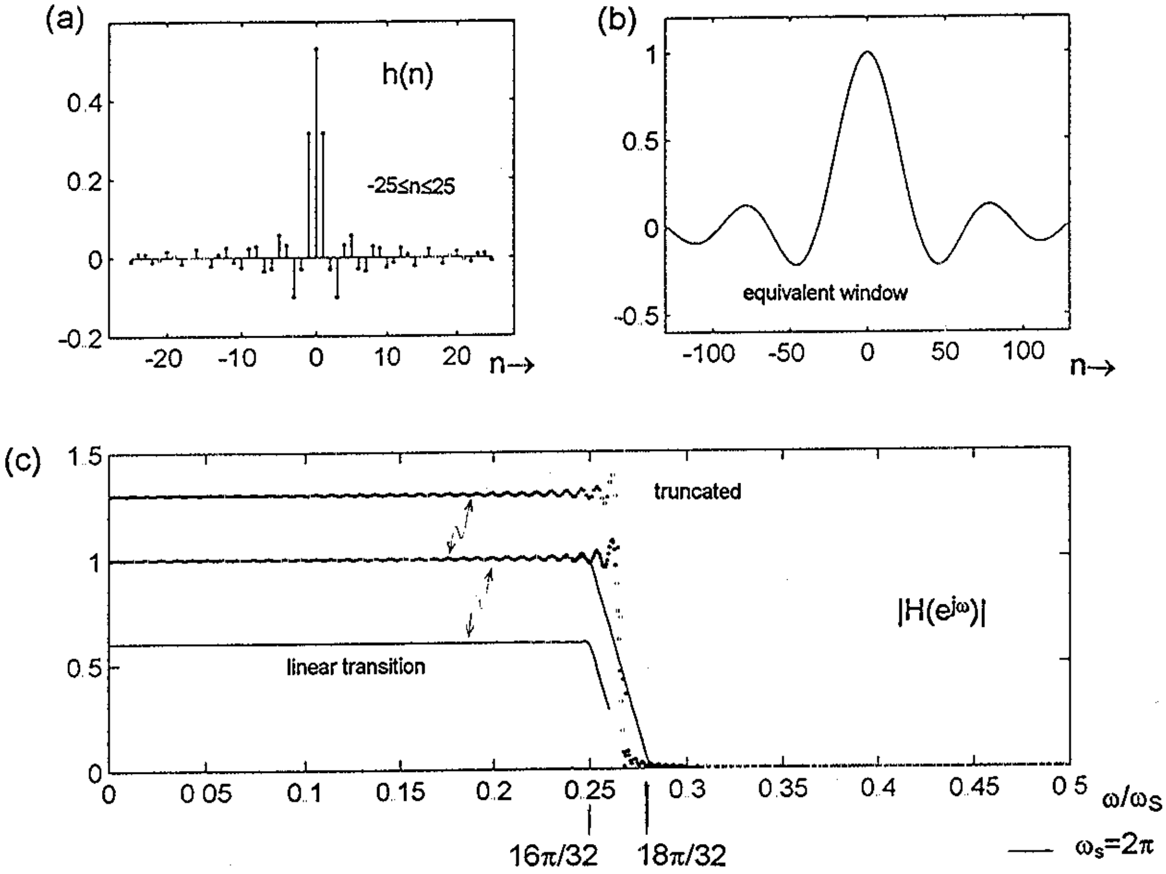


Fig. 6 Here in (c) we see a drastic reduction of Gibbs phenomenon through the choice of a linear transition region in (a) we look at the taps close to the center. Note that the taps corresponding to the linear transition design (lines) are tapering faster than the truncated design (dots). In (b) we divide all 255 taps of the linear transition design by the corresponding taps of the truncated design. It is no surprise that it is a sinc (we are just reversing the previous multiplication)

Fig. 7 shows another choice of shape for the transition region - a piece of a cosine This choice has the advantage of matching the slope (to zero) at the edges of the transition band, unlike the linear segment which presented a discontinuous slope at the ends Once again, it is possible to write an expression for the cosine segment, and to integrate equation (4). Once again this is tedious so we will look for an easier way, and will accept a result that is not quite general, but perfectly adequate for making our point.

Instead of choosing an arbitrary cosine, we will choose a cosine that is a natural expansion function of the DTFT. [For example, see equations (11-13).] In particular, we will consider 1/2 + (1/2)cos(16ω), which is plotted in Fig. 7 as $H_C(e^{j\omega})$. In fact we know that:

$$H_C(e^{j\omega}) = 0.25\ e^{-16j\omega} + 0.5 + 0.25\ e^{16j\omega} \qquad (15a)$$

which has impulse response, by inspection, as:

$$\begin{aligned} h_C(-16) &= 0.25 \\ h_C(0) &= 0.50 \\ h_C(+16) &= 0.25 \\ h_C(n) &= 0 \qquad n \neq -16,\ 0,\ \text{or} +16 \end{aligned} \qquad (15b)$$

so this is a known filter, sometimes called a "comb filter" because of its shape. But we only want a small portion of this response, the part from π/2 to 9π/16. Accordingly we need to multiply $H_C(e^{j\omega})$ by some $H_B(e^{j\omega})$, which is a bandpass from π/2 to 9π/16 in this case. We need to make this filter (by subtracting low-pass filters) and we need to do the multiply (by convolution in the time domain).

Suppose $H_{L1}(e^{j\omega})$ is a low-pass with a cutoff at π/2 and $H_{L2}(e^{j\omega})$ is a low-pass with cutoff at 9π/16, both of which have impulse response, $h_{L1}(n)$ and $h_{L2}(n)$, obtained from equation (14). We can form the desired bandpass:

$$H_B(e^{j\omega}) = H_{L2}(e^{j\omega}) - H_{L1}(e^{j\omega}) \qquad (16a)$$

or:

$$h_B(n) = h_{L2}(n) - h_{L1}(n) \qquad (16b)$$

and the little cosine transition region is found by multiplying $H_C(e^{j\omega})$ with $H_B(e^{j\omega})$ which we do by convolution in time:

$$h_{CB}(n) = h_C(n) * h_B(n) \qquad (17)$$

We have then only to add on the rectangular low-pass below π/2 to obtain $H(e^{j\omega})$ as:

$$H(e^{j\omega}) = H_{L1}(e^{j\omega}) + H_{CB}(e^{j\omega}) \qquad (18a)$$

which we have by the impulse response:

$$h(n) = h_{L1}(n) + h_{CB}(n) \qquad (18b)$$

Here we will likely want to truncate h(n) to the same length as $h_{L1}(n)$.

The resulting magnitude response, shown in Fig. 7, along with the same comparison sharp cutoff low-pass at 9π/16, is seen to be an improvement. Because the cosine is actually a sharper cutoff than the linear transition, some Gibbs phenomenon remains.

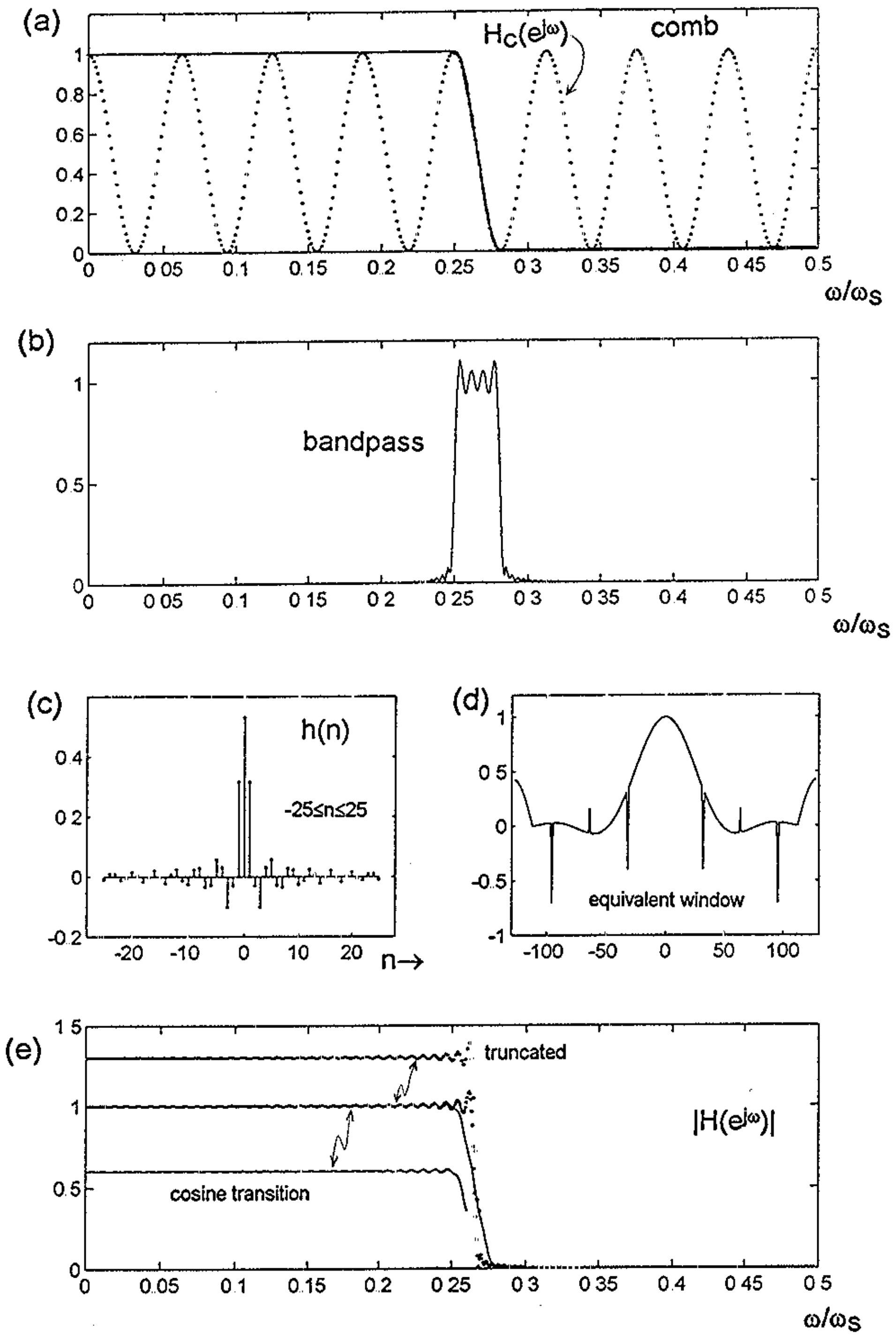


Fig. 7 A small segment of the comb filter (a - dotted), between 1/4 (=0.25) and 9/32 (=0.28125) is isolated by (length 255) bandpass filter (b) and becomes transition region of the overall desired response (a - solid line). This causes the taps to taper some (c) which is better seen by the "equivalent window" of (d). The resulting filter (e - solid), also length 255, shows much less Gibbs phenomenon as compared to the truncated design (e - dotted). While this filter shows more Gibbs phenomenon than Fig. 6 (c), the cutoff is much sharper.

2b-4 Windowing the Impulse Response

In Fig. 6 and Fig. 7 we plotted the impulse responses of the filters which we obtained through the modification of the desired filter response from a sharp to a gradual cutoff. This modification was one of our ploys for reducing the Gibbs phenomenon. For reference, we also plotted the impulse response of the corresponding sharp cutoff filters. What we found was that the impulse responses became tapered (gradually reduced to smaller and smaller values) at the ends as a result of making the transition band more gradual (less sharp).

It might not be evident that this is consistent with the "uncertainty relationships" that governs corresponding descriptions in Fourier-transform domains. In tapering the ends of the impulse response, we are actually de-emphasizing the ends and thereby emphasizing the middle, thus making the time-domain description effectively shorter. Thus we are locating the time domain with more precision. In return we find that the one "event" of interest in the frequency domain, the transition from a passband to a stopband, is less precisely defined.

Since we found an intentional reduction of the specified sharpness of the transition band to result (through the normal filter design procedure) in a tapering of the ends of the impulse response, a second ploy for reducing the Gibbs phenomenon might well be to first design the filter with the sharp transition, and then to artificially taper the impulse response. We do this by multiplying the original filter coefficients by what is called a "window." In fact, we can view the truncation from infinite length to finite length as a windowing with a rectangular window. Thus we wish to try a different window that is of finite length, but non-rectangular - tapering at the ends in fact. [Note that in using a linear transition we were effectively using a sinc window to taper the impulse response - see Fig. 6b.]

Without doubt the simplest of the windows of this type that is still effective is the so-called Hamming window. The result of using this window gives us perhaps 95% or more of what we intended a window to accomplish. [Of course finding a window that is slightly better than the Hamming window for a particular application is something we must consider. As with the matter of finding the best filter coefficients themselves, the "cost" of doing something "absolutely right" averages to almost nothing.] Here and in many other studies, the Hamming window has an advantage of having a simple mathematical form (a raised cycle of a cosine) and thus being easy to understand, and as such, it sheds much light on windows in general.

The Hamming window is one full cycle of a raised cosine. Taking this cycle from $t=-\pi$ to $t=\pi$, we could choose:

$$W_H(t) = 0.54 + 0.46\cos(t) \tag{19a}$$

as sketched in Fig. 8a. The proportion of 0.54 constant and 0.46 cosine is determined in a simple manner so as to minimize the first sidelobe in the CTFT of the time window. When we go to make this into a length N discrete time window, we need to sample the full cycle exactly once in N samples. Thus the samples are $2\pi/N$ apart. Since we are

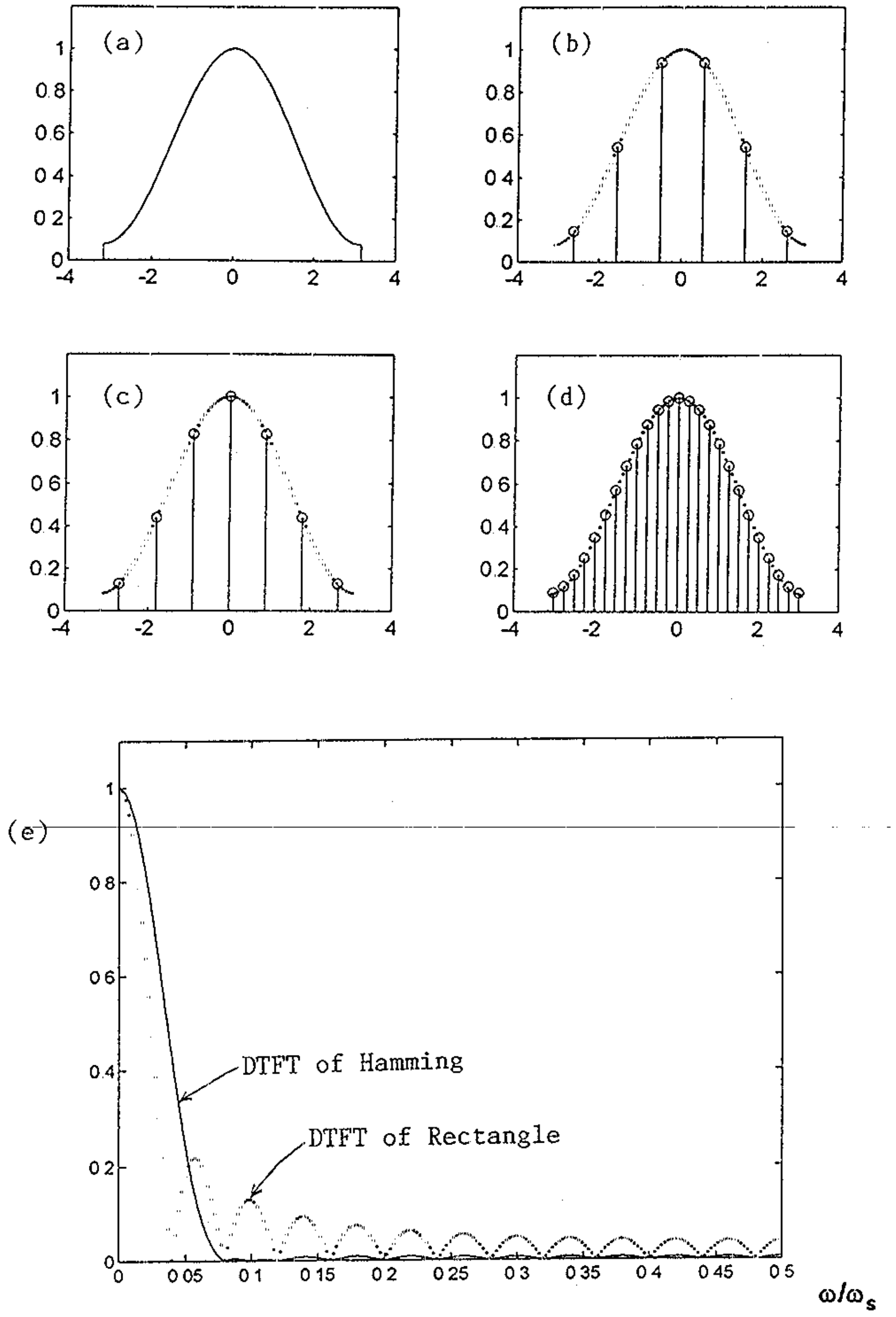


Fig. 8 ı he continuous-time Hamming window in (a) is a raised cosine. Proper sampling of exactly one cycle is shown in (b) for an even number of samples, 6, and in (c) for an odd number of samples, 7. In (d) we have a length 25 Hamming window. The magnitude of the DTFT of the length 25 Hamming window is compared to that of a length 25 rectangular window in (e). We note most importantly that the transform of the Hamming window is wider, but its sidelobes are only about 1% as compared to the sidelobes (as much as 22%) of the rectangular window. Note the untypical first sidelobe of the Hamming window which has an extra midlobe zero (seen at 0.1) with the Hamming window defined as we have here.

going to use this window to weight filter taps, for filters that are almost always supposed to be linear phase, we want to keep the window itself symmetric (linear phase). If N is odd, the center sample is 1. If N is even than there is no center sample - two samples symmetric about the center are close to 1. There are no samples taken at π or at $-\pi$. If such a sample were taken, then both would have to be included (for symmetry), and this would be one full cycle plus one extra sample. [Many sources make this error. It is of little consequence - just wrong.] Thus a general form for the discrete time Hamming window could be [2]:

$$W_H(n) = 0.54 + 0.46 \cos(\pi/N + 2n\pi/N) \quad n = 0,1,\ldots N-1 \tag{19b}$$

some examples of discrete Hamming windows are shown in Figures 8b through 8d.

Fig. 8e shows the DTFT of a length 25 Hamming window, along with the DTFT of a length 25 rectangular window. The Hamming window differs from the rectangular window in two important ways. First, its main lobe is twice as wide. Secondly, its first sidelobe is very small, about 1%, as compared to the 22% of the rectangular window.

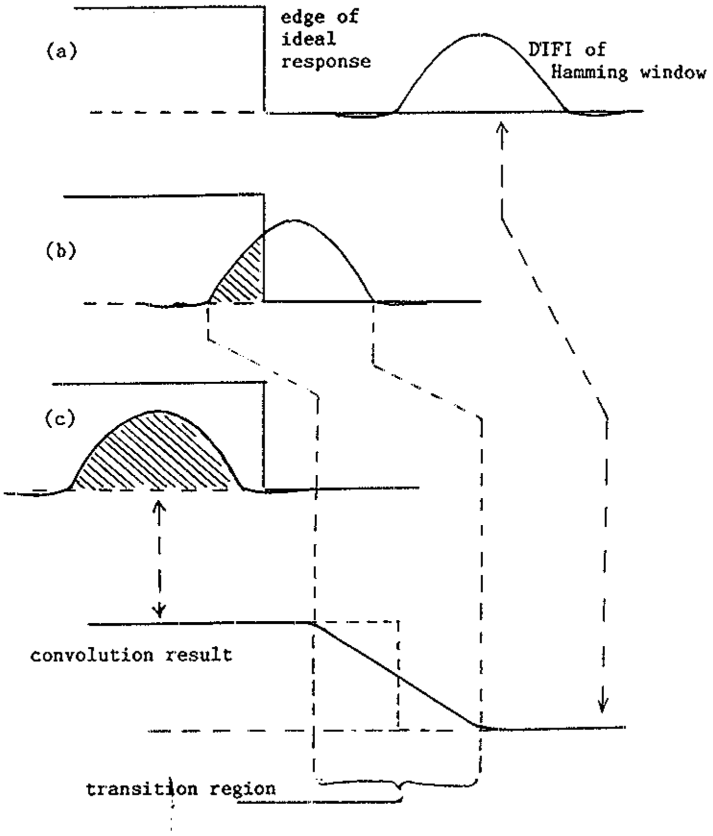


Fig. 9 Here, in comparison to Fig. 4, the convolution of the DTFT of a Hamming window with the edge of the ideal filter does little more than stretch out the transition region. This is because the sidelobes are so small. The transition region is twice as wide, however.

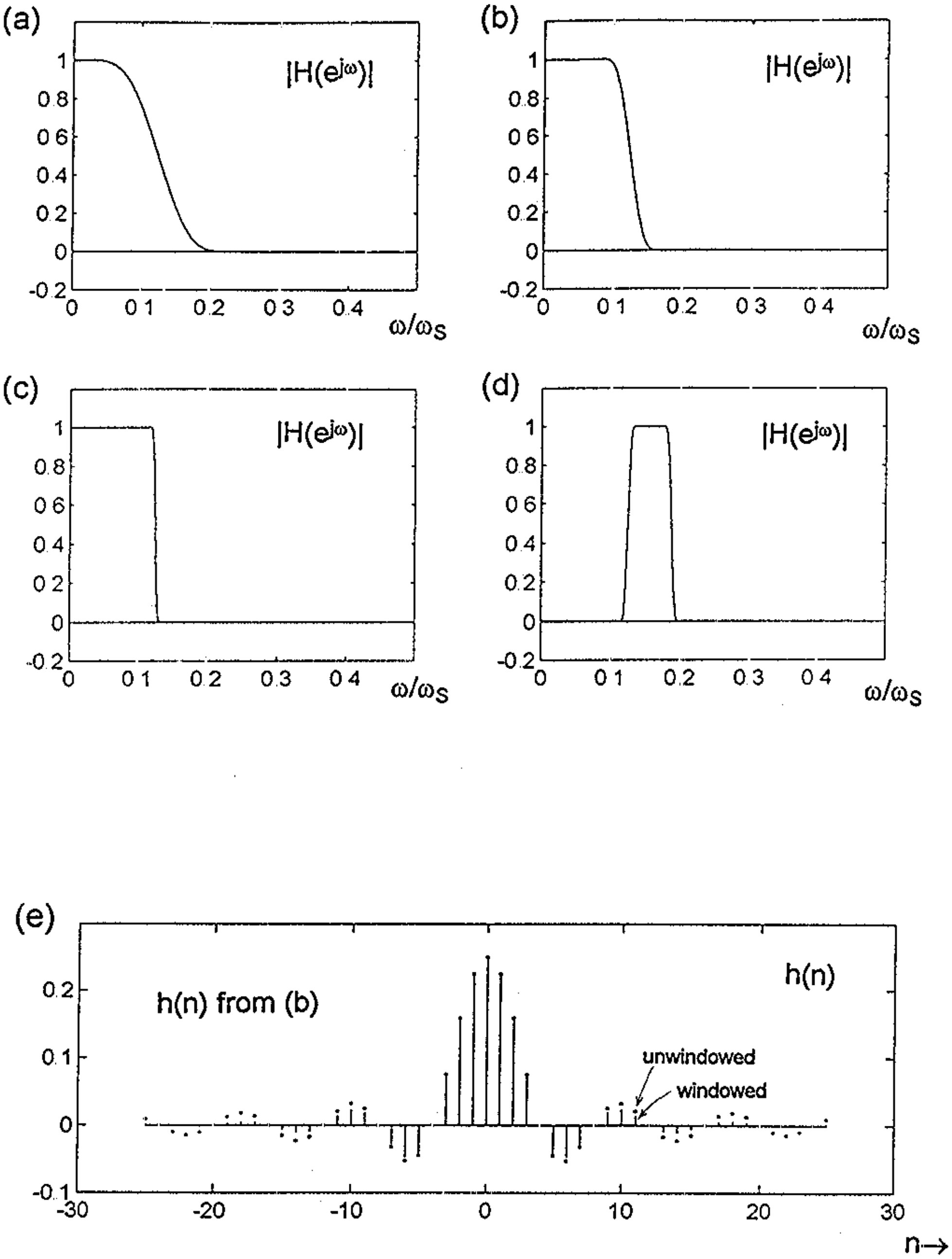


Fig. 10 Here we show the filters of Fig. 5, but have applied a Hamming window to each case. The lengths again are 21, 51, 311, and 201. In (e) we show the impulse response corresponding to (b) as lines, along with an overplot of the unwindowed impulse response (dot).

The significance of the wider main lobe and very small sidelobe can be seen when we now consider that multiplication of an infinite duration impulse response by the Hamming window will result in the convolution of the DTFT of the Hamming window with a sharp edge (Fig. 9). Reformulating our original Gibbs phenomenon discussion, we see that as the main lobe passes through the sharp edge, the transition is drawn out by a factor of two. But once this transition is done, there is no sidelobe of significance to knock the frequency response back down. For the most part, the Gibbs phenomenon is gone. The price paid is a slower cutoff rate.

Fig. 10 shows four examples of the use of the Hamming window (a simple, tap-by-tap multiply of the impulse response and the Hamming window), which can be compared to the corresponding, rectangular window (simple truncation) cases seen in Fig. 5.

2c. FREQUENCY SAMPLING - I

Many engineers have memorized two notions about the "frequency sampling" method of FIR filter design. First, it involves taking the inverse DFT of frequency samples. Second, it is not very useful. The first notion is correct, but incomplete in not involving essential details and useful modifications. The second notion is not unreasonable if the first were the whole story. What we will find is that the method as given in this section offers us little that is new - except insight. In Section 3b where we generalize the method, it can offer some extremely useful capabilities - for example, setting an adjustable zero exactly on a frequency we want to reject completely.

2c-1 Getting the Phase Specification Right

Earlier we mentioned that when we do not pay specific attention to phase, we are likely to end up with a zero-phase (non-causal) filter or worse. Ignoring the phase with frequency sampling can lead to nonsense. Suppose for example that we want a low-pass filter, and we intend to begin with samples of some desired frequency response. We might suppose that these samples will begin with a series of ones (low frequencies - the passband) and then end up with a series of zeros (high frequencies - the stopband). Yet we must remember that the DFT, like its "parent" the DTFT, has a negative side, which is, in the case of the DFT however, wrapped around to the upper portion of the positive side, by virtue of the chosen definition. [We are accustomed to having the DTFT defined on an interval $-\pi$ to $+\pi$, while the DFT is defined on the interval 0 to $+2\pi$, with its samples indexed k=0 to k=N-1. Thus the upper half of the DFT is the negative frequency side of the DTFT. This strange situation is likely a consequence of the fact that when we go to a discrete description, even and odd case would need to be handled differently, unless we do count from 0 to N-1.]

Now, recognizing the disguised version of the negative side of the spectrum, we might try the following samples for a desired low-pass filter of length 31:

D(k) = 1 k=0,1,2,3,4,5,6,7 (passband - positive frequencies)
D(k) = 0 k=8,9,10......22,23 (stopband) (20a)
D(k) = 1 k=24,25.....29,30 (passband - negative frequencies)

Note that there are eight samples of value 1 on the low end, and only seven samples of value 1 on the high side. This is correct. But note that we are ignoring phase. Yet we are now in a position to easily obtain h(n) as the inverse DFT of these D(k), using equation (8). Fig. 11a shows us the result, which looks a bit strange, since we are accustomed to seeing a sinc-like shape for low-pass impulse responses. Fig. 11b

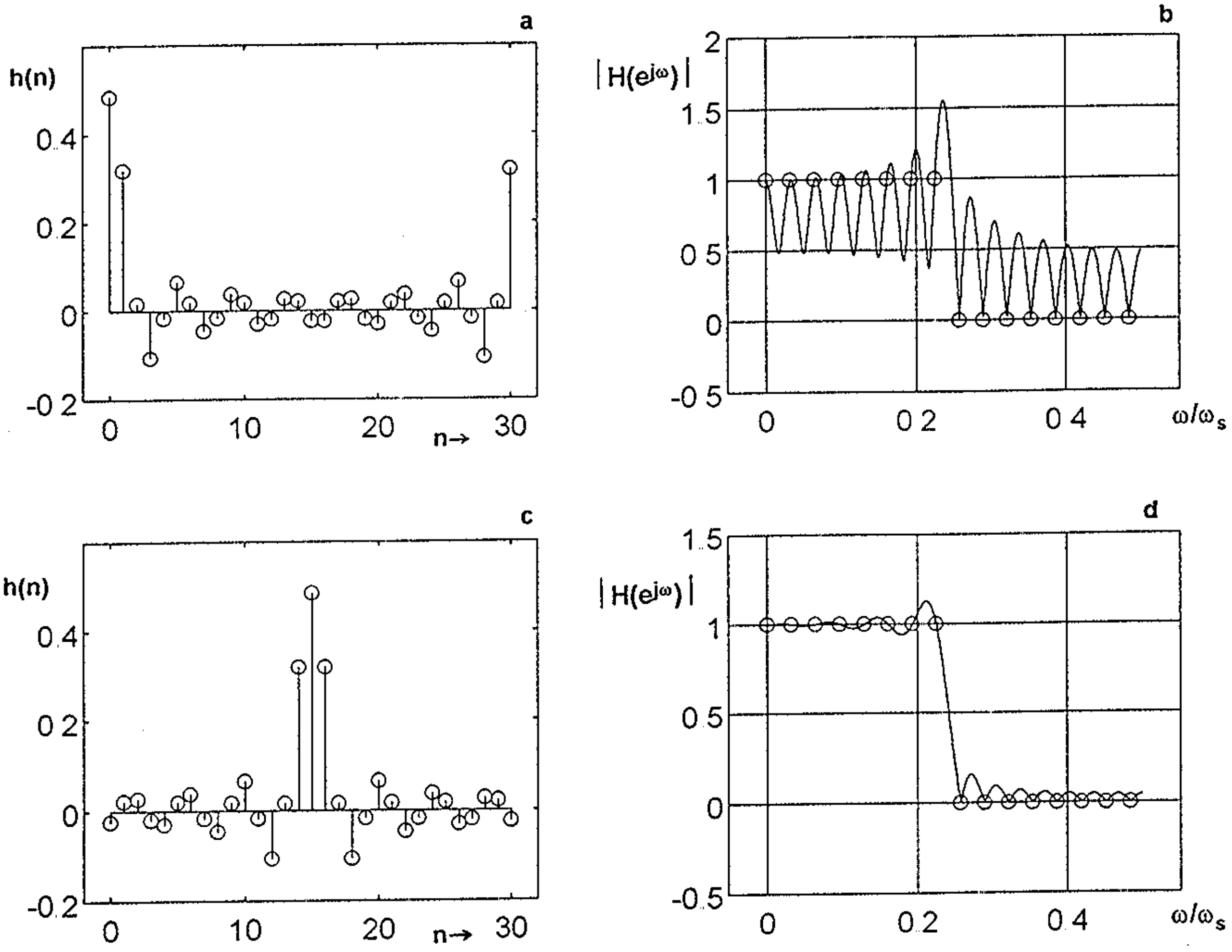


Fig. 11 Since the DFT is defined on n=0,1,2,.....(N-1), we can not design a zero-phase filter. If we try to do so by making all the samples 1 or zero, we get the rotated impulse response (a) with corresponding magnitude response (b) which does go through the specified points, but which has too much ripple. Including to correct phase rotates the impulse response correctly (c) to a sinc-like shape, and the corresponding magnitude response (d) is what we actually had in mind.

shows the corresponding magnitude response. Here we use the DTFT, equation (2), to find the response for all ω. Note that the response, as promised, does go through the specified samples (eight passband 1's and eight stopband 0's). It just was not supposed to ripple so much.

Intuitively a good engineer would likely recognize that the impulse response was somehow accidentally rotated. This can be fixed as in Fig. 11c so that it looks sinc-like, and the corresponding magnitude response in Fig. 11d verifies a better result. But note well that Gibbs phenomenon is present here as well.

Intuitive fixes are all well and good, at least as opposed to having no fix at all. Yet we have seen that for a length N FIR filter, where n runs from 0 to N-1, the linear phase was $e^{-j[(N-1)/2]\omega}$. Here our frequency samples are $\omega=(2\pi/N)k$, where k=0 to k=N-1. So, choosing our samples with the correct phase would give (N=31 here):

$$\begin{array}{ll} D(k) = 1 \cdot e^{-j[(N-1)/N]\pi k} & k=0,1,2,3,4,5,6,7 \\ D(k) = 0 & k=8,9,\ldots\ldots 22,23 \\ D(k) = 1 \cdot e^{-j[(N-1)/N]\pi k} & k=24,25\ldots\ldots 29,30 \end{array} \qquad (20b)$$

This is correct, and using the inverse DFT of the samples of equation (20b) in fact reestablishes the result of the intuitive fix of Fig. 11c and Fig. 11d.

But there is yet another barb to avoid. In the case of an even number of samples, the phase for the samples k=(N+2)/2 to k=N-1 (the second half of the samples) are inverted. This is due to a zero at z=-1 in the z-plane that is automatic for even length. Other unit circle zeros (and there will generally be many such zeros for the usual type of filters that have stopbands) are paired in complex conjugates such that a phase shift of 2π results, and is the same as zero additional phase. For example, if the filter of equation (20b) were to be length 30 instead of length 31, we might choose (N=30 here):

$$\begin{array}{ll} D(k) = 1 \cdot e^{-j[(N-1)/N]\pi k} & k=0,1,2,3,4,5,6,7 \\ D(k) = 0 & k=8,9,\ldots\ldots 21,22 \\ D(k) = -1 \cdot e^{-j[(N-1)/N]\pi k} & k=23,24\ldots\ldots 28,29 \end{array} \qquad (20c)$$

The automatic zero at z=-1 with an even order filter applies to all design procedures, not just to frequency sampling. One important consequence is that we can't have even length high-pass or notch filters, but only low-pass or bandpass types. That is, we can't have a non-zero specification at half the sampling frequency while at the same time employing a length that insists on zero response at half the sampling frequency.

2c-2 Controlling the Gibbs Phenomenon

We saw in Fig. 11d that the same Gibbs phenomenon that plagued the inverse DTFT designs continues over to the frequency sampling designs. Controlling the Gibbs phenomenon here is very similar to the inverse DTFT case. We could, for example, try a Hamming window in a straightforward manner, and it would work. Alternatively we can consider the modification of the specified response to make it less sharp as we did in Section 2b-3. In the case of frequency sampling design, we think in terms of choosing transition band samples. This works, and intuitively we recognize that in specifying one or more samples in the transition band, since all the samples are equally spaced here, we are going to widen the transition band.

Fig. 12 shows a length 19 low-pass filter where the magnitudes of the 19 frequency samples are chosen so that the first five and last four at 1, with the middle 10 samples equal to zero. We note the considerable ripple about the transition edge, and a corresponding stopband rejection that is only -16db in the first lobe. Increasing the length of the filter will not improve the stopband. Fig. 13 shows a case where the passband is defined by sample magnitudes 1, 1, 1, .85, .5, .15, 0, 0, 0, 0, 0, 0, 0, 0, .15, .5, .85, 1, and 1. So a sharp transition between two samples is softened to include a range of four samples spaces total. The result is outstanding in that the stopband rejection excess -43 db, but clearly the transition region is far less sharp

We will revisit frequency sampling in Section 3b.

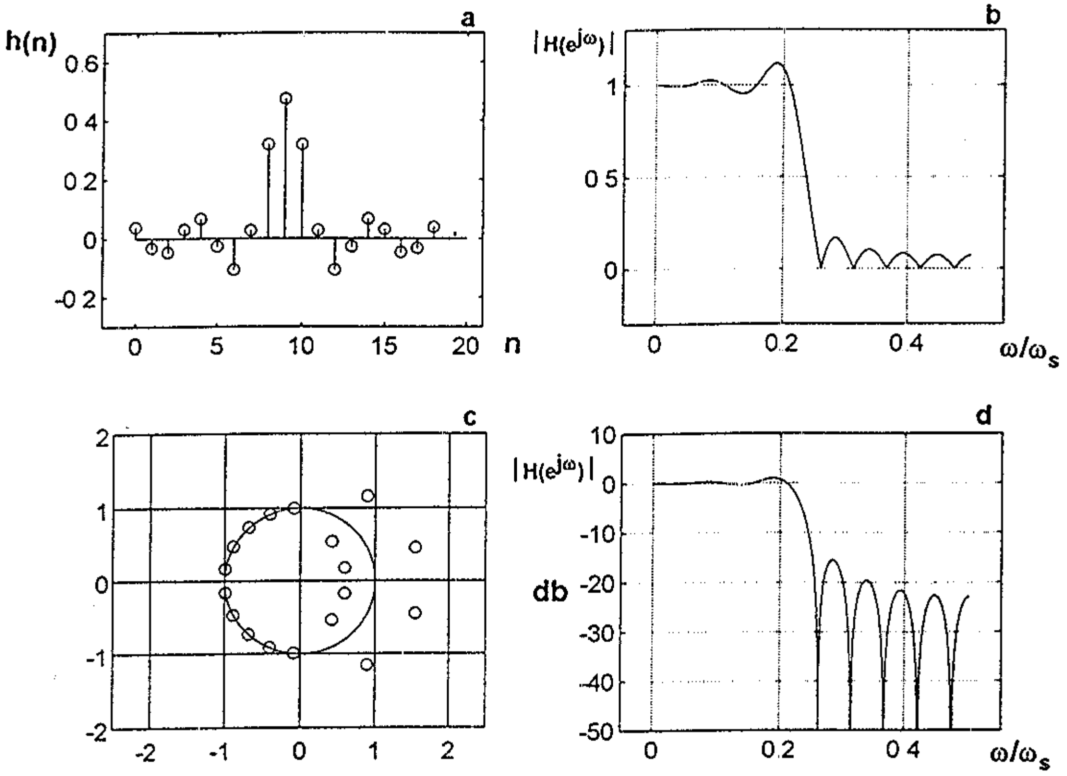


Fig. 12 Here a length 19 filter with samples of magnitude: 1,1,1,1,1,0 0 0 0 0 0 0 0 0 0 1 1 1 1 (sudden transitions) is shown. Linear phase is included in the specification actually employed here. Gibbs phenomenon, and the associated large sidelobes in the stopband are seen.

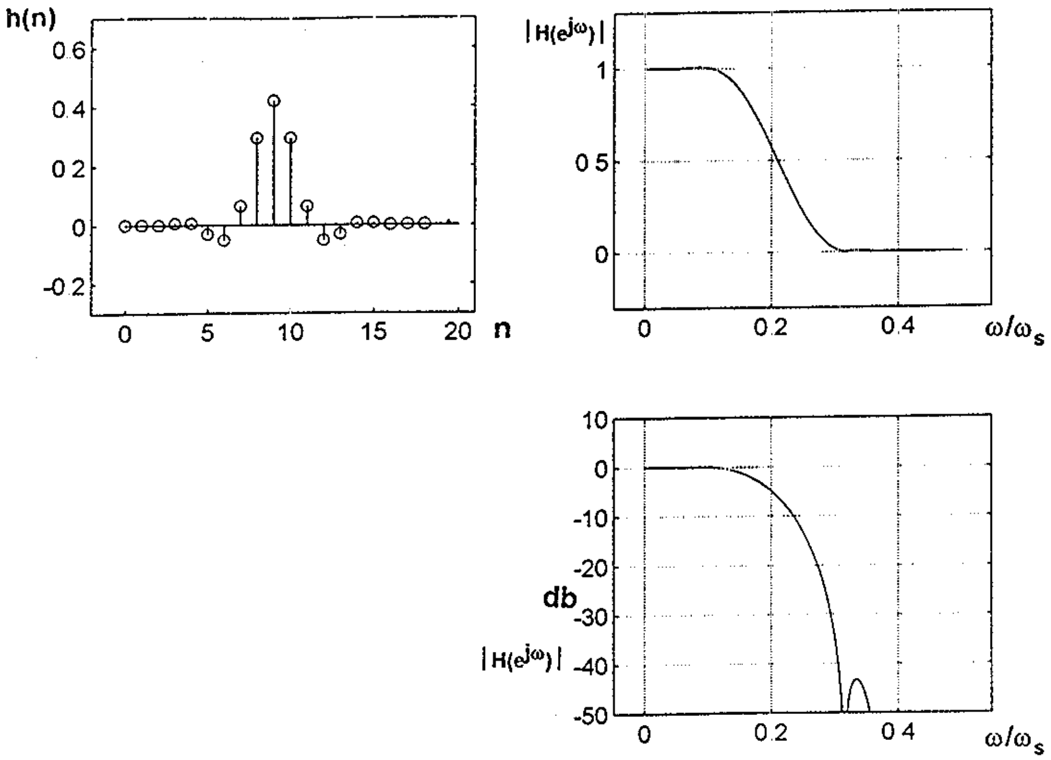


Fig. 13 Length 19 with frequency samples of magnitude 1 1 1 .85 .5 .15 0 0 0 0 0 0 0 0 .15 .5 .85 1 1 (gradual transition) removes Gibbs phenomenon, providing much better stopband rejection, but at the usual price of a much less sharp cutoff region.

2d. BILINEAR Z-TRANSFORM

This design method is completely different from what we have done above with FIR filters. What it has in common with the other methods in Section 2 is that the response is specified in the frequency domain. This method designs IIR filters, and is based on prototype analog filters. What one does is first choose an analog prototype, and adjust its frequency specifications a bit, and then make a substitution:

$$H(z) = T(s) \Big|_{s \leftarrow (2/T)(z-1)/(z+1)} \qquad (21)$$

Where T(s) is the transfer function of an analog filter and H(z) is the corresponding digital filter. Here, T is the sampling time, or $1/f_S$, f_S being the sampling frequency. Because this is a substitution, we hope to preserve the general shape of the frequency response.

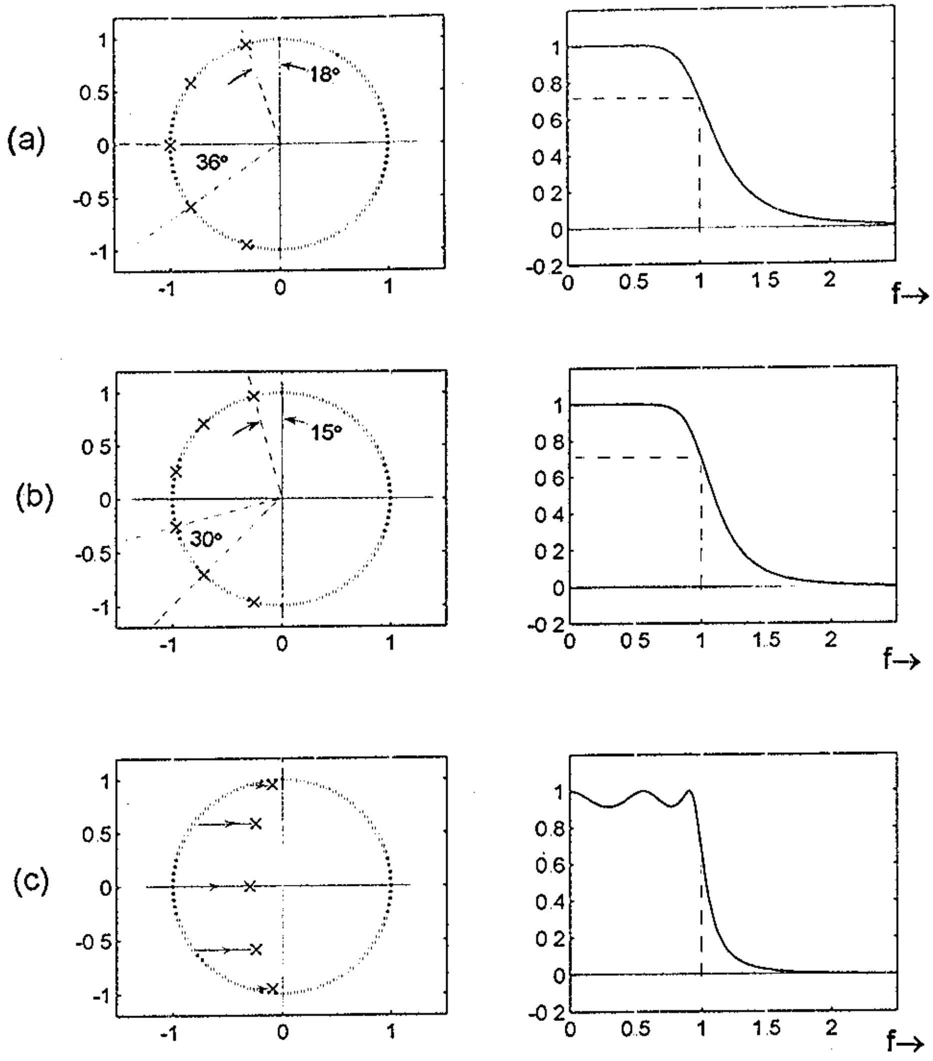
2d-1 Analog Filters

Analog filters come in great variety, with properties associated with the frequency dependent impedances of capacitors and inductors, or their equivalents (i.e., active filters). Associated with their widespread use over the years is a comprehensive mathematical foundation relating performance parameters to actual component values. It is quite natural that digital filtering has at time followed these same paths (perhaps originally with the idea of simulation rather than actual filtering). Yet the direct conversion of a well-cataloged analog filter to a digital version using something like equation (21) is extremely attractive. Here we will again be concerned mainly with low-pass filters. Conversions to other types is not generally difficult (Section 5a).

Not surprisingly, when we need a low-pass filter it is almost always the case that we want a passband of 1 and a stopband of 0, along with an acceptable transition band. In the analog art, there is a filter characteristic called "Butterworth" that is also called "maximally flat" which has an attractively flat passband and a monotonic transition and stopband (just a nice filter). It is difficult to imagine filtering applications where one would want an non-flat passband ("tone controls" and "equalizers" perhaps) so why would anyone choose anything other than Butterworth? The answer is that if we allow some "ripple" in the passband, we may achieve a superior transition-band performance (sharper cutoff) with no increase in resources. One class of filters that offers this option (an equiripple passband) followed by a sharper cutoff is the "Chebyshev" type. It is easy to calculate the Butterworth design data, and the Chebyshev design data can be easily derived from the Butterworth, along with a little computer-aided checking.

Analog filters are described by poles and zeros in the s-plane. The poles of an analog Butterworth low-pass lie on a circle centered about zero in the s-plane, and the radius of the circle is exactly the frequency at which the magnitude response of the filter falls to $1/\sqrt{2}$ (approximately -3db) of the DC gain (essentially the passband response). An Nth order filter has N poles, all equally spaced, and all lying in the left half of the s-plane (stability). One way to set the poles is to begin with 2N equally spaced points about our chosen circle. Now we rotate these points so that all points have complex conjugates, and so that no points are on the imaginary axis. Then we erase the set of points that are in the right half of the s-plane. The remaining points are uniquely determined and represent the poles of the Butterworth filter. In order to obtain a corresponding Chebyshev low-pass, we have only to reduce the real part of all the Butterworth poles by some factor. For example, we could perhaps make the real part of the Chebyshev poles 1/3 of the Butterworth values. While there exist precise formulas for the amount of Chebyshev ripple that comes from a particular order and reduction factor, we almost never have a value of ripple in mind. (We don't want ripple - there is an amount we will tolerate, and we would be delighted to end up with less.) Accordingly, we probably will just find an acceptable level of ripple by adjusting the reduction factor by trial-and-error. Fig. 14 reviews these procedures and shows some examples.

Once we have located the poles, p_k, it is a simple matter of writing down the analog transfer function T(s):



<u>Fig. 14</u> Placement of poles for analog low-pass filters. In (a) we have a 5th-Order Butterworth while in (b) we have a 6th-Order Butterworth. By simply moving the Butterworth poles toward the jΩ-axis, done by multiplying all the real parts by the same factor less than 1 (0.3 is used here), a Chebyshev low-pass is obtained (c). Note the faster cutoff rate of the Chebyshev 5th-order as compared to the Butterworth 5th-order, achieved at the price of more passband ripple.

$$T(s) = \frac{1}{(s-p_1)(s-p_2)\ \dots\ (s-p_N)} \tag{22}$$

We are now in a position to convert this to a digital version. (Or perhaps we have another T(s) which we have obtained in some manner.)

2d-2 IIR Digital Filters

Above we looked at the transfer functions of digital filters mainly in their from of frequency response functions $H(e^{j\omega}) = H(z=e^{j\omega})$. In this case, the coefficients of the transfer function were the actual impulse response values. The same will not be true of IIR digital filters. A typical IIR filter has a transfer function for M zeros and N poles as:

$$H(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \ldots\ b_Mz^{-(M-1)}}{a_0 + a_1z^{-1} + a_2z^{-2} + \ldots\ a_Nz^{-(N-1)}} \qquad (23a)$$

which is shown in polynomial form. It could also be converted to factored form as:

$$H(z) = \frac{(z-Z_1)(z-Z_2)(z-Z_3)\ldots\ldots(z-Z_M)}{(z-P_1)(z-P_2)(z-P_3)\ldots\ldots(z-P_N)} \qquad (23b)$$

where the Z's are the digital zeros and the P's are the digital poles. For the most part we will need the actual coefficients a_n and b_n as in equation (23a) to determine and realize the actual filter.

2d-3 Bilinear z-Transform

In as much as equation (21) has defined the substitution procedure, and we have the analog poles at hand, we might suppose that we would start by multiplying out equation (22), substituting using equation (21), and arriving at the form of equation (23a). However it is easily seen that if, for example we have 13 poles then the leading term in the denominator of equation (22) will be s^{13}, and so on. Making the substitution of equation (21) would thus lead to an algebra nightmare of some form. Instead it will be much easier to just map the analog poles (and zeros if there were any) one at a time from the s-plane to the z-plane. The mapping equation derives from the substitution of equation (21):

$$s = (2/T)\ [\ (z-1)\ /\ (z+1)\ ] \qquad (24a)$$

or

$$z = (2+sT)/(2-sT) \qquad (24b)$$

We will need to first understand mapping in frequencies, as derived from equation (24a), and then will use equation (24b) to avoid algebraic problems.

To understand the frequency mapping, recall that the z-plane and the s-plane map in general as $z = e^{sT}$. We see a problem. Somehow there have to be two different frequencies involved, one analog, and the other digital. We can thus write down:

$$j\Omega_a = (2/T)(e^{j\Omega_d T}-1)/(e^{j\Omega_d T}+1) \qquad (24c)$$

where Ω_a is the analog frequency and Ω_d is the digital frequency, both in radians/sec. This is easily simplified to the equation:

$$\Omega_a T/2 = \tan(\Omega_d T/2) \qquad (24d)$$

which is the "prewarping" equation. Actually it describes the relationship between analog frequencies from 0 to infinity and digital frequencies from 0 to half the sampling frequency. In this mapping, the entire frequency response of an analog filter is mapped into the upper half of the unit circle from z=1 to z=-1. The mapping of frequency is of course non-linear, close to linear for low frequencies, and highly compressed for high frequencies (Fig. 15). We note that this can lead to frequency

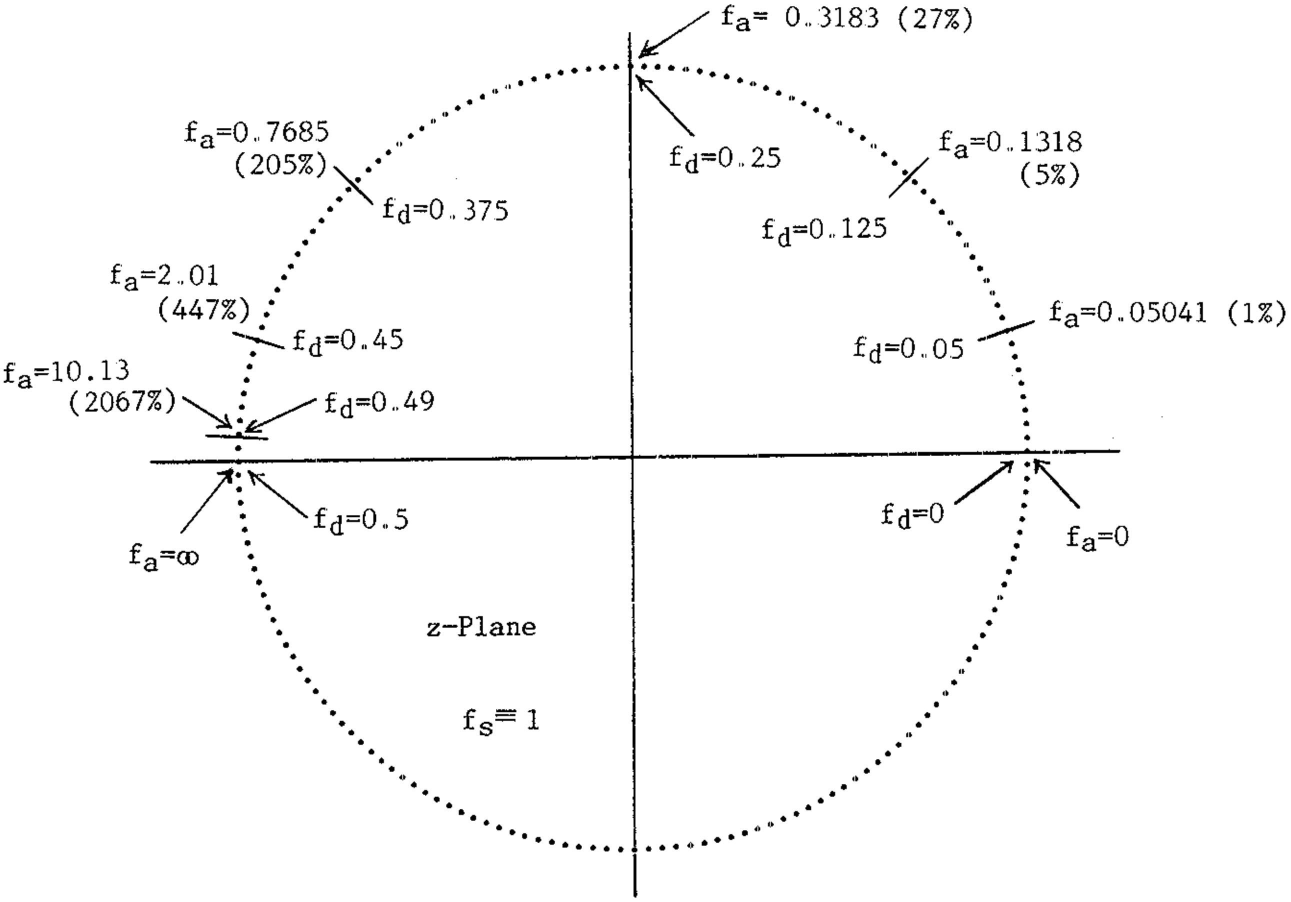


Fig. 15 The Bilinear z-Transform warps frequencies between the analog and digital domains. Equation (24d) can be written as $f_A = (f_S/\pi)\tan(\pi f_D/f_S)$ or as $f_A/f_S=(1/\pi)\tan(\pi f_D/f_S)$. As digital frequency runs from 0 to $f_S/2$ (the upper half of the unit circle, analog frequency runs from 0 to infinity, becoming more and more compressed as shown.

responses that are sharper in the digital case than they are in the corresponding analog case: usually a good thing. For example, a first-order analog low-pass can be transformed in a way that places the analog cutoff well around the unit circle (Fig. 16).

Once we recognize the general nature of the frequency mapping, equation (24d), it's utility is largely one of allowing us to set exactly one and only one point frequency exactly where we want it. Often this particular frequency is á cutoff frequency, but for such filters as bandpass or notches, we would more likely choose center frequencies. This is illustrated in Fig. 16, and will also be revisited in the second example below.

In Fig. 16, which was first-order, we were easily able to make the desired substitution. Yet above we alluded to the algebraic monstrosity of higher order. Fig. 17 shows an example where we desire to have a digital low-pass with a cutoff frequency $f_C = 5f_S/16$ (just a bit above a quarter of the sampling frequency), based on a 13th-order analog, Butterworth.

$\Omega_D = (3/8)\ \Omega_S = (3/8)(2\pi/T) = 2.356/T$
(choose digital cutoff)

$\Omega_A = (2/T)\ \mathrm{Tan}(3\pi/8) = 4.828/T$
(warp this for analog prototype)

$T(s) = 1\ /\ (s + \Omega_C)$
(analog prototype: -3db at Ω_C)
$\Omega_C = \Omega_A$
(match at cutoff, so.....)

$T(s) = 1\ /\ (\ s + 4.828/T\ )$

$$H(z) = \frac{1}{\frac{2}{T}\,\frac{z-1}{z+1} + \frac{4.828}{T}}$$

$$= \frac{T(z+1)}{6.828\ z + 2.828}$$

(substituted into equation 21)

Pole at $z = -2.828/6.828 = -0.414$
Zero at $z = -1$

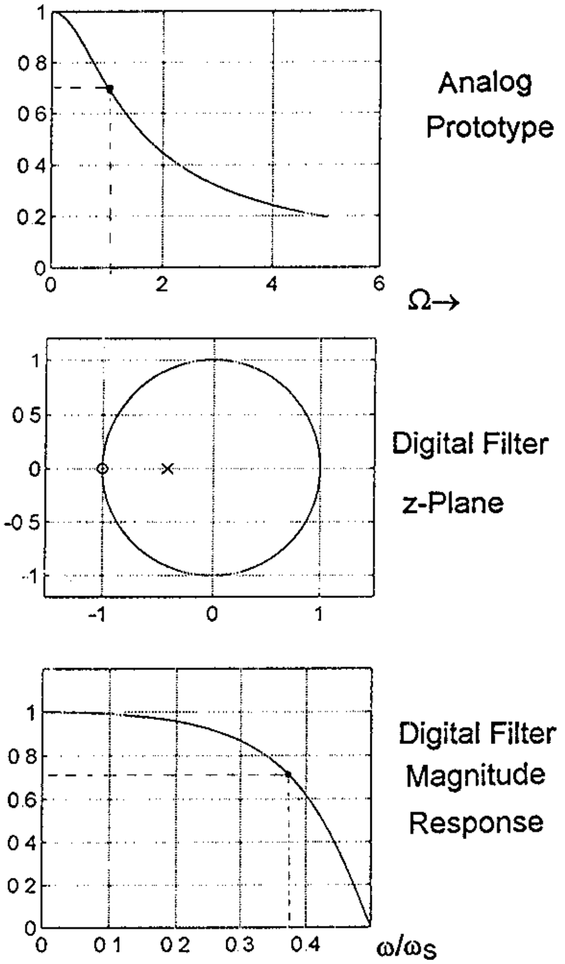
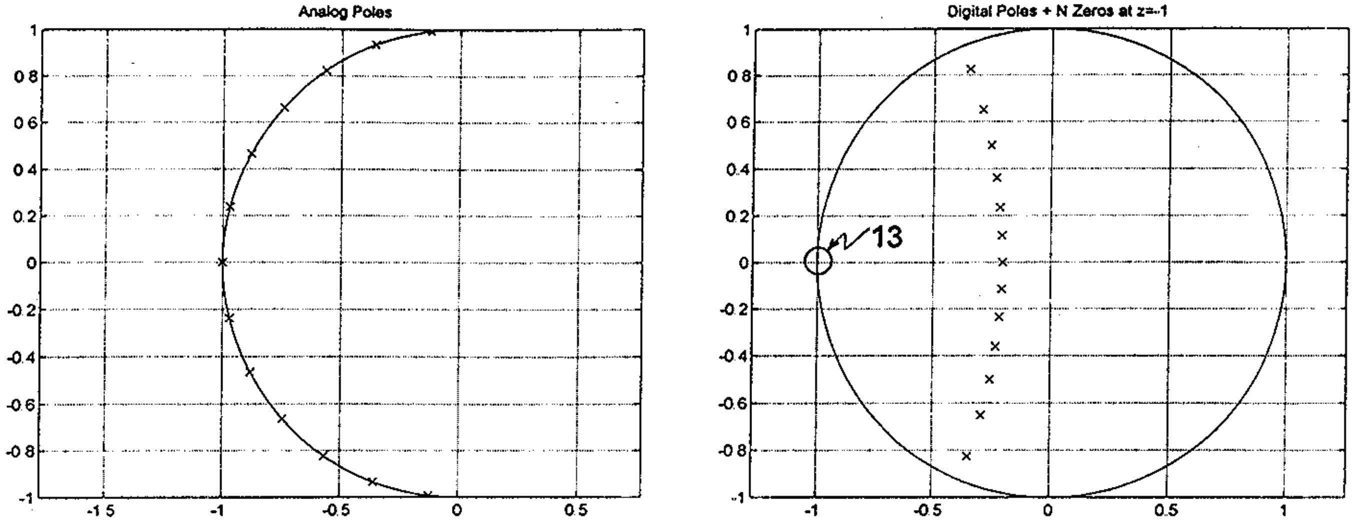


Fig. 16 A first-order Bilinear z-Transform example shows how "prewarping" is done, that direct substitution, equation (21) is possible in this simple case, and that a much sharper digital filter results here (because the digital cutoff is set fairly close to $f_S/2$).



Analog poles at radius 1 are scaled to $(2/T)\tan(5\pi/16) = 0.4764/T$.
The poles are then mapped as $z = (2 + sT)/(2 - sT)$

The resulting z-plane poles then yield H(z) as:

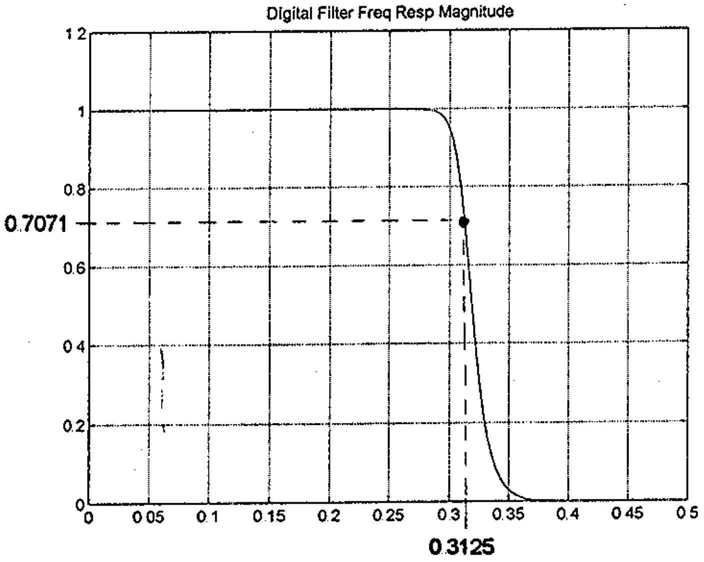
Fig. 17

13 Zeros at z=-1
or $(z+1)^{13}$

$$H(z) = N(z)/D(z)$$

$$N(z) = 1 + 13z^{-1} + 78z^{-2} + 286z^{-3} + 715z^{-4} + 1287z^{-5} + 1716z^{-6} + 1716z^{-7} + 1287z^{-8} + 715z^{-9} + 286z^{-10} + 78z^{-11} + 13z^{-12} + z^{-13}$$

$$D(z) = 1 + 3.24292929750979z^{-1} + 6.39126412353914z^{-2} + 8.50779056006800z^{-3} + 8.49496536419325z^{-4} + 6.51222905271538z^{-5} + 3.91994023831812z^{-6} + 1.85343402485817z^{-7} + 0.68541851572719z^{-8} + 0.19457274486280z^{-9} + 0.04108536356270z^{-10} + 0.00608720461132z^{-11} + 0.00056590398728z^{-12} + 0.00002486558069z^{-13}$$



(End of Filter Element - Part 1)

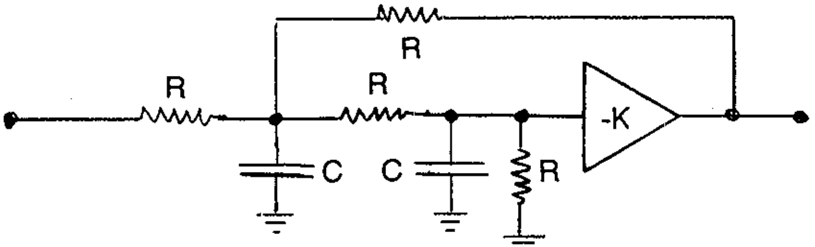


Fig. 1 The Standard Negative-Gain VCVS Low-Pass

negative feedback reduced the effects of component tolerances, but the high gain required severely strained the op-amp to the point where performance suffered (often the circuit was unstable.)

Here our interest concerns a "problem left to the reader" in Section 5-1. There we noted that the pole locus included unstable poles for K more negative than -5 "through a curious path." Accordingly we now want to solve the implied homework problem.

We recall that the transfer function of the circuit of Fig. 1 is:

$$T(s) = \frac{-K/R^2C^2}{s^2 + 5s/RC + (5+k)/R^2C^2} \qquad (1)$$

From this we find the pole radius:

$$\omega_0 = \sqrt{(5+K)} / RC \qquad (2)$$

and the Q is given by:

$$Q = \sqrt{(5+K)} / 5 \qquad (3)$$

Knowing the pole radius and the Q is sufficient to define the positions of the poles. We note that the pole radius depends on RC as well as on K, but the Q depends only on K. Normally we think of useful Q as being a real number greater than zero and less than infinity. A value of Q equal to infinity is a sinewave oscillator, and negative values of Q correspond to unstable networks. What is curious about the NG-VCVS case is that clearly, for K more negative than -5, the Q will become imaginary. What could this possibly mean?

Here we can do two things to make our job easier. First, we will let RC=1. This does result in some loss of generality, but will preserve the essential features. Secondly, we will look at a couple of simpler networks which do not involve imaginary Q - just as points of reference.

The reference configurations here will be the Sallen-Key (Positive Gain VCVS) and the state-variable. We will use the subscripts SK for Sallen-Key and SV for state-variable. Accordingly, in addition to equation (1) the other two transfer functions of interest are:

$$T_{SK}(s) = \frac{K_{SK}/R^2C^2}{s^2 + (3-K_{SK})s/RC + 1/R^2C^2} \tag{4}$$

$$T_{SV}(s) = \frac{-1/R^2C^2}{s^2 + K_{SV}\,s/RC + 1/R^2C^2} \tag{5}$$

Where K_{SK} is the ordinary Sallen-Key positive gain amplifier and K_{SV} is the gain from the bandpass back to the input summer. It is already clear that the Sallen-Key and the state-variable poles are similarly controlled by their respective gains and will be the same if $K_{SK} = (3-K_{SV})$. In particular, their Q's are purely real.

Looking at the denominators of equations (1), (4) and (5), and normalizing to RC=1, we obtain:

$$D(s) \;=\; s^2 + 5s + (5+K) \tag{6}$$

$$D_{SK}(s) \;=\; s^2 + (3-K_{SK})s + 1 \tag{7}$$

$$D_{SV}(s) \;=\; s^2 + K_{SV}\,s + 1 \tag{8}$$

There as a strong temptation of further normalize equation (6) to a unity pole radius. It will turn out that this is dangerous (of limited range of validity) but we will do it anyway. This makes equation (6) become:

$$D(s) \;=\; s^2 + [5 / \sqrt{(5+K)}\,]\, s + 1 \tag{9}$$

Comparing equations (7), (8), and (9) with the "standard form" of a denominator:

$$D_s(s) \;=\; s^2 + (\omega_0/Q)\, s + \omega_0^2 \tag{10}$$

we see that all three are in "standard form" with ω_0 equal to unity and Q's as given by the standard design equations for the configurations. Everything looks fine!

We are now in a position to calculate and plot the "root loci" of the denominators. This is just a plot showing how the poles are positioned as a function of a particular parameter, which is the associated gain K here. The pole loci for Sallen-Key and for state-variable are shown in Fig. 2 and Fig. 3. As noted, these two are closely related, and we immediately identify with the familiar, most useful application range: the semicircle in the left half plane.

At this point, we also bring in the NG-VCVS for consideration, and indeed, its root locus looks fairly manageable (Fig. 4). We note that we do need very large values of K to get poles that are on the circle in the left half plane, and yet which approach the jω-axis closely (high Q). (This relates exactly to what we have mentioned about good passive and poor active sensitivity on the part of the NG-VCVS.) Smaller value of K give complex conjugate poles on the circle, and we eventually reach K=1.25 (Q=1/2) where we have two real poles at -1. As K becomes more negative than 1.25, we see real poles, one moving in toward zero, and the other heading out toward -∞, as K approaches -5.

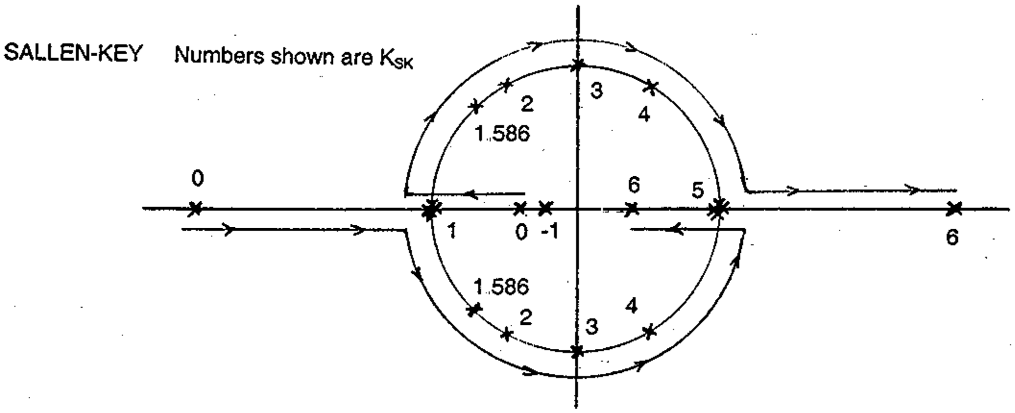


Fig. 2 Root Locus (pole positions vs. K_{SK}) for Sallen Key

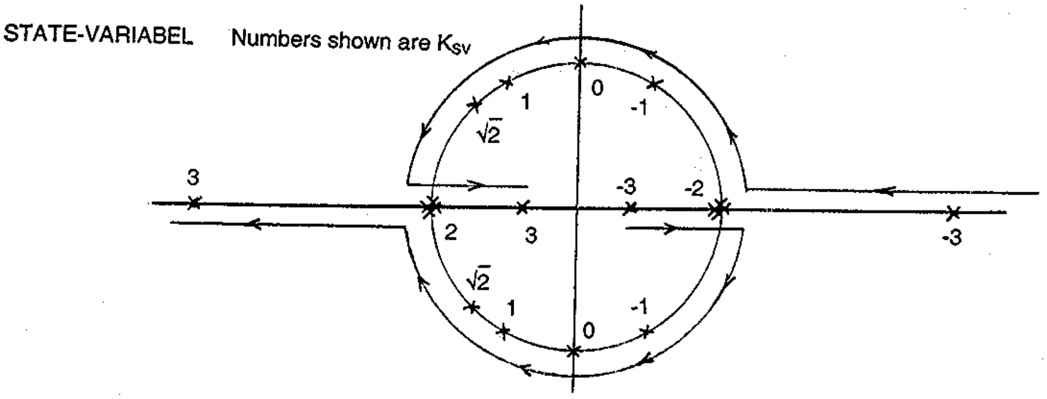
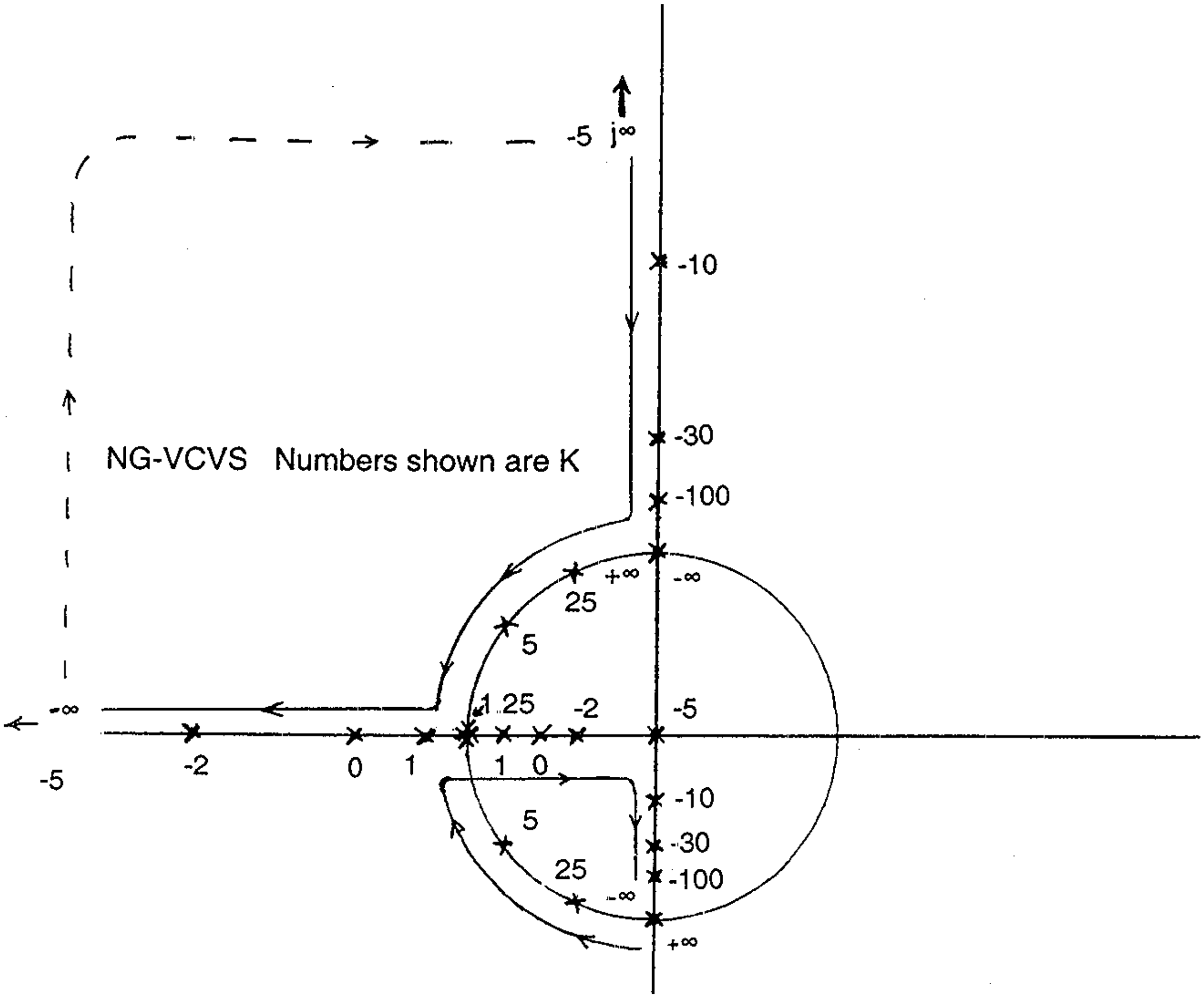


Fig. 3 Root Locus (pole positions vs. K_{SV}) for state-variable

As in the case of Sallen-Key and state variable, we are mainly interested in complex conjugate poles on the circle in the left half plane, and to a limited degree, in real negative poles. Beyond this range, the pole loci are more of a curiosity than anything else. Note as well that changing the sign of K in any of the configurations might involve a significant modification to the original circuit. Even a K=0 is a problem: the poles are properly placed, but there is no output! Sallen-Key gives us stable complex conjugate poles (on circle) for K_{SK}=1 to K_{SK}=3. The same range for state-variable uses K_{SV}=2 down to K_{SV}=0. For NG-VCVS, we use K=1.25 to K=∞. So, are we 100% happy?

The remaining problem appears if we look at NG-VCVS for gains more negative than -5. Note that this means an actual positive gain greater than +5 in Fig. 1. What happens as K goes from -5 to -∞ ? Shockingly, the pole at -∞, jumps to +j∞, and then comes in toward +j as K approaches -∞. The other pole, which was at s=0 when K=-5, moves downward toward -j as K approaches -∞. The poles end up where they started when K was at +∞. Note that during this time, the poles are not complex conjugates. In fact, the migration along the imaginary axis is very reminiscent of what we commonly see occurring for real axis poles. It is almost as though we somehow got rotated by 90°, and this we could suppose to be a consequence of the Q becoming imaginary for K more negative than -5 [see equation (3)]. Is this a problem? If so, is there a mistake, and where? We warned that there was a problem.



<u>Fig. 4</u> Root Locus (pole positions vs. K for NG-VCVS)

If we look back to equation (1), we see that when K becomes more negative than -5, the sign of the last term of the denominator becomes negative. Since the product R^2C^2 must be positive (as must RC in order to correspond to physical reality), any attempt to normalize the last term to +1 must fail. It amounts to having an imaginary pole radius. The change of sign automatically disqualifies the network from stability, according to the Routh criterion. In addition, the square root of this -1 (i.e., j) folds into the middle term, as we noted, implying the imaginary Q. This accounts for the rotation. Another way to see this is suppose that we have a denominator:

$$D_1(s) = s^2 + bs + c \tag{11a}$$

which we can compare with:

$$D_2(s) = s^2 + jbs - c \tag{11b}$$

where it is easy to show that the roots of $D_2(s)$ are those of $D_1(s)$ multiplied by j.

We need not insist upon using the "double normalization" which we used to not only set RC=1, but to also make the pole radius 1. That is, we use equation (6) instead of equation (9). Such a choice results in a pole locus as seen in Fig 5. Here the poles for values of K from +∞ down to 1.25 have a real part of -2.5. This corresponds to the left half-plane semicircle in Fig. 4. As K continues from +1.25 negative to approach -5, we obtain real poles (as in Fig. 4). As K crosses -5 to even more negative values, we do not get the anomalous rotation, but we certainly do get an unstable network as the real pole crosses zero into the right half plane. The practical consequences are the same from either view. In practice, we would just use the design equations, choosing Q and calculating K [equation (3)] and then choose RC for the actual desired pole radius [equation (2)].

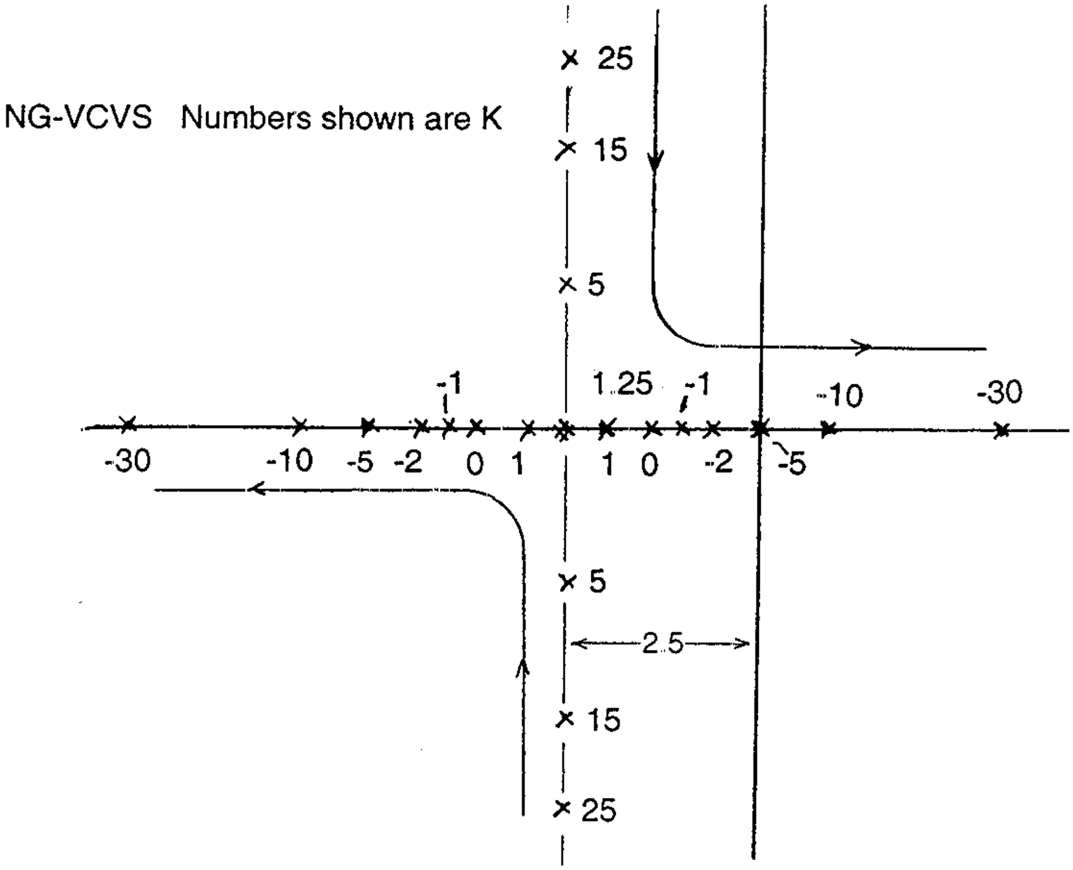


Fig. 5 An alternative pole locus for NG-VCVS

Electronotes, Vol. 20, No. 197, April 2001
Published by B. Hutchins, 1016 Hanshaw Rd., Ithaca, NY 14850 (607)-257-8010