ELECTRONOTES 198

NEWSLETTER OF THE

MUSICAL ENGINEERING GROUP

1016 Hanshaw Rd., Ithaca, NY 14850

Volume 20, No. 198 June 2001

GROUP ANNOUNCEMENTS

Contents of EN#198

In this issue, we continue the presentation on Digital Filtering - the Basic Elements of Digital Signal Processing. Part 3 will conclude the digital filtering material next issue, and we expect then to move on to sampling. We also have an interesting analog corner this issue relating to the measurement of Q.

Analog Signal Processing Corner

Measuring Q With Decrement -by Bernie Hutchins

In Analog Signal Processing (ASP), EN#192, page 32, in a discussion of the time response of an analog bandpass filter (i.e., "ringing") we left a "problem to the reader" in the form of suggesting that the Q of the bandpass can be determined from the decay properties by means of a method known as "log decrement." The conventional "solution" to this problem is fairly well known, and in fact has appeared in our Application Notes: AN-279, May 5, 1983, "Measuring the Q of Bandpass Filters," which in turn relies on results of AN-272, Feb 25, 1983, "Time Response of Bandpass Filters." The AN-279 result, the conventional result, is:

$$Q = -\pi / \ln(d) \qquad (1)$$

(Continues on Page 51)

3. FREQUENCY DOMAIN BASED DESIGNS - GENERAL METHODS

In Section 2, we introduced frequency domain based filter design and developed many useful tools and concepts. In this Section 3, we will look at three quite general methods which will build on the material of Section 2. Section 3a will involve weighted, integrated, least-squared error, of which the inverse DTFT is a special case. We shall also see the idea of "windowing" re-emerge here briefly. Section 3b concerns a generalized view of frequency sampling, of which the inverse DFT is a special case. One form of this generalized frequency sampling, in the limit of a large number of samples, reproduces the integrated least squared error result. Finally, in Section 3c we will look at the "equi-ripple" method, which involves an iterative design procedure rather than a closed form calculation. Fortunately, there is a powerful "alternation theorem" that tells us when we have found the right answer, regardless of the procedure.

3a. WEIGHTED, INTEGRATED, LEAST-SQUARED ERROR FOR FIR FILTER DESIGN

In Section 2b we discussed FIR filter design starting with an inverse DTFT, and we then refining the method a bit. We mentioned that this was a special case of minimizing the integrated squared error in the frequency domain. Here we will examine the method and show how an option for weighting the error make this perhaps the nicest method of those generally available for designing "ordinary" types of filters [3,4,5]. One reason for this is that the options within the weighted integrated square error design include (as special cases) two other popular design methods.

The first of these included methods is the Inverse DTFT technique This is a classic method presented in most DSP books. Secondly we find the refinement to this method where this sudden time-domain truncation is softened with a variety of tapered windows (such as the Hamming window). This windowing reduces passband and stopband rippling at some cost: reduction in the cutoff rate of the transition bands. We will show in one example that the Hamming window result can fall out of the least squared design by appropriate tinkering with the "don't care" region. This is not to suggest that the windowing results can be easily duplicated with least squares, but only that there exists within least squares some range of outcomes that can be similar to windowing. Thus in some sense, least squares is more general.

Perhaps the most important reason for our enthusiasm for the weighted least square procedure, which we will present here, is that it includes all previous cases, and also, it gives us additional control in useful ways. Many filter design procedures involve closed-form calculations, but are still in a sense iterative, in that the designer looks at a current approximation; and then plays with the parameters for some result that is still closer to the desired response. That is, there is a trial-and-error iteration involved. In some cases, we simply guess new parameters, and then as a result, we may find a variety of performance changes to evaluate (and subsequently trade-off). With the weighted least squares, we usually have a very good idea which parameters to adjust

to achieve a particular performance improvement. That is, with the weighting option, improving the performance is generally a more intuitive, more obvious choice.

In particular, if we want to improve (i.e., flatten) the passband, we give that more weight. If we want to improve the stopband (increase the rejection), we can give that more weight. If we want to do both, we can increase the length of the filter, and/or perhaps relax the transition region. Further, a classic "don't care" band option is here simply a matter of weighting the error in a particular band by zero. All of this works fairly nicely in most cases.

3a-1 Theory of Minimizing Integrated Square Error

We want to design an FIR filter by finding the impulse response h(n) for n=0 to n=N-1. The filter thus has a frequency response as given by equation (2). We also assume that we already have some suitable notion of the response of the filter we desire, which we call $D(e^{j\omega})$. $H(e^{j\omega})$ and $D(e^{j\omega})$ are defined for all ω on a continuous range from $-\pi$ to $+\pi$. For each ω there is an associated error, the difference between the desired response and the one we actually intend to achieve, which we can call $E(e^{j\omega})$. Thus:

$$E(e^{j\omega}) = D(e^{j\omega}) - H(e^{j\omega}) = D(e^{j\omega}) - \sum_{n=0}^{N-1} h(n)\, e^{-jn\omega} \qquad (25)$$

This error may be large or small, and can be expected to vary with frequency. We thus need some sort of way of quantifying an overall error over the full range from $-\pi$ to $+\pi$. Although not by any means the only error criterion or even necessarily the best, the method of totaling the squared error is probably the best known and generally used. We can weight and "sum" this squared error by integration, and call it J as given by:

$$J = \int_{-\pi}^{\pi} \{ W(\omega) [D(e^{j\omega}) - \sum_{n=0}^{N-1} h(n)\, e^{-jn\omega}] \}^2 \, d\omega \qquad (26)$$

For obvious reasons, this is called "integrated squared error".

Minimizing J is not an easy thing to do until we invoke the powerful "orthogonality principle." This is most easily understood if we consider a geometric analogy shown in Fig. 18. We note that the exponential functions in the summation for our frequency response [equation (2)] correspond to an orthogonal vector space of functions. Compare this now to an ordinary 3-dimensional vector space. Suppose we want to represent some ideal vector in 3-space, but only have two dimensions available. That is, we want to represent a point in space by a point on a plane such that the error is minimized. This we do intuitively by making the error orthogonal (perpendicular) to the vector directions available (the plane) as indicated in Fig. 18.

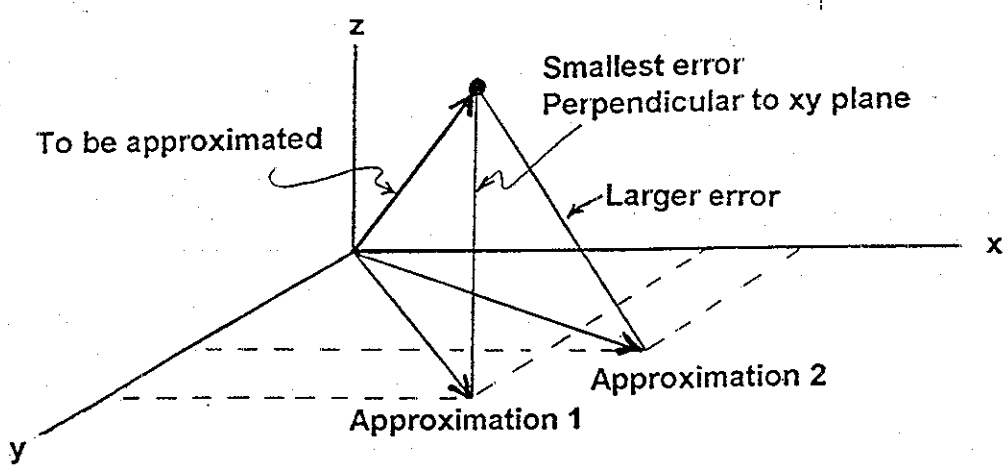


Fig. 18 Perpendicular (orthogonal) error is smallest

Here we have reached a reasonable notion of error, and we understand orthogonality of functions in terms of an appropriately defined inner product being zero. In this case, the appropriate inner product is obtained by multiplying two functions together and integrating this product from $-\pi$ to $+\pi$. Note that any two exponential functions $e^{jn\omega}$ and $e^{jm\omega}$ are orthogonal for $m \neq n$.

In terms of having the error orthogonal to the exponential functions, we have:

$$\int_{-\pi}^{\pi} E(e^{j\omega})\, e^{jm\omega}\, d\omega = 0 \qquad (27)$$

For our particular case, we use equation (25) and the concept of weighting from equation (26) to arrive at:

$$\int_{-\pi}^{\pi} W(\omega) \left[D(e^{j\omega}) - \sum_{n=0}^{N-1} h(n)\, e^{-jn\omega} \right] e^{jm\omega}\, d\omega = 0 \qquad (28)$$

This is the equation we need to solve. For specificity at this point, we will put in a desired function corresponding to a linear-phase FIR filter:

$$D(e^{j\omega}) = A(\omega) e^{-j[(N-1)/2]\omega} \qquad (29)$$

where $A(\omega)$ is the amplitude. The amplitude function can be considered as yet another way to represent a transfer function. The way we interpret this is in terms of a pure linear phase $e^{-j[(N-1)/2]\omega}$ times an "amplitude" function $A(\omega)$ which is the sum of cosines, as indicated. This we compare to a more classic multiplicative decomposition:

$$D(e^{j\omega}) = e^{j\phi(\omega)} \cdot |D(e^{j\omega})| \tag{30}$$

in terms of a phase $\phi(\omega)$ and a magnitude $|D(e^{j\omega})|$. From equations (29) and (30) it is clear that $A(\omega)$ and $|H(e^{j\omega})|$ are not the same thing unless $A(\omega)$ is always positive. This would likely only be true for filters that have no stopbands (perhaps some kind of amplitude equalizer). However, other filters such as common low-pass, band-pass, and high-pass will have bands where we intend to make the response approximate zero. In making it approximate zero, we need to cross zero one or more times, and this corresponds to zeros on the unit circle, and corresponding jumps of π in phase, interrupting any pure linear phase.

Another way to look at it is to write:

$$e^{j\phi(\omega)} |D(e^{j\omega})| = D(e^{j\omega}) = e^{-j[(N-1)/2]\omega} \; \text{sgn}\{A(\omega)\} \; \text{sgn}\{A(\omega)\} \; A(\omega) \tag{31}$$

where $\text{sgn}\{A(\omega)\}$ is the sign of $A(\omega)$, and hence $\text{sgn}\{A(\omega)\}A(\omega) = |D(e^{j\omega})|$ and we then see that the actual phase is:

$$e^{j\phi(\omega)} = e^{-j[(N-1)/2]\omega} \; \text{sgn}\{A(\omega)\} \tag{32}$$

from which we see that the actual phase is the pure linear phase as flipped (hence π phase jumps) by the sign of $A(\omega)$. It is often convenient to work with $A(\omega)$.

We can now write equation (28) as:

$$\int_{-\pi}^{\pi} W(\omega) \, A(\omega) e^{-j[(N-1)/2]\omega} \, e^{jm\omega} \, d\omega =$$

$$\sum_{n=0}^{N-1} h(n) \int_{-\pi}^{\pi} W(\omega) \, e^{-jn\omega} \, e^{jm\omega} \, d\omega \tag{33}$$

Happily, this equation can be put in the form of a set of N equations in N unknowns of the form:

$$d = Mh \tag{34}$$

where h is the impulse response to be determined, d is a length N vector whose elements are given by the left side of equation (33) for m = 0 to N-1, and M is an N×N matrix whose elements are given by the integral on the right side of equation (33) for m=0 to N-1 and n=0 to N-1. For some simple (yet useful) cases, these integrals are very easy to do, resulting in sinc functions evaluated at bandedges.

It will be most convenient to be even more specific here. We will consider a three-band filter as suggested in Fig.19. This filter has a band from 0 to ω_1 which has an amplitude A_1 and weight W_1, a second band from ω_1 to ω_2 that has amplitude A_2 and weight W_2, and a third band from ω_2 to π that has amplitude A_3 and weight W_3, along with a corresponding response for negative frequencies as shown. Note that this means that over each band, $A(\omega)$ and $W(\omega)$ are constants which move outside the integrals, leaving us only exponentials to integrate (giving of course more exponentials). In addition, integrating from $-\pi$ to $+\pi$ helps by providing exponentials that can be combined to sines.

We thus have:

$$W_3A_3 \int_{-\pi}^{-\omega_2} e^{-j[(N-1)/2]\omega}\, e^{jm\omega}\, d\omega$$

$$+\ W_2A_2 \int_{-\omega_2}^{-\omega_1} e^{-j[(N-1)/2]\omega}\, e^{jm\omega}\, d\omega$$

$$+\ W_1A_1 \int_{-\omega_1}^{+\omega_1} e^{-j[(N-1)/2]\omega}\, e^{jm\omega}\, d\omega$$

$$+\ W_2A_2 \int_{+\omega_1}^{+\omega_2} e^{-j[(N-1)/2]\omega}\, e^{jm\omega}\, d\omega$$

$$+\ W_3A_3 \int_{\omega_2}^{+\pi} e^{-j[(N-1)/2]\omega}\, e^{jm\omega}\, d\omega$$

$$= \sum_{n=0}^{N-1} h(n) \left\{ W_3 \int_{-\pi}^{-\omega_2} e^{-jn\omega}\, e^{jm\omega}\, d\omega \right.$$

$$+\ W_2 \int_{-\omega_2}^{-\omega_1} e^{-jn\omega}\, e^{jm\omega}\, d\omega + W_1 \int_{-\omega_1}^{\omega_1} e^{-jn\omega}\, e^{jm\omega}\, d\omega$$

$$\left. +\ W_2 \int_{\omega_1}^{\omega_2} e^{-jn\omega}\, e^{jm\omega}\, d\omega + W_3 \int_{\omega_2}^{\pi} e^{-jn\omega}\, e^{jm\omega}\, d\omega \right\} \qquad (35)$$

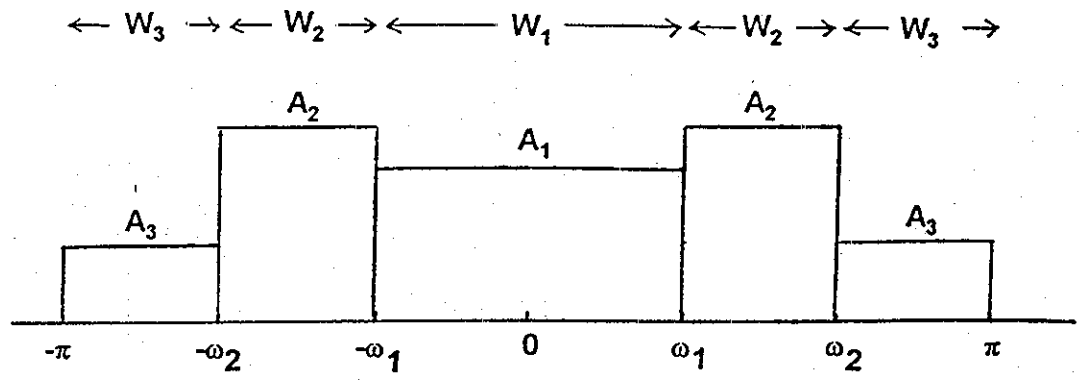


Fig. 19 Three-Band Filter

Equation (35) is tedious, but simple to do. The left side (upper portion) evaluates to:

$$d(m) = 2\omega_2(W_2A_2-W_3A_3)\text{sinc}[m - (N-1)/2]\omega_2$$
$$+2\omega_1(W_1A_1-W_2A_2)\text{sinc}[m - (N-1)/2]\omega_1$$
$$+2\pi W_3A_3\text{sinc}[m - (N-1)/2]\pi \qquad (36)$$

The matrix elements of M from equation (9) are obtained from the right side of equation (35), the terms in { }, as:

$$M(n,m) = 2(W_2-W_3)\omega_2\text{sinc}(m-n)\omega_2$$
$$+ 2(W_1-W_2)\omega_1\text{sinc}(m-n)\omega_1$$
$$+ 2W_3\pi\text{sinc}(m-n)\pi \qquad (37)$$

This completes the evaluation of integrals, and all that generally remains is to write a program to evaluate equations (36) and (37), and then calculate:

$$h = M^{-1}d \qquad (38)$$

3a-2 Examples Using Weighted Integrated Least Squared Error

Our first example shows a length 31 FIR low-pass filter chosen to have a cutoff of 0.175 of the sampling rate. Here we have chosen two bands. The first band from 0 to 0.175 is to have a desired value of magnitude 1, while the second band from 0.175

to 0.5 is to have magnitude 0. The weight on the error in both bands is to be 1. The result seen in Fig. 20 is exactly the result we get if we had used the inverse DTFT, complete with Gibbs phenomenon peaking. This reproduction of the previous method was achieved using equation (38) through the use of equal weighting. In fact, it is easy to show that for this case, where the weighting is 1 from $-\pi$ to π, that the M matrix is simply 2π times the identity matrix, and this leads easily to the inverse DTFT.

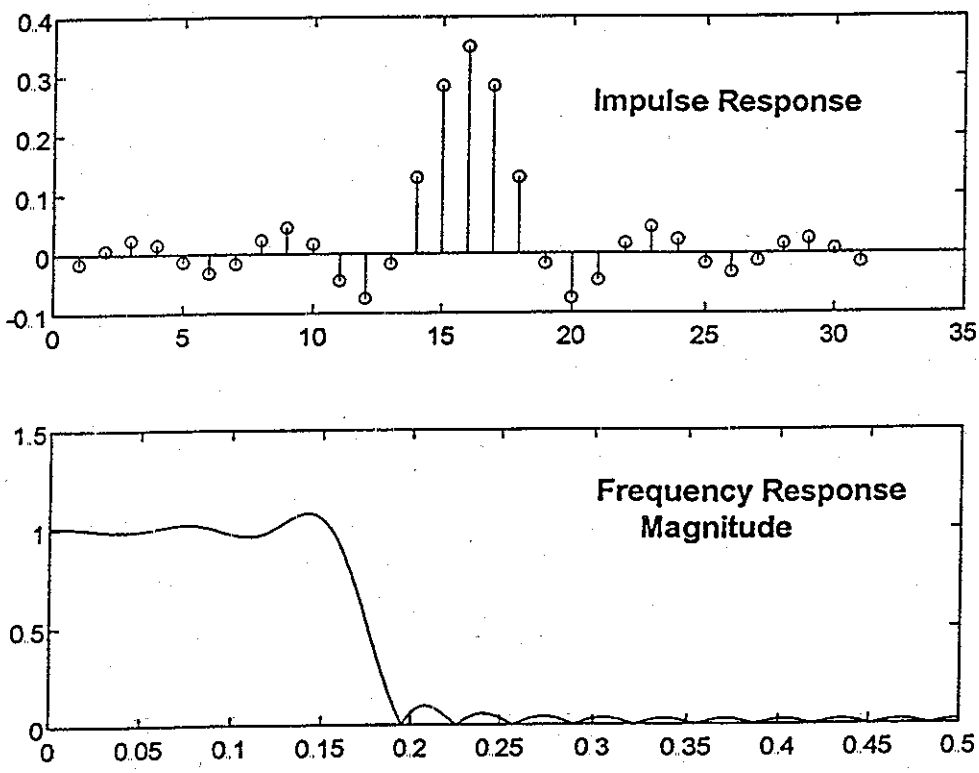


Fig. 20 By using no transition band, and using equal weighting, a result equivalent to inverse DTFT is achieved.

In Fig. 21 we find the advantage of the new method. Here we have a low-pass formed from a three band approach. The first band has magnitude 1 from 0 to 0.15, and we weight the error on this band at 1. This is followed by a second band from 0.15 to 0.2. We don't care what magnitude we assign to this band because we are going to weight the error on this band at 0. In fact, this is often called a "don't care" or "transition" band. The third band, from 0.3 to 0.5 is assigned a magnitude of 0 and a weight of 1. The distinction between assigning a magnitude (some desired response level) and a weight (how much we care about not getting this value over a particular band) must be understood.

It is not evident that essentially ignoring the error on a particular band will lead to a favorable result, in the other bands, or more particularly in the band with zero weight. Accordingly we will look at the results obtained, and always be cautions about what happens in our don't care band, as we may care if it gets too out of hand. But Fig. 21 does show a favorable result: the hoped for reduction in Gibbs phenomenon, and a

less sharp but otherwise normal-looking transition band. The point that the result we get is similar to previous results is born out by looking at the "equivalent window" also shown in Fig. 21. If we had been using a window to control Gibbs phenomenon, we would have taken h1 from Fig. 20, multiplied it point-by-point with a window (e.g., Hamming) and arrived at something like the more tapered set of taps in Fig. 21. Thus we can divide the impulse response of Fig. 21, point-by-point by the impulse response of Fig. 20, to get the equivalent window. The point to note is that it is somewhat similar to the Hamming window, or other useful tapered windows. We are kind of doing the same thing.

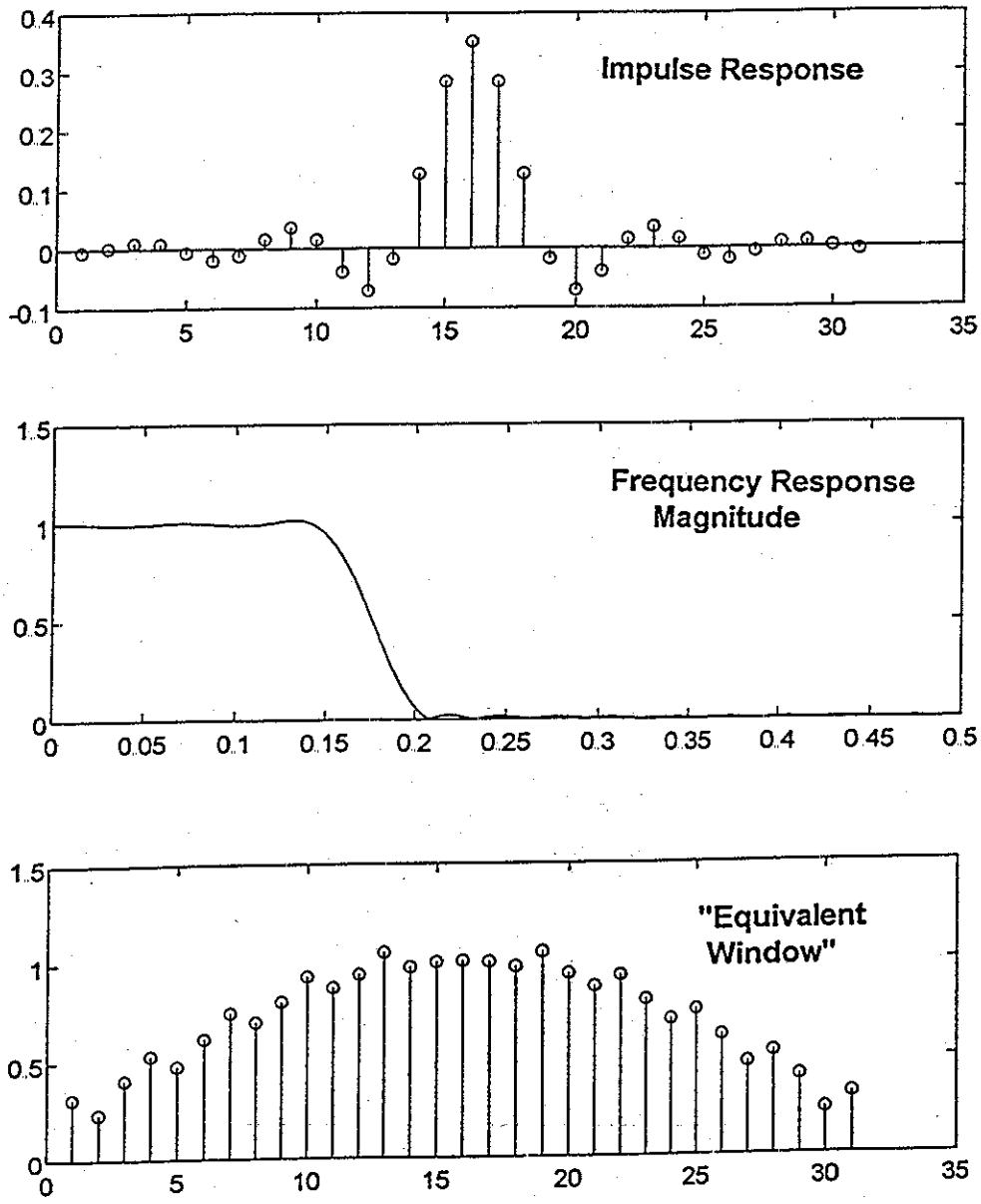


Fig. 21 A don't care region (zero weight on the error) results in a wider transition region and a reduction of Gibbs phenomenon, similar to using a window.

The point about the equivalent window in Fig. 21 leads to the question as to whether or not a particular choice of don't care region could lead to an actual Hamming window. This is possible, although here only an example will be given to indicate the truth of this claim. This is shown in Fig. 22. Here we started with a length 63 low-pass design with a cutoff at 0.25 with no transition region. This was then windowed with a Hamming window. Next, by trial and error, a don't care region was manipulated to give the same impulse response as that obtained with the Hamming window. An excellent agreement was found when the don't care region was from 0.2245 to 0.2755. In this case, the taps agreed to three or four decimal places. Fig. 22 actually shows the final result of dividing the taps using don't care by the sharp transition result. We expect to see the Hamming window, and we do. [When computing equivalent windows, one must be careful not to divide by 0, or even to divide two very small numbers (that are essentially zero). In this example, all odd taps except for the center are supposed to be zero in both design approaches (due to the choice of cutoff at 1/4). In actual computer calculation, these "zeros" might be 10^{-8} and 10^{-12} for example. Thus when an equivalent window is computed, steps may be necessary to remove accidental spikes. Here we have just plotted zero in these positions.]

Figures 23a and 23b show what is perhaps the most useful contribution of the use of weighted error. Here we take a low-pass with a don't care band from 0.165 to 0.185. The difference is that in Fig. 23a, the weight of the error in the stopband is set to 1000 (passband weight =1) while Fig. 23b shows weight 1000 in the passband (stopband weight =1). These are quite different filters. We note the much improved flattening of the band with the greater weight. In addition, the cutoff region is displaced away from the band with the greater weight, a side effect which might require adjustments. It is clear however that weighting allows us to redistribute error between bands.

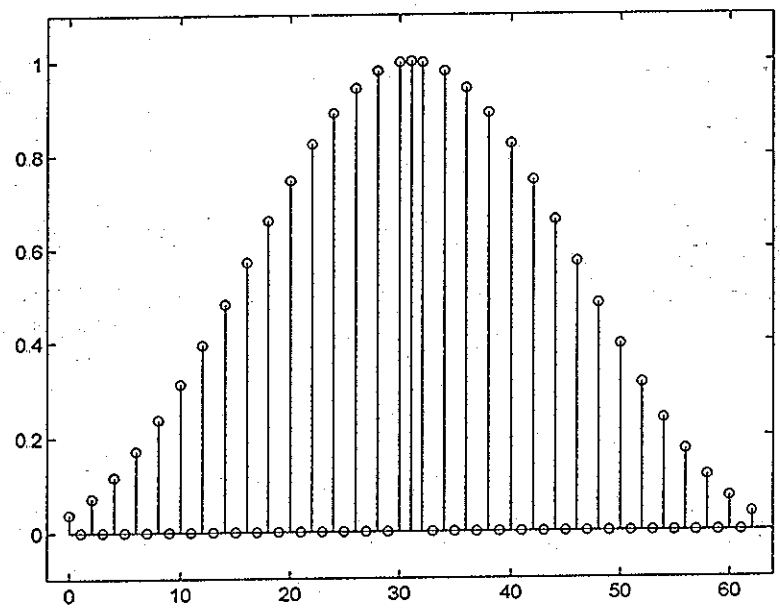


Fig. 22 The integrated least squares method can be manipulated into a result equivalent to the Hamming window.

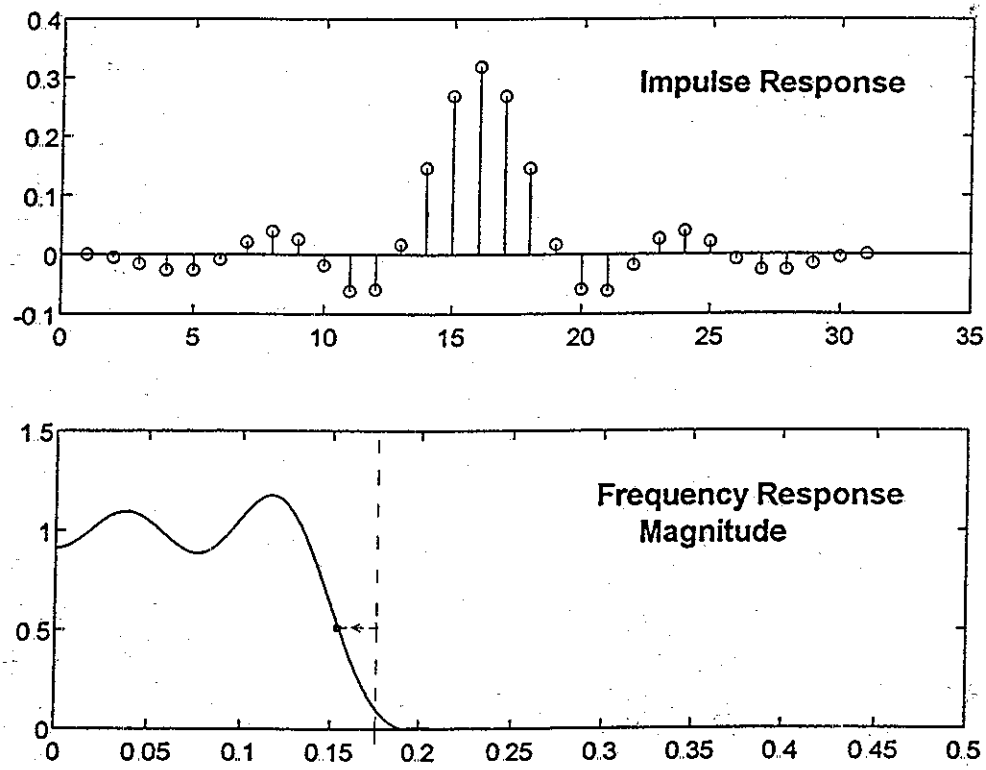
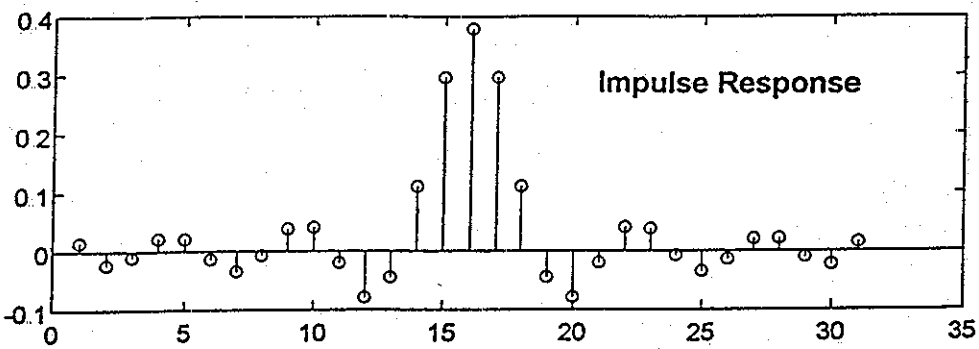


Fig. 23a Here weighting the error in the stopband 1000 times larger than the passband greatly reduces stopband ripple. The cutoff is shifted downward as well, relative to an expected 0.175.



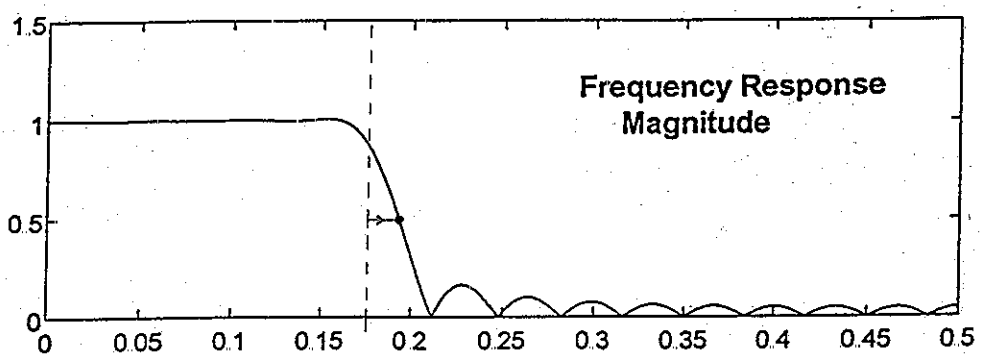


Fig. 23b Here weighting the error in the passband 1000 times larger than the stopband greatly reduces passband ripple. The cutoff is shifted upward as well, relative to an expected 0.175.

3b GENERALIZING FREQUENCY SAMPLING

In Section 2, the idea of using the inverse DFT of samples of a frequency response rather than the inverse DTFT of the continuous response was introduced. There we studied problems with setting up the correct phase response for the samples, and the effects of using one or more transition band samples. Here we will be generalizing these frequency sampling ideas [6]. First, we will look at the case where the samples are not equally spaced. Then we will look at the case where we have an excess of samples relative to the desired length of the filter, and will employ a minimization of squared error of the same nature as that used in Section 3a.

3b-1 Unequal Spacing of Frequency Samples

Frequency sampling in a more general sense need not require equally spaced samples. Instead, we seek a length N impulse response h(n) that is related to N frequency samples by N equations in N unknowns. If we set up and solve this problem, one of its special cases will be equally spaced samples and a matrix representing the N×N equations that is the same as the DFT matrix.

Before setting up the equations, we should mention two important consequences of using unequal spacing. First, there will be no automatic way of generating the samples from specified passbands. Instead, we will in general need an input vector f of frequencies and a corresponding vector of amplitudes. Secondly, we need to recognize that there can be severe consequences of choosing the unequal spaced points.

The basis for the method here goes back to equation (2). We simply have in mind N frequencies ω_k with N corresponding versions of equation (2), thus giving us our N equations with N knowns (H), and N unknowns (h). While this works for any N frequencies and corresponding samples, we will be looking here for choices of sample values that give us real and symmetric (linear phase) impulse responses. The same phase inversion for samples from π to 2π that was required for even length in Section 2c-1 is also needed for even length here.

As an example, if we have N=4, equation (2) yields four equations:

$$H(e^{j\omega_1}) = h_0 + h_1e^{-j\omega_1} + h_2e^{-2j\omega_1} + h_3e^{-3j\omega_1} \quad (39a)$$

$$H(e^{j\omega_2}) = h_0 + h_1e^{-j\omega_2} + h_2e^{-2j\omega_2} + h_3e^{-3j\omega_2} \quad (39b)$$

$$H(e^{j\omega_3}) = h_0 + h_1e^{-j\omega_3} + h_2e^{-2j\omega_3} + h_3e^{-3j\omega_3} \quad (39c)$$

$$H(e^{j\omega_4}) = h_0 + h_1e^{-j\omega_4} + h_2e^{-2j\omega_4} + h_3e^{-3j\omega_4} \quad (39d)$$

which are in matrix form:

$$H = Eh \tag{40}$$

where

$$E(k,n) = e^{-jn\omega_k} \tag{41}$$

Inverting equation (40) we get:

$$h = E^{-1}H \tag{42}$$

Note that if the frequencies are equally spaced we would have $\omega_k = (2\pi/N)k$ and the matrix M becomes:

$$E(k,n) = e^{-j(2\pi/N)nk} \tag{43}$$

which is the DFT matrix.

3b-2 Sampling the Amplitude Function

It is frequently easier and more computationally efficient to work with the sampling of the amplitude function [equation (29), etc.] rather than with a sampling of the frequency response itself. With this approach, we can automatically includes the linear phase. Also, we only need to specify the lower half of the response: 0 to π. In addition, we are in general then only solving half as many equations with half as many unknowns, meaning an inversion of the matrix of only about 1/4 the original size.

Samples may be chosen for the amplitude function for the odd length case:

$$A(\omega_k) = a_0 + a_1 \cos(\omega_k) + a_2 \cos(2\omega_k) + \ldots \tag{44a}$$

while for even length, the amplitude function is:

$$A(\omega_k) = a_1 \cos(\omega_k/2) + a_2 \cos(3\omega_k/2) + a_3 \cos(5\omega_k/2) + \ldots \tag{44b}$$

(See also section 3c-9.) Thus L samples can result in either a length 2L-1 odd length filter or a length 2L even length filter. The extra tap in the even length filter is merely a consequence of the automatic zero at z=-1 that occurs with even length.

3b-3 Examples of Unequal Spacing

When we do frequency sampling with equal spacing, we can only specify samples at frequencies that are constrained by our choice of sampling frequency and number of samples. If there is one particular frequency to which we would like to give specific attention, it is unlikely to be available. When we use unequally spaced samples, we make sure to include the special frequency in the overall set, and see what happens.

This situation often occurs when we want to place zeros in a stopband. In a typical case we might have a general low-pass requirement, perhaps to remove all frequencies above 10 kHz. Perhaps this is thought of as lowering the amplitude of moderate levels of wideband noise in this region. But we might also know that there is a strong interfering component at, say, 15.7 kHz. No filter that we have designed or ever can design can have zero magnitude response except at isolated frequencies. Rather we expect a stopband to consist of isolated zeros and small, non-zero lobes between them. The response is generally small, but not zero. This generally small response is effective at reducing generally small input components to negligible levels. However, a strong narrow-band input component may be made only small (not negligible) as it passes through the filter, unless by luck a zero of the filter's response is exactly on that frequency. With unequal (i.e., arbitrary) spacing of samples we can put a zero exactly on a desired frequency, and hope that nothing drastic will happen elsewhere.

For Fig. 24a, we have chosen equally spaced samples, and have stopband zeros at 0.24, 0.28, 0.32, 0.36, 0.40, 0.44, and 0.48. Now suppose that we learn that there is a strong interfering component at frequency 0.39. We see from the magnitude response that this frequency is partially passed by the response lobe between 0.36 and 0.40. By using unequally spaced samples, we can move the zero from 0.40 to exactly 0.39. We see from Fig. 24b that this completely rejects the 0.39 frequency as expected, but also effects the stopband lobes slightly. There is little change in the passband to note. Likely this is a very successful result overall.

Another idea that occurs to us would be to keep the sample at 0.4, and insert an additional zero at 0.39, necessarily increasing the length of the filter by two taps. Surprisingly this has a rather large effect overall as is seen in Fig. 24c. What we get is additional flattening in the stopband, and increased ripple in the passband. We see that requiring two close samples to be zero (or just the same) will have an effect on the derivative of the magnitude response as well as the magnitude itself. This response in Fig. 24c may be what we want, but just moving one sample as in Fig. 24b may be a preferred approach. Of course, any time we design a filter we need to carefully examine the resulting response. With unequal spacing of samples, we need to be additionally careful.

Fig. 25 shows a case where a seemingly slight change in spacing (only 25%) has a drastic effect. Here the first 15 passband samples, and the first six of 15 stopband samples are spaced at in interval of 0.016. The last nine stopband sample are spaced at 0.020. Perhaps we suppose that once we get the response nicely into the stopband, we could move the zeros a bit further apart. What we do see however is that as the spacing is increased, the response shoots back up so that the lobes reach the level of the passband itself. This likely amounts to a nasty surprise.

3b-4 Using More Equations than Unknowns: Least Squared Error Again

In some cases, it is useful to consider a number of frequency sampling points that is greater than the length of the filter to be designed. For example, if the filter is of length N, we might still want to take M sample points where M is greater than N. In such a

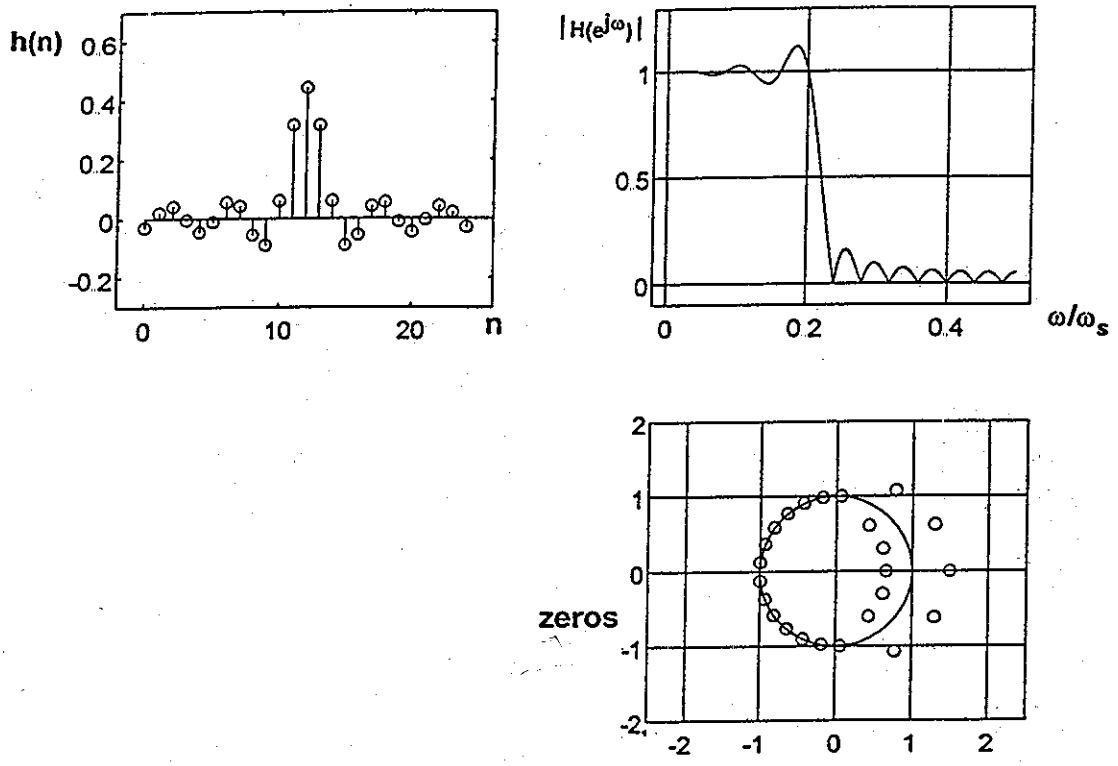


Fig. 24a A length 25 filter with one sample at $0.4f_s$

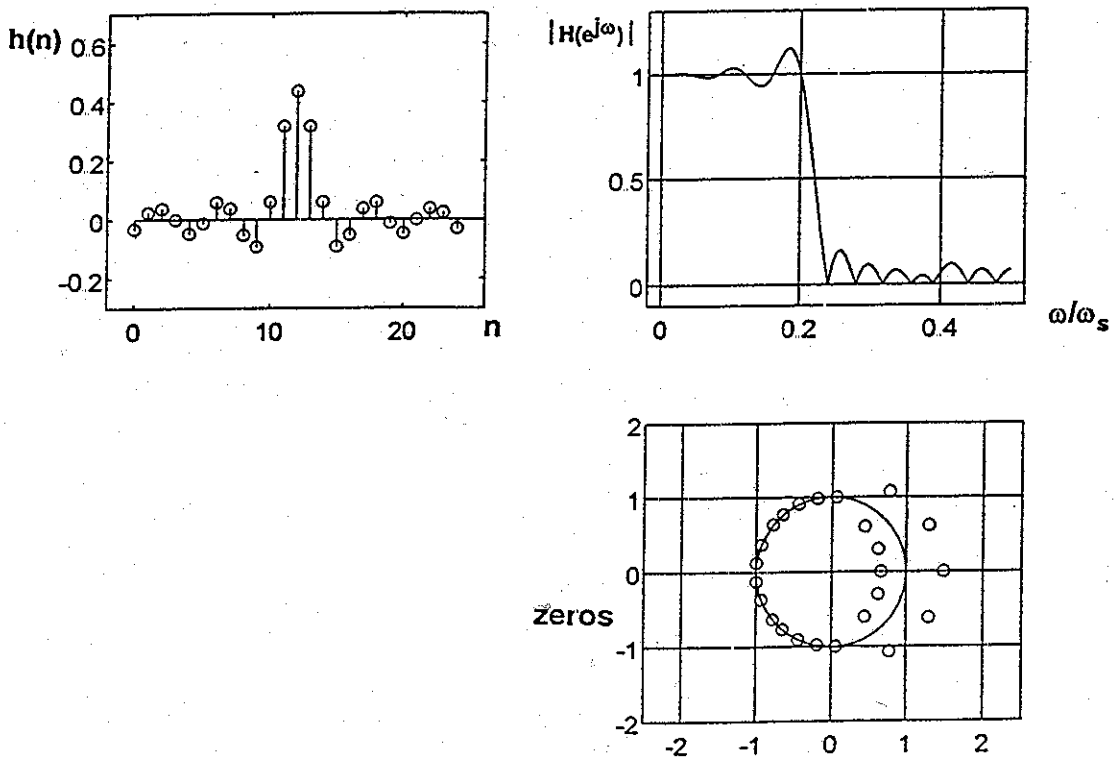


Fig. 24b A length 25 filter with one sample moved to $0.39f_s$

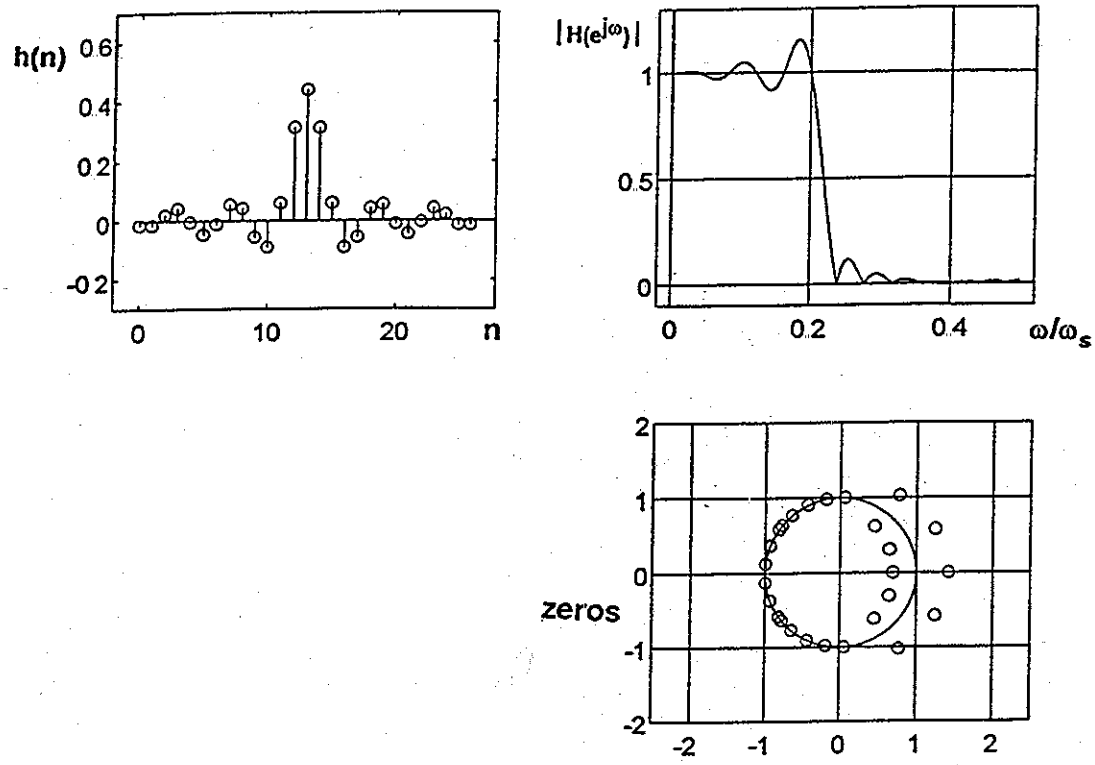


Fig. 24c Two Close Samples at 0.39 and 0.40

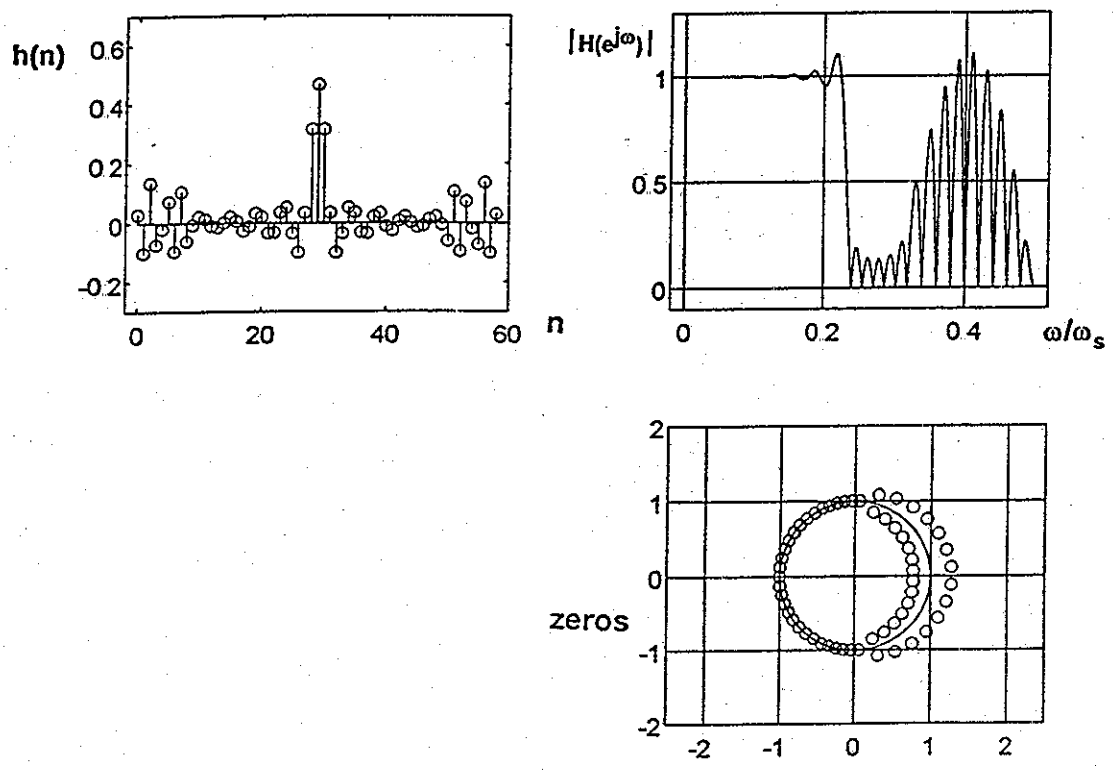


Fig. 25 Even a Slight Change of Spacing Can Lead to Drastic Changes

case, we have more equations that unknowns, and we can not solve these in the usual way (matrix inversion) but must instead use the least square procedure or pseudo-inverse. In the case where M=N, we can fit the response exactly to the M points for zero error at each point and zero error total. For the case of M>N, we expect the curve in general to not go exactly through any of the specified points. In this case of M>N, in general, there is an error at each point, and we seek to minimize this error in the least squares sense.

This least square procedure is easily solved in terms of the use of a "pseudo-inverse." The phase is set in a similar way to previous examples, but it is important to recognize that while the frequency samples divide 2π by M, it is still N, the length of the filter we end up with, that determines the linear phase term. The matrix E corresponding to the coefficients of the equations is set up in a manner similar to previous cases, but we recognize that we have M rows by N columns. Here the samples are equally spaced, but we must still use the matrix (instead of the inverse DFT) because in general $N \neq M$. The major difference is thus that, since E is not square, the pseudo-inverse, $(E^tE)^{-1}E^t$, is used instead of just E^{-1}.

The use of an excess number of samples and the corresponding LMS solution can usually give results superior to the result from N samples in N unknowns. Fig. 26a shows a case of a length 19 filter obtained from 19 equally spaced frequency samples. This amounts to ordinary (inverse DTF) frequency sampling, with a nominal cutoff of 0.25. Fig. 26b shows the corresponding case of a length 19 filter based on 200 frequency samples. The difference is not astoundingly better, but it does appear significantly better in three ways: it has less passband ripple, better stopband rejection, and the cutoff (-6db) is more precisely obtained at 0.25. Since locating the cutoff is a matter of defining a transition between two samples, we naturally do a better job if we have a denser set of samples in frequency.

Fig. 27 shows another application of the use of excessive frequency samples. In this case we are trying to produce (imitate) a Butterworth magnitude response but obtain linear phase. Normally we would produce Butterworth filters with IIR methods, which do not have linear phase. In addition, if we decided to duplicate a Butterworth magnitude using FIR (zeros, no poles), we would expect to need a much higher order filter, and this in itself is interesting and important to know about. Integrating the squared error would make it necessary to mathematically write down the mathematical expression for the Butterworth magnitude and then do the integral [equation (4)]. This would be difficult. But it is easy to compute the desired Butterworth magnitude and input samples of this to a frequency sampling design procedure (adding linear phase as usual). This is seen in Fig. 27 for 500 frequency samples of a 12th-order Butterworth low-pass, reduced to a length 12 filter (Fig. 27a), a length 25 filter (Fig. 27b) and a length 50 filter (Fig. 27c). The answer to the question is that you seem to need something like a 50th-order FIR to do the job of a 12th-order IIR, as far as magnitude is concerned.

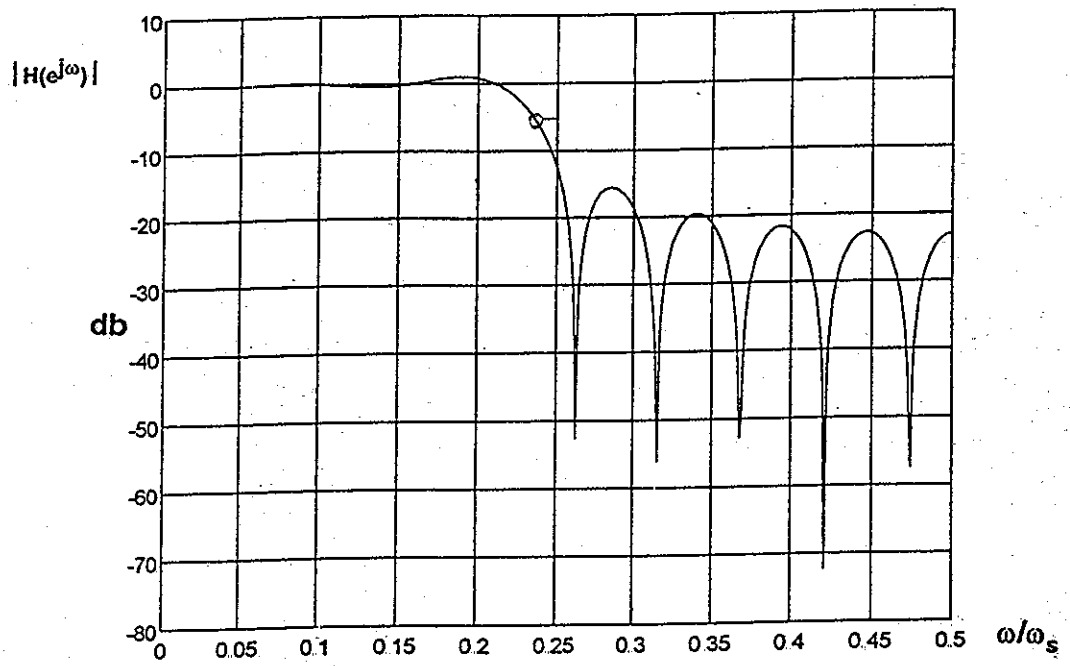


Fig. 26a A length 19 filter from 19 frequency samples

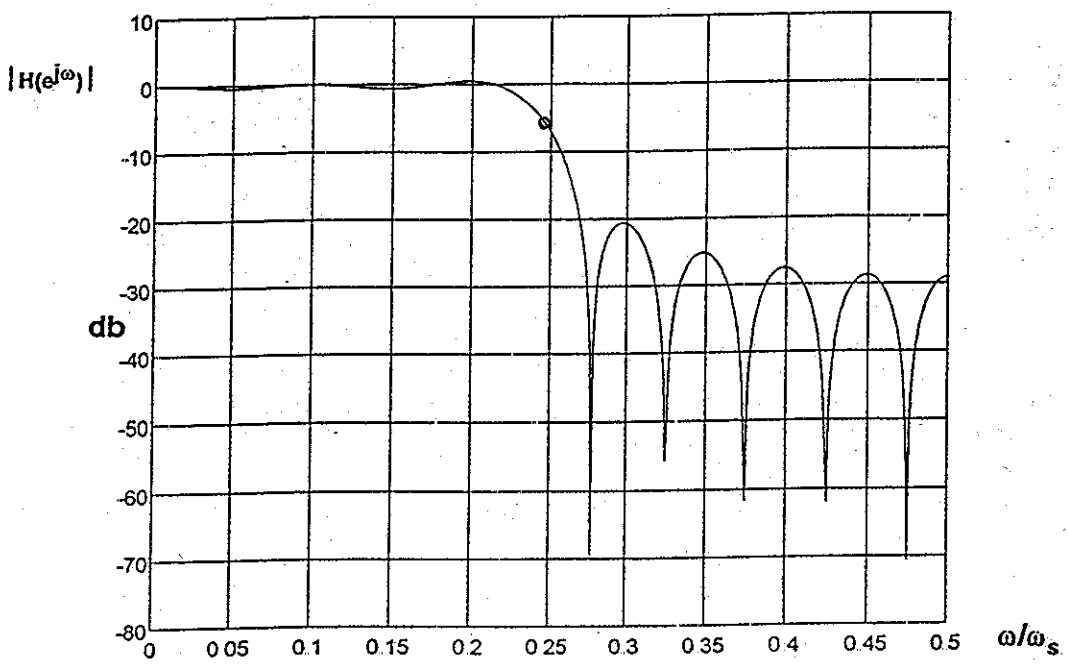


Fig. 26b A length 19 filter from 200 frequency samples

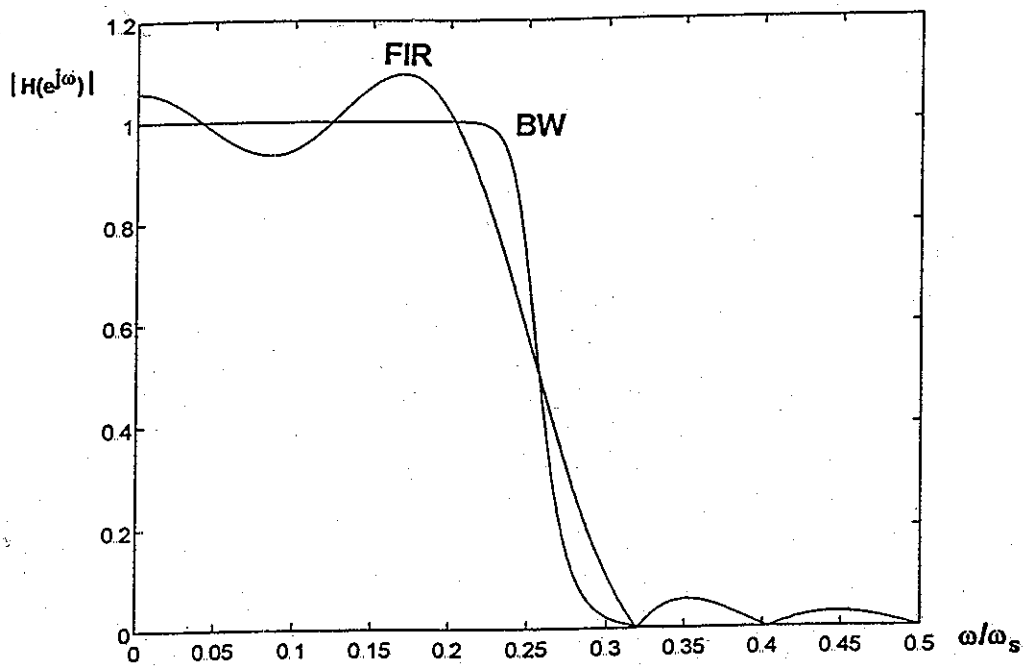


Fig. 27a A length 12 FIR filter trying to be a 12th order-IIR Butterworth

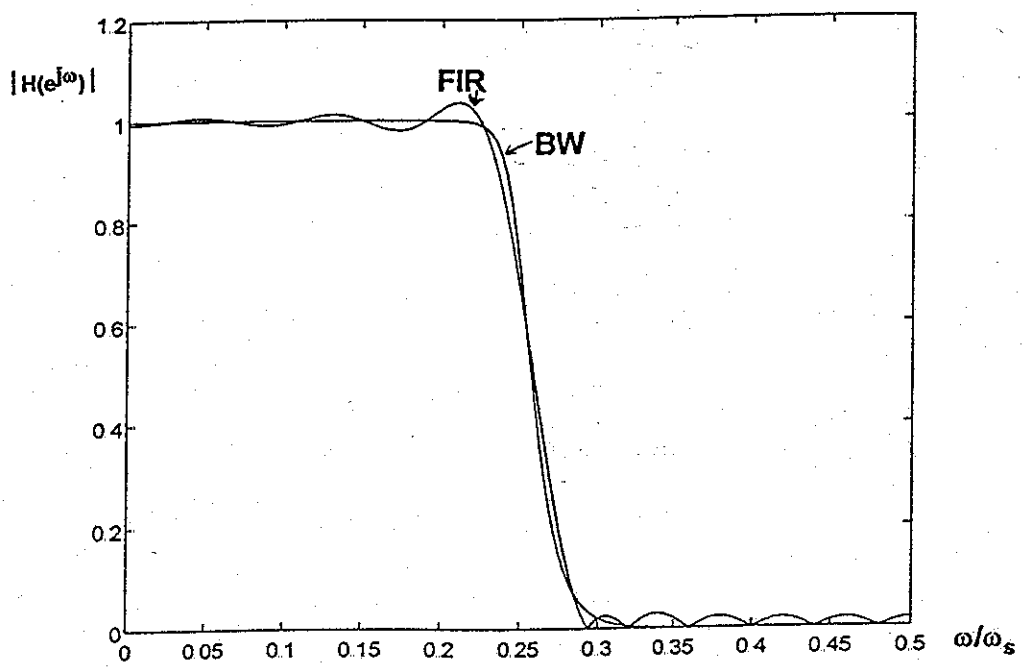


Fig. 27b A length 25 FIR trying to be a 12th-order Butterworth

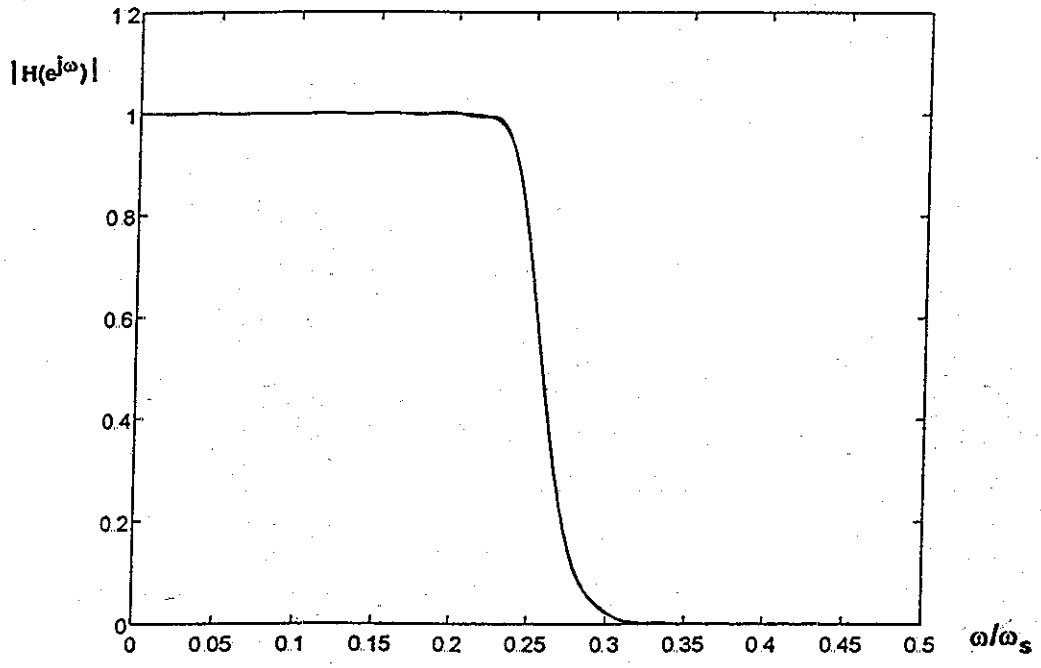


Fig. 27c A length 50 FIR filter trying to be a 12th order-IIR Butterworth

3b-5 Weighted Least Squares

The method of least squares in Section 3b-4 above can be modified to accept a weight on the error in different bands, or even on individual samples if we wish. We simply have to determine a weight vector W much as we do an amplitude vector for each sample. Next, representing W as a diagonal matrix, we can find h by using $(E^tW\,E)^{-1}E^tW$, replacing $(E^tE)^{-1}E^t$ for the unweighted case, replacing E^{-1} for the case where M=N. Perhaps to our surprise, the actual weighting only works when M>N and only well when M is something like (at least) twice N. For M=N, the weighting has no effect. For example, a weight vector of 1000 on the passband and 1 on the stopband will give exactly the same thing as 1 on the passband and 1 on the stopband (etc.). No flattening of the passband occurs as a result of the larger weighting. The reason for this is that when M=N, there is an exact, zero-error solution (the designed response goes exactly through the desired response). In consequence, no weighting of this zero error makes any difference.

Choosing M>>N, we can use the weighting vector much as we did the weighting with integrated least squared error design. If M=N and we need to flatten a band, unequal spacing of samples rather than weighting is suggested.

Weighting of the least squares frequency sampling approach is similar to the case of integrated squared error. In fact, as is likely intuitively clear, if we start out with a large excess of frequency samples, we can get a result that is virtually identical to integrated squared error. This is because the summation of a very dense set of squared errors is virtually the same as integrating the squared error.

3c. EQUI-RIPPLE DESIGN

Among the most popular of the FIR digital filter design methods is the one variously called "equiripple," "Parks-McClellan," or "Remez" [7-11] While there is no closed form procedure for designing equiripple filters, there is a powerful "alternation theorem" that allows us to recognize the unique best filter meeting our specifications when we do find it. This is important (knowing when to stop) since there may be several or many different ways of moving toward the right answer.

By combining the alternation theorem and some additional (fairly simple) mathematical considerations, we can enhance our understanding of what to expect from the design. While some designs look like neat, expected, reasonable results, others (particularly for filters of three or more bands) may have interesting features that may look suspicious until we carefully check to see that they do meet the mathematical requirements.

The equi-ripple method, like the others we have studied, can be thought of as an approach to handling Gibbs phenomenon. In previous methods, we achieved reduced ripple (improvement) by methods which in turn result in a less sharp cutoff (worse). Another thing that we noticed was that the ripple, prior to applying the various ripple reduction procedures, was preferentially concentrated close to the transition regions. With equi-ripple, we are essentially keeping the ripple, but distributing it equally over entire bands. In turn, we expect to better retain a sharper cutoff rate. This is entirely analogous to the case where a Chebyshev characteristic, rather than a Butterworth, is chosen. For the same order filter, we are willing to tolerate limited ripple for the reward of a sharper cutoff (see Fig. 14).

As mentioned, the design procedure is iterative. Yet we do see features we have seen above. In particular, we will have a "don't care" region for transition bands, and each iteration involves a curve fitting to samples that is very similar to frequency sampling with uneven spacings.

3c-1 The Alternation Theorem

The "alternation theorem" tells us how to recognize the unique, best equiripple approximation. To determine that the response is the unique, best approximation, we need to see that the response has R+1 extremal points, where R is the number of degrees of freedom in the design. This is the theorem. We now need to say what we mean by extremal points, and what we mean by degrees of freedom.

The idea of degrees of freedom is the easiest. This is something we get to choose. Most simply, in a length N FIR filter there are N tap weights (coefficients) which we get to choose, and hence, N degrees of freedom. However, we are most often operating under a "linear phase" constraint, and certain required symmetries constrain certain tap weights to be equal, and accordingly, the number of degrees of freedom is reduced, typically by about 1/2. A length N filter with N even has only N/2 unique tap weights, and hence R=N/2. If the length N is odd, then (N-1)/2 taps are paired by symmetry, and the very center is unpaired. This gives (N-1)/2 + 1 = (N+1)/2 unique values, and hence R = (N+1)/2.

It is the usual practice to combine symmetrical taps so that their contribution to the amplitude is represented by a cosine [see for example equations (11-13)]. This permits us to largely dispense with simple (but easy-to-muddle) rules involving parameters that need careful definition. Instead we simply observe that the amplitude of each cosine involved in the expansion is a degree of freedom. Accordingly, we look for a number of extremal points that is one more than the number of cosines. We only need to be careful to recognize (count) cos(0) as a cosine.

As for extremal points, these occur at frequencies where the error (difference from ideal, non-rippled, case) is of maximum magnitude. These must occur on bands that are defined (never in transition bands), and they must alternate in sign (even while jumping over transition bands). For the reader who has not studied the consequences of this theorem carefully, the examples of Fig. 28 may provide a useful measure of understanding.

Fig. 28a is a fairly conventional cases of a two-band filter (low-pass in this case). It is designed with a transition region between 0.2 and 0.223. The length is 20 so we have R=10 cosines and we find 11 extremals. Here we are plotting the magnitude, so it is necessary to recognize that actual sign changes occur in the stopband such that the sign of the error does alternate. Fig. 28b shows a case of a three band filter of length 23, thus having 12 cosines and 13 extremals. This has "don't care" transition regions between 0.1 and 0.15, and between 0.3 and 0.35, with desired magnitudes of 1, 0, and 2.

Fig. 28c shows a five band filter of length 23 with 12 cosines and 13 extremals. The transition bands are between 0.05 and 0.1, between 0.15 and 0.2, between 0.25 and 0.3, and between 0.35 and 0.4, with magnitudes on the defined bands of 1, 0, 1, 0, and 1. Here we see the values of the alternation theorem in telling us that the resulting response, despite a curious wiggle in the third passband, is none the less unique and best (and is a better approximation to 1 in this band than would be a full sized ripple).

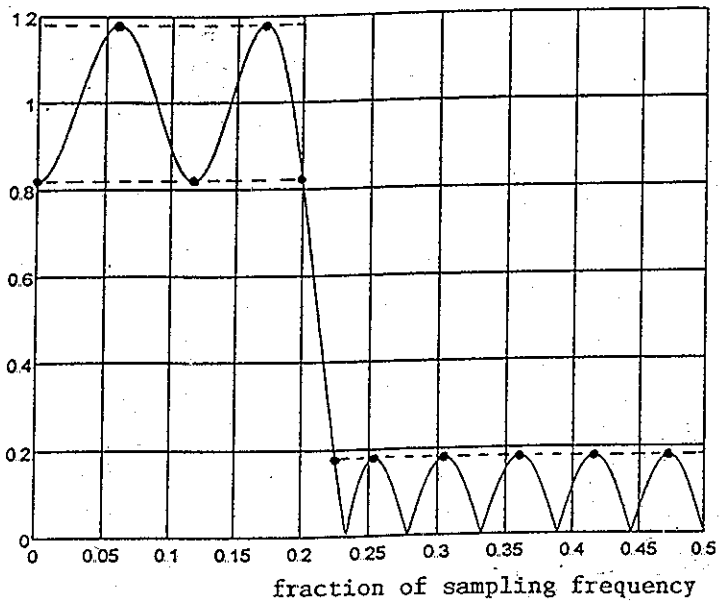


Fig. 28a A two-band low-pass of length 20 having R=10 cosines and 11 extremal frequencies (solid dots). Note that magnitude is plotted so the signs of the error in the stopband do alternate.

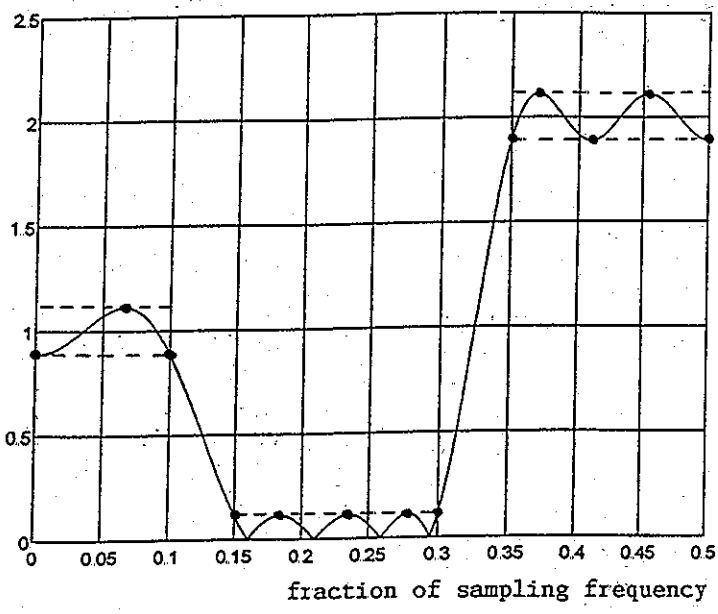


Fig. 28b A three-band filter of length 23 having R=12 cosines and 13 extremal frequencies (solid dots).

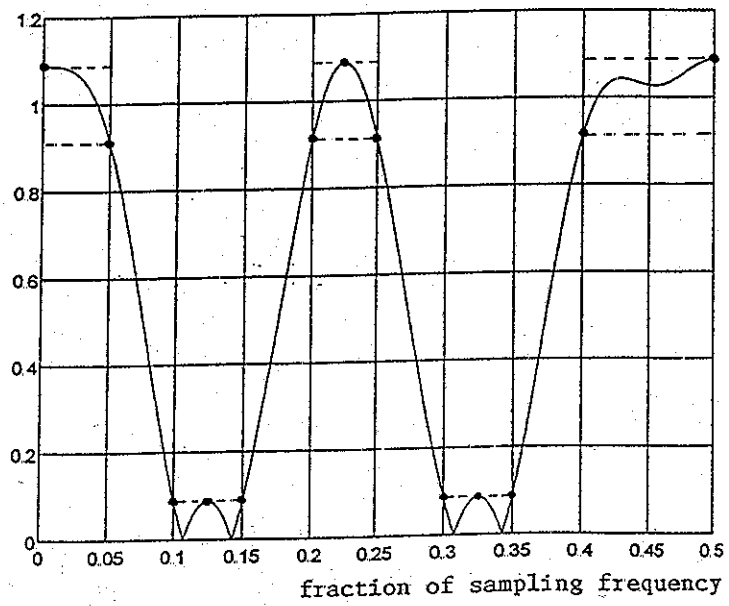


Fig. 28c A five-band filter of length 23 has R=12 cosines and 13 extremal frequencies. The filter has a suspicious "wiggle" in the final passband, so we rely on the alternation theorem to assure us that the response is optimal.

3c-2. Additional Mathematical Considerations

Some Basic Facts From Which to Argue

We can establish the following facts:

(1) The filter has R cosines. It must have R+1 (or more) extremals (Alternation Theorem).

(2) The response has at most R flat points. One of these is at 0 and another is at half the sampling frequency. This leaves at most R-2 internal flat points. (See Fig. 29)

(3) Extremals must occur either at band edges, or internal flat points.

(4) A filter with M bands has 2M band edges.

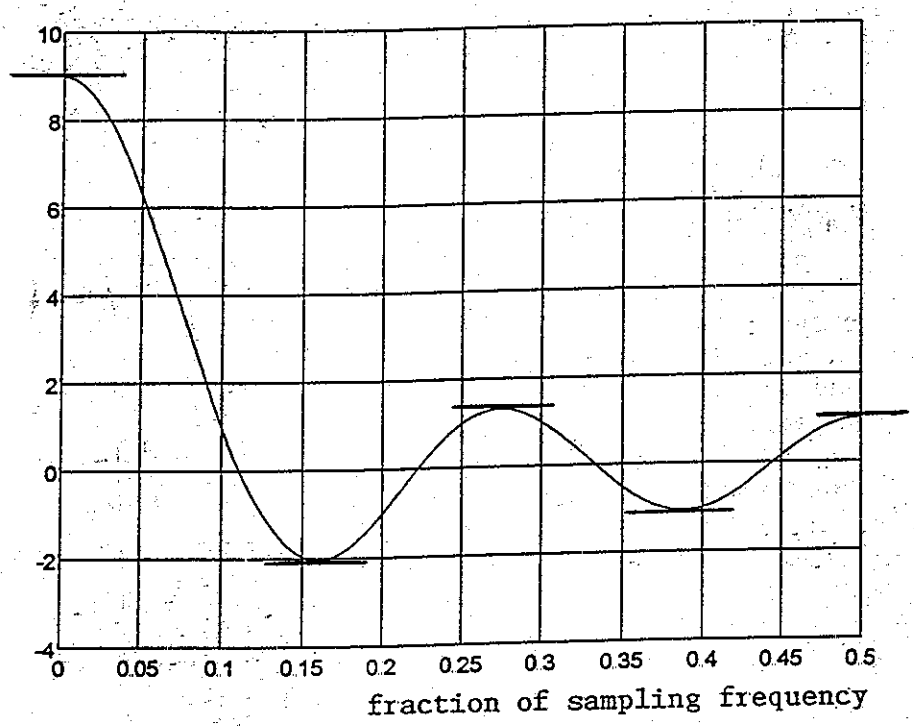


Fig. 29 Frequency response (not magnitude) of a length 9, non-causal moving average filter (before division by 9). The response has R=5 cosines and 5 flat points. Two of the flat points are at the ends, while R-2=3 are internal.

3c-3 The Maximum Number of Extremals for a Two-Band Filter is R+2

For a two-band filter, we have R-2 internal flat points and 4 band edges for possible extremals. This sets a maximum of R+2. Thus a two-band filter has either R+1 (minimum required) or R+2 (extraripple) extremals. Figures 30a through 30d show results that include these two cases. All of these cases have a length of 15 and a

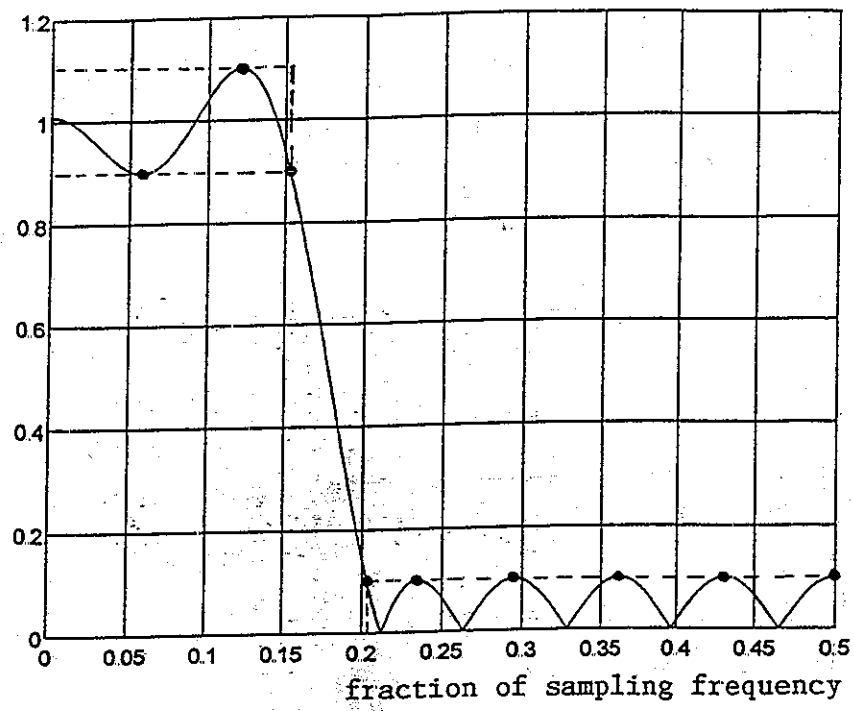


Fig. 30a A length 15 filter with 8 cosines and 9 extremals. The endpoint at 0 is not an extremal. There are three extremals in (or on the edge of) the passband. The transition region is from 0.1530 to 0.2030.

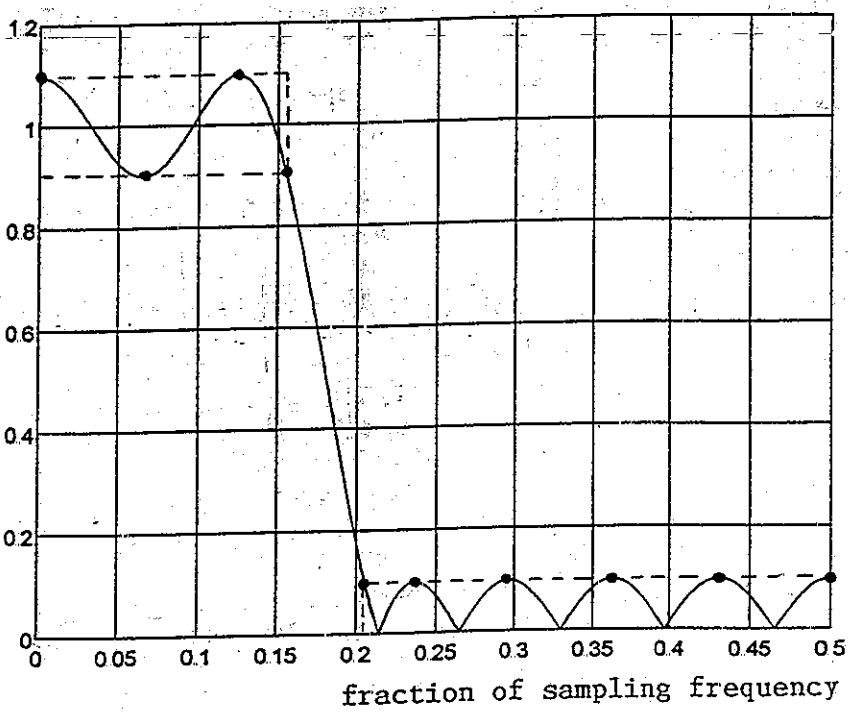


Fig. 30b A length 15 filter with 8 cosines and 10 extremals. The endpoint at 0 has become an extremal. This is an "extraripple" case. The exact same filter results if we just change the length to 17. The transition region is from 0.1555 to 0.2055.

transition bandwidth of 0.05. The differences are in the position of the upper edge of the passband, which varies from 0.1530 for Fig. 30a to 0.1605 for the case of Fig. 30d. As the most general trend, we see the 9 extremals redistribute from three in the passband (including bandedge), six in the stopband (including bandedge), Fig. 30a, to four in the passband, five in the stopband (Fig. 30c and Fig. 30d). This is what we expect to happen as the transition band moves up, widening the passband, and narrowing the stopband. Yet Fig. 30b stands out as having an extra extremal.

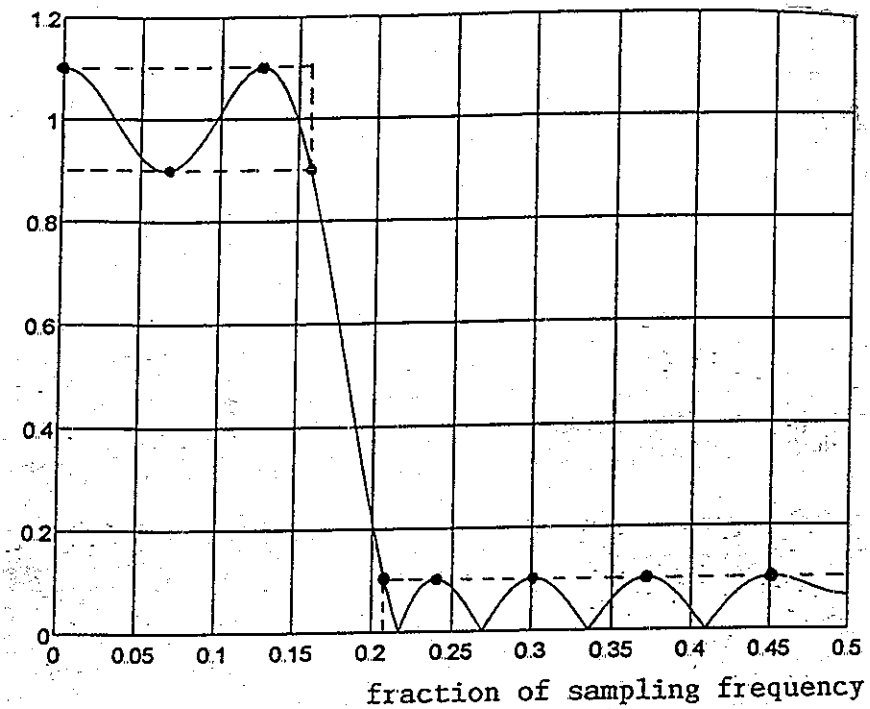


Fig. 30c A length 15 filter with 8 cosines and (back to) 9 extremals. The four passband extremals remain, but one disappears from the stopband, relative to Fig. 30b. The endpoint at 0.5 is no longer an extremal. The transition region is from 0.1580 to 0.2080.

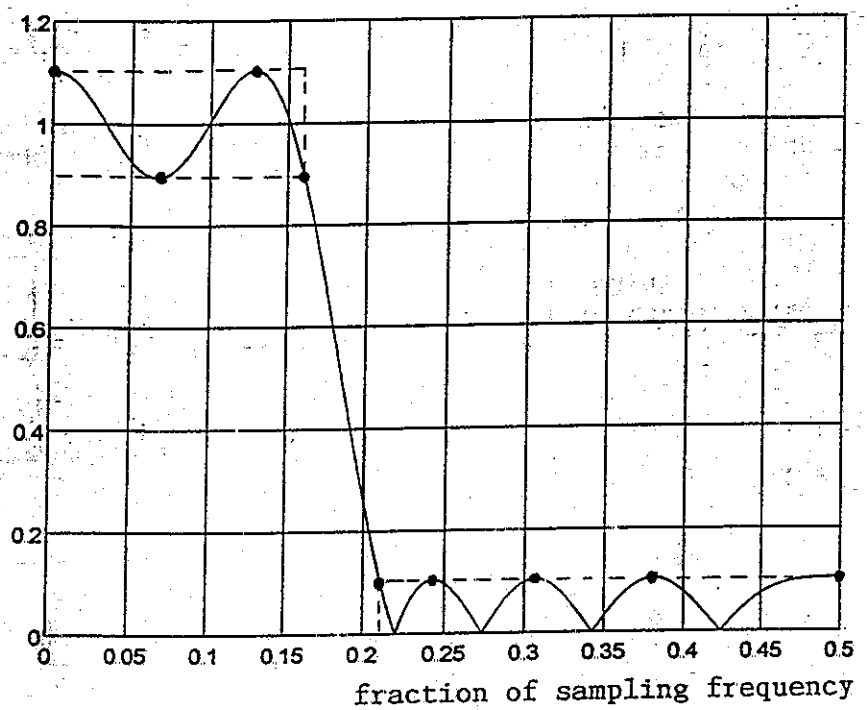


Fig. 30d A length 15 filter with 8 cosines and 9 extremals. Here the endpoint at 0 has again become an extremal, but there are only 5 stopband extremals as compared to Fig. 30a and Fig. 30b. The transition region is from 0.1605 to 0.2105.

In one sense, we can see Fig. 30b as a more or less inevitable "accident" where an extremal, in moving from one band to the other, momentarily appears in both (Table 1). More insight comes from simply trying to increase the length of the filter, leaving all the frequency bands the same. Changing from length 15 to length 17 gives us exactly the same filter. To be more specific, what happens is that the two extra tap weights on the

ends, that come into the increase from length 15 to length 17, calculate to be zero. They are not needed. If we look at the filter as length 15, there is an extra extremal. If we look at it as length 17 (the end taps just happen to be zero), we have 9 cosines and 10 extremals - nothing unusual.

TABLE 1

Figure	Passband Edge	Passband Extremals	Stopband Extremals	Total Extremals
30a	0.1530	3	6	9
30b	0.1555	4	6	10
30c	0.1580	4	5	9
30d	0.1605	4	5	9

3c-4 Transition Band Edges Must be Extremals for Two-Band Filter

The two band filter must have R+1 extremals, and only R-2 internal flat points are available. Thus, there must be 3 band edge extremals. Two of these must be the edges of the transition band. One end point (0 or π) must be a third extremal. [In the extraripple case, both 0 and π will be extremals, for a total of R+2, as in Fig. 30b.]

If it were the case that there were three band edge extremals, and that one of them was at 0, a second at π, with the third being one of the transition band edges, then not only would the other transition band edge not be an extremal, but the internal flat point closes to this second transition band edge would also not be an extremal, as the sign of the error would not alternate. That is, removing one of the edges would remove two extremals. This would result in one extremal fewer than the minimum required. Since all the internal flat points are already extremals, this can not be made up.

3c-5 Transition Band is Monotonic for Two-Band Filter

Suppose that the transition band of a two-band filter is not monotonic. This means that the response must turn upward and back downward in mid transition band. This requires two internal flat points being used up, leaving (R-2)-2 = R-4 for extremals. If all four band edges are extremals, this only gives (R-4)+4 = R extremals, and R+1 are required.

3c-6 A Three-Band Filter Need Not Have Monotonic Transition Regions

A three band filter has two transition regions, and one of these need not be monotonic, although both often will be monotonic. The non-monotonic case comes from the use of transition band widths that are substantially different.

That the non-monotonic case can occur is evident since the two internal flat points that would have to appear in the transition region are unavailable for extremals (by definition, extremals can never occur in the transition region), but these lost extremals can be made up by having two extra band edges become extremals.

When one transition region is narrow and monotonic, the response must have a very large slope in that region, which means that the response must involve cosines that are large and/or which wiggle fast. This overall "drive" toward sharp transition in a narrow region can not be simply "turned off" outside that region, since the amplitudes and frequencies are constant over all of 0 to π. In the defined bands, relatively flat response is possible through compensating slopes. However, if the other transition band is much wider than the first, the response, in an intentional "don't care" region, can oscillate wildly. Fig. 31 is an interesting three band response where we have a transition band or "don't care" region where we likely "do care" when we see the mischief that occurs. If necessary, this blow-up can be controlled with one or several narrow guide bands in the transition region.

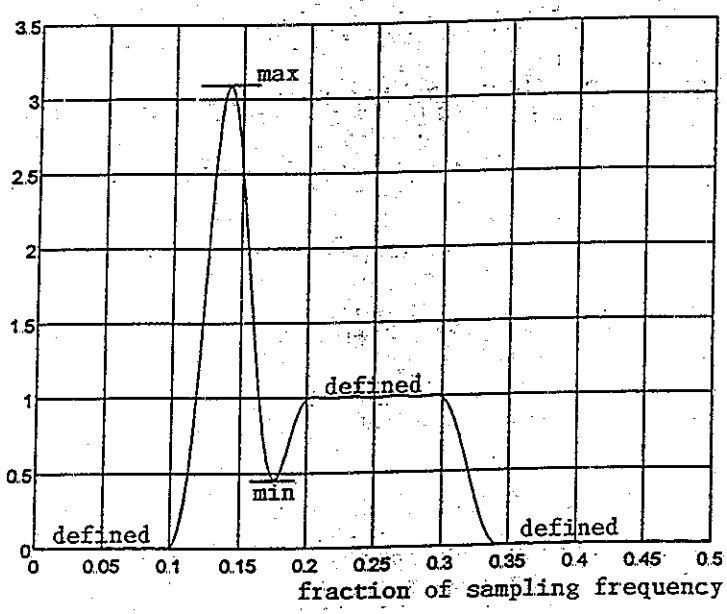


Fig. 31 This three band filter has unequal transition (don't care) regions, and the wider one from 0.1 to 0.2 develops two flat points (local max or min) which are not seen in the narrower band fro 0.30 to 0.34.

3c-7 A Five Band Filter

Filters with more than two bands can have more than R+2 extremals since more band edges become available for potential extremals. Fig. 32 shows a five-band filter of length 25, which should have 13 cosines and at least 14 extremals. In fact, it has 17 extremals total, three more than minimum. As with previous examples of extra extremals, we observe that we get this exact same filter if we design for lengths 27, 29, and 31. That is, no additional non-zero coefficients appear on the ends until we reach length 33. Here we have symmetry about $0.25f_S$ as well as additional symmetry about $0.125f_S$ and about $0.375f_S$. Without additional symmetry, additional extremals do not appear (Fig. 28c)

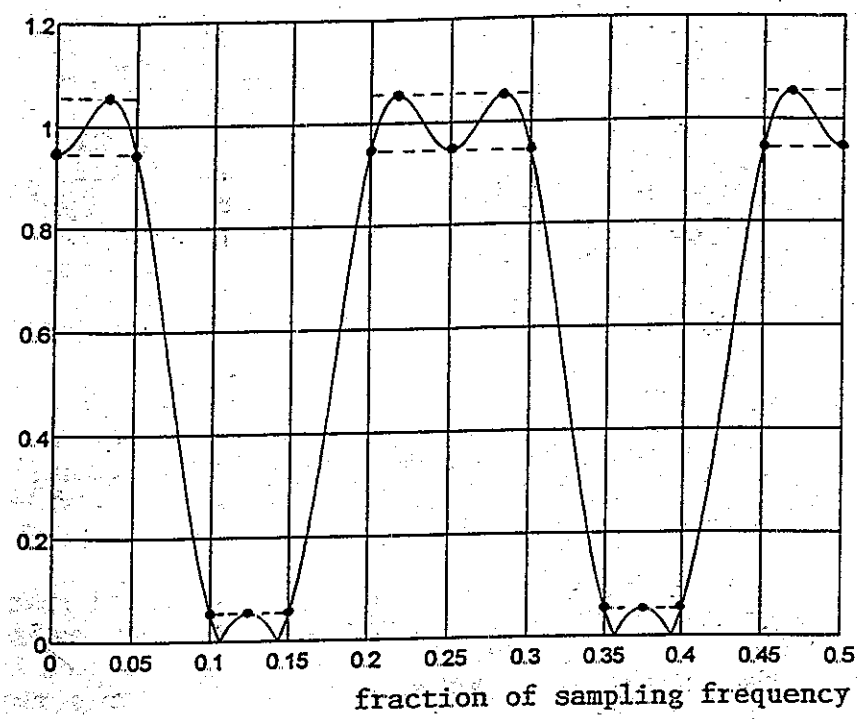


Fig. 32 This 5-band filter is length 25 and has 13 cosines. It must have 14 or more extremals. In fact, it has 17 extremals, which is acceptable. Because of the symmetry, only every fourth cosine of those available is actually used. Thus we get this exact same filter for lengths 27, 29, and 31. Length 31 would have 16 cosines and require 17 extremals, which is what we have here.

3c-8 Starting a Simple Example

Let's start by supposing that we want a linear-phase low-pass filter of length 5 which is equiripple about a passband equal to 1 for frequencies from 0 to 0.2π, and equiripple about a stopband equal to 0 for frequencies from 0.4π to π. Accordingly, the region from 0.2π to 0.4π is a "don't care" or "transition" band.

It is convenient to design a zero-phase non-causal filter, and to make it causal, we will then shift all the taps to positive times - a manipulation common for most FIR design methods. Since this is a length 5 linear phase filter, we can write its frequency response as:

$$H(e^{j\omega}) = h(-2)e^{2j\omega} + h(-1)e^{j\omega} + h(0) + h(1)e^{-j\omega} + h(2)e^{-2j\omega} \qquad (45a)$$

Since h(n) = h(-n) for our linear-phase filter, it is convenient to use:

$$H(e^{j\omega}) = a_0 + a_1 \cos(\omega) + a_2 \cos(2\omega) \qquad (45b)$$

where $a_0=h(0)$, $a_1=2h(1)$, $a_2=2h(2)$. This conversion is useful as it clearly shows the frequency response to be the sum of cosines (a Fourier series with the usual roles of time and frequency reversed). Perhaps more importantly here, we have reduced the problem from 5 unknowns to only three, which is important for hand calculations.

3c-9 Relationship to Frequency Sampling

From equations (45a) and (45b) we see that all possible filters (with the linear phase as specified) can be examined by trying different values of a_0, a_1, and a_2. Accordingly, three instances of equation (45b) are sufficient to determine the three amplitude coefficients, a_0, a_1, and a_2. To obtain three such equations, it is only necessary to know three value of ω and three corresponding desired values of $H(e^{j\omega})$ as:

$$H(e^{j\omega 1}) = a_0 + a_1 \cos(\omega_1) + a_2 \cos(2\omega_1) \tag{46a}$$

$$H(e^{j\omega 2}) = a_0 + a_1 \cos(\omega_2) + a_2 \cos(2\omega_2) \tag{46b}$$

$$H(e^{j\omega 3}) = a_0 + a_1 \cos(\omega_3) + a_2 \cos(2\omega_3) \tag{46c}$$

which are, in matrix form:

$$\begin{bmatrix} H(e^{j\omega 1}) \\ H(e^{j\omega 2}) \\ H(e^{j\omega 3}) \end{bmatrix} = \begin{bmatrix} 1 & \cos(\omega_1) & \cos(2\omega_1) \\ 1 & \cos(\omega_2) & \cos(2\omega_2) \\ 1 & \cos(\omega_3) & \cos(2\omega_3) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \tag{47}$$

and in shorthand form:

$$H = Ma \tag{48a}$$

The solution to this set of three equations is of course:

$$a = M^{-1} H \tag{48b}$$

This is one of the many forms of frequency sampling design. Without careful study, we can not say too much about what sort of response we will get and how much it might resemble some idealized filter from which the samples $H(e^{j\omega})$ may have been obtained. In fact, all we can say is that the resulting filter will have a response that goes exactly through the three sample points.

3c-10 The Equiripple Design Procedure

The equiripple design, like the frequency sampling design, involves the setup and solution of a number of equations. In fact, we shall see that there is an extra equation (and an extra unknown, the magnitude of the ripple) in this case. However, we do not expect to solve the problem with one set of equations, but rather expect to have to iterate toward the unique best solution. Further, we have in the equiripple case the alternation theorem to let us know how well we are doing as we iterate.

The first step in the procedure is to guess a set of "extremal frequencies" and then make the frequency response pass through these points. In some cases, particularly those involving filters of fairly high order, a "grid" of equally spaced points on defined bands of the specification serves as a first guess. At other times (such as in the simple case here) some additional educated guessing proves useful.

We might well proceed as follows. We know that the equiripple response should "alternate" between points where the error is positive and points where the error is negative. Thus we might think of trying three equations as follows:

$$H(e^{j\omega 1}) + \delta = a_0 + a_1 \cos(\omega_1) + a_2 \cos(2\omega_1) \tag{49a}$$

$$H(e^{j\omega 2}) - \delta = a_0 + a_1 \cos(\omega_2) + a_2 \cos(2\omega_2) \tag{49b}$$

$$H(e^{j\omega 3}) + \delta = a_0 + a_1 \cos(\omega_3) + a_2 \cos(2\omega_3) \tag{49c}$$

Here we would be thinking of $H(e^{j\omega})$ as a desired response (often 1 for a passband and 0 for a stopband), δ is the "error," and the ω_k are trial extremal frequencies. However, this is obviously just the same frequency sampling problem that we had above. The problem is that we don't know the error δ - it's not a constant or even a design parameter. [We could of course make the response go through any error if we specify it, but here we don't know what the minimized maximum error will end up being until we find the alternation theorem satisfied. Thus we have another unknown, and will need another equation.

Here, what we already found about equiripple design starts to make sense. Since we have another unknown, we need another equation, and thus, we need one more extremal frequency. In frequency sampling, we had R equations corresponding to the R amplitude coefficients a_n, and R values of the desired response. Note that R is the number of degrees of freedom (the number of cosines - counting cos(0) of course). The alternation theorem tells us that we must have R+1 (or more) extremal frequencies. Here we have just found that adding δ as an unknown requires an additional equation (and corresponding extremal frequency) beyond the R equations of frequency sampling. Accordingly, the requirement of R+1 extremals (a demand of a relatively difficult theorem) can be seen to be much more simply the requirement of N equations for N unknowns.

In the present example, we have three cosines amplitudes (R=3) and the error δ as unknowns. We can therefore add a fourth equation to those of equation (49), and at the same time, move the δ's more officially to the right side of the equations with the other unknowns.

$$H(e^{j\omega 1}) = a_0 + a_1 \cos(\omega_1) + a_2 \cos(2\omega_1) - \delta \quad (50a)$$

$$H(e^{j\omega 2}) = a_0 + a_1 \cos(\omega_2) + a_2 \cos(2\omega_2) + \delta \quad (50b)$$

$$H(e^{j\omega 3}) = a_0 + a_1 \cos(\omega_3) + a_2 \cos(2\omega_3) - \delta \quad (50c)$$

$$H(e^{j\omega 4}) = a_0 + a_1 \cos(\omega_4) + a_2 \cos(2\omega_4) + \delta \quad (50d)$$

In matrix form, these are:

$$\begin{bmatrix} H(e^{j\omega 1}) \\ H(e^{j\omega 2}) \\ H(e^{j\omega 3}) \\ H(e^{j\omega 4}) \end{bmatrix} = \begin{bmatrix} 1 & \cos(\omega_1) & \cos(2\omega_1) & -1 \\ 1 & \cos(\omega_2) & \cos(2\omega_2) & +1 \\ 1 & \cos(\omega_3) & \cos(2\omega_3) & -1 \\ 1 & \cos(\omega_4) & \cos(2\omega_4) & +1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} \quad (51)$$

and once again we have a shortcut:

$$H = Mb \quad (52a)$$

with inverse:

$$b = M^{-1}H \quad (52b)$$

where b is the vector $[a_0 \;\; a_1 \;\; a_2 \;\; \delta]$, H is the desired response, and M is the matrix, all as in equation (51). Note that equation (52b) is not the solution to the design problem, but rather only one iteration whose progress toward a final solution must be examined, using the alternation theorem.

3c-11 Iterating the Design

We will begin with a guess as to where the extremal frequencies are located. Suppose we recognize that in a two-band filter the transition band edges will be extremals, so we choose 0.2π and 0.4π as two of our extremal points. For the other two, we try 0 and π, the endpoints. That is, we are expecting some response as in Fig. 33.

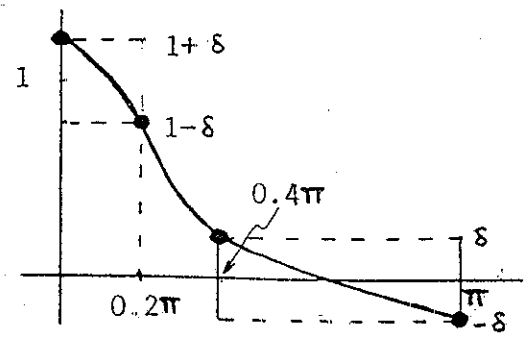


Fig. 33 A First Guess: Extremal Frequencies at 0, 0.2π , 0.4π , and π

We thus start with four equations:

$$a_0 + a_1 \cos(0) + a_2 \cos(2 \cdot 0) = 1 + \delta \quad (53a)$$

$$a_0 + a_1 \cos(0.2\pi) + a_2 \cos(0.4\pi) = 1 - \delta \quad (53b)$$

$$a_0 + a_1 \cos(0.4\pi) + a_2 \cos(0.8\pi) = \delta \quad (53c)$$

$$a_0 + a_1 \cos(\pi) + a_2 \cos(2\pi) = -\delta \quad (53d)$$

which are, in matrix form:

$$\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 0.809 & 0.309 & 1 \\ 1 & 0.309 & -0.809 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (54)$$

which are solved (using any convenient matrix inversion) to give:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 0.2019 \\ 0.6667 \\ 0.2981 \\ 0.1667 \end{bmatrix} \quad (55)$$

Which has the corresponding impulse response [equations (45a) and (45b)] of:

$$h = [\, 0.1491 \;\; 0.3333 \;\; 0.2019 \;\; 0.3333 \;\; 0.1491 \,] \quad (56)$$

With this result of the first iteration, we can calculate the actual frequency response, using some convenient program, or just calculating equation (45b). The result is shown in Fig. 34.

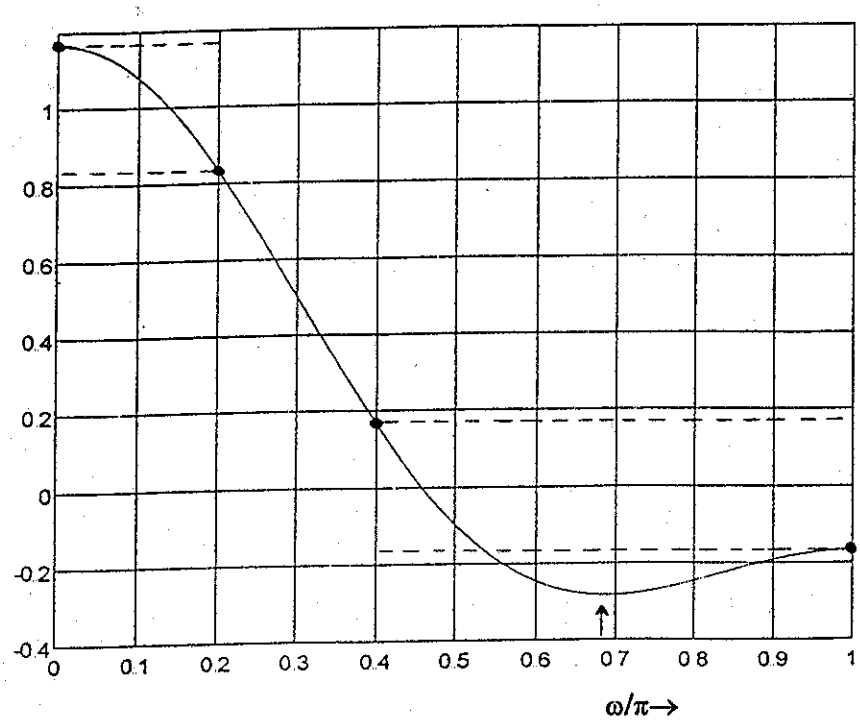


Fig. 34 Result of first iteration is a failure. The error exceeds the set ripple at about 0.68π. Frequencies shown as fractions of π.

From Fig. 34 we see that we have not guessed correctly. True enough, the response does go through the points specified, but these are not the extremals because there is an error larger than the current δ, and it occurs at a frequency of about 0.68π. We might suppose that the actual error is in choosing the response at π as $-\delta$ instead as $+\delta$. Indeed, solving for this endpoint as $+\delta$ will reduce the stopband error, but it would not be an extremal, as the sign of the error would not alternate. Instead, we do something very logical. We remove π, and in turn include 0.68π in our next trial set. The second iteration is represented by the matrix equations:

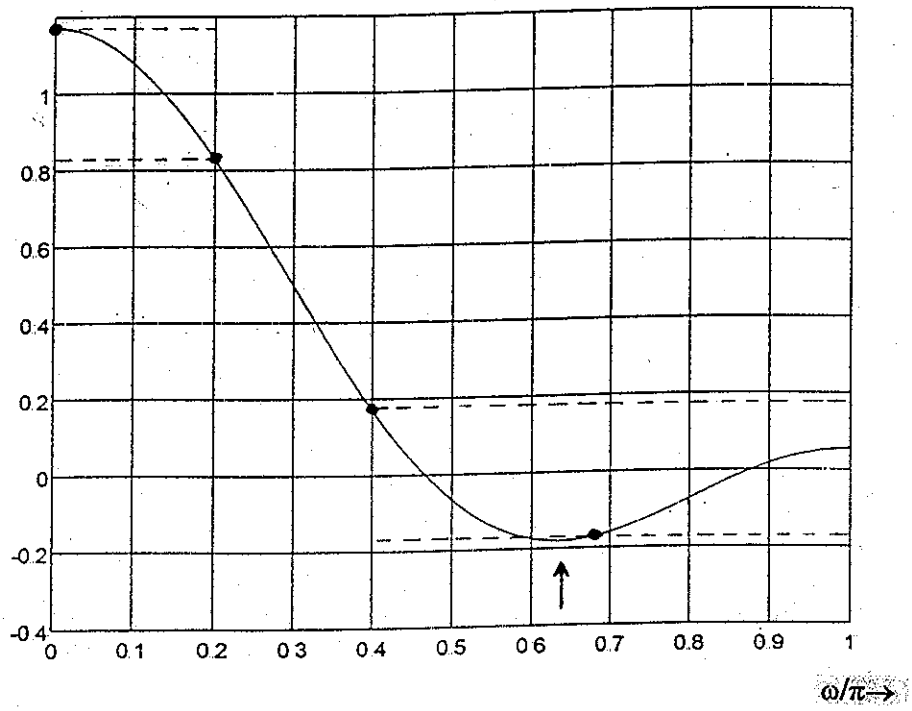
$$\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 0.809 & 0.309 & 1 \\ 1 & 0.309 & -0.809 & -1 \\ 1 & -0.536 & -0.426 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad (57)$$

Where the only changes are the two middle elements of the bottom row, which are $\cos(0.68\pi)$ and $\cos(1.36\pi)$ respectively. The solution is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 0.2331 \\ 0.5579 \\ 0.3397 \\ 0.1706 \end{bmatrix} \tag{58}$$

Notice that the error δ has increased from 0.1667 to 0.1706. This is in fact expected and necessary as we move toward the minimum, maximum error. The frequency response that results from this iteration is seen in Fig. 35. We note that we have a much better result, but the error is still excessive. Here we see that there is an error greater than the current δ at about 0.64 . [Note that this is an actual shift in the curve, and not a matter of our failing to correctly examine the error after the first iteration.]

Fig. 35 A second iteration is much better. The response now goes through the new trial extremal at 0.68π, but there is still a slightly larger error at approximately 0.64π.



So 0.64π replaces 0.68π and the matrix equation for the third iteration is:

$$\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 0.809 & 0.309 & 1 \\ 1 & 0.309 & -0.809 & -1 \\ 1 & -0.426 & -0.637 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{59}$$

Where the only changes are the two middle elements of the bottom row, which are $\cos(0.64\pi)$ and $\cos(1.28\pi)$ respectively. The solution is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \delta \end{bmatrix} = \begin{bmatrix} 0.2808 \\ 0.5461 \\ 0.3442 \\ 0.1711 \end{bmatrix} \quad (60)$$

Here we notice that the error has again increased, slightly, from 0.1706 to 0.1711. Again, this is what we expect. The frequency response that results from this iteration is seen in Fig. 36. This is pretty much the right answer, and we will not need to improve on it through the use of a fourth iteration. There are now four extremals, all of which have the maximum error. This has an impulse response:

$$h = [\, 0.1721 \;\; 0.2730 \;\; 0.2808 \;\; 0.2730 \;\; 0.1721 \,] \quad (61)$$

This is the same result we get if we use a filter design program for this method.

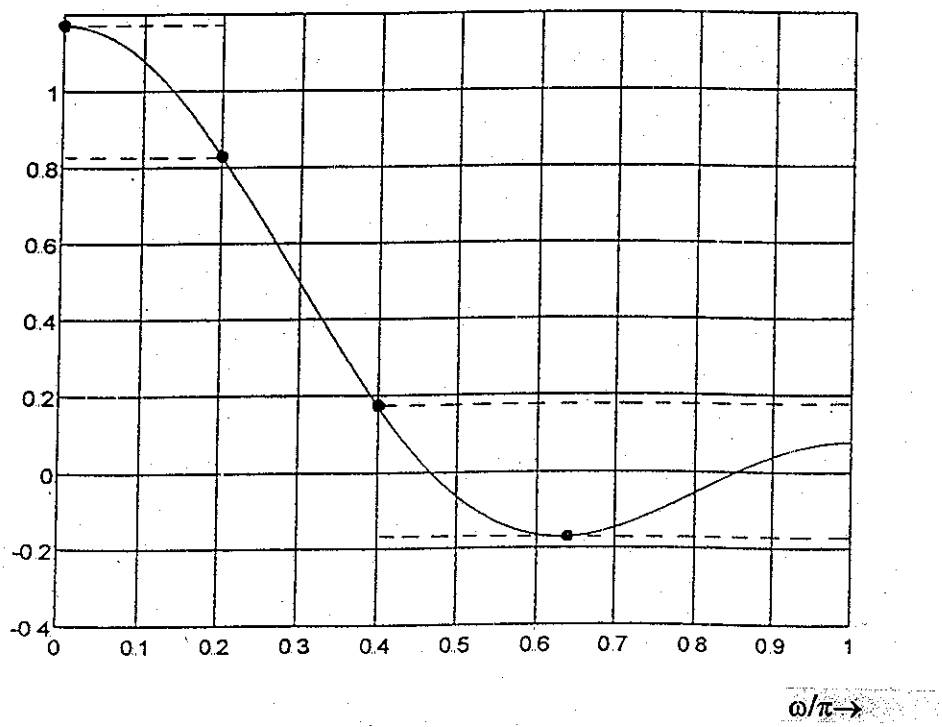


Fig. 36 A third iteration is essentially perfect.

4. TIME DOMAIN FILTER DESIGN METHODS

4a. IMPULSE INVARIANT DESIGN METHOD

4a-1 Overview

There is perhaps no digital filter design method that is easier to state than the Impulse Invariant (IIV) method. As with other Infinite Impulse Response (IIR) methods, notably Bilinear z-Transform, we start with an analog prototype filter. [Caution: the terms IIV and IIR are easily confused.] If this analog filter has an impulse response g(t), then the corresponding IIV digital filter has in impulse response:

$$h(n) = g(nT) \tag{62a}$$

where T is the sampling interval chosen. With Bilinear z-transform, we preserved the shape of the frequency response. With IIV, we preserve the shape of the impulse response. From here on, it gets more complicated in practice.

One difficulty implicit in equation (62a) is that it describes the sampling of an analog waveform g(t). In consequence, we must look at this in relation to sampling principles. We are accustomed to applying sampling ideas to what we think of as "signals," and we do not usually think of the impulse response of a filter as a signal, but we can, and here we need to do so. What if anything do we know about the bandwidth of g(t)? Well we know exactly that it corresponds to the frequency response of our analog prototype filter. Further, we know that these filters are not absolutely bandlimited. In Fig. 14 we have looked at low-pass filters and see that these have a response that is appropriately concentrated around low frequencies, but it does not go completely to zero as frequencies go higher and higher. We know from our knowledge of sampling that h(n) as in equation (62a) will have a frequency response consisting of the original response, and replicas of this response, spaced at intervals of 1/T. That is, the response will be aliased (Fig. 37), and this may be a serious problem. In fact, if we try to design a filter based on an analog prototype that does not at least <u>try</u> to go to zero at infinity, we anticipate no success. Accordingly, the IIV method is restricted to low-pass and bandpass type responses, while high-pass and notch types are absolutely ruled out (Fig. 37). And even so, we must be careful of low-pass and bandpass - to keep the sampling interval low enough.

This replicate and overlap phenomenon with IIV is often simply called "aliasing." This usage of the term aliasing is slightly different from the usual, as it applies to the filter's response, and not to anything that happens to a signal passing through the filter. It will become clear that we can enforce the IIV principle, equation (62a), even when the responses are hopelessly overlapped (aliased). We just don't get very good filtering. With IIV we are always going to have aliasing to some degree, but not in the sense that any additional frequencies are created in the baseband.

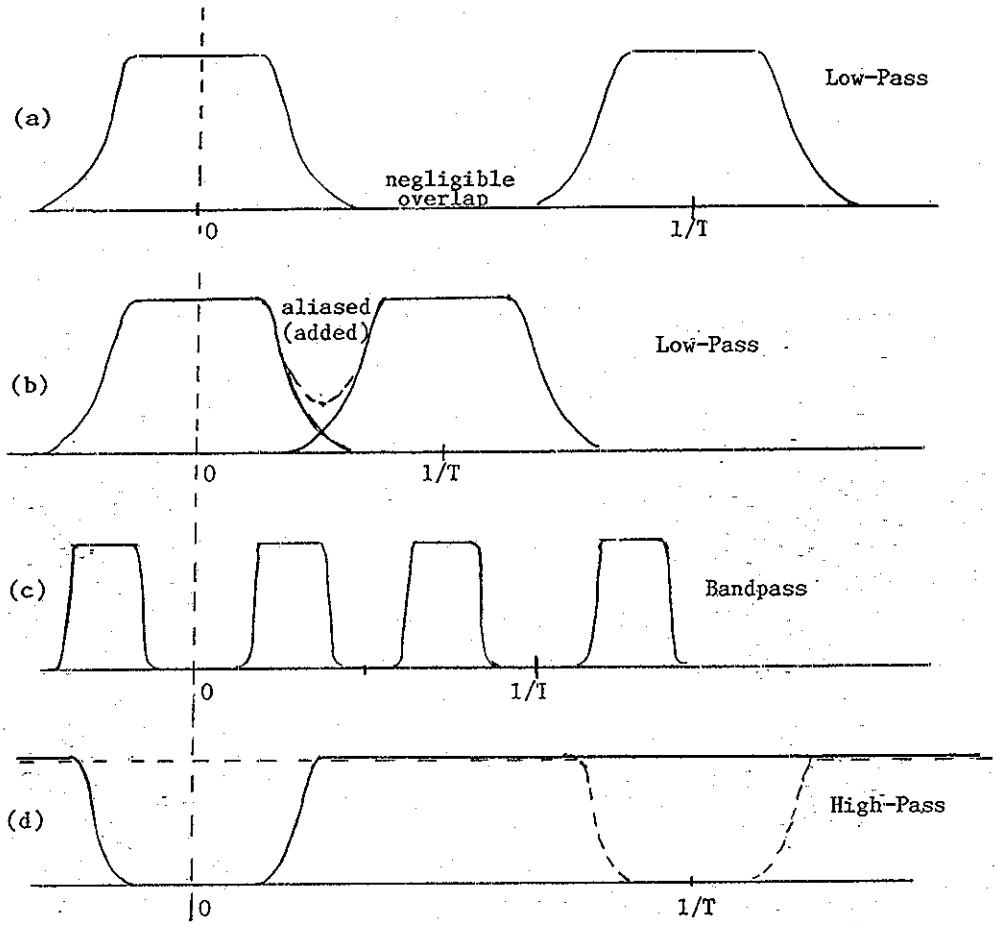


Fig. 37. In (a) we see an appropriate choice of T for a low-pass IIV design, while in (b) there is overlap which spoils the stopband. In (c) we have an acceptable bandpass. In (d) we have a high-pass attempt which aliases to a constant for all frequencies.

Recall however from our study of Bilinear z-Transform that we do not expect to begin with a prototype g(t), but instead, with the transfer function T(s), or even just from the poles (Fig. 14). Further, we do not really want to end up with h(n) in this IIR case, but instead we probably want the equivalent transfer function H(z). [In the FIR case, the impulse response h(n) also gave us the filter coefficients used in practice. In IIR, we have the denominator to complicate matters.] Accordingly we can recast equation (62a) as:

$$H(z) = Z\left[\, L^{-1}\{\, T(s)\,\}\,\Big|_{t=nT}\,\right] \qquad (62b)$$

In practice, we will likely start with a knowledge of the pole positions, and from them, obtain T(s) as in equation (23b). We then take the inverse Laplace transform of T(s) to get g(t). Then g(t) is sampled to get h(n), and H(z) is the z-transform of h(n). Thus we see equation (62b) as a somewhat more informed version of equation (62a) but otherwise the same.

The clear bottleneck in this procedure is the inverse Laplace transform. This is easy for first-order or second-order, but becomes rapidly more difficult. Yet it is when we consider how to do the inverse Laplace transform that we find that the IIV procedure itself becomes one of breaking the filter into parallel sections, and passing through the IIV transformation with terms in parallel.

Suppose we have a low-pass filter with only poles to deal with. We can easily form T(s) from the poles, in cascade form as:

$$T(s) = \prod_{k=1}^{M} \frac{c_k}{(s - p_k)} \tag{63}$$

where p_k is the kth pole of M total. The c_k can all be taken to be one, or to any values that achieve an overall gain that is satisfactory. To simplify matters (but as importantly, because it corresponds to practical cases), we will assume from this point that all poles are first-order ("simple poles"). That is, we may have as many poles as we want, and we certainly expect complex conjugate pairs, but there is never more than one pole in any position. This cascade or product form of T(s) is not well suited to inverse Laplace transform. Instead we would need the parallel or summation form:

$$T(s) = \sum_{k=1}^{M} \frac{d_k}{(s - p_k)} \tag{64}$$

This is a classic "partial fraction" expansion. Here the coefficients d_k can be obtained by the "residue" method (remember we allow only first-order poles):

$$d_k = T(s)(s-p_k)\Big|_{s=p_k} \tag{65}$$

Alternatively we recognize that nearly all our poles come in complex pairs (a pole p_k implies a conjugate pole p_k^*). In such a case, the partial fraction coefficient corresponding to p_k^* is d_k^*. We then find that two conjugate terms in the sum of equation (64) can result in a second-order term with purely real coefficients [see equation (74)]. So we want to take a slightly broader view of equation (64) to assume that it includes mostly (all, or all but one) such pairable terms. That is, we will, if convenient, do the IIV transformation on second-order parallel terms instead of just first-order terms.

The transforms in equation (62b), Laplace transform and z-transform, are linear. So once we achieve equation (64), or an equivalent form with second-order parallel terms, we can sample and z-transform these individually, and then sum the z-transforms. Inherently, we are going to get a parallel structure for our digital filter at this point (to be derived later):

$$H(z) = \sum_{k=1}^{M} \frac{d_k}{1 - e^{p_k T} z^{-1}} \tag{66}$$

We may in fact prefer to use that structure in a realization. Or, we can multiply out the terms for a cascade form.

Some interesting results with IIV are that the analog poles arrive in the z-plane by the absurdly simple mapping of:

$$p_{zk} = e^{p_{sk} T} \tag{67}$$

where p_{sk} are the analog poles and p_{zk} are the digital poles. Since we look at our low-pass as having only poles, we might assume that the design procedure is only a matter of mapping poles, just as we did in Bilinear z-transform. Unfortunately, there are zeros that appear. These can be seen to result from the repeated cross multiplication of the terms when putting equation (66) in cascade (product) form. And, we don't know how to easily locate these zeros, except to note that we hope they will end up in the very general vicinity of z=-1. The fact that they do not end up at exactly z=-1 (as does happen with Bilinear z-Transform) is another way to understand the aliasing of the response.

4a-2 IIV at First-Order

Consider the simple first-order analog low-pass, an R-C filter as shown in Fig. 38a. This has transfer function:

$$T(s) = (1/RC) / (s + 1/RC) \tag{68}$$

which has a pole at s=-1/RC, and an impulse response:

$$g(t) = (1/RC)e^{-(t/RC)} u(t) \tag{69}$$

where u(t) is the usual unit step which is equal to 1 for $t \geq 0$, 0 otherwise. The magnitude of the frequency response is shown in Fig. 38b. The impulse response of the digital filter is thus:

$$h(n) = g(nT) = (1/RC)e^{-(nT/RC)} u(n) \tag{70}$$

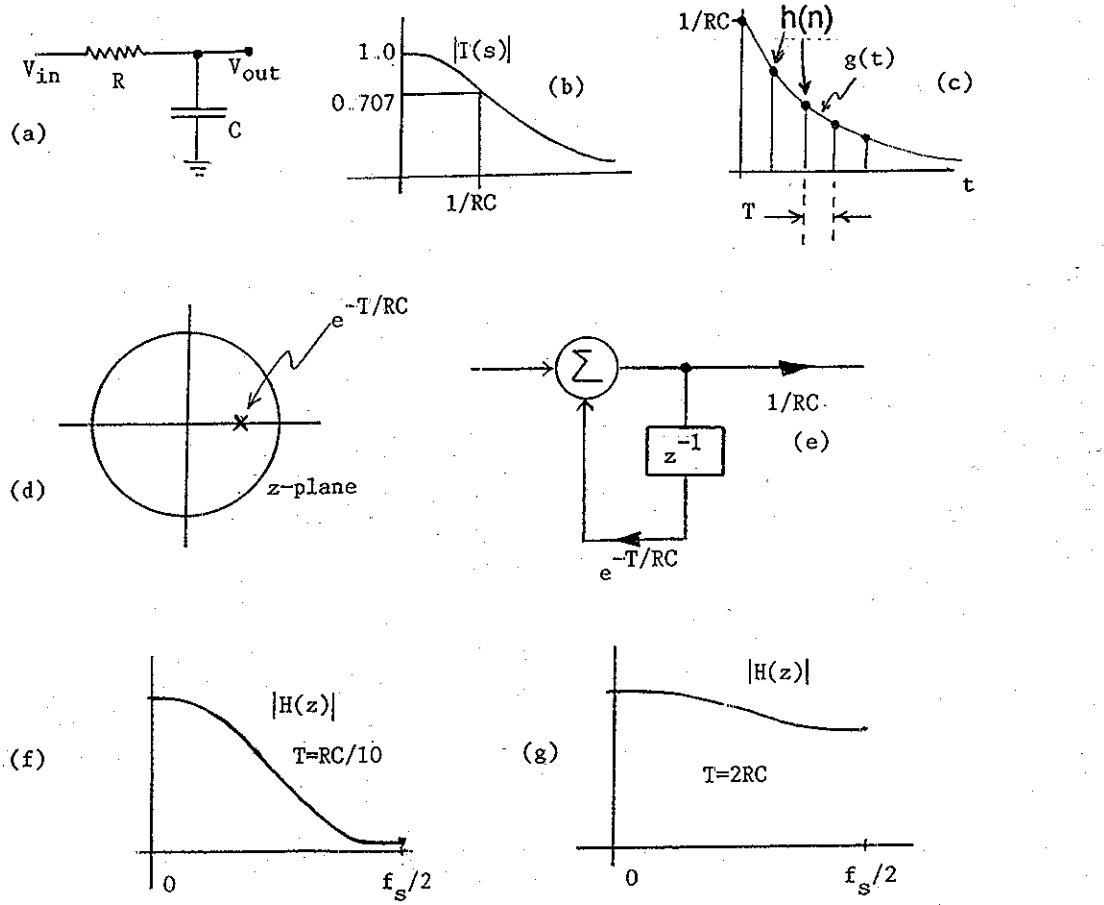


Fig. 38 IIV at first-order. The simple R-C low-pass (a), (b) and (c) is transformed to a digital filter with pole (d) and network (e) such that h(n) for (e) is g(nT) in (c). Unless T is small enough, as in (f), the result may be a very poor low-pass (g).

as shown in Fig. 38c. The z-transform of h(n) is:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} = \sum_{n=0}^{\infty} (1/RC)(e^{-T/RC})^n z^{-n} = \frac{1/RC}{1 - e^{-T/RC} z^{-1}} \qquad (71)$$

which results from summing the geometric series. Equation (71) will serve to justify the form of equation (66) as well. This H(z) has a pole at $z = e^{-T/RC}$ (Fig. 38d). Fig. 38e shows the network for H(z) while Fig. 38f and Fig. 38g show magnitude response for H(z) for different values of T. We see here that when T is very small (RC/10) that the poles is at 0.9048, close to z=+1, and there is minimal aliasing, and the magnitude response at z=-1 is small (a reasonable low-pass for first-order). On the other hand, when T is large (2RC) the pole is at 0.1353, closer to z=0 than to z=1, and a very poor low-pass response is the result. We interpret this result as "aliasing," but it is seen also simply as a poor placement of the pole in the z-plane.

4a-3 IIV at Second-Order and Higher

A second-order analog low-pass would have the form:

$$T(s) = 1 / [(s-p)(s-p^*)] \tag{72}$$

where p and p* are a pair of conjugate poles. If we take $p = -\sigma + j\Omega$, then using the residue equation (65) we obtain the partial fraction expansion of T(s) as:

$$T(s) = \frac{-j/2\Omega}{s - (\sigma+j\Omega)} + \frac{j/2\Omega}{s - (\sigma-j\Omega)} \tag{73}$$

Now, exactly as we did for first order, these two sections can be transformed in parallel. [Note: that the first-order pole was purely real, but the math is the same for a complex pole.]

$$H(z) = \frac{j/2\Omega}{1 - e^{(\sigma+j\Omega)T}z^{-1}} + \frac{-j/2\Omega}{1 - e^{(\sigma-j\Omega)T}z^{-1}}$$

$$= \frac{(1/\Omega)e^{\sigma T}\sin(\Omega T)z^{-1}}{1 - 2e^{\sigma T}\cos(\Omega T)z^{-1} + e^{2\sigma T}z^{-2}} \tag{74}$$

The network corresponding to this transfer function is shown in Fig. 39a.

The impulse response of the analog low-pass is the inverse Laplace transform of equation (72) and is given by:

$$g(t) = (1/\Omega)e^{\sigma t}\sin(\Omega t) \tag{75}$$

Choosing σ=-0.1 and Ω=1, we obtain the curve shown in Fig. 39b. The impulse response of the digital filter is easily obtained by simulation. We would first assume a value for T and calculate the coefficients. Then assuming that an impulse (a sample of value 1 at n = 0, and samples of value 0 for all other n) arrives at the input, we trace the resulting output. Thus the data at w_1 is moved to w_2, that at w_0 is moved to w_1, and the resulting summation is computed, replacing the moved w_0. In Fig. 39b we see overplots for T=0.3 and T=1. Note that both obey h(n) = g(nT). But the difference is again in the magnitude responses, not the impulse responses, and we see (Fig. 39c) that there is considerable aliasing for the larger value of T.

Above we mentioned the problem with zeros, and as yet, for the first and second-order examples we have shown, no zeros appeared. In consequence, we could have just used equation (67) to map the analog poles to digital. The zeros will appear at 3rd order and above, and most filters of practical interest will be of such higher orders.

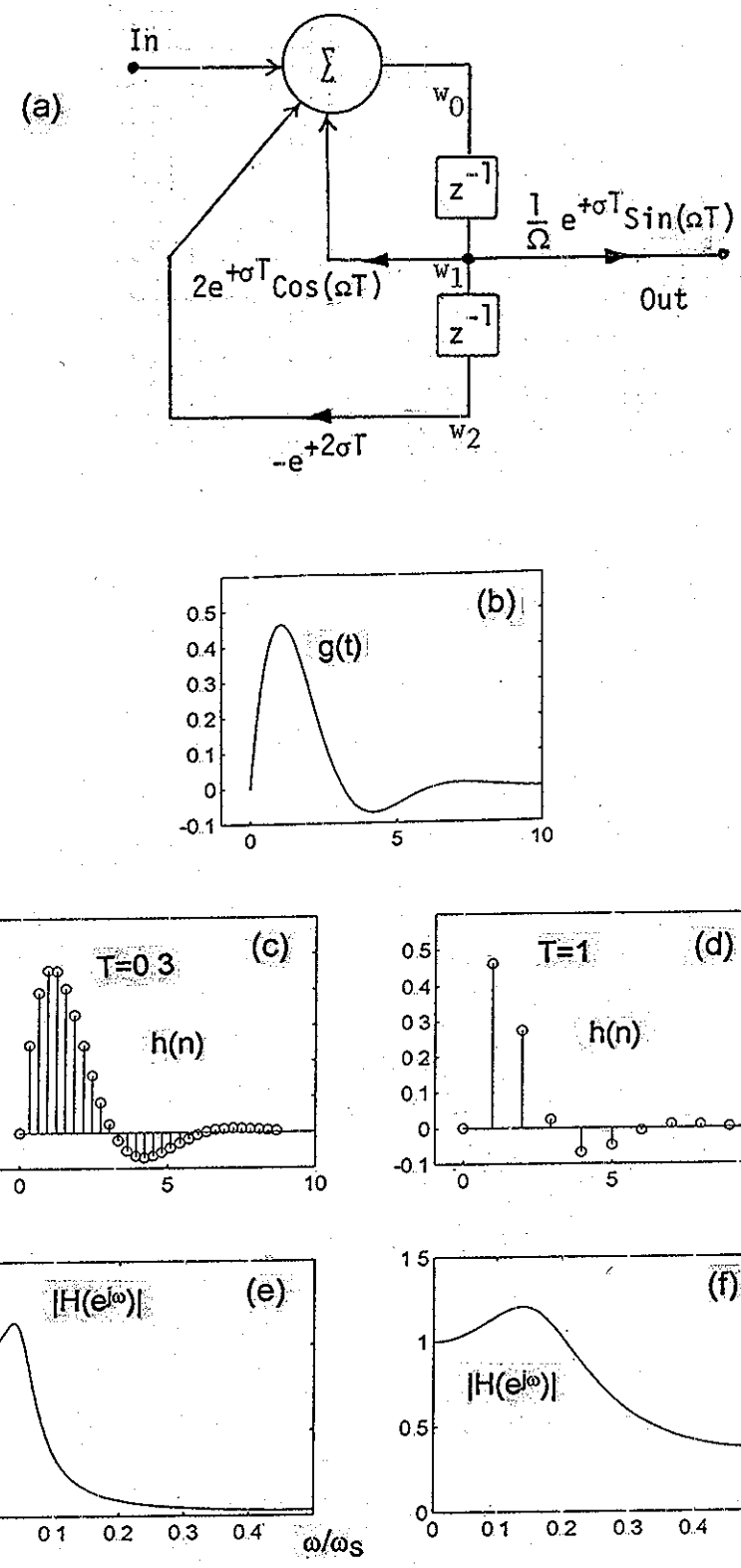


Fig. 39 The second order network corresponding to equation (74) is seen in (a). The filter coefficients depend on σ, Ω, and of course, on T. For σ=-0.6 and Ω=1, the analog impulse response is seen in Fig. 39b. Then we choose T=0.3 (c), (e) or T=1 (d), (f). We see that the denser sampling of the impulse response (T=0.3) results in a superior magnitude response.

When approaching a higher order filter, we may well think in terms of staying with first-order sections rather than combining to second-order sections where possible. This is because modern math and signal processing software readily handles complex numbers, even in denominators.

In addition to impulse-invariance, we sometimes see filters select for invariance to other time-domain responses. In particular, there is step invariance:

$$[z/(z-1)]H(z) = Z\left[L^{-1}\{ T(s)/s \} \Big|_{t=nT} \right] \tag{76}$$

and ramp invariance:

$$[z/(z-1)^2]H(z) = Z\left[L^{-1}\{ T(s)/s^2 \} \Big|_{t=nT} \right] \tag{77}$$

4b. INTERPOLATOR BASED FILTER DESIGN

4b-1 The Implementation of Interpolation

It is often the case that we have a set of samples representing some signal. For various reasons, we may desire to have a somewhat different set of samples representing the same signal. For example, we might wish to have samples with twice the density in time (twice the sampling rate), or perhaps a sampling rate that is 25% higher. Since we believe that under the right conditions, we can recover a continuous time signal from its samples, we can expect to be able to make the sort of changes suggested here. We also hope to be able to make these changes without actually going back to the analog world and then resampling. What we require is digital interpolation. We want to use the samples we have, which contain all the information we need, and from them obtain the alternative set of samples. This is a digital low-pass filtering, and fairly general filter design procedures can be used to obtain the necessary filters. Some methods of interpolation are, however, at least initially presented in the time domain.

Possibly the simplest type of interpolation is linear interpolation. That is, we assume that any and all possible samples that might be supposed to exist between any two sample points must lie on a straight line that connects those two points. While this is unlikely to be exactly right, we recognize that all interpolation is in a sense guessing, and linear interpolation is a guess that is probably not terribly wrong. From this example, we get two ideas. First, we can think of possible implementation of interpolation as a time domain procedure. Secondly, we might want to consider better

interpolation procedures by fitting higher order polynomials to data points. (A straight line is of course a first-order polynomial.)

Now, it might seem reasonable that if we intend to accomplish interpolation, and if we have previously found low-pass filtering to be some type of interpolation, and if we then actually do interpolation in the time domain, that the result should be somehow equivalent to low-pass filtering. On the other hand, fitting curves to points (solving simultaneous equations) seems unlikely to end in a filtering procedure. It is thus a surprise that not only can this curve fitting be done by filter structures, but also we get linear phase FIR filters as a result.

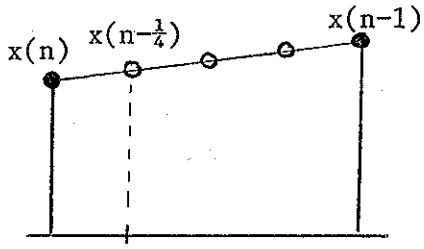
Typically the design is understood by actually solving for the (unknown) coefficients of a polynomial of degree N, fit to N+1 (known) points. These coefficients are of course different for different sets of input points. However, when we then calculate the value of a replacement sample, at some selected position with respect to the original samples, we find that the result is simply a weighted sum of the original samples. That is, the coefficients are an intermediate step of which we do not need to keep track. The replacement samples are calculated in terms of the coefficients, but the coefficients were calculated in terms of the original samples. So the replacement samples are found in terms of the original samples. For example, with linear interpolation, the value of a replacement sample 1/4 of the way from a first sample to the second, is calculated as 3/4 the closer sample and 1/4 the further sample, regardless of the actual sample values (Fig. 40a). This allows us to calculate the set of samples that is offset from the originals by 1/4. Likewise, we could for example calculate the samples offset by 1/2, and by 3/4 (Fig. 40b). These could be sequenced into the intermediate positions, at four times the original rate, by the rotary switch (commutator) shown in Fig. 40b. Interleaving these three filters and feeding in samples with three padded zeros results in exactly an FIR filter and produces a 4:1 interpolation (Fig. 40c). This is the so-called "polyphase" implementation. In our example, we would end up with an impulse response that is 1/4, 2/4, 3/4, 4/4, 3/4, 2/4, and 1/4 (Fig. 40d). This is a low-pass filter. In fact, the impulse response is the convolution of two length four rectangles, so the frequency response is the square of the frequency response of a length four "moving average."

This procedure of designing filters for each offset and combining then into poly-phase structures is part of the "mulit-rate" DSP art, and is time-consuming, even though relatively straightforward. Here we are more interested in the resulting filters, and would like to look at the results rather than the details.

4b-2 Impulse Response as Response to an Impulse

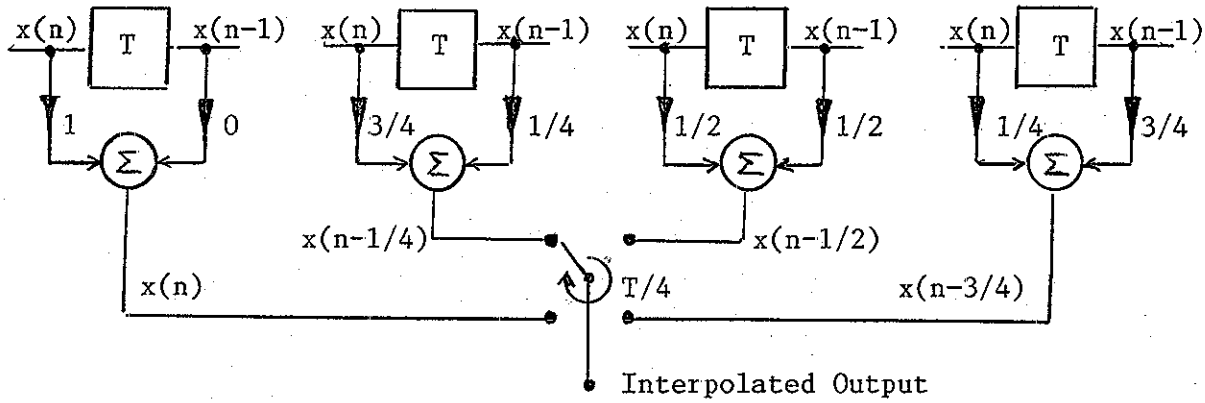
Suppose we are asked to find the impulse response of a linear interpolator. The answer (which sounds like a trick) is that it is the response of a linear interpolator to an impulse. What we are saying is, that if an impulse arrive at the input, what should the output be? It is easy to see that this should be the triangular shape we found in Fig.

(a)

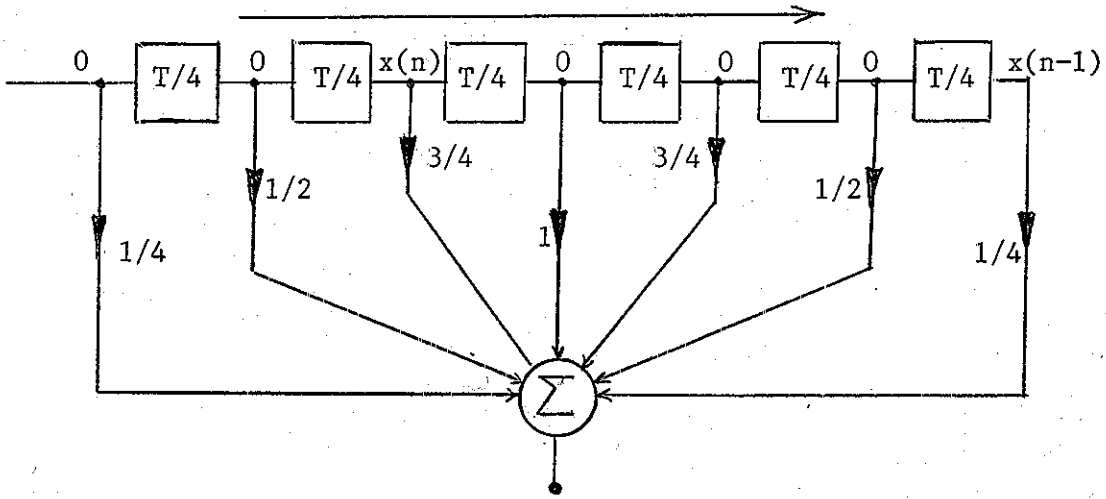


$x(n-\frac{1}{4}) = (3/4)x(n) + (1/4)x(n-1)$

(b)



(c)



(d)

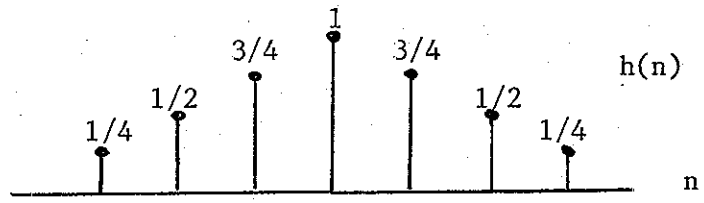


Fig. 40 Development from polynomial fit to impulse response h(n). In (a) we see the linear fit, in (b) we have the filters computing the offset samples, in (c) we have sequencing through the use of a zero-padded input, and in (d), the impulse response.

40d. There we are placing three intermediate samples between all original samples. With an input impulse, for most all outputs, these added samples are zeros stuck between zeros. In the one case of the actual impulse, there is a ramp upward, and a ramp downward.

Fig. 41 as a slightly different and more general view of the linear interpolator. In finding the impulse response, we have only two non-zero instances of fitting a straight line to two points: the impulse on the left, and the impulse on the right. Combining these and offsetting them in time in the proper way, we get our triangle shape. We also show non-productive segments on either side of the triangle. We have then only to sample this triangle, as in Fig. 40d. Note that the more points we interpolate, the lower will be the low-pass cutoff, relative to the sampling frequency.

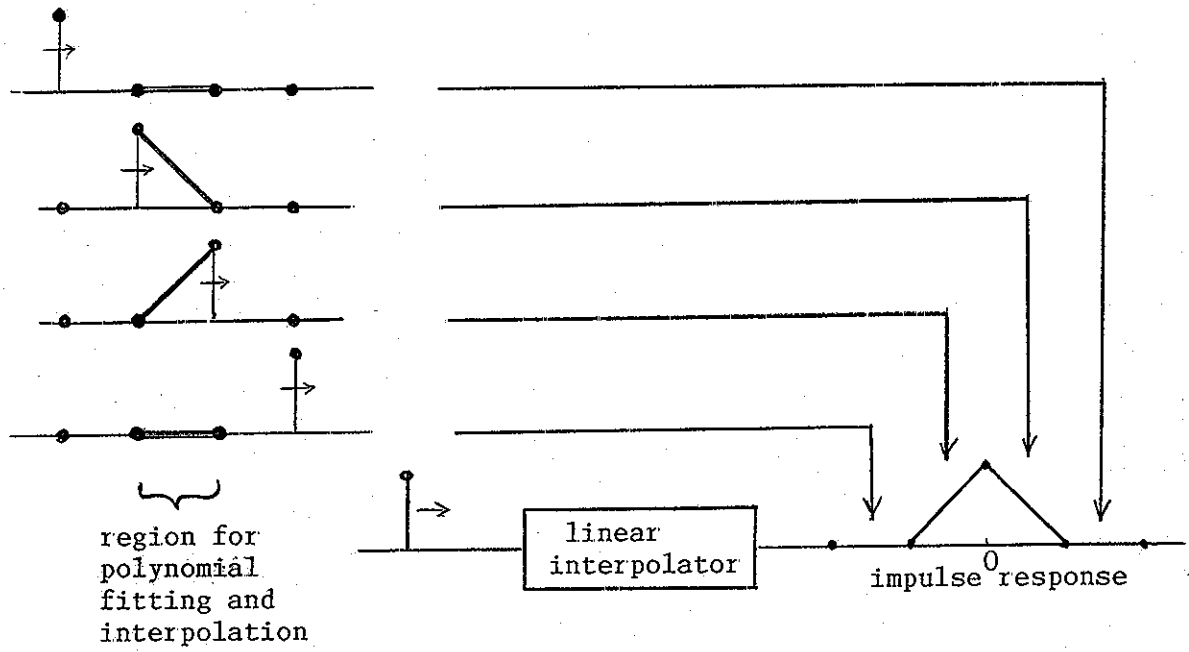
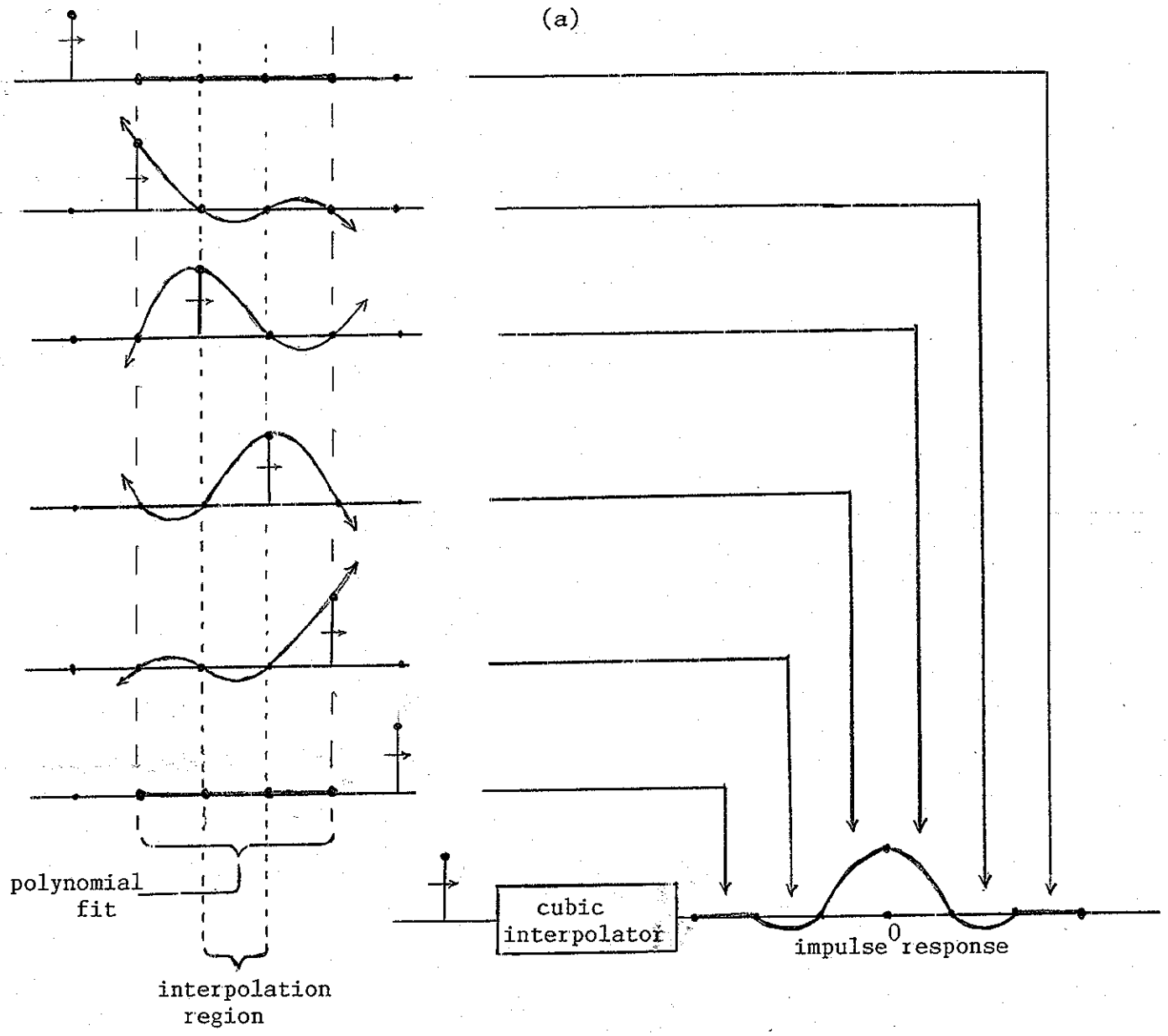


Fig. 41 The choice of linear interpolation results in the triangular impulse response shown. The corresponding digital filters are obtained by sampling this response.

4b-3 Cubic Interpolation

We have now found several ways to look at polynomial interpolation filters, and we understand linear interpolation very well. What happens with higher order polynomials? Well, things can get rather confusing for a number of reasons. If we go to 2nd-order (fitting a parabola to three points), we now have two possible regions (either side of the middle point) from which to extract additional samples. We might comfortably look for points half way in either direction about the middle. But even when we decide this, is there an easy way to find the impulse response? Is the impulse response a parabola that is fit to the points (-1,0), (0,1), and (1,0)? No, because in fitting the parabola to three points, there are three instances where the impulse is involved, just as there were two in Fig. 41.



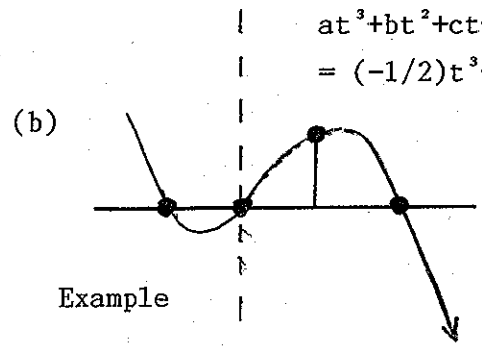


Fig. 42 Impulse response of cubic interpolator

If we jump to cubic interpolation (fitting a cubic, 3rd-order polynomial to four points), we can resolve one issue by extracting new samples from the region between the second and third point. Then we use a procedure very similar to that shown in Fig. 41 to find the response of the cubic interpolator to an impulse. Note that there are four points to fit, so there are four different cubics which can be generated as the impulse passes through (just as there were two lines in the 1st-order case). Fig. 42 shows this situation, the four cubics flanked by two of the non-productive cases, one on either side. Here while the fit is to four points, it is the region between the second and third that is used for the impulse response.

It is perhaps clear how the actual cubics are calculated - we simply fit the curve to the particular position of the impulse. This is actually nothing more than what is called "Lagrange interpolation" and simple product formulas for the polynomials are published. However, note that we can easily calculate the curves. For example, the third cubic, taken on the time interval -1 to +2, is of the form $at^3 + bt^2 + ct + d$, so we have four equations:

$$0 = -a + b - c + d \tag{78a}$$

$$0 = d \tag{78b}$$

$$1 = a + b + c + d \tag{78c}$$

$$0 = 8a + 4b + 2c + d \tag{78d}$$

which are easily solved to give the cubic:

$$-(1/2)t^3 + (1/2)t^2 + t \tag{79}$$

At this point, suppose we propose to interpolate by a factor of four. We would evaluate the polynomial in Fig. 42b at t=0, 1/4, 2/4, 3/4, and 4/4, giving impulse response value of 0, 0.2734, 0.5625, 0.8203, and 1. The upper polynomial evaluated on -1 to +2, with the impulse at -1, would give:

$$-(1/6)t^3 + (1/2)t^2 - (1/3)t \tag{80}$$

to give impulse response value at t=0, 1/4, 2/4, 3/4 , and 1 of 0 -0.0547, -0.0625, -0.03906, and 0. These are all the values that are unique that we need. Fig. 43a shows the filter while Fig. 43b shows the magnitude response.

Note that the impulse response for the cubic interpolator begins to resemble a sinc, but is finite length. This suggests that we once again are dealing with a low-pass filter. It is also true that if we continue to higher and higher order polynomials, the impulse response converges to a sinc.

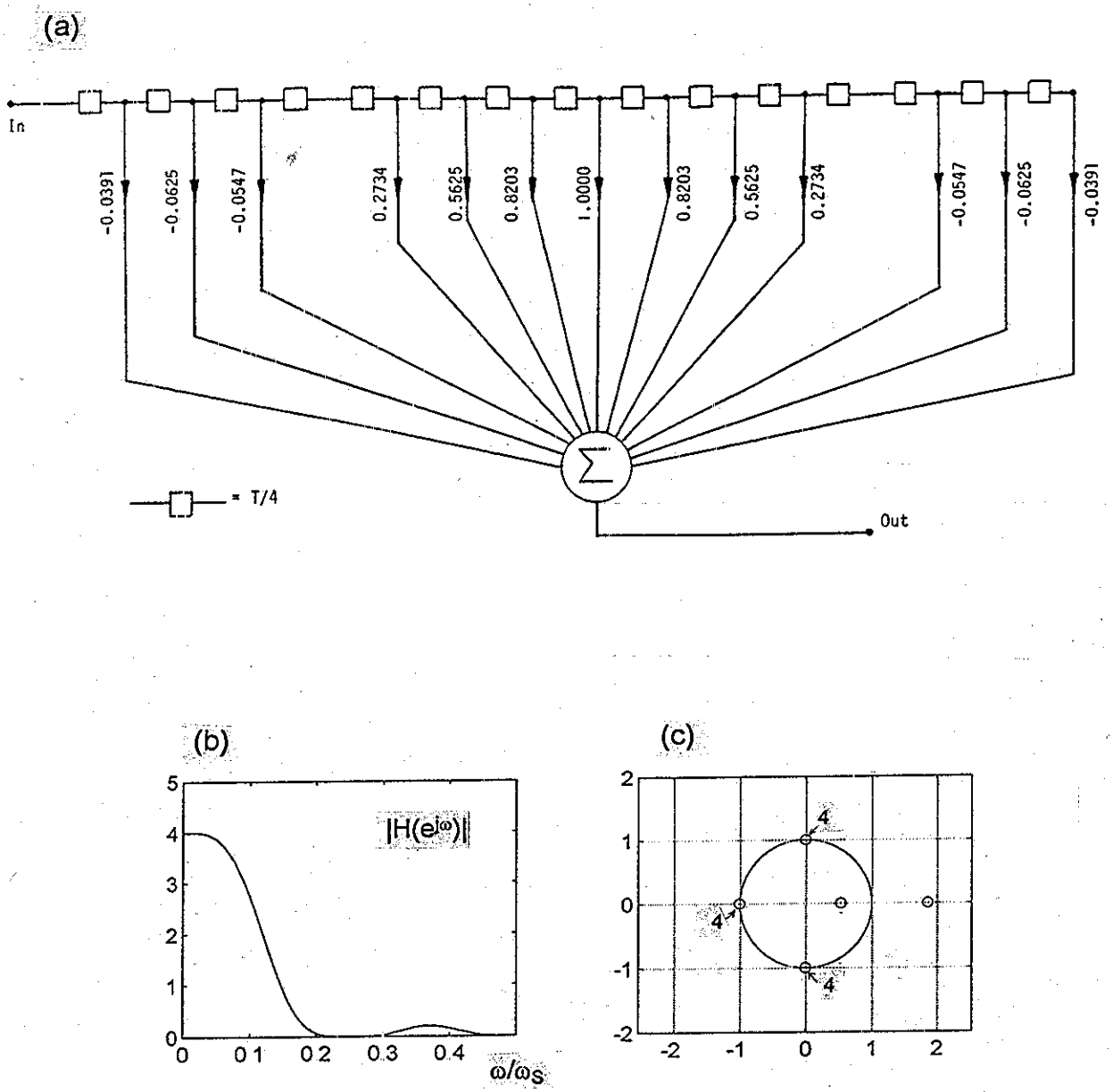


Fig. 43 FIR filter for cubic interpolation (a), with frequency response (b), and with zeros in z-plane (c). Note the arrangement and order of the zeros as shown, which suggests a frequency domain procedure for finding the same filter.

[Filters to be concluded next issue]

Measuring Q with Decrement (Continued from Page 1)

where d is a "decrement" to be described below. Note that we take the natural log of the decrement, and this is the origin of the "log decrement" terminology. This standard result is almost certainly the solution to the problem we had in mind when ASP was written.

Yet as with many such notions, tempered through the experiences of many intervening years, and viewed relative to the tools we have available today, we can solve a more interesting and more useful problem. Specifically, we want to show that the "log decrement" method (i.e., the laboratory observation) is most needed in the case where the simple formula of equation (1) is likely to have a significant error. That is, it works poorest in the cases where it is most needed! Wonderful. What we really need is to stop calling it the "log decrement" method, which alludes to the approximate solution, equation (1), and instead solve for a more exact equation, based on the decrement. Simple enough? Not really, because the equation is difficult to solve - except numerically. Happily, today, a program such as Matlab® will quickly grind out the answer iteratively and spit it out.

First of all, why would we use the decrement method? Let's review a bit. A bandpass filter might have a generic transfer function:

$$T(s) = \frac{A s \omega_0}{s^2 + (\omega_0/Q)s + \omega_0^2} \qquad (2)$$

which using inverse Laplace transform, has an impulse response:

$$g(t) = A\,\omega_0\, e^{-(\omega_0/2Q)t}\left[\cos(\omega_0\sqrt{1 - 1/4Q^2})\,t - (1/\sqrt{4Q^2-1})\sin(\omega_0\sqrt{1 - 1/4Q^2})\,t\right] \qquad (3)$$

which is an exponentially-decaying sinusoidal waveform. If we want to measure the Q of a particular bandpass, we could consider either frequency-domain methods or time-domain methods, and we would probably do well to also consider whether the Q is expected to be low or high. Perhaps a value of Q=10 might be a dividing line, but the general advisory would be to always consider a variety of methods (we are not locked out by making an initial choice). These choices are discussed in ASP and in AN-279, and here, we will concentrate on the decrement method, in line with solving the implied homework problem.

Fig. 1 shows a set of four impulse responses for various bandpass filters. Here we note right away that the decrement we are talking about is the ratio of a current peak to the previous peak (shown in Fig. 1b). This is the observation we need to make. Since it is a ratio which we expect to be everywhere the same, we could measure it between any two consecutive peaks. This allows us to simplify the mathematics considerably. We can always choose a starting time in the waveform and adjust the height of the impulse exciting the filter such that a height of 1 occurs at the reference point chosen. Taking this to be time zero, we can simplify equation (3) as:

$$g(t) = e^{-(\omega_o/2Q)t} \cos(\omega_0 \sqrt{1-1/4Q^2})\, t \qquad (4)$$

This is still sufficiently general to be completely valid for an observed decrement. It is equation (4) that we have used to plot Fig. 1.

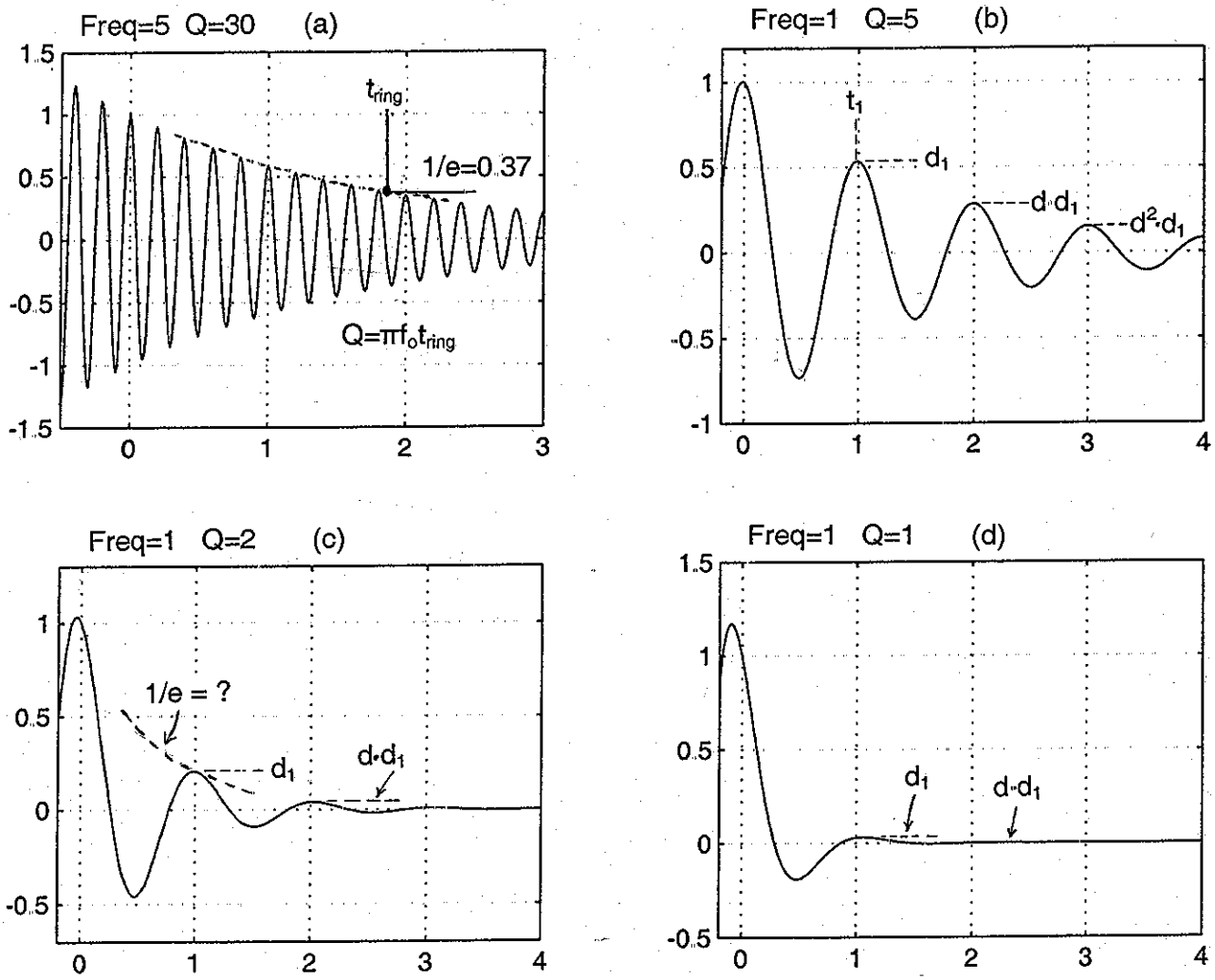


Fig. 1 Ringing Bandpass. In (a) we have a high-Q typical example where we can "see" the exponential decay envelope quite well. In (b) we have a lower Q where it is difficult to see the envelope. The notion of a decrement is illustrated in this example, and note that the first maximum is slightly to the left of (not at) time zero. Cases (c) and (d) show even lower Q's where a decrement method is almost essential.

What we see is that overall, the plots look quite different. Fig. 1a shows a more-or-less textbook example of a decaying sinusoidal waveform. In this case, we consider the Q to be high. It is easy for us to see the "envelope" of the decay. We could easily measure Q based on this ring time as described in ASP. On the other hand, if we were measuring the Q as the center frequency divided by the 3db bandwidth, the filter is so sharp that the 3db frequencies might be very difficult to separate out. We would use the method of measuring the "skirts." Fig. 2b shows a Q of 5, starting to get low. In this case, we could probably easily measure the Q based on the 3db frequencies. But in the time domain, it would be harder to "see" the decay envelope. The situation of "seeing" the decay envelope is clearly worse in the Q=2 and Q=1 case. They just die too fast (it's hard, perhaps impossible to see the 1/e point of the envelope). But, it should be fairly easy to measure the decrement. The situation is summarized in Table 1.

TABLE 1

	Frequency Domain	Time Domain
Low-Q	$Q = \omega_0/(\omega_{upper} - \omega_{lower})$	decrement method
High-Q	use "skirts"	ringing - decay envelope

These methods are all highly reliable, unless we try to use a log-decrement method at low Q, which is exactly where we need it most. To see why, we simply solve the original problem.

We rely on the decay envelope being governed by:

$$e^{-\omega_0 t/2Q} \tag{5}$$

and we take the time between peaks as being $2\pi/\omega_0$. Plugging this time into equation (5), we get a decrement between supposed peaks of:

$$d = e^{-\pi/Q} \tag{6a}$$

or

$$\ln(d) = -\pi/Q \tag{6b}$$

and

$$Q = -\pi/\ln(d) \tag{6c}$$

and we have it. The problem is that the frequency can not be approximated well by ω_0 except for high Q. So the method works well for high Q - just where we don't need it usually!

One approach to getting a more accurate formula might be to use the damped frequency:

$$\omega_0 \sqrt{1 - 1/4Q^2} \tag{7}$$

instead of ω_0. So the decay time of interest is $2\pi / \omega_0 \sqrt{1 - 1/4Q^2}$. this looks problematic, since we need to solve for Q. Here we tried an often used approximation for the square root of a number close to 1: $\sqrt{1-\Delta} \approx 1-\Delta/2$. Following the same procedure as for the log decrement method, we get:

$$d = e^{-(\pi/Q)/(\sqrt{1 - 1/4Q^2})} \approx e^{-(\pi/Q)/(1 - 1/8Q^2)} \tag{8}$$

This leads to a quadratic in Q:

$$Q^2 + (\pi/ \ln(d)) 8Q - 1 = 0 \tag{9}$$

Which has the solution (using the quadratic formula):

$$Q = - \pi/(2\ln(d)) - \pi/(2\ln(d)) \sqrt{1 + \ln(d)^2/2\pi^2} \tag{10}$$

Here we have (expediently) chosen the (-) sign from the quadratic formula so as to match solutions with the log decrement formula as d approaches 1 (and for positive Q!). Note that equation (10) can be thought of as splitting equation (6c) into two equal parts and then applying a correction factor to the second part. We shall see how well this works a bit later.

It is possible, and not particularly difficult (freshman calculus) to find the solution that is completely right. To do this, we need only find the relative maximums of equation (4). Differentiating equation (4) with respect to time, setting the derivative to zero, and solving for t we obtain:

$$t_1 = (1 / \omega_0\sqrt{1 - 1/4Q^2}) \{ \tan^{-1}[-1 / (2Q \omega_0 \sqrt{1 - 1/4Q^2})] +2\pi \} \tag{11}$$

Note that we have added 2π to the arctan in order to find the first peak (t_1 in Fig. 1b) beyond the maximum that is near zero. This results in a simple calculation of t_1, and plugging back into equation (4) we get d_1. Thus we arrive at a straightforward pair of equations for deriving d_1 given the Q and the frequency ω_0 (both obtainable easily from the transfer function or just from the poles).

But hold on - something looks strange. What we have done seems to be correct, but perhaps it is not very useful; and isn't the solution in equation (11) trying to tell us something? In fact, we can recast equation (11) in a more general form:

$$t_m = (1 / \omega_0\sqrt{1 - 1/4Q^2}) \{ \tan^{-1}[-1 / (2Q \omega_0 \sqrt{1 - 1/4Q^2})] +2m\pi \} \tag{12}$$

This we can do because the arctangent is periodic with period 2π. [Indeed, we used this fact to get t_1 in equation (11).] At this point, two facts become obvious: First, t=0 is

not a maximum of g(t) of equation (4). The actual maximum occurs slightly before t=0 [m=0 in equation (12)]. The reason for this is that while t=0 is clearly a maximum of the cosine, g(t) is subject everywhere to the decaying exponential term. So at flat points of the cosine, g(t) is sloping downward. Thus g(t) must be sloping upward, initially, for negative times (see Fig. 1b for example). The second obvious observation is that all maxima are spaced at time interval:

$$t_p = 2\pi / \omega_0\sqrt{1 - 1/4Q^2} \qquad (13)$$

In fact, this periodicity with t_p is pervasive in the simplifying developments that follow.

Note in particular that t_1 is not t_p, and d_1 is the decrement from g(0)=1 to $g(t_1)$, not the decrement d that we really want (from one maximum to the next). The significance of this is that, as a matter of convenience, it is d, measured from one maximum to the next, that we expect to be able to measure. But we do have a good number of ways of calculating the actual d. For example, a fundamental approach would be:

(1) Calculate t_0 and t_1 from equation (12). (Or any two consecutive t_m will do.)
(2) Calculate $g(t_0)$ and $g(t_1)$ using equation (4).
(3) $d = g(t_1) / g(t_0)$.

This is an important method because it is based on our exact calculations using calculus and because it corresponds exactly to the laboratory measurement we expect to make.

But it turns out that things continue to simplify. Not only are the maximum points spaced at t_p and decremented by d, but any two points separated by t_p are decremented by d (zero crossings being an exception of course). Thus, for example, since g(0)=1 in equation (4), it follows that $d=g(t_p)$. However, even further simplified, note that the cosine is the same value at intervals of t_p, so all that remains is:

$$d = e^{-(\pi/Q)/(\sqrt{1 - 1/4Q^2})} \qquad (14)$$

which is familiar. It is the "guess" we made in equation (8), but here it returns as much more than a guess. The periodicity of the cosine term assures us that the exponential term determines the decrement. Still, equation (14) is difficult to invert. Or is it?

Because we have an equation in the forward direction (d from Q), we can always guess a value of Q, calculate d, and see if this was the d we have in mind. More efficiently, we use a program that searches through a range of Q until we encounter the d specified. An example program in Matlab® is shown below. What this program does is start with Q=0.501 and searches up until the actual decrement matches the desired decrement (well enough). In most cases, for low Q and a fast PC, the calculation time is only a second or so. Based on code similar to that of the program dectoq.m, we can easily write programs to calculate tables and to plot Q as a function of d over various ranges. (Note that in plotting an increasing Q against an increasing d, it is much more efficient to begin the search of Q with the Q for the previous value of d. It is then usually only a few steps upward to the next Q, as compared with an increasingly long search if Q is reset to 0.5 at the start of each search.) Table 2 shows some typical results, and Fig. 2, Fig. 3, and Fig. 4 show some typical plots.

PROGRAM dectoq.m

```
function [Q,Qapprox,Qlogdec]=dectoq(d)
% Calculate Q from the measured decrement (d).
% Also compare to an approximate method.
% Also compare to the classical log decrement.

w0=2*pi;
Q=0.5;    % critical damping
d1=0;

while dd<d
  Q=Q+0.001;
  fr=sqrt(1-1/(4*Q^2));
  dd=exp(-pi/(Q*fr));
end                          %    Q is now located

ld=log(d);
Qlogdec=-pi/ld;

tps=2*(pi^2);
lds=ld^2;
Qapprox= -(pi/(2*ld))*(1 + sqrt(1 + lds/tps));
```

TABLE 2

decrement d	Q	Qapprox	Qlogdec
0.9	29.82	29.82	29.82
0.5	4.56	4.56	4.53
0.2	2.02	2.01	1.95
0.1	1.45	1.45	1.36
0.05	1.16	1.16	1.05
0.02	0.95	0.94	0.80
0.01	0.85	0.83	0.68
0.005	0.78	0.76	0.59
0.002	0.71	0.69	0.51
0.001	0.68	0.65	0.45
0.0005	0.65	0.62	0.41
0.0002	0.62	0.58	0.37
0.0001	0.61	0.56	0.34
0.00001	0.57	0.52	0.27
0.000001	0.55	0.49	0.23
0.0000001	0.54	0.46	0.19
0.00000001	0.53	0.45	0.17
0.000000001	0.52	0.44	0.15

Errors Exceeding 5%

Impossible Q's Less than 0.5

The overwhelming result seen in the table and in the figures is that the log decrement method falls apart badly at low Q. Note that a decrement to 10% already has an error exceeding 5% for the Q. A couple of useful data points to keep in mind are that a decrement to about 0.5 is a Q of about 5 while a decrement to 0.2 is a Q of about 2. (Engineers are often so obsessed with calculations that we forget to look at the results!) Perhaps we can say that if the decrement is less than 0.5, don't even think about log decrement. The approximate method is much better. Note that its error reaches 5% only for a decrement to about 0.0002, which is pretty hard to measure in the first place (by no means impossible - we can click to different scales on an oscilloscope!). But, the approximate method is far superior to the log decrement method, and works very well for many case of interest. The problem is - you are very unlikely to remember the formula, or exactly how to derive it!

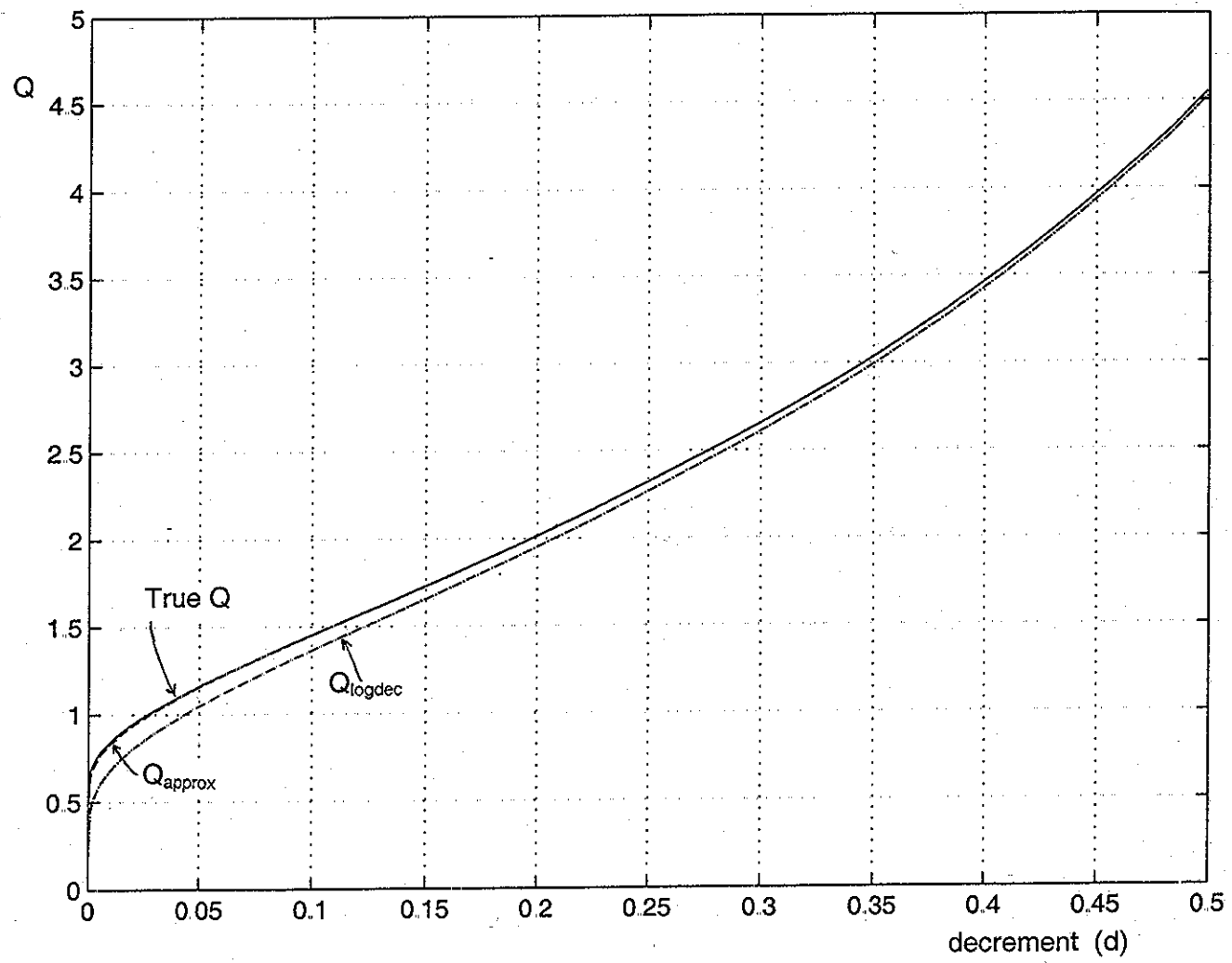


Fig. 2 True Q (solid), Q_{approx} (dashed) and Q_{logdec} (dash-dot) for a range of d from 0 to 0.5. Note the convergence of the methods for higher Q's.

So there is great virtue in doing it exactly right. You will need to remember that you have to calculate the exponential over one time period of the reduced frequency, and you will have to iterate the calculation. Or, you can just use the graphs here - we have purposely put on the grids with this in mind.

Some final comments relate to actual measurement procedures and to the fact that in many cases, the Q is so low that it is hard to see the ringing. One thing to keep in mind is that we are looking for relative values, and that we can measure separate peaks on different scales of the scope. One must be careful to assure that the amplitudes are measured relative to zero. Because low Q cases decay rapidly to zero, it should be

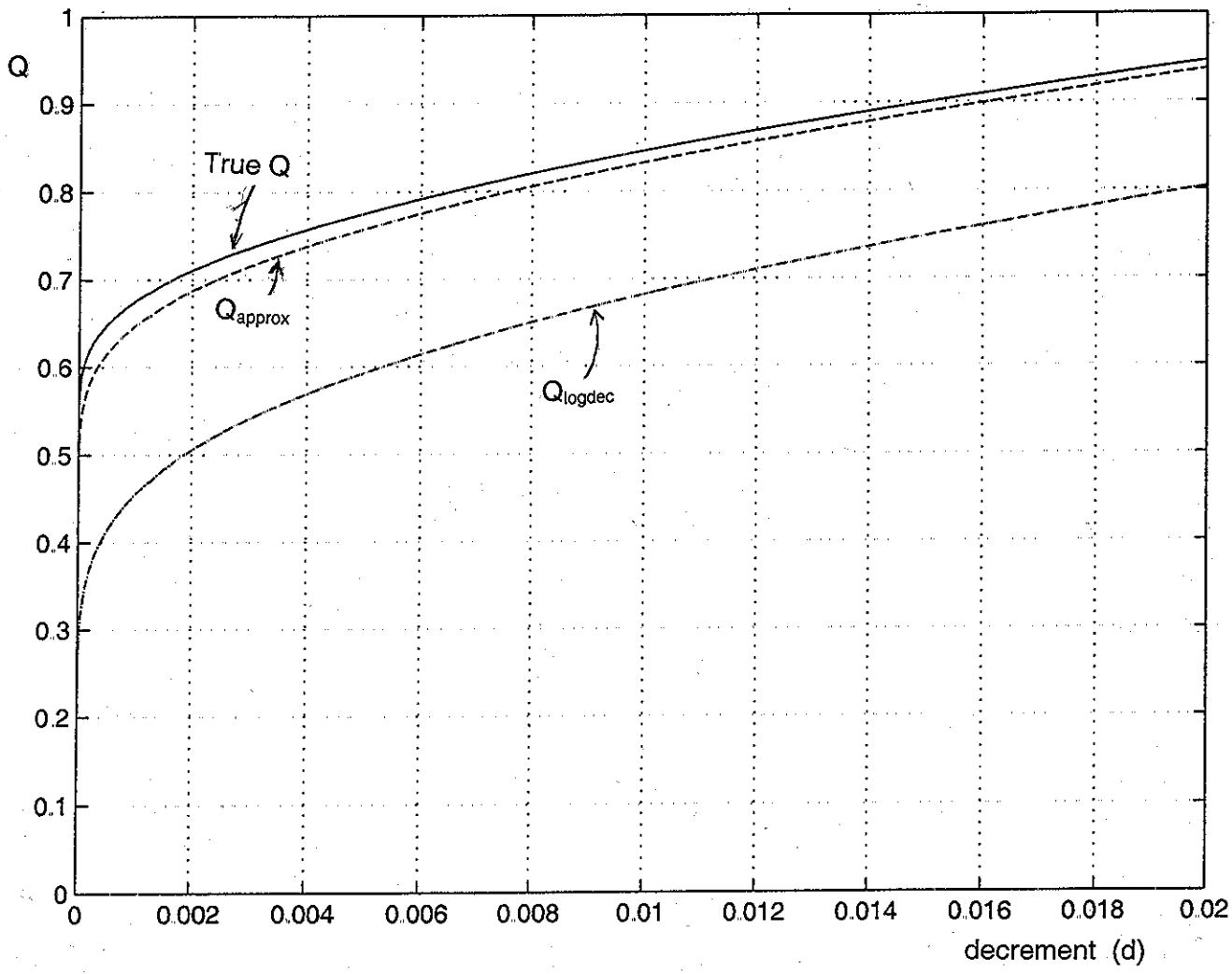


Fig. 3 True Q (solid), Q_{approx} (dashed) and Q_{logdec} (dash-dot) for a range of d from 0 to 0.02. We note the approximate method is much better than the log decrement. In fact, the log decrement is heading for Q=0 at d=0 while the true Q goes to 0.5 at d=0 (critical damping). The true Q calculation is the one recommended.

easy to measure an amplitude relative to this asymptotic decay to zero. This is particularly important in the event that there are DC offsets in the filter's output. In cases where the second maximum of a series is still disconcertingly small, why not try the minimum following the maximum (for example, this might be useful for the case of Fig. 1d). The amplitude of the minimum is of course determined by a decrement obtained with equation (14), but evaluated at $t_p/2$, which amounts to a factor of 1/2 multiplying the exponent. Again, caution is urged in measuring relative to the asymptotic zero.

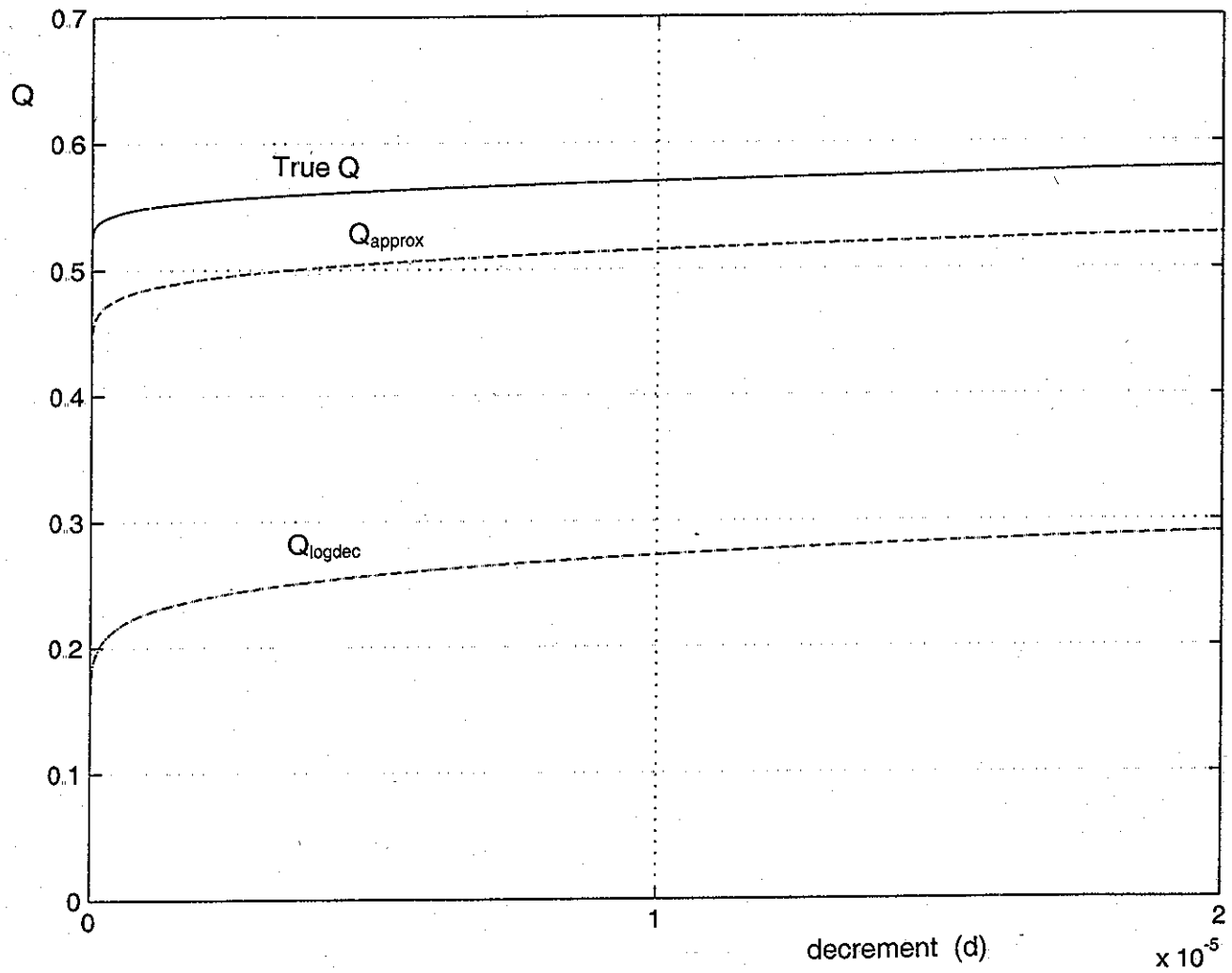


Fig. 4 True Q (solid), Q_{approx} (dashed) and Q_{logdec} (dash-dot) for a range of d from 0 to 0.00002. We note the approximate method is much better than the log decrement, but shows a significant error in this range. (Keep in mind however that we might never see, or be concerned with, such small ringing.) Note again that the log decrement is heading for Q=0 at d=0 while the true Q goes to 0.5 at d=0 (critical damping). The true Q calculation is the one that is highly recommended here.

Who cares? Always a good question. As a general rule, we like to have more than one way to understand things, and we like more than one way to measure things (to check our measurements). So it is nice to be able to use as many of the methods in Table 1 for measuring Q as we are able. In general, if the bandpass filter of interest is on the bench, we envision having a function generator attached to the input and an oscilloscope to the output. Then you can imagine yourself clicking knobs here and there to jump from one method to the other. Likely we would find one of the methods the most agreeable. In fact, jumping from frequency domain to time domain measurements might well be just a matter of switching from sine to pulse, dropping the input frequency a couple or decades, and slowing the sweep of the scope a couple of decades. We do this almost automatically.

At times however, we must use time-domain methods. We may only have access to the output of the filter. It may not be an actual filter, but some sort of communications link that is somehow ringing, and we need to characterize it as a network with a certain Q. In such cases, we may see the ringing in response to what we believe are rectangular transitions at the input (the channel is transmitting pulses).

In fact, in looking through my books to find out if anyone actually worked out the correct relationship between decrement and Q, I was unable to even find "log decrement" in any electronic books. (I didn't look very hard actually.) But I remembered where I saw log decrement first - in a freshman physics lab. Indeed, you will probably find a lot more information on time domain measurements of damping in books on mechanical oscillations, rather than in electronics books. In the physics lab, we have easy ways of achieving displacements. We move a pendulum and release it or push it from a standing position. But, the equivalent of a sinusoidal oscillator is not so convenient. Imagine a physics lab bench that oscillates sinusoidally over a wide frequency range - it can be done, but not easily. We electrical engineers have it easy - just a couple of clicks of the function generator!

In summary, "log decrement" method should be avoided because it is inaccurate when needed most (low Q). Yet the true decrement method works very well. For mechanical experiments, time domain measurement of Q may be essential. For electrical engineers, we need to have a good feel for the Q of a ringing device based on the decrement.

****

Electronotes, Vol. 20, No. 198, June 2001

Published by B. Hutchins, 1016 Hanshaw Rd., Ithaca, NY 14850 (607)-257-8010