An Approach to the Approximation Problem for Nonrecursive Digital Filters

LAWRENCE R. RABINER, Member, IEEE
Bell Telephone Laboratories, Inc.
Murray Hill, N. J. 07974

BERNARD GOLD, Senior Member, IEEE
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, Mass.

C. A. McGONEGAL
Bell Telephone Laboratories, Inc.
Murray Hill, N. J. 07974

Abstract

A direct design procedure for nonrecursive digital filters, based primarily on the frequency-response characteristic of the desired filters, is presented. An optimization technique is used to minimize the maximum deviation of the synthesized filter from the ideal filter over some frequence range. Using this *frequency-sampling* technique, a wide variety of low-pass and bandpass filters have been designed, as well as several wide-band differentiators. Some experimental results on truncation of the filter coefficients are also presented. A brief discussion of the technique of nonuniform sampling is also included.

Manuscript received January 19, 1970.

Introduction

Nonrecursive digital filters have finite-duration impulse response and consequently contain no poles (only zeros) in the finite z-plane. The approximation problem is that of finding suitable approximations to various idealized filter transfer functions. A designer may be interested in approximating either the magnitude, or the phase, or both magnitude and phase of this ideal filter. A few examples of typical ideal filters are shown in Fig. 1(A) through (F). Fig. 1(A) shows an ideal low-pass filter while Fig. 1(B) through (D) show ideal high-pass, bandpass, and band-elimination filters. Fig. 1(E) shows the response of an ideal differentiator while Fig. 1(F) shows the phase response of an ideal Hilbert transformer which allows the two outputs to be in phase quadrature.

The approximation problem for recursive digital filters (having infinite-duration impulse response, and poles as well as zeros) has been treated extensively [1], [2]. Mathematically, in the recursive case the realizable approximation can be expressed as the ratio of two trigonometric polynomials, leading to filter designs based on classical analog filter theory. This leads, for example, to fairly sophisticated design techniques for Butterworth, Chebyshev, and elliptic filters to yield good magnitude response approximations. For nonrecursive digital filters the realizable approximations are trigonometric polynomials. Thus, the class of approximations is more constrained. The most widely used approach towards approximating the frequency domain filter characteristic is based on approximating the infinite-duration impulse response of the ideal filters by the finite-duration impulse response of the nonrecursive realization. The most significant result in this connection is the Gibbs phenomenon, illustrated in Fig. 2, which shows the resultant frequency response obtained when the "ideal" (infinite) impulse response corresponding to Fig. 1(A) is symmetrically truncated. As is well known, the amount of error or "overshoot" in the vicinity of the discontinuity does not diminish, even as the response is increased in duration. Recognition of this fact has prompted workers in the field to seek ways to decrease the ripple by decreasing the severity of the discontinuity. This can be accomplished by introducing a time-limited window function $w(n)$ having a z-transform $W(z)$. From the complex convolution theorem the z-transform of the product $h(n)\,w(n)$ is given by

$$F(z) = \frac{1}{2\pi j}\oint H(z/v)W(v)v^{-1}dv \qquad (1)$$

where $h(n)$ is the ideal impulse response and $H(z)$ is its z-transform. Thus, multiplying $h(n)$ by a window corresponds to smoothing the spectrum. Careful choice of a window can result in a frequency-response function with appreciably less in-band and out-of-band ripple, as can be seen by comparing Figs. 2 and 3.

Kaiser [1] has introduced a set of windows (which we

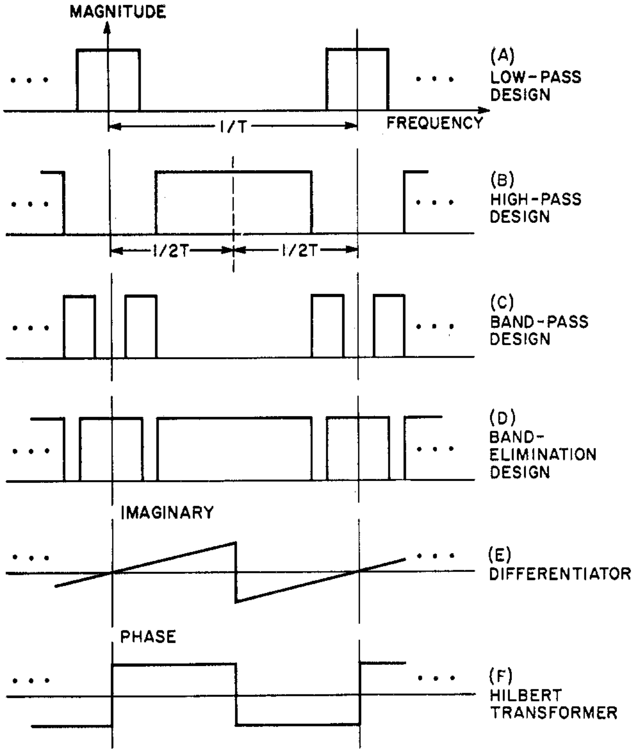


Fig. 1. Examples of typical ideal filters.

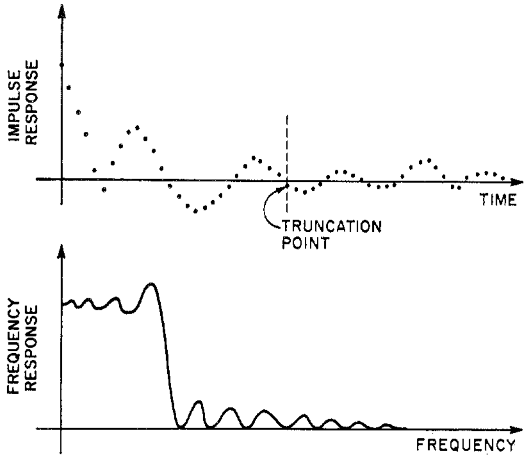
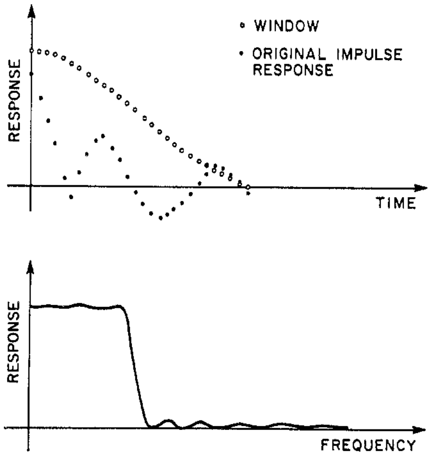


Fig. 2. An example of the effects of truncating the impulse response on the frequency response.

Fig. 3. Reduction of overshoot in the frequency response by windowing the truncated impulse response.



shall call Kaiser windows) which are very close to optimum. By adjusting a parameter of the window, the sidelobes can be diminished at the cost of increased transition bandwidth. Helms [3] recently proposed the Dolph–Chebyshev window because it has good spectral properties and because its parameters can be readily determined directly.

Window functions have also found great use in spectral analysis of random functions, but this subject will not be specifically discussed in this paper.

Design of nonrecursive filters from frequency-response specifications has been considered by Martin [4], who specified initial values of the frequency response at selected frequencies, leaving unspecified values of the frequency response in preselected transition bands. He then used a minimization procedure to solve for final values of the frequency response at equally spaced frequencies. The criterion used for the minimization was that the maximum deviation of the continuous frequency reponse from the ideal frequency response be minimized for both in-band and out-of-band frequencies. Martin obtained useful results for small values of N (the number of impulse-response samples) for the case of low-pass filters and for wide-band differentiators.

A recent paper by Gold and Jordan [5] introduced a somewhat different approach to the approximation problem for nonrecursive digital filters. In this approach the frequency response is specified exactly at N equispaced frequencies. If it is assumed that the number of frequency samples is equal to the number of samples in the impulse response, then the (continuous) frequency response is exactly determined. A simple example is shown in Fig. 4, where an ideal rectangular low-pass filter is sampled at equally spaced frequencies, resulting in a continuous frequency response with overshoot. (The transition band in Fig. 4 is the frequency range between the last in-band sample and the first out-of-band sample.) The impulse response corresponding to this frequency sampled filter is now no longer truncated, but rather *aliased* or *folded.* This fact should be well noted, as it serves to delineate sharply between this method (the sampling method) and the window method. It is not clear to us whether truncation or aliasing of an infinite impulse response is an intrinsically better procedure; however, this theoretical distinction makes it awkward to formulate the sampling method in terms of the window method. Our reasons for the rather extensive study of the sampling method to be presented in this paper are the following.

1) The designer trying to design filters to approximate a given ideal shape in the frequency domain need never

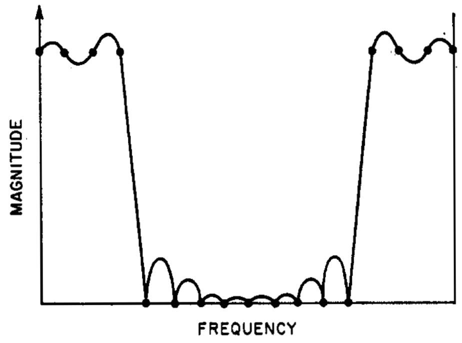


Fig. 4. The continuous frequency response of a filter derived from its frequency samples.

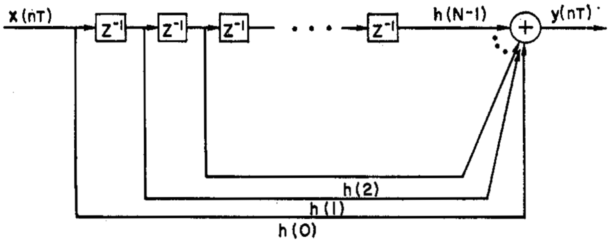
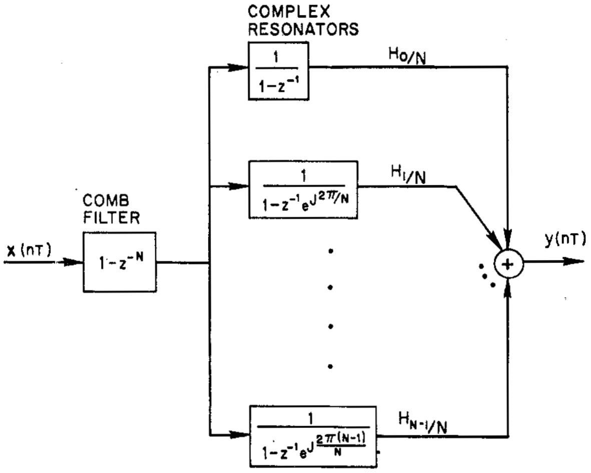


Fig. 5. Direct convolution realization of nonrecursive digital filter.

Fig. 6. Frequency sampling realization of nonrecursive digital filter.



concern himself with an impulse response. This is intuitively appealing for filters with sufficiently long impulse responses (i.e., greater than about 30 samples), so that high-speed convolution using the fast Fourier transform is used for synthesis since the design results can be applied directly to yield the synthesis.

2) The sampling procedure is capable of being exploited to yield an "optimum" filter. As discussed above, the window technique results in a tradeoff between overshoot and transition bandwidth. By contrast, in the sampling technique, once the designer has chosen a transition bandwidth, he can, in a practical sense, calculate the best filter that will have such a transition bandwidth. As will be seen, this leads to quite efficient designs.

In the Gold and Jordan paper [5], results were obtained only for a few low-pass filters. Also, the computer optimization technique was semiautomatic, requiring an online interactive display oscilloscope. The computations needed for optimization were fairly lengthy and somewhat inaccurate.

In the present paper, the design method and optimization are treated more generally, and described in detail. The procedure has been fully automated and made computationally efficient. As a result, it has been possible to generate extensive design data, applicable in many cases to "cookbook" design. An analysis of low-pass and bandpass filters, as well as of wide-band differentiators, is presented. Numerical comparisons are made between the window and sampling methods for low-pass filters and differentiators. The effects of finite register length are discussed and a few results presented. Finally, the theory for a nonuniform sampling procedure is presented and a few numerical results are given.

Synthesis Techniques for Nonrecursive Filters

Before presenting the formalism of our design technique, it is worth discussing the filter synthesis question heuristically. We know of three useful ways of synthesizing a nonrecursive digital filter.

1) Direct Convolution: The impulse response of the filter is explicitly found and the filter is realized via the computation

$$y(n) = \sum_{m=0}^{N-1} h(m)x(n-m) \qquad (2)$$

where $h(m)$ represents the filter impulse response, $x(n)$ is the input sequence, and $y(n)$ is the output sequence. The realization of (2) is shown in Fig. 5. The limits in (2) imply that $h(m)$ is of duration N, so that $h(m)=0$ for $m \geq N$.

2) Fast Convolution: Here only values of the frequency response of the filter need to be explicitly found. First the discrete Fourier transform of $x(n)$ (suitably augmented with zero-valued time samples) is computed, then multiplied by samples of the filter frequency response, and then the product is inverse transformed to yield the output.

3) Frequency Sampling: Here the sampling theorem is specifically realized as a digital network [8]. As seen in Fig. 6, this network consists of a comb filter in cascade with a set of parallel complex exponential resonators, the outputs of which are suitably weighted and added to form the output.

Formulation of the Frequency Sampling Method of Filter Design

The sampling technique described in this paper can be applied to a finite set of samples of the z-transform of a filter evaluated anywhere in the z-plane. For the most part we will restrict ourselves to the case where the sample points are equally spaced around the unit circle, and the sample values represent values of the continuous frequency response of the filter. Later in this paper we will consider the more general case and, in particular, will examine the case of nonuniform frequency spacing of the samples.

For the case of uniformly spaced frequency samples the design procedure consists of a sequence of computations which can be summarized as follows.

1) Choose a set of frequencies at which the sampled frequency response is specified. The values of the sampled frequency response at some of these frequencies are generally left as parameters of the design problem. For the uniform frequency sampling considered here, the choice of a set of frequencies is merely the choice of a value for N, the number of impulse-response samples, and an initial frequency. Once N has been chosen, the frequency spacing between samples is $\Delta f = 1/NT$, where T is the sampling period. The choice of values of the frequency response at the sample frequencies is dictated by the ideal filter being approximated.

2) Obtain values of the continuous frequency response of the filter as a function of the filter parameters. The continuous frequency response can be determined as a function of the frequency samples, either as an explicit equation (i.e., the sampling theorem), or implicitly in terms of the fact Fourier transform algorithm (FFT) [9] or the chirp z-transform algorithm (CZT) [10].

3) Once the interpolated frequency response is obtained, a program automatically readjusts the filter parameters (the unspecified frequency samples) while searching for a minimum of some filter characteristic.

4) When the minimum has been obtained and verified, the final values of the free parameters are then used in the realization along with the fixed frequency samples.

There are a wide variety of filter problems where the designer requires a sharp cut-off amplitude characteristic and, preferably, a linear phase characteristic. For this reason, one of our aims was to obtain an interpolated frequency response which was pure real except for a linear phase shift. To achieve this goal requires careful consideration of the parameter N and the specific frequency positions of the samples. As a result, we found it useful to formulate the sampling theorem for four cases.

Case A: N even, frequency samples at

$$f_k = \frac{k}{NT}, \qquad k = 0, 1, 2, \cdots, N-1.$$

Case B: N even, frequency samples at

$$f_k = \frac{k+\frac{1}{2}}{NT}, \qquad k = 0, 1, 2, \cdots, N-1.$$

Case C: N odd, frequency samples at

$$f_k = \frac{k}{NT}, \qquad k = 0, 1, 2, \cdots, N-1.$$

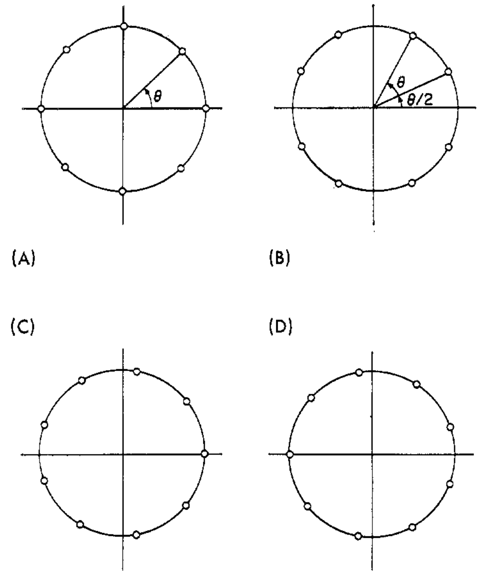


Fig. 7. The four possible orientations for uniformly spaced frequency samples. (A) and (C) show type-1 data whereas (B) and (D) show type-2 data.

Case D: N odd, frequency samples at

$$f_k = \frac{k+\frac{1}{2}}{NT}, \qquad k = 0, 1, 2, \cdots, N-1.$$

Fig. 7 illustrates these four cases, the circles representing the sampling points around the unit circle in the z-plane. For Cases A and B, N is 8, whereas for Cases C and D, N is 9. The data of Cases A and C will henceforth be referred to as type-1 data; whereas the data for Cases B and D will be referred to as type-2 data. The difference between the two types reflects the initial frequency at which the frequency response is sampled.

Derivation of Sampling Theorem for Case A

Given a finite-duration filter impulse response $h(0)$, $h(1), \cdots, h(N-1)$, the z-transform of this filter is

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}. \qquad (3)$$

Since $h(n)$ is of finite duration, it can be represented in terms of its discrete Fourier transform (DFT) H_k, $k=0, 1, \cdots, N-1$, as follows:

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H_k e^{j2\pi kn/N} \qquad (4)$$

where

$$H_k = H(z) \big|_{z=e^{j2\pi k/N}}. \qquad (5)$$

Substituting (4) into (3) and interchanging sums, we observe that the sum over the n index can be evaluated in closed form so that

$$H(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{H_k}{1 - z^{-1}e^{j2\pi k/N}}. \quad (6)$$

Evaluating (6) on the unit circle where $z = e^{j\omega T}$ leads to the interpolated frequency response

$$H(e^{j\omega T}) = \frac{\exp\left[-\frac{j\omega NT}{2}\left(1 - \frac{1}{N}\right)\right]}{N} \cdot \sum_{'=0}^{N-1} \frac{H_k e^{-j\pi k/N} \sin\left(\frac{\omega NT}{2}\right)}{\sin\left(\frac{\omega T}{2} - \frac{\pi k}{N}\right)}. \quad (7)$$

Let us now examine in detail the implications of (4) through (7). If the initial set of frequency samples H_k is chosen so that H_k is a real, symmetric sequence (i.e., $H_k = H_{N-k}$), then the interpolated frequency response cannot be pure real. A small oscillatory imaginary component of amplitude

$$A = \frac{1}{N} \sum_{k=0}^{N-1} H_k(-1)^k \quad (8)$$

will be part of the interpolated frequency response. In many cases the amplitude A is very small and can be tolerated. In other cases one is forced to look to other techniques for designing pure real nonrecursive filters.

One simple way of alleviating the problem of having an imaginary component (other than a linear phase shift) in the interpolated frequency response is suggested by (7). By making the substitution

$$H_k = G_k e^{j\pi k/N}, \quad (9)$$

the summation in (7) becomes pure real, and $H(e^{j\omega T})$ is real except for the linear phase-shift term outside the sum. A physical interpretation of the significance of the substitution of (9) can be obtained by examining the impulse response corresponding to this set of frequency samples. If the set G_k is chosen such that $G_{N/2} = 0$ and $G_k = -G_{N-k}$, then the impulse response $h(n)$ can be written as

$$h(n) = \frac{G_0}{N} + \sum_{k=1}^{(N/2)-1} \frac{2G_k}{N} \cos\left(\frac{\pi}{N}k + \frac{2\pi}{N}nk\right). \quad (10)$$

It is easily shown that $h(n)$ is a real sequence with the symmetry property

$$h(n) = h(N - 1 - n) \quad n = 0, 1, 2, \cdots, \frac{N}{2} - 1. \quad (11)$$

It should be noted that this is not the usual symmetry property of an N-point sequence. A typical impulse response is shown in Fig. 8 for the case $N = 16$. As seen in Fig. 8, the origin of symmetry of the impulse response lies midway between samples representing a delay of a noninteger number of samples. This half-sample delay can also be verified from (7) where the linear phase-shift term has a component equivalent to half a sample delay.

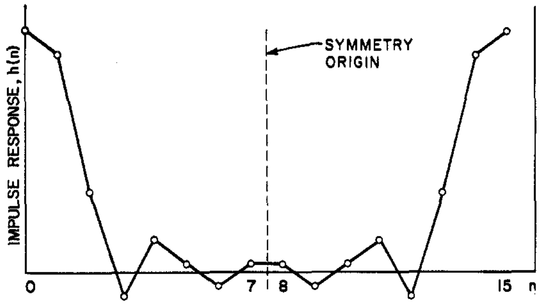


Fig. 8. A typical impulse response for Case A, showing the half-sample delay obtained for this case.

Sampling Theorem for Case C

The derivation of the sampling theorem for N odd is the same as for N even, leading to (7). However, for N odd, choosing the set of frequency samples H_k to be real and symmetric yields a real and symmetric impulse response whose origin of symmetry falls on a sampling point. Thus a pure real interpolated frequency response can be attained for this case. It is easy to show that the continuous frequency response is real by first deriving the impulse response and then computing the frequency response. The impulse response can be written as

$$h(n) = \frac{H_0}{N} + \sum_{k=1}^{(N-1)/2} \frac{2H_k}{N} \cos\left(\frac{2\pi}{N}nk\right). \quad (12)$$

The impulse response is a real and symmetric function with a unique peak at $n = 0$. By rotating the impulse response $(N-1)/2$ samples, i.e., replacing $h(n)$ by $h[(n-(N-1)/2) \bmod N]$, so that the peak occurs at $n = (N-1)/2$, and translating the entire impulse response by $(N-1)/2$ samples, the frequency response can be written as

$$H(e^{j\omega T}) = \sum_{n=-(N-1)/2}^{(N-1)/2} 2h(n) \cos(n\omega T) \quad (13)$$

which is purely real.

Summary of Computation Procedure for Cases A and C

The continuous frequency response can be computed directly from (7) for either N odd or even. However, our method of computation differs in that the FFT algorithm is used instead. We now present the detailed steps used to obtain the interpolated frequency response from the set of N frequency samples.

1) Given N, the designer must determine how fine an interpolation should be used. For the designs we investigated, where N varied from 15 to 256, we found that 16 N sample values of $H(e^{j\omega T})$ lead to reliable computations and results; i.e., 16 to 1 interpolation was used.

2) Given the set of N values of H_k, the FFT is used to compute $h(n)$, the inverse DFT of H_k. For both N odd and N even the set H_k which was used was real and symmetric; therefore $h(n)$ is real in all cases and symmetric for N odd.

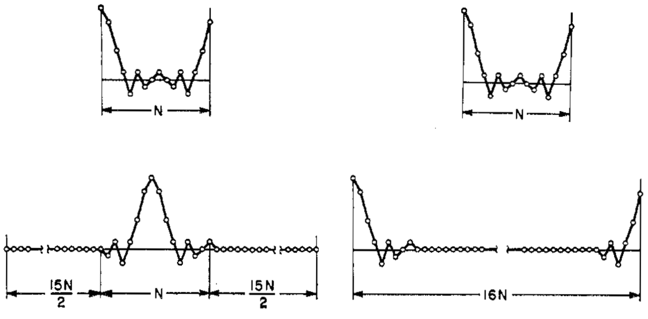


Fig. 9. Computational methods for obtaining the interpolated frequency response.

3) In order to obtain values of the interpolated frequency response one of two procedures is followed. Either a) $h(n)$ is rotated by $N/2$ samples (N even) or $[(N-1)/2]$ samples (N odd) to remove the sharp edges of the impulse response, and then 15 N zero-valued samples are symmetrically placed around the impulse response [as illustrated in Fig. 9(A)]; or b) $h(n)$ is split around the $(N/2)$nd sample value, and 15 N zero-valued samples are placed between the two pieces of the impulse response [as illustrated in Fig. 9(B)]. The zero-augmented sequences of Fig. 9(A) and (B) are transformed using the FFT to give the interpolated frequency responses. These two procedures can easily be shown to yield identical results, the differences being primarily computational ones.

Sampling Theorem for Case B

If the set of frequency samples is evaluated at $f_k = (k+\frac{1}{2})/NT$, $k=0, 1, \cdots, N-1$, and if this set is defined as F_k, then following a development similar to (4) through (7), we obtain

$$H(z) = \frac{1 + z^{-N}}{(-N)} \sum_{k=0}^{N-1} \frac{F_k}{1 - z^{-1} \exp\left[\dfrac{j2\pi}{N}(k+\frac{1}{2})\right]} \cdot \qquad (14)$$

Evaluating (14) on the unit circle gives

$$H(e^{j\omega T}) = \exp\left[-\frac{j\omega NT}{2}\left(1 - \frac{1}{N}\right)\right] \cdot \sum_{k=0}^{N-1} \frac{jF_k \cos\left(\dfrac{\omega NT}{2}\right) \exp\left[-\dfrac{j\pi}{N}(k+\frac{1}{2})\right]}{\sin\left(\dfrac{\omega T}{2} - \dfrac{\pi}{N}(k+\frac{1}{2})\right)} . \qquad (15)$$

To perform the computation of (15) using the FFT requires a somewhat different procedure than for the previous Cases A and C. This is because in order to compute an inverse DFT using the FFT, it is assumed that the frequency position of the first sample is 0 Hz, whereas in Case B it is $1/(2NT)$ Hz. Therefore, the procedure is the following.

1) Shift the F_k by an angle of π/N clockwise thereby aligning the samples as required by the FFT.

2) Perform the FFT, obtaining a complex impulse response.

3) Either rotate the impulse response by $N/2$ samples and symmetrically augment with zero-valued samples, or split the impulse response at the center and fill in with 15 N zero-valued samples between the two halves of the impulse response.

4) Compute the 16 N point FFT to obtain an interpolated frequency response.

5) Rotate the frequency response data by an angle of π/N (8 samples for a 16 to 1 interpolation) counterclockwise, thereby compensating the original shift and producing the desired result.

The importance of data of Case B is that the interpolated frequency response, when the frequency samples form a real and symmetric set, is pure real. This can be proven from (15), but it is more easily shown to be true by examining the impulse response. For the conditions of Case B the symmetry of the frequency samples can be written

$$F_k = F_{N-1-k}. \qquad (16)$$

Therefore the complex impulse response corresponding to step 2 above is

$$f(n) = \sum_{k=0}^{(N/2)-1} \frac{2F_k}{N} e^{-j\pi n/N} \cos\left(\frac{2\pi}{N} n(k+\frac{1}{2})\right) \qquad (17)$$
$$n = 0, 1, 2, \cdots, N-1.$$

From (17) we see that the real part of $f(n)$ is symmetric, the imaginary part is antisymmetric, and $f(N/2)$ is identically zero. Therefore, the impulse response is technically of duration $(N-1)$ samples, although there are N independent frequency samples. It is, therefore, easy to find an axis of symmetry which coincides with a sample point. Thus the interpolated frequency response corresponding to step 4 above is real. For this case alone the original frequency samples, the true filter impulse response, and the interpolated frequency-response samples are all real.

Sampling Theorem for Case D

The development for Case D is identical to that for Case B. For the set F_k real and symmetric, the interpolated frequency response has a small and imaginary component similar to that of Case A discussed earlier. By making the set F_k complex, a real interpolated frequency response can be obtained as seen previously. Because of the similarity of this case to Case A no further discussion is necessary.

Rationale for Minimization Algorithm

There are several reasons why the different Cases A, B, C, and D are of interest. First, by inspection of (7) and (15), it is seen that when $H(e^{j\omega T})$ is real, it consists of a sum of elementary functions of the form

$$\sin\left(\frac{\omega NT}{2}\right) \Big/ \sin\left(\frac{\omega T}{2} - \theta\right). \qquad (18)$$

In the design of, for example, a low-pass filter one would choose the frequency samples which occur in the passband to have value 1.0 and those which occur in the stopband to have value 0.0. The values of the frequency samples which occur in the transition band would be chosen according to some criterion. It is intuitively appealing to picture that the transition values found for any given optimum design produce functions of the form of (18) with ripples which cancel the ripples caused by the fixed samples. As the number of transition values is increased, it is easy to picture ever finer cancellation. Thus it is useful to obtain a real $H(e^{j\omega T})$.

Another reason for sampling at different frequencies (type-1 and type-2 data) arises when the designer chooses his bandwidth. If the frequency samples are to form an even function, then for Cases A and C, the bandwidth must contain an odd number of samples (the sample at frequency f_k is balanced by a sample at frequency f_{N-k}, except for the sample at $f=0$). Similarly for Cases B and D the bandwidth must contain an even number of frequency samples. Hence sampling at different frequencies provides additional flexibility to the designer. It also turns out that for small bandwidths (in terms of number of in-band frequency samples) the sidelobe ripple cancellation is more efficient when the bandwidth is an even number of samples than when it is an odd number. Furthermore, as will be explained later, a convenient design for band-pass filters is based on rotation of low-pass prototypes. As such, the existence of data from all cases is of great value.

At this point, we now turn to a discussion of the optimization techniques which we have used.

The Minimization Algorithm

From (7) and (15) we observe that $H(e^{j\omega T})$ is a *linear* function of the samples H_k or F_k. In all of our problems most of the H_k or F_k will be preset, and the remaining few (the transition coefficients) will be varied until the maximum sidelobe is a minimum. Fig. 10 shows the typical specification for a low-pass filter. In this example, there are $2\ BW-1$ samples preset to 1.0, $2\ M$ transition samples, and the remaining samples are preset to 0.0. Symmetry considerations reduce the number of independent transition samples to M. Let us denote the transition coefficients by T_1, T_2, T_3, etc. Then Fig. 11 shows how $H(e^{j\omega_1 T})$, $H(e^{j\omega_2 T})$, etc. might vary with any one transition coefficient. All such variations are linear. It has been shown [11] that the upper envelope of these straight lines forms a convex function. (In Fig. 11 the upper envelope is drawn with heavy lines.) It has also been shown [12] that a convex function has a unique minimum (a local minimum is a global minimum). From this it follows that a procedure which searches for the minimum value of a maximum sidelobe must converge; i.e., the search will *not* result in a false minimum.

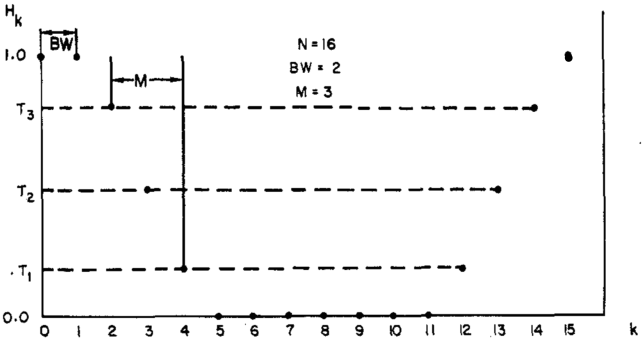
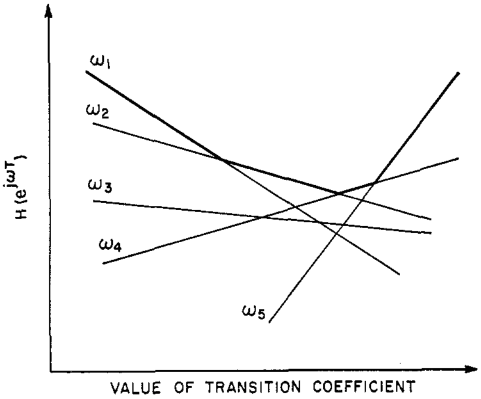


Fig. 10. Typical specifications for type-1 low-pass filters.

Fig. 11. Curves showing the linear variation of the frequency response versus transition coefficient at any given frequency.



The above reasoning may be extended to more than one dimension. $H(e^{j\omega_1 T})$, $H(e^{j\omega_2 T})$, etc., can be plotted as a hyperline of the transition coefficients, T_1, T_2, etc. The upper envelope of the different hyperlines is a convex hypersurface and leads to the same result as before, namely, that a minimax search procedure as a function of T_1, T_2, etc., will converge.

The assurance of ultimate convergence does not necessarily mean that any given search procedure is feasible in terms of computer running time. Now is a good time to stress the discrete nature of our interpolation technique. This discreteness has two important effects. First, it makes it more or less impossible to locate and measure an *exact* minimax of the continuous function $H(e^{j\omega T})$. Experimentally, this is not bothersome; if a sufficient number of ω's are used, the computed result is within a fraction of a decibel of the exact answer. Second, the discreteness helps us by discretizing, in a sense, the convex hypersurface into a connected set of hyperlines. This is true because for small variations in T_1, T_2, etc., the (discrete) frequency position of the maximum sidelobe remains fixed. Thus, over this small variation, the *maximum* $H(e^{j\omega T})$ is a linear function of T_1, T_2, etc., which shows that the convex surface is really a connected set of hyperlines. When the frequency position of the maximum sidelobe changes, the slope of the resultant hyperline of steepest descent changes.

The above reasoning suggests the following search procedure.

1) Always begin with a one-dimensional search. For example, if it is desired to optimize over three transition coefficients, T_1, T_2, and T_3, begin by setting $T_3=T_2=1$ and searching for the value of T_1 in the range 0.0 to 1.0 which yields a minimax. This value is labeled as point A in Fig. 12.

2) Now go to two dimensions. Let $T_2=1.0$ and the value of T_1 obtained from step 1 define a point on a two-dimensional line. To find another point, perturb T_2 slightly from its preset value of unity (to a slightly smaller value) and repeat the one-dimensional search, varying T_1, as before. This new two-dimensional point (point B in Fig. 12), along with the previous one, determines the appropriate straight line (the path of steepest descent) along which to do the full two-dimensional search.

3) A simple search is now made along the line found in step 2, yielding a minimum of $H(e^{j\omega T})$ (point C in Fig. 12). A new path of steepest descent is obtained by varying T_1 and keeping T_2 fixed at the value of point C, yielding point D; then perturbing T_2 slightly and again varying T_1 yielding point E. A simple search is made along the new line yielding a minimum at point F. If the difference between the values of $H(e^{j\omega T})$ at the minima of the searches along the lines of steepest descent (points C and F) is less than some prescribed threshold, the search is ended and point F is the two-dimensional solution. Otherwise the procedures of step 3 are iterated to yield refinements of the path of steepest descent until two consecutive searches yield minima whose difference satisfies the threshold condition. Practically it has been found that a two-dimensional search has always terminated within three iterations when the threshold is set to 0.1 dB.

4) Now go to three dimensions. Let $T_3=1.0$ and the two-dimensional result of step 3 define a point on a three-dimensional line. To find another point on the line, perturb T_3 slightly (to a smaller value) and repeat the two-dimensional search of steps 1 through 3. We now have two points on a three-dimensional line along which we can search for a minimum. At the minimum a new three-dimensional line of steepest descent is obtained and a new search is conducted. The search procedure is terminated when the difference in minima between two consecutive three-dimensional searches is less than a prescribed threshold.

Clearly the search procedure is more time consuming as the dimensionality increases; in fact, it is reasonable to expect that the search time is roughly an exponential function of the dimensionality. We have found experimentally that a four-dimensional search is attainable (within 300 seconds on a CDC-6600 computer), and that all searches have indeed converged.

A useful check on the convergence of the search can be made by examining sidelobes other than the minimax. In general, for M transition coefficients (M-dimensional search) there are $(M+1)$ *equal* minimax sidelobes. The proof of this assertion will not be given here, but philosophically it is similar to the proof given by Papoulis [13] in his discussion of elliptic filters.

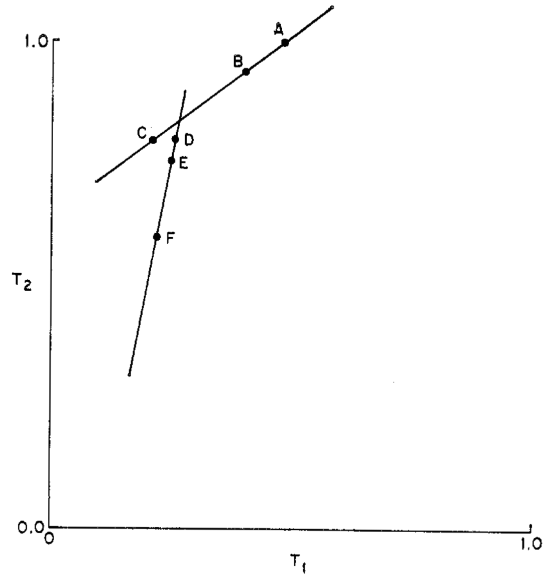


Fig. 12. Illustration of the path followed in a typical search for two optimum transition coefficients.

Note from (7), (15), and Fig. 10, that there are only M variable values of H_k or F_k; the remaining values are preset and remain fixed. This implies that during the course of a search (which may involve thousands of computations of (7) and (15) before convergence) increased computational efficiency results from separating (7) and (15) into two sums, namely, into those terms with the preset H_k or F_k and those terms with the variable H_k or F_k. The first sum may be evaluated once and stored in a table. The second sum consists of very few terms (one to four) and can either be rapidly computed for all values of the (discrete) interpolation for each step in the search or else broken into separate terms, each involving one transition coefficient, and also stored in tables. Using the second alternative, the (discrete) interpolation function is formed for the various values of T_1, T_2, etc., by multiplying the various tables by the appropriate transition coefficients and adding the results. This procedure is uneconomical of computer storage but exceedingly economical of computer running time.

Figs. 13 and 14 show typical examples of the results of a three-dimensional search for type-1 low-pass filters. Fig. 13(A) shows the entire frequency response with $N=64$, $BW=16$, and transition coefficients $T_1=0.030957$, $T_2=0.275570$, and $T_3=0.744348$. For this filter, ripple peaks 1, 2, 5, and 6 are equal within 0.54 dB. Fig. 13(B) shows an expanded view of the frequency response of Fig. 13(A). The first plot in Fig. 13(b) shows the nature of the in-band ripple. The greatly magnified vertical scale is in thousandths of a decibel. The ripple is very small near 0 frequency and increases steadily until the edge of the

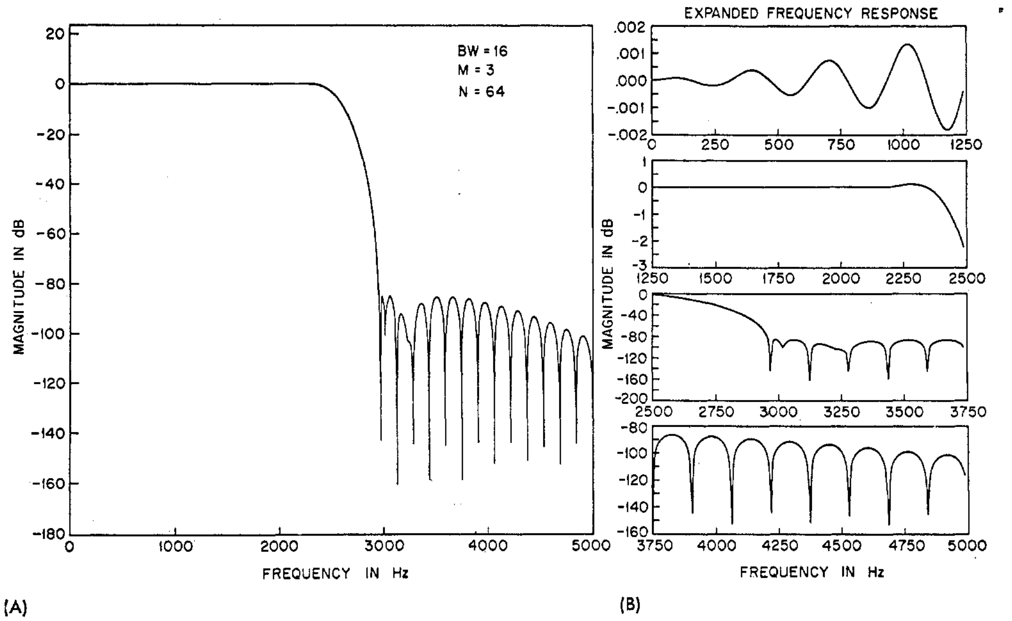
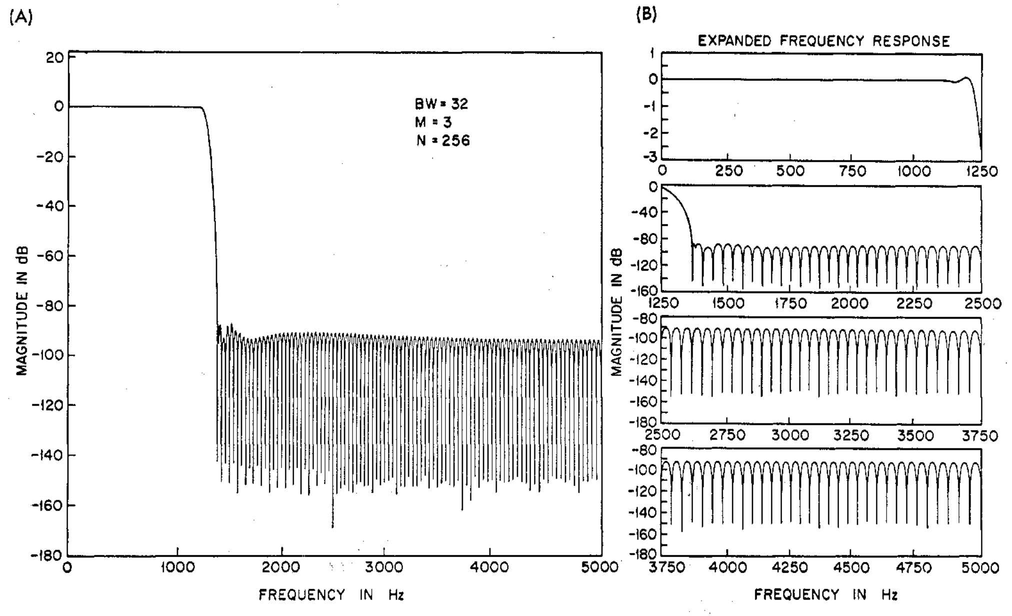


Fig. 13. Interpolated frequency response for low-pass filter with three transition coefficients, for small value of N.

Fig. 14. Interpolated frequency response for low-pass filter with three transition coefficients, for large value of N.



transition band at which point it reaches about 0.1 dB. The next three plots of Fig. 13(B) show the transition band and the out-of-band ripple. It is seen that the peak height of the out-of-band ripple decreases steadily as the frequency gets farther away from the edge of the transition band. This is due to the (sin ω/ω) type interpolation falloff from each of the nonzero values of H_k. Finally it is noted that the minimax solution of Fig. 13 is -85.01 dB.

Fig. 14 shows results for a larger value of N, this case being identical to the one discussed by Gold and Jordan [5]. For this set of data BW is 32, N is 256, and the transition coefficients are $T_1=0.025779$, $T_2=0.251635$, and $T_3=0.723071$. Fig. 14(A) shows the entire frequency response for this filter while (B) shows expanded horizontal and vertical scales. The minimax solution is -87.89 dB, and ripple peaks 1, 2, 4, and 5 are equal with 0.35 dB.

Results

Using the method explained in the previous sections, we have designed a large number of low-pass filters, bandpass filters, and wide-band differentiators. For low-pass filters we have considered type-1 and type-2 data for various values of N, BW, and M, as defined earlier in

TABLE I

Low-Pass Filter Design, One Transition Coefficient (Type-1 Data, *N* Even)

BW	Minimax	T_1
	$N=16$	
1	−39.75363827	0.42631836
2	−37.61346340	0.40397949
3	−36.57721567	0.39454346
4	−35.87249756	0.38916626
5	−35.31695461	0.38840332
6	−35.51951933	0.40155639
	$N=32$	
1	−42.24728918	0.42856445
2	−41.29370594	0.40773926
3	−41.03810358	0.39662476
4	−40.93496323	0.38925171
6	−40.85183477	0.37897949
8	−40.75032616	0.36990356
10	−40.54562140	0.35928955
12	−39.93450451	0.34487915
14	−38.91993237	0.34407349
	$N=64$	
1	−42.96059322	0.42882080
2	−42.30815172	0.40830689
3	−42.32423735	0.39807129
4	−42.43565893	0.39177246
5	−42.55461407	0.38742065
6	−42.66526604	0.38416748
10	−43.01104736	0.37609863
14	−43.28309965	0.37089233
18	−43.56508827	0.36605225
22	−43.96245098	0.35977783
26	−44.60516977	0.34813232
30	−43.81448936	0.29973144
	$N=128$	
1	−43.15302420	0.42889404
2	−42.59092569	0.40847778
3	−42.67634487	0.39838257
4	−42.84038544	0.39226685
5	−42.99805641	0.38812256
7	−43.25537014	0.38281250
10	−43.52547789	0.37826538
18	−43.93180990	0.37251587
26	−44.18097305	0.36941528
34	−44.40153408	0.36686401
42	−44.67161417	0.36394653
50	−45.17186594	0.35902100
58	−46.92415667	0.34273681
62	−49.46289873	0.28751221
	$N=256$	
1	−43.20187807	0.42891235
2	−42.66346216	0.40852051
3	−42.76789284	0.39846802
4	−42.94498634	0.39239502
5	−43.11306858	0.38829956
7	−43.38275146	0.38311157
10	−43.65783644	0.37877197
11	−43.72631788	0.37778931
18	−44.03770208	0.37368774
34	−44.34437370	0.37011490
50	−44.50268936	0.36840210
58	−44.56697893	0.36773071
66	−44.62925625	0.36708985
82	−44.76955509	0.36568604
98	−44.99645948	0.36352539
106	−45.22021151	0.36150513
114	−45.73309422	0.35722656
122	−47.75400352	0.34083862
123	−48.54518938	0.33485107
124	−50.05162334	0.32495117
125	−53.51222181	0.30643310
126	−55.34463978	0.28125000

TABLE II

Low-Pass Filter Design, Two Transition Coefficients (Type-1 Data, *N* Even)

BW	Minimax	T_1	T_2
	$N=16$		
1	−65.27693653	0.10703125	0.60559357
2	−62.85937929	0.12384644	0.62291631
3	−62.96594906	0.12827148	0.62855407
4	−66.03942485	0.12130127	0.61952704
5	−71.73997498	0.11066284	0.60979204
	$N=32$		
1	−67.37020397	0.09610596	0.59045212
2	−63.93104696	0.11263428	0.60560235
3	−62.49787903	0.11931763	0.61192546
5	−61.28204536	0.12541504	0.61824023
7	−60.82049131	0.12907715	0.62307031
9	−59.74928617	0.12068481	0.60685586
11	−62.48683357	0.13004150	0.62821502
13	−70.64571857	0.11017914	0.60670943
	$N=64$		
1	−70.26372528	0.09376831	0.58789222
2	−67.20729542	0.10411987	0.59421778
3	−65.80684280	0.10850220	0.59666158
4	−64.95227051	0.11038818	0.59730067
5	−64.42742348	0.11113281	0.59698496
9	−63.41714096	0.10936890	0.59088884
13	−62.72142410	0.10828857	0.58738641
17	−62.37051868	0.11031494	0.58968142
21	−62.04848146	0.11254273	0.59249461
25	−61.88074064	0.11994629	0.60564501
29	−70.05681992	0.10717773	0.59842159
	$N=128$		
1	−70.58992958	0.09445190	0.58900996
2	−68.62421608	0.10349731	0.59379058
3	−67.66701698	0.10701294	0.59506081
4	−66.95196629	0.10685425	0.59298926
6	−66.32718945	0.10596924	0.58953845
9	−66.01315498	0.10471191	0.58593906
17	−65.89422417	0.10288086	0.58097354
25	−65.92644215	0.10182495	0.57812308
33	−65.95577812	0.10096436	0.57576437
41	−65.97698021	0.10094604	0.57451694
49	−65.67919827	0.09865112	0.56927420
57	−64.61514568	0.09845581	0.56604486
61	−71.76589394	0.10496826	0.59452277
	$N=256$		
1	−70.65072060	0.09458618	0.58923281
2	−68.87253571	0.10375977	0.59425391
3	−68.10910606	0.10658569	0.59466635
4	−67.56728649	0.10690918	0.59322120
6	−67.11176777	0.10612793	0.58998313
9	−66.98584080	0.10502319	0.58671387
10	−66.96896267	0.10457764	0.58571956
17	−67.16303158	0.10323486	0.58217779
33	−67.59191895	0.10212250	0.57908094
49	−67.88601303	0.10168457	0.57769495
57	−67.98678684	0.10133057	0.57690264
65	−68.10634232	0.10109253	0.57627586
81	−68.26243591	0.09982910	0.57389784
97	−68.50218678	0.09773560	0.56999568
105	−68.69239998	0.09583130	0.56641188
113	−68.79911518	0.09375000	0.56186695
121	−68.76781750	0.08519897	0.54300842
122	−68.42778111	0.08584595	0.54292957
123	−68.08275509	0.08768921	0.54538271
124	−68.77609253	0.09545975	0.56272902
125	−72.14476395	0.09437866	0.56816792

TABLE III

Low-Pass Filter Design, Three Transition Coefficients (Type-1 Data, N Even

BW	Minimax	T_1	T_2	T_3
		$N=16$		
1	−96.63068199	0.01597290	0.19530278	0.67931499
2	−95.47730064	0.01951294	0.22385191	0.70432347
3	−98.72853756	0.01724854	0.21873067	0.70148393
4	−114.30107403	0.01259155	0.19831225	0.68572509
		$N=32$		
1	−93.11873436	0.01735230	0.20052231	0.68302930
2	−89.36560249	0.02354126	0.23959557	0.71593525
4	−86.97191620	0.02770996	0.26135787	0.73350248
6	−86.69376850	0.02871094	0.26670884	0.73796855
8	−88.41957283	0.02705688	0.26084303	0.73367810
10	−92.86338043	0.02340698	0.24691517	0.72345444
12	−115.13009739	0.01320190	0.20258948	0.68939742
		$N=64$		
1	−94.71326447	0.01544800	0.19125221	0.67535861
2	−89.93906212	0.02057495	0.22634942	0.70507613
3	−87.99901295	0.02438354	0.24519492	0.72204262
4	−87.48722935	0.02581177	0.25236063	0.72570913
8	−85.52755356	0.03010864	0.27213724	0.74181077
12	−85.35023785	0.02996826	0.27101526	0.74040381
16	−85.01383400	0.03095703	0.27556998	0.74434815
20	−85.68937778	0.02975464	0.27059980	0.74029023
24	−86.59522438	0.02882080	0.26751875	0.73850916
28	−115.01544189	0.01322021	0.20274639	0.68953717
		$N=128$		
1	−94.00015545	0.01566772	0.19122512	0.67492092
2	−92.48471928	0.01967163	0.22183722	0.70122715
3	−90.42631149	0.02271729	0.23824591	0.71450669
5	−88.68881035	0.02398071	0.24430176	0.71859482
8	−87.40693378	0.02510986	0.24892636	0.72166583
16	−86.15447140	0.02633057	0.25347440	0.72460093
24	−85.83375359	0.02739258	0.25816799	0.72843530
32	−85.56298637	0.02763062	0.25894149	0.72886454
40	−84.62227345	0.03014526	0.26996954	0.73824701
48	−85.57633972	0.02913208	0.26652967	0.73584086
56	−86.70246506	0.02863769	0.26648645	0.73743919
60	−114.79892921	0.01326904	0.20291216	0.68961047
		$N=256$		
1	−92.07104015	0.10647949	0.19387524	0.67664281
2	−93.63089275	0.01963501	0.22197911	0.70144920
3	−87.43575478	0.01908569	0.22085960	0.69990539
5	−89.86921692	0.02305298	0.24117076	0.71635813
8	−89.21122360	0.02479248	0.24843111	0.72164702
9	−88.34475231	0.02329712	0.24253562	0.71679420
16	−88.42712784	0.02444458	0.24629538	0.71900030
32	−87.89452744	0.02577896	0.25163493	0.72307099
48	−87.84068012	0.02421875	0.24359358	0.71550480
56	−86.96756554	0.02345581	0.23957232	0.71177494
64	−87.60656548	0.02396851	0.24199281	0.71380179
80	−87.11819744	0.02351685	0.23926844	0.71103085
96	−86.78892708	0.02435913	0.24219392	0.71293931
104	−85.55295181	0.02552490	0.24590992	0.71524908
112	−85.86081982	0.02607422	0.24926456	0.71857490
120	−88.45293331	0.02683105	0.25909273	0.73130690
121	−90.09288883	0.02561035	0.25523207	0.72916388
122	−93.23881817	0.02344360	0.24778644	0.72456966
123	−99.37811375	0.01946106	0.23070314	0.71099759
124	−113.12398720	0.01351929	0.20394843	0.69037794

TABLE IV

Low-Pass Filter Design, Four Transition Coefficients (Type-1 Data, N Even)

BW	Minimax	T_1	T_2	T_3	T_4
			$N=16$		
1	−127.30743676	0.00131836	0.03717696	0.25469056	0.71883166
			$N=128$		
16	−108.29668730	0.00606079	0.09324160	0.40820056	0.82096794

Fig. 10. For bandpass filters and differentiators we have considered type-1 data only.

Before proceeding to a discussion of the data, a few general remarks can be made about the results for low-pass filters.

1) For each filter there are three design parameters: N, M, and BW. A derived parameter, percentage bandwidth, defined as the ratio of in-band frequency bandwidth to half the sampling frequency, is of great value in visualizing the results.

2) For most cases the minimax lies between −40 and −50 dB for a single transition point, between −65 and −75 dB for two transition points, between −85 and −95 dB for three transition points, and about −105 dB for four transition points. To a rough approximation, adding a transition sample reduces the sidelobes by about 20 dB.

3) If the designer wants parameters that are not tabulated, he can find approximate values of the transition coefficients by linear interpolation of the tabulated values. Experimentally, we have found that the deviation of the result obtained by linear interpolation will be less than 6 dB from the optimum.

Low-Pass Filters

The data for type-1 low-pass filters, for N even, are tabulated in Tables I through IV. This set of data corresponds to Case A discussed previously with the frequency samples H_k constituting a real and symmetric set.

TABLE V

Low-Pass Filter Design, One Transition Coefficient (Type-1 Data, *N* Odd)

BW	Minimax	T_1
	$N=15$	
1	−42.30932283	0.43378296
2	−41.26299286	0.41793823
3	−41.25333786	0.41047363
4	−41.94907713	0.40405884
5	−44.37124538	0.39268189
6	−56.01416588	0.35766525
	$N=33$	
1	−43.03163004	0.42994995
2	−42.42527962	0.41042481
3	−42.40898275	0.40141601
4	−42.45948601	0.39641724
6	−42.52403450	0.39161377
8	−42.44085121	0.39039917
10	−42.11079407	0.39192505
12	−41.92705250	0.39420166
14	−44.69430351	0.38552246
15	−56.18293285	0.35360718
	$N=65$	
1	−43.16935968	0.42919312
2	−42.61945581	0.40903320
3	−42.70906305	0.39920654
4	−42.86997318	0.39335937
5	−43.01999664	0.38950806
6	−43.14578819	0.38679809
10	−43.44808340	0.38129272
14	−43.54684496	0.37946167
18	−43.48173618	0.37955322
22	−43.19538212	0.38162842
26	−42.44725609	0.38746948
30	−44.76228619	0.38417358
31	−59.21673775	0.35282745
	$N=125$	
1	−43.20501566	0.42899170
2	−42.66971111	0.40867310
3	−42.77438974	0.39868774
4	−42.95051050	0.39268189
6	−43.25854683	0.38579101
8	−43.47917461	0.38195801
10	−43.63750410	0.37954102
18	−43.95589399	0.37518311
26	−44.05913115	0.37384033
34	−44.05672455	0.37371826
42	−43.94708776	0.37470093
50	−43.58473492	0.37797851
58	−42.14925432	0.39086304
59	−42.60623264	0.39063110
60	−44.78062010	0.38383713
61	−56.22547865	0.35263062

TABLE VI

Low-Pass Filter Design, Two Transition Coefficients (Type-1 Data, N Odd)

BW	Minimax	T_1	T_2
	$N=15$		
1	−70.60540585	0.09500122	0.58995418
2	−69.26168156	0.10319824	0.59357118
3	−69.91973495	0.10083618	0.58594327
4	−75.51172256	0.08407593	0.55715312
5	−103.46078300	0.05180206	0.49917424
	$N=33$		
1	−70.60967541	0.09497070	0.58985167
2	−68.16726971	0.10585937	0.59743846
3	−67.13149548	0.10937500	0.59911696
5	−66.53917217	0.10965576	0.59674101
7	−67.23387909	0.10902100	0.59417456
9	−67.85412312	0.10502930	0.58771575
11	−69.08597469	0.10219727	0.58216391
13	−75.86953640	0.08137207	0.54712777
14	−104.04059029	0.05029373	0.49149549
	$N=65$		
1	−70.66014957	0.09472656	0.58945943
2	−68.89622307	0.10404663	0.59476127
3	−67.90234470	0.10720215	0.59577449
4	−67.24003792	0.10726929	0.59415763
5	−66.86065960	0.10689087	0.59253047
9	−66.27561188	0.10548706	0.58845983
13	−65.96417046	0.10466309	0.58660485
17	−66.16404629	0.10649414	0.58862042
21	−66.76456833	0.10701904	0.58894575
25	−68.13407993	0.10327148	0.58320831
29	−75.98313046	0.08069458	0.54500379
30	−104.92083740	0.04978485	0.48965181
	$N=125$		
1	−70.68010235	0.09464722	0.58933268
2	−68.94157696	0.10390015	0.59450024
3	−68.19352627	0.10682373	0.59508549
5	−67.34261131	0.10668945	0.59187505
7	−67.09767151	0.10587158	0.58921869
9	−67.05801296	0.10523682	0.58738706
17	−67.17504501	0.10372925	0.58358265
25	−67.22918987	0.10316772	0.58224835
33	−67.11609936	0.10303955	0.58198956
41	−66.71271324	0.10313721	0.58245499
49	−66.62364197	0.10561523	0.58629534
57	−69.28378487	0.10061646	0.57812192
58	−70.35782337	0.09663696	0.57121235
59	−75.94700718	0.08054886	0.54451285
60	−104.09012318	0.04991760	0.48963264

Values of minimax and transition coefficients are tabulated as functions of *N* and *M*.

The data for type-1 low-pass filters, for *N* odd, are tabulated in Tables V through VII. This set of data corresponds to Case C discussed previously.

The data for type-2 low-pass filters, for *N* even, are tabulated in Tables VIII through X. This set of data corresponds to Case B discussed previously. The data of these tables are shown graphically in Figs. 15 through 20. The horizontal axis for each of these figures is the percentage bandwidth defined (for type-2 data) as

$$\text{percentage bandwidth} = \frac{2BW - 1}{N}\cdot \qquad (19)$$

Figs. 15, 17, and 19 show the one-, two-, and three-dimensional minimax; Figs. 16, 18, and 20 show values of transition coefficients for one, two, and three transition coefficients.

The curves of minimax all show sharp drops for both large and small values of percentage bandwidth. The drop

TABLE VII

Low-Pass Filter Design, Three Transition Coefficients (Type-1 Data, *N* Odd)

BW	Minimax	T_1	T_2	T_3
		$N=15$		
1	−94.61166191	0.01455078	0.18457882	0.66897613
2	−104.99813080	0.01000977	0.17360713	0.65951526
3	−114.90719318	0.00873413	0.16397310	0.64711264
4	−157.29257584	0.00378799	0.12393963	0.60181154
		$N=33$		
1	−96.03734779	0.01373291	0.18448586	0.67025933
2	−86.96916771	0.01668701	0.20723432	0.68914992
4	−87.86485004	0.01990967	0.22577646	0.70374222
6	−93.33241367	0.02140503	0.23353566	0.70954533
8	−92.88408661	0.02062988	0.22815933	0.70362590
10	−99.76651382	0.01605835	0.20567451	0.68306885
12	−115.53731537	0.00855103	0.16082642	0.63829710
13	−160.64276695	0.00366821	0.12056040	0.59255887
		$N=65$		
1	−95.01340866	0.01552124	0.19101259	0.67492051
2	−93.53645134	0.01989136	0.22329262	0.70260139
3	−91.72289371	0.02214966	0.23609223	0.71288030
4	−89.08071899	0.02115479	0.23418217	0.71154775
8	−88.25607777	0.02576904	0.25203440	0.72436684
12	−88.23531914	0.02576904	0.25178881	0.72372888
16	−89.70906830	0.02368774	0.24385557	0.71742143
20	−84.97463989	0.01913452	0.22072406	0.69678377
24	−87.32207489	0.02396240	0.23899360	0.71008776
28	−115.10305405	0.00938721	0.16513107	0.64142790
29	−142.13555908	0.00352402	0.11881758	0.58955571
		$N=125$		
1	−94.76369476	0.01556396	0.19093541	0.67475127
2	−93.68562794	0.01968994	0.22232235	0.70177618
4	−88.66252041	0.02420654	0.24380589	0.71788831
6	−90.23277760	0.02402344	0.24543860	0.71966323
8	−87.43502617	0.02286987	0.23940384	0.71406829
16	−89.34566116	0.02507324	0.24955546	0.72206742
24	−87.72590733	0.02415771	0.24472233	0.71761565
32	−89.26216030	0.02522583	0.24994760	0.72208238
40	−88.83863926	0.02439575	0.24590832	0.71850440
48	−88.96858311	0.02440186	0.24453094	0.71649557
56	−93.85659409	0.01801758	0.21212543	0.68652023
57	−103.89656162	0.01030884	0.17459164	0.65248157
58	−110.04844761	0.00630951	0.14743829	0.62397790
59	−155.93431473	0.00361786	0.11937160	0.58988331

TABLE VIII

Low-Pass Filter Design, One Transition Coefficient (Type-2 Data, N Even)

BW	Minimax	T_1
	$N=16$	
1	−51.60668707	0.26674805
2	−47.48000240	0.32149048
3	−45.19746828	0.34810181
4	−44.32862616	0.36308594
5	−45.68347692	0.36661987
6	−56.63700199	0.34327393
	$N=32$	
1	−52.64991188	0.26073609
2	−49.39390278	0.30878296
3	−47.72596645	0.32984619
4	−46.68811989	0.34217529
6	−45.33436489	0.35704956
8	−44.30730963	0.36750488
10	−43.11168003	0.37810669
12	−42.97900438	0.38465576
14	−56.32780266	0.35030518
	$N=64$	
1	−52.90375662	0.25923462
2	−49.74046421	0.30603638
3	−48.38088989	0.32510986
4	−47.47863007	0.33595581
5	−46.88655186	0.34287720
6	−46.46230555	0.34774170
10	−45.46141434	0.35859375
14	−44.85988188	0.36470337
18	−44.34302616	0.36983643
22	−43.69835377	0.37586059
26	−42.45641375	0.38624268
30	−56.25024033	0.35200195
	$N=128$	
1	−52.96778202	0.25885620
2	−49.82771969	0.30534668
3	−48.51341629	0.32404785
4	−47.67455149	0.33443604
5	−47.11462021	0.34100952
7	−46.43420267	0.34880371
10	−45.88529110	0.35493774
18	−45.21660566	0.36182251
26	−44.87959814	0.36521607
34	−44.61497784	0.36784058
42	−44.32706451	0.37066040
50	−43.87646437	0.37500000
58	−42.30969715	0.38807373
62	−56.23294735	0.35241699
	$N=256$	
1	−52.98314095	0.25876465
2	−49.84995031	0.30517578
3	−48.54066038	0.32379761
4	−47.72320795	0.33405762
5	−47.17156410	0.34054565
7	−46.50723314	0.34817505
11	−45.86754894	0.35531616
18	−45.39137316	0.36023559
34	−44.99831438	0.36420288
50	−44.81827116	0.36600342
66	−44.68390656	0.36733399
82	−44.54332495	0.36871338
98	−44.33441591	0.37075806
114	−43.77572870	0.37611694
122	−42.27385855	0.38851929
123	−42.17442465	0.39050293
124	−42.62321901	0.39036255
125	−44.78945160	0.38365478
126	−56.23000479	0.35252228

in minimax for the high percentage bandwidth is caused by the fact that very few ripples need to be canceled in the small out-of-band frequency range. The drop for the low percentage bandwidth is caused by the fact that there are very few contributions to the ripple; hence the small amount of ripple in the large out-of-band region is more perfectly canceled than for larger values of percentage bandwidth.

Before proceeding to the data on bandpass filters, two comments seem worthwhile.

1) As seen from Tables I through X or from Figs. 15 through 20, for a broad range of values of percentage bandwidth, values of minimax and transition coefficients do not change much, i.e., the curves tend to be flat topped.

2) For small values of bandwidth, ripple cancellation for type-2 filters is superior to ripple cancellation for

TABLE IX

Low-Pass Filter Design, Two Transition Coefficients (Type-2 Data, N Even)

BW	Minimax	T_1	T_2
		$N=16$	
1	−77.26126766	0.05309448	0.41784180
2	−73.81026745	0.07175293	0.49369211
3	−73.02352142	0.07862549	0.51966134
4	−77.95156193	0.07042847	0.51158076
5	−105.23953247	0.04587402	0.46967784
		$N=32$	
1	−80.49464130	0.04725342	0.40357383
2	−73.92513466	0.07094727	0.49129255
3	−72.40863037	0.08012695	0.52153983
5	−70.95047379	0.08935547	0.54805908
7	−70.22383976	0.09403687	0.56031410
9	−69.94402790	0.09628906	0.56637987
11	−70.82423878	0.09323731	0.56226952
13	−104.85642624	0.04882812	0.48479068
		$N=64$	
1	−80.80974960	0.04658203	0.40168723
2	−75.11772251	0.06759644	0.48390015
3	−72.66662025	0.07886963	0.51850058
4	−71.85610867	0.08393555	0.53379876
5	−71.34401417	0.08721924	0.54311474
9	−70.32861614	0.09371948	0.56020256
13	−69.34809303	0.09761963	0.56903714
17	−68.06440258	0.10051880	0.57543691
21	−67.99149132	0.10289307	0.58007699
25	−69.32065105	0.10068359	0.57729656
29	−105.72862339	0.04923706	0.48767025
		$N=128$	
1	−80.89347839	0.04639893	0.40117195
2	−77.22580583	0.06295776	0.47399521
3	−73.43786240	0.07648926	0.51361278
4	−71.93675232	0.08345947	0.53266251
6	−71.10850430	0.08880615	0.54769675
9	−70.53600121	0.09255371	0.55752959
17	−69.95890045	0.09628906	0.56676912
25	−69.29977322	0.09834595	0.57137301
33	−68.75139713	0.10077515	0.57594641
41	−67.89687920	0.10183716	0.57863142
49	−66.76120186	0.10264282	0.58123560
57	−69.21525860	0.10157471	0.57946395
61	−104.57432938	0.04970703	0.48900685
		$N=256$	
1	−80.73009777	0.04661255	0.40166772
2	−77.22607231	0.06292725	0.47389125
3	−73.46642208	0.07637329	0.51333544
4	−71.95929623	0.08333740	0.53236954
6	−71.13770962	0.08861694	0.54725409
10	−70.46772671	0.09299927	0.55873035
17	−70.05028152	0.09572144	0.56546700
33	−69.59198570	0.09783325	0.57040469
49	−69.15290451	0.09915161	0.57313522
65	−68.87001705	0.10036621	0.57542419
81	−68.56850624	0.10156250	0.57766514
97	−67.91495895	0.10211182	0.57929190
113	−66.49593639	0.10307617	0.58240311
121	−69.16745663	0.10181885	0.58003297
122	−69.47800255	0.10039673	0.57756963
123	−70.38166904	0.09647827	0.57081524
124	−76.36186981	0.08026733	0.54389865
125	−104.06292439	0.04988327	0.48943934

TABLE X

Low-Pass Filter Design, Three Transition Coefficients (Type-2 Data, N Even)

BW	Minimax	T_1	T_2	T_3
		$N=16$		
1	−101.40284348	0.00737915	0.11050435	0.49920845
2	−104.84970379	0.00474243	0.11504207	0.54899404
3	−109.42771912	0.00368042	0.11363929	0.56198983
4	−129.92168427	0.00327759	0.10690445	0.58824614
		$N=32$		
1	−101.36804008	0.00750122	0.11125334	0.50005367
2	−99.62673664	0.01146851	0.15483406	0.59491490
4	−90.25220776	0.01828613	0.20058013	0.66114353
6	−90.54358006	0.01939697	0.21148202	0.67786618
8	−91.16841984	0.01898193	0.21139913	0.68027127
10	−106.64640045	0.01017456	0.16962101	0.64299580
12	−155.63167953	0.00352173	0.11656092	0.58328406
		$N=64$		
1	−108.25328732	0.00635376	0.10549329	0.49324515
2	−96.89522362	0.01149292	0.15247338	0.59059079
3	−93.50934601	0.01464844	0.17720349	0.63073228
4	−91.45420170	0.01686401	0.19265675	0.65259480
8	−89.21100140	0.02095337	0.22033124	0.68792394
12	−91.85765266	0.02094727	0.22664372	0.69828908
16	−91.86564636	0.02175903	0.23164135	0.70385697
20	−93.00161743	0.02034302	0.22588693	0.69939827
24	−90.38130093	0.02063599	0.22318429	0.69521828
28	−138.57364845	0.00465088	0.12720165	0.59773274
		$N=128$		
1	−109.42113113	0.00633545	0.10540650	0.49310051
2	−99.17228699	0.01088257	0.14983535	0.58804236
3	−94.18891430	0.01414185	0.17453122	0.62785978
5	−91.83029461	0.01736450	0.19833205	0.66187917
8	−89.42102623	0.02021484	0.21561932	0.68272648
16	−87.40464020	0.02316284	0.23232431	0.70110354
24	−88.17846870	0.01965942	0.22132125	0.69493783
32	−90.52461393	0.02318115	0.23939702	0.71174777
40	−90.07130432	0.02406006	0.24375106	0.71589811
48	−90.97514629	0.02312622	0.23958630	0.71239267
56	−88.51494217	0.02221069	0.23021766	0.70169053
60	−147.38232040	0.00330200	0.11666346	0.58657999
		$N=256$		
1	−109.40897179	0.00632324	0.10524188	0.49282388
2	−103.89405537	0.01022339	0.14791087	0.58696792
3	−93.76251125	0.01429443	0.17487576	0.62793383
5	−91.26832485	0.01748047	0.19825172	0.66139144
9	−88.72007465	0.02074585	0.21817298	0.68543746
16	−88.28249550	0.02236328	0.22890808	0.69809890
32	−91.24586964	0.02164307	0.23139950	0.70409545
48	−91.01594257	0.02251587	0.23625686	0.70892584
64	−90.69410038	0.02330322	0.24007374	0.71251523
80	−89.23360157	0.02214355	0.23625978	0.70987380
96	−89.70531368	0.02456665	0.24640362	0.71851692
112	−90.55468941	0.02351685	0.24150463	0.71425258
120	−89.99984550	0.02116089	0.22640173	0.69891484
121	−101.17623997	0.01504517	0.19942922	0.67579737
122	−100.22463131	0.00847168	0.16540943	0.64397531
123	−100.61959362	0.00181885	0.12179494	0.59890476
124	−162.43343735	0.00360641	0.11918625	0.58952593

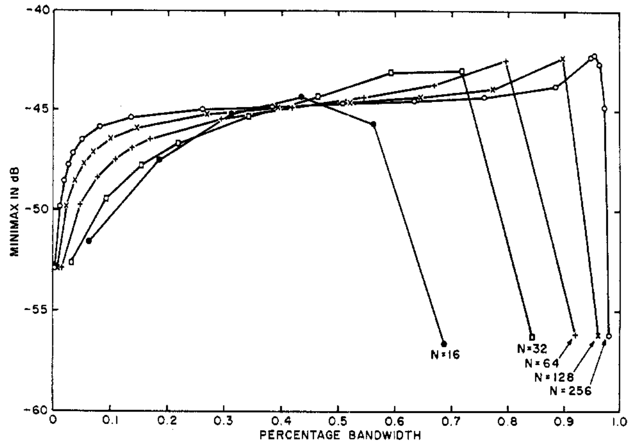


Fig. 15. The minimax as a function of percentage bandwidth for type-2 low-pass filters with one transition coefficient.

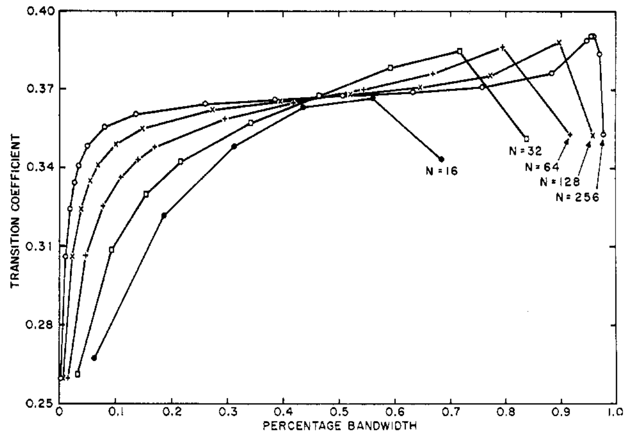
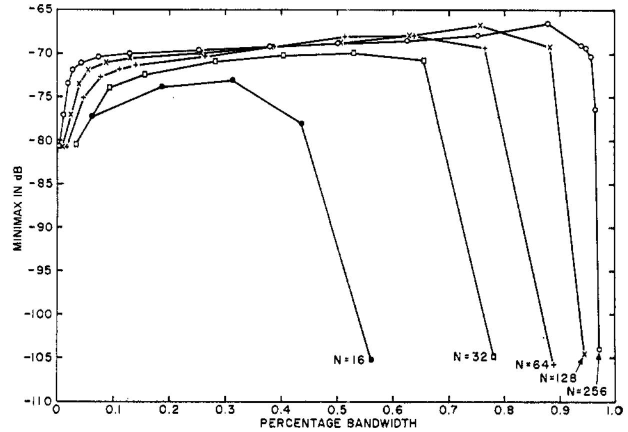


Fig. 16. The value of the transition coefficient as a function of percentage bandwidth for type-2 low-pass filters with one transition coefficient.

Fig. 17. The minimax as a function of percentage bandwidth for type-2 low-pass filters with two transition coefficients.



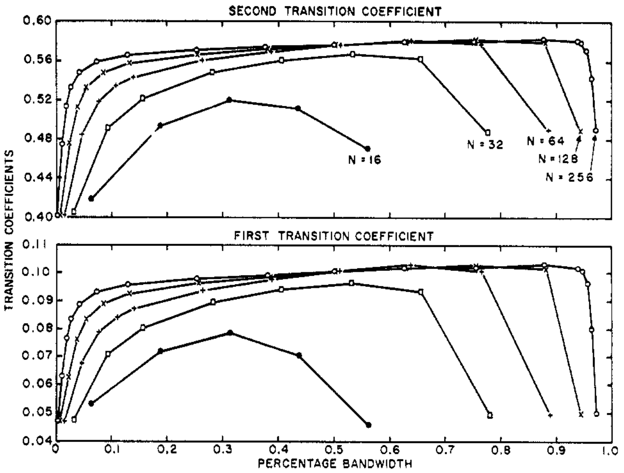


Fig. 18. The values of the transition coefficients as a function of percentage bandwidth for type-2 low-pass filters with two transition coefficients.

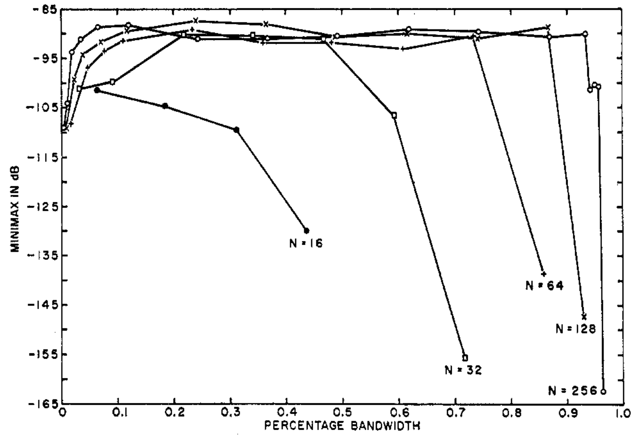
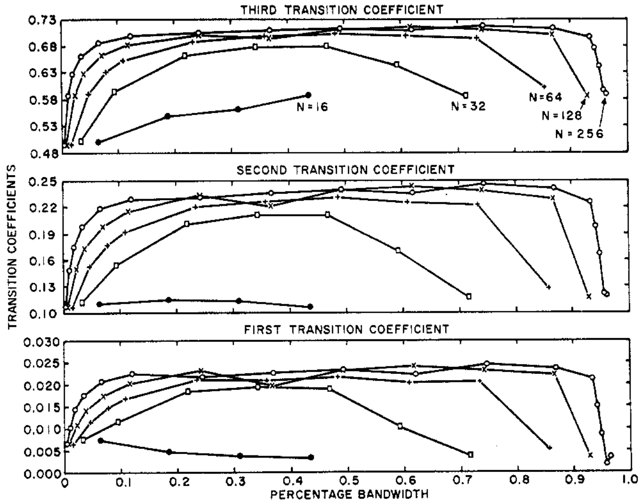


Fig. 19. The minimax as a function of percentage bandwidth for type-2 low-pass filters with three transition coefficients.

Fig. 20. The values of the transition coefficients as a function of percentage bandwidth for type-2 low-pass filters with three transition coefficients.



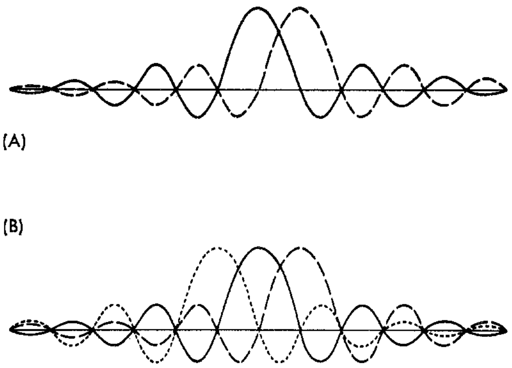


Fig. 21. The summation of an even number (2) and an odd number (3) of sin (x)/x curves.

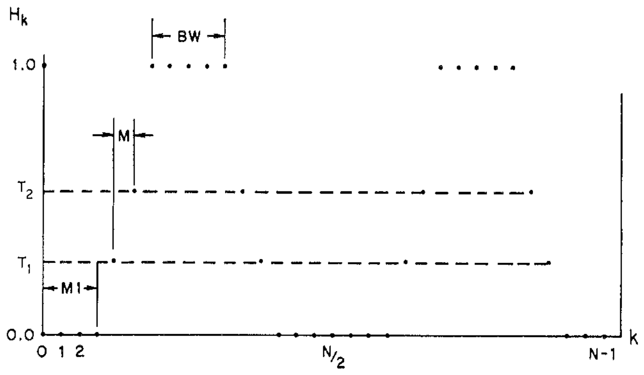


Fig. 22. Typical specifications for a type-1 bandpass filter.

type-1 filters. This can best be explained by referring to Fig. 21. In Fig. 21(A) the ripple from two (sin ω/ω) functions is shown. This case corresponds to a type-2 filter with no transition samples. The ripple peaks from each of the functions tend to cancel uniformly. In Fig. 21(B) the ripple from three (sin ω/ω) functions is shown. This case corresponds to a type-1 filter where the odd term comes from the unpaired frequency sample at zero frequency. The sidelobes from the additional (sin ω/ω) function are seen to add uniformly to all the ripples from Case A. Thus before trying to cancel the ripple with the transition coefficients, the ripple of Fig. 21(A) is significantly less than the ripple of (B). Experimentally it turns out that ripple cancellation for the data of Fig. 21(A) is also much better than for the data of (B). The reason all type-2 filters are not better than all type-1 filters is that as the number of elementary (sin ω/ω) functions increase, the difference in ripple heights between the sum of an even number and the sum of an odd number of such functions becomes smaller and smaller and is negligible for larger bandwidths.

Bandpass Filters

The nomenclature for defining a bandpass filter in terms of its frequency samples is given in Fig. 22; in addition to the parameters *N*, *BW*, and *M*, there is also the center frequency of the filter. We have defined the parameter *M*1 as the number of zero-valued samples preceding the first transition sample. Furthermore, for all cases considered, the bandpass filter samples were considered to be symmetrical about the center frequency. This arbitrary constraint is desirable for computational purposes since it reduces the number of variables by one half. In general, nonsymmetric transition samples lead to a somewhat lower minimax sidelobe, but this advantage seems canceled out by the increased computational cost.

We have approached the design problem in two ways. First, given a version of the optimization program, one can choose the parameters *M*, *N*, *BW*, and *M*1 and run the program to give any desired optimum bandpass filter. We have tabulated the results of a few runs for various values of *N*, *M*1, and *BW*, for one, two, and three symmetric transition coefficients. These data are shown in Tables XI through XIII. The most striking observation from these tables is the difference in minimax between odd and even values of bandwidth for small values of bandwidth. This effect is similar to the one discussed earlier for low-pass filters, and it is worthwhile for the designer to keep it in mind.

The second approach to the design of bandpass filters is to define "suboptimum" bandpass filters, which are derived very simply from the low-pass prototype by appropriately rotating the low-pass frequency samples (including the optimized transition coefficients) to the desired center frequency. An example is given in Fig. 23; the sampled passbands of the derived filter are identical with those of the low-pass, but at different locations. The resulting interpolated bandpass response can be obtained by adding the interpolated low-pass response which has been rotated counterclockwise to the same response rotated clockwise. Therefore, it is clear that the suboptimum filter minimax can never be more than 6 dB worse than the low-pass prototype. However, a truly optimum bandpass filter, as designed by our first approach, may be better than this low-pass prototype; therefore, there is no guarantee that suboptimum bandpass filters are within 6 dB of the optimum. Our experimental results show that a 3-dB loss of suboptimum (relative to optimum) is the usual case.

By allowing rotations of an integer $+\frac{1}{2}$ number of samples, as well as integer rotations, one can design either type-1 or type-2 bandpass filters from either type-1 or type-2 low-pass prototypes. It can be shown that in many cases one of the possible frequency transformations is superior to other transformations. A schematized example is shown in Fig. 24. Fig. 24(A) shows a type-2 low-pass filter with a double frequency ripple peak pass near the band edge. (This situation is typical of many type-2 low-pass filters.) The result of a frequency transformation of an integer number of rotations is shown in Fig. 24(B). The sidelobes add almost everywhere in the out-of-band region. The resulting design is a type-2 bandpass filter. The result of a frequency transformation of an integer

TABLE XI

Bandpass Filter Designs, One Transition Coefficient (Type-1 Data, N Even)

BW	$M1$	Minimax	T_1
		$N=16$	
3	2	−34.175276	0.45593262
		$N=32$	
5	2	−35.767563	0.40270386
5	3	−35.668758	0.39149780
5	4	−36.150829	0.39191895
5	5	−36.577216	0.39454346
6	2	−47.610400	0.30016480
6	3	−49.023605	0.30545654
6	4	−50.470645	0.30634766
6	5	−50.470645	0.30634766
7	2	−35.624309	0.40203247
7	3	−35.433974	0.38943481
7	4	−35.872498	0.38916626
7	5	−35.433974	0.38943481
		$N=128$	
13	10	−41.259604	0.38093262
18	20	−46.386827	0.34979248
19	20	−42.807318	0.37623901
20	20	−46.300830	0.35097046
30	8	−45.528528	0.35111694
35	2	−40.743021	0.35415039
35	8	−42.548930	0.36517944
35	16	−43.227605	0.36619873
35	20	−42.548930	0.36517944

TABLE XII

Bandpass Filter Designs, Two Transition Coefficients (Type-1 Data, N Even)

BW	$M1$	Minimax	T_1	T_2
		$N=16$		
1	2	−74.261646	0.09519653	0.59498585
		$N=32$		
3	2	−60.824677	0.11812134	0.61574359
3	3	−61.426695	0.11731567	0.61224050
3	4	−62.291398	0.12105713	0.61825728
3	5	−62.859379	0.12384644	0.62291631
4	2	−80.477118	0.05566406	0.45630774
4	3	−76.145997	0.06130981	0.46821324
4	4	−79.328351	0.05487671	0.45372102
4	5	−79.328351	0.05487671	0.45372102
5	2	−59.856571	0.12357788	0.62173523
5	3	−61.337764	0.12523804	0.62286495
5	4	−62.965949	0.12827148	0.62855407
5	5	−61.337764	0.12523804	0.62286495
		$N=128$		
11	10	−62.184930	0.11203003	0.59676391
16	20	−70.834468	0.09031982	0.55215414
17	20	−63.214016	0.10917969	0.59054608
18	20	−70.729173	0.09111938	0.55425376
28	8	−69.713386	0.09022827	0.55326966
33	2	−62.526547	0.10723877	0.58803040
33	8	−61.611602	0.11864624	0.60405885
33	16	−62.165339	0.11191406	0.59243833
33	20	−61.611602	0.11864624	0.60405885

TABLE XIII

Bandpass Filter Designs, Three Transition Coefficients (Type-1 Data, N Even)

BW	$M1$	Minimax	T_1	T_2	T_3
			$N=32$		
1	2	−92.550184	0.01790771	0.20360144	0.68572443
1	3	−90.696831	0.01699829	0.19934001	0.68240900
1	4	−90.352082	0.01849976	0.20508501	0.68649875
1	5	−96.630682	0.01597290	0.19530278	0.67931499
2	2	−113.580111	0.00490723	0.09406196	0.47584767
2	3	−110.531433	0.00568848	0.10004413	0.48475702
2	4	−113.033724	0.00422363	0.08800332	0.46619571
2	5	−113.033724	0.00422363	0.08800332	0.46619571
3	2	−89.015630	0.02207642	0.23341023	0.71117572
3	3	−85.383010	0.02587891	0.24743934	0.72135515
3	4	−95.484849	0.01950073	0.22380012	0.70428223
3	5	−85.383010	0.02587891	0.24743934	0.72135515
			$N=128$		
9	10	−86.703837	0.02807617	0.26306832	0.73460394
14	20	−90.302061	0.01867676	0.20640635	0.67233389
15	20	−84.654966	0.03029785	0.27145008	0.74047619
16	20	−90.789648	0.01875610	0.20830675	0.67549529
26	8	−91.905838	0.01835937	0.21150551	0.68240265
31	2	−83.873835	0.02974243	0.27190412	0.74186481
31	8	−84.547179	0.02946167	0.26787226	0.73722876
31	16	−85.064596	0.03010254	0.27143276	0.74060358
31	20	−84.547179	0.02946167	0.26787226	0.73722876

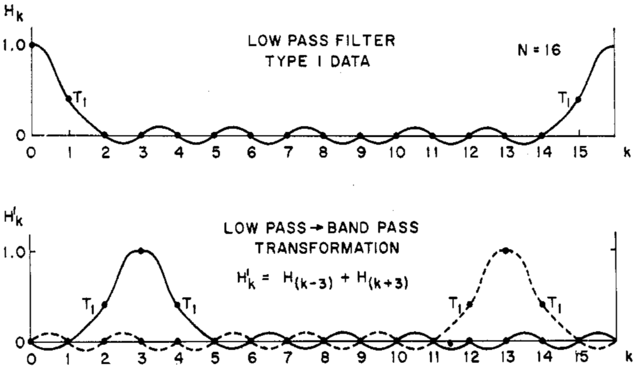
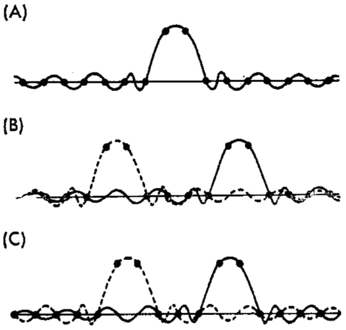


Fig. 23. An illustration of the low-pass to bandpass transformation.

Fig. 24. Transformation of a type-2 (N even) low-pass filter (A) to both type-2 (B) and type-1 (C) bandpass filters. (B) Integer number of rotations. (C) Integer+1/2 number of rotations.



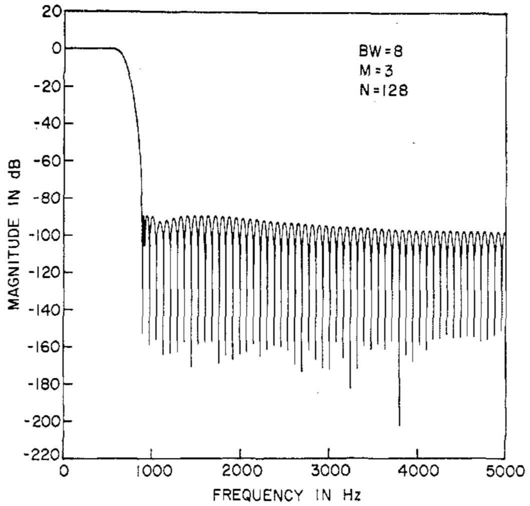


Fig. 25. The frequency response for a type-2 low-pass filter.

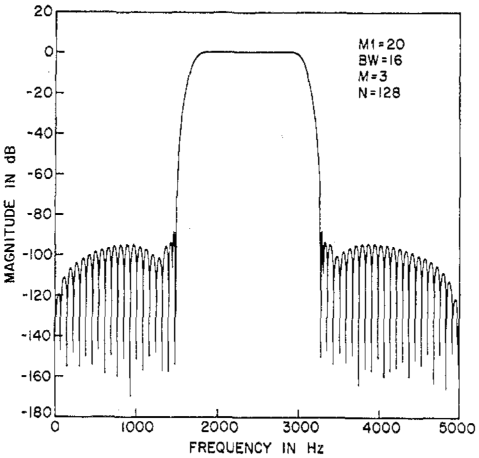


Fig. 27. The frequency response for a type-1 bandpass filter obtained from a transformation of the type-2 low-pass filter of Fig. 25 (integer+1/2 number of rotations).

Fig. 26. The frequency response for a type-2 bandpass filter obtained from a transformation of the type-2 low-pass filter of Fig. 25 (integer number of rotations).

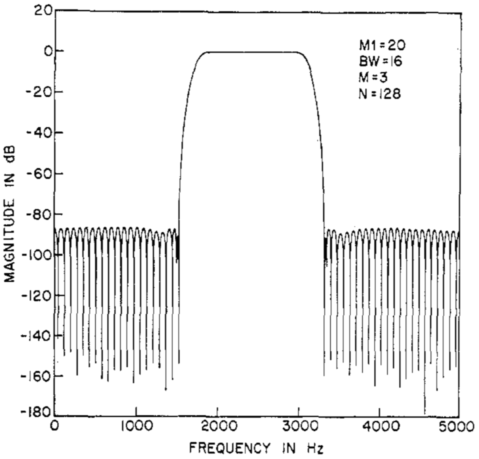
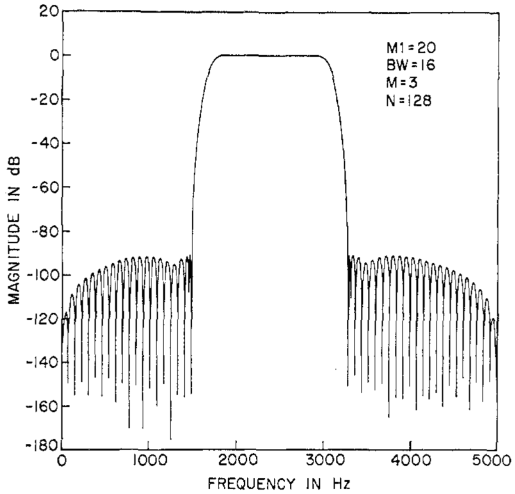


Fig. 28. The frequency response for an optimum type-1 bandpass filter.



$+\frac{1}{2}$ number of rotations is shown in Fig. 24(C). The sidelobes cancel almost everywhere. The resulting design is a type-1 bandpass filter, the characteristics of which are superior to the filter in part (B).

A practical demonstration of these ideas is shown in Figs. 25 through 27. Fig. 25 shows an optimum type-2 low-pass design. The result of an integer number of rotations is shown in Fig. 26; the result of an integer $+\frac{1}{2}$ number of rotations is shown in Fig. 27. The minimax of the low-pass design is -89.4 dB, and the ripple envelope falls to -98 dB at high frequency. The peak ripple of Fig. 26 is about -86 dB, and the ripple envelope falls to -87 dB at high frequency. The peak ripple of Fig. 27 is -89.2 dB, and the ripple envelope falls rapidly at high frequency to about -119 dB. Thus the second frequency transformation is far superior to the first in this case.

For comparison purposes Fig. 28 shows the optimum type-1 bandpass filter as designed by our first approach. The minimax is -90.8 dB, and the ripple envelope drops off rapidly at high frequencies to about -120 dB. It is clear that the suboptimum filter of Fig. 27 is quite similar to the optimum design.

Comparison Between Window and Sampling Design

Direct comparison between the classical window design technique and the frequency-sampling technique described here is difficult; however, we have enough numerical design information to present some comparisons for the low-pass filter case. In addition, since Kaiser [2] in his exposition of the window method surveys the work of himself and others on wide-band differentiators, we thought it useful to design a few differentiators, using the sampling method for further comparisons.

Low-Pass Design Comparisons

The window function introduced by Kaiser has properties very close to those of the prolate spheroidal window [14] and is thus quite close to optimum, given the constraints of a window function design. Kaiser has given an approximate formula for the number of terms N required for a 0.01 percent of peak overshoot in the response. (For the design of a low-pass filter this corresponds to a peak ripple of -80 dB.) The number of terms required is

$$\begin{aligned} N &\approx \frac{11.5}{\text{percentage transition band—windowing}} \\ &= \frac{11.5}{(\omega_2 - \omega_1)/(\omega_s/2)} \cdot \end{aligned} \tag{20}$$

For the sampling technique the percentage transition band, assuming three transition points, is

$$\begin{matrix}\text{percentage transition}\\ \text{band—sampling}\end{matrix} = \frac{4}{N/2} = \frac{8}{N} \cdot \tag{21}$$

All of the low-pass designs of the sampling method have their peak ripple lower than -85 dB, hence somewhat better than the design constraint of (20). Yet (20) implies that for a given N, to achieve -80 dB peak ripple requires a percentage transition band of about $11.5/N$ for Kaiser's window, or about 50 percent bigger than that required by the sampling technique. It should be pointed out, however, that, according to Kaiser, the in-band ripple characteristics of filters designed using the Kaiser windows are equally as good as the out-of-band characteristics. No such claim can be made for the sampling technique because no constraint was placed on the in-band ripple in the design. However, we have found that the largest value of in-band ripple for any of the filters we have designed was less than 0.15 dB.

Helms recently proposed another window possessing certain desirable properties—the Dolph–Chebyshev window. For this window the number of terms N needed to achieve a peak ripple of -80 dB is

$$N \approx \frac{13.0}{(\omega_2 - \omega_1)/(\omega_s/2)} \tag{22}$$

where $(\omega_2 - \omega_1)/(\omega_s/2)$ is the percentage transition band of the filter. Equation (22) shows that the window requires slightly more terms (larger N) than the equivalent Kaiser window and again about 50 percent more terms than the sampling method to achieve this design constraint.

Wide-Band Differentiators

As mentioned earlier, the sampling technique is amenable to filter designs other than standard low-pass or bandpass filters. To illustrate how to apply this procedure to a more general frequency-response characteristic, various wide-band differentiators were designed.

The basic design used data for type-1 filters. Since the ideal frequency response for a differentiator has characteristic response

$$H(\omega) = j\omega, \tag{23}$$

the frequency samples H_k were set to values

$$H_k = H(e^{j2\pi k/N}) = \begin{cases} j\dfrac{k}{N} & k = 0, 1, \cdots, L \\ -\dfrac{j(N-k)}{N} & \\ \quad k = N-L, \cdots, N-1 & \\ \text{optimally chosen, all other } k. & \end{cases} \tag{24}$$

A single value of 19 was used for N in order to compare the resulting differentiators with those described by Kaiser [1].

The design criterion used was one which sought to minimize either the maximum absolute deviation or the maximum absolute relative deviation between the interpolated frequency response and the ideal differentiator frequency response over some specified range. For the case studied ($N = 19$) there were seven fixed values of H_k and three variable samples. Various normalized in-band frequency ranges were used for the minimization. These frequency ranges included:

1) 0 to 0.737 full band
2) 0 to 0.789 full band
3) 0 to 0.842 full band.

The resulting differentiators are tabulated with respect to the maximum absolute error and transition coefficients in Table XIV. A typical interpolated frequency response and the absolute error for a minimum absolute error differentiator in the range 0 to 0.737 full band are shown in Fig. 29. The peak error in this range is 0.00019 and occurs at a normalized frequency near the edge of the differentiator band. However, as seen in Fig. 29, the peak error remains large even for low frequencies. Fig. 30 shows the frequency response and absolute error for a minimum relative error differentiator. The peak error here is 0.0003; however the error is much smaller at low frequencies (on the order of 10^{-5} to 10^{-4}) and remains small for most of the frequency range.

Kaiser [1] has compared six techniques for designing nonrecursive wide-band differentiators. The best result among those presented uses a Kaiser window ($\omega_a\tau = 6.0$) with differentiation bandwidth of about 0.8 full band and

TABLE XIV

Differentiator Design (Type-1 Data, $N=19$)

Percent Band-width	Peak Error	T_1	T_2	T_3
		Minimized Absolute In-Band Error		
0.737	0.0001891	0.37163696	0.76372207	0.73665305
0.789	0.0010745	0.42243652	0.80468043	0.73684211
0.842	0.0051854	0.48053589	0.83691982	0.73684211
		Minimized Relative In-Band Error		
0.737	0.0003032	0.36164551	0.75508157	0.73642816
0.789	0.0017759	0.39659424	0.78972235	0.73752642
0.842	0.0136256	0.43627930	0.82847971	0.73684211

Fig. 29. The frequency response (A) and absolute error curve (B) for a wide-band differentiator whose transition coefficients were chosen so as to minimize the maximum absolute error.

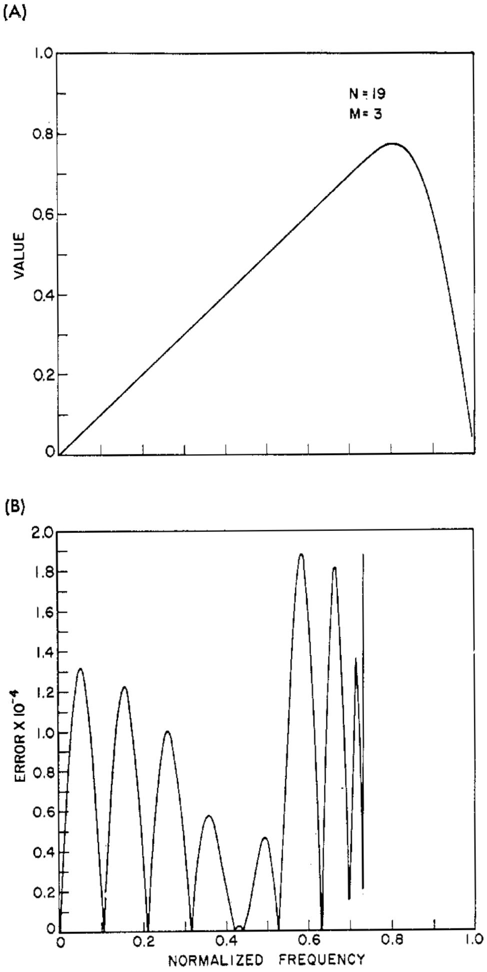
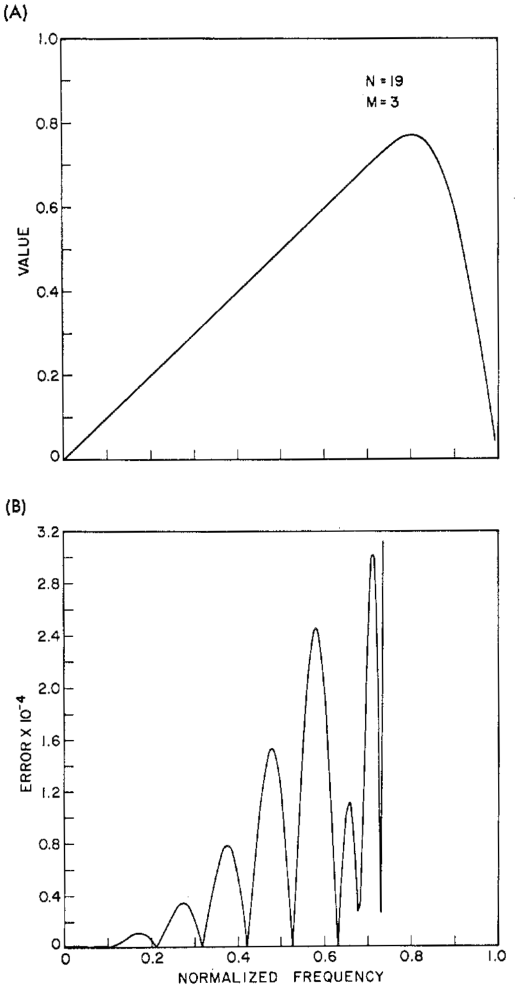


Fig. 30. The frequency response (A) and relative error curve (B) for a wide-band differentiator whose transition coefficients were chosen so as to minimize the maximum relative error.



peak error of 0.0016. Both the peak error and the shape of the error curve are similar to the minimum relative error design presented in Table XIV for a bandwidth of 0.789 full band. Hence in this case these very different techniques yield filter designs which are quite similar.

Experimental Results Obtained When Finite Register Length Is Taken into Account

In performing the search for the optimum filter designs, a 60-bit word length computer (CDC-6600) was used and the results checked with a 36-bit word length machine (GE-635). We can therefore assume no significant truncation errors occurred in this computation. However, the synthesis of a given filter could conceivably be performed on an 18-, 16-, or even 12-bit machine, or perhaps with special purpose hardware where the shortest possible word length is desirable. Much work has been done recently on the subject of the effects of finite register length. This work can roughly be divided into two parts:

1) truncation of the parameters, which changes the filter shape;
2) truncation of the variables, which introduces noise into the output.

In an earlier section we saw that there are three standard nonrecursive filter realizations: direct convolution, frequency sampling, and fast convolution. Weinstein [15] has treated the latter two realizations for case 2, both theoretically and experimentally. Noise in the direct-convolution realization is easily computed by assuming that each multiplication introduces an independent noise of variance $E_0^2/12$, where E_0 is a single quantization level. The total noise variance is $E_0^2 \cdot N/12$ where N is the number of multiplications in the realization, i.e., the length of the filter impulse response.

For parameter truncation simple models are not readily available so that theoretical prediction cannot safely be made. Therefore we performed measurements for the standard realizations.

1) Direct Convolution: The impulse response of several of the type-1 designs of low-pass filters was accurately computed and the coefficients were then truncated. Values for N of 16 and 32 were used since a direct convolution realization would not generally be used for larger values of N. The results of truncation are shown in Table XV. The maintenance of at least −80 dB rejection required 17 bits, and the maintenance of −73 dB rejection required 14 bits for three transition samples.

2) Frequency Sampling: The frequency samples for several type-1 low-pass filters were truncated. Since most of the frequency samples for the low-pass case were either 0 or 1.0, only three coefficients were actually affected by the truncation. The results of truncation are presented in Table XVI. Truncation to 17 bits did not seriously affect the peak ripple. The maintenance of at least −75 dB rejection required only 11 bits for the coefficients. It should be noted that the coefficients of the resonators in the frequency sampling realization (Fig. 6) were not truncated. Hence the results here are an overbound on the actual results of coefficient truncation.

TABLE XV

Truncation of Frequency Samples of Type-1 Low-Pass Filters, Three Transition Coefficients

Number of Bits	Minimax	Number of Bits	Minimax
$N=16$, $BW=1$		$N=64$, $BW=28$	
36	−96.63	36	−115.24
17	−95.05	17	−112.80
14	−88.60	14	−111.43
11	−92.74	11	−89.84
8	−75.57	8	−70.75
5	−38.86	5	−62.35
$N=16$, $BW=4$		$N=128$, $BW=8$	
36	−114.12	36	−87.41
17	−113.53	17	−87.35
14	−113.33	14	−83.95
11	−76.51	11	−75.71
8	−59.29	8	−72.10
5	−41.81	5	−62.66
$N=32$, $BW=2$		$N=128$, $BW=60$	
36	−89.37	36	−114.44
17	−88.51	17	−110.66
14	−84.75	14	−99.74
11	−88.23	11	−89.86
8	−75.99	8	−70.75
5	−59.03	5	−62.35
$N=32$, $BW=12$		$N=256$, $BW=8$	
36	−112.50	36	−89.21
17	−111.57	17	−88.41
14	−109.60	14	−87.23
11	−79.18	11	−75.80
8	−62.24	8	−72.10
5	−62.34	5	−62.66
$N=64$, $BW=4$		$N=256$, $BW=124$	
36	−87.48	36	−112.80
17	−87.39	17	−109.22
14	−86.45	14	−100.43
11	−81.26	11	−82.60
8	−59.53	8	−64.07
5	−39.72	5	−62.35

3) Fast Convolution: The effects of truncation are straightforward. Each of the interpolated frequency response coefficients are truncated; hence coefficients falling below the quantization level are truncated to have 0 value.

Nonuniform Frequency Samples

In this section we will show that a finite-duration impulse-response filter could be designed from frequency samples placed *anywhere* in the z-plane. Whereas in the

TABLE XVI

Truncation of Impulse-Response Coefficients of Type-1 Low-Pass Filters, Three Transition Coefficients

Number of Bits	Minimax	Number of Bits	Minimax
$N=16$, $BW=1$		$N=32$, $BW=4$	
36	−96.63	36	−86.97
17	−96.33	17	−85.74
14	−83.67	14	−75.23
11	−67.34	11	−58.75
8	−51.14	8	−39.76
5	−33.18	5	−31.44
$N=16$, $BW=2$		$N=32$, $BW=6$	
36	−95.47	36	−86.69
17	−92.73	17	−82.39
14	−78.52	14	−73.92
11	−67.26	11	−55.41
8	−46.06	8	−36.03
5	−27.32	5	−24.76
$N=16$, $BW=3$		$N=32$, $BW=8$	
36	−98.70	36	−88.41
17	−94.57	17	−83.14
14	−73.63	14	−78.70
11	−55.53	11	−56.21
8	−38.73	8	−38.11
5	−39.71	5	−27.86
$N=16$, $BW=4$		$N=32$, $BW=10$	
36	−114.12	36	−92.85
17	−95.15	17	−87.03
14	−83.53	14	−74.11
11	−63.30	11	−53.65
8	−42.20	8	−36.31
5	−29.31	5	−25.38
$N=32$, $BW=2$		$N=32$, $BW=12$	
36	−89.37	36	−112.50
17	−88.26	17	−89.05
14	−82.35	14	−68.20
11	−62.82	11	−61.38
8	−50.18	8	−44.26
5	−30.21	5	−26.32

previous sections we have restricted ourselves to the case of uniformly spaced samples around the unit circle, in this section we will discuss an extension of the techniques to nonuniformly spaced samples around the unit circle.

Let $h(n)$, $n=0, 1, \cdots, N-1$, be the impulse response of a nonrecursive filter with z-transform $H(z)$. It can be shown that N independent values of $H(z)$ can be specified for this filter by writing $H(z)$ in the form [16]

$$H(z) = \sum_{k=0}^{N-1} \frac{H_k}{\alpha_k} \frac{\prod_{i=0}^{N-1} (1 - z^{-1}a_i)}{(1 - z^{-1}a_k)} \tag{25}$$

where

$$\alpha_k = \prod_{i=0,\, i\neq k}^{N-1} (1 - a_i/a_k) \tag{26}$$

and $\{a_k\}$ are the z-plane positions at which $H(a_k)=H_k$. $H(z)$ in (25) can be shown to be an $(N-1)$st order polynomial in z^{-1}. Thus the design of a nonrecursive filter can be thought of in terms of deriving suitable values for $\{a_k\}$ and $\{H_k\}$ of (25).

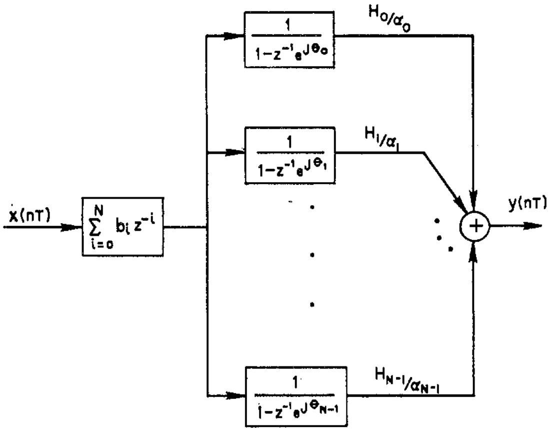

x(nT)
$\sum_{i=0}^{N} b_i z^{-i}$
$\frac{1}{1-z^{-1}e^{J\theta_0}}$
H_0/α_0
$\frac{1}{1-z^{-1}e^{J\theta_1}}$
H_1/α_1
$\frac{1}{1-z^{-1}e^{J\theta_{N-1}}}$
H_{N-1}/α_{N-1}
+
y(nT)

Fig. 31. A method of realization of the nonuniform sampled nonrecursive digital filter.

In this section we will consider the nonuniform case where

$$a_k = e^{j2\pi\theta_k/N}, \tag{27}$$

i.e., nonuniformly spaced samples around the unit circle. For this set of samples (25) can be manipulated into the form

$$H(z) = \sum_{k=0}^{N-1} \frac{H_k \sum_{i=0}^{N-1} b_i z^{-i}}{\alpha_k(1 - z^{-1}e^{j2\pi\theta_k/N})} \tag{28}$$

where

$$\alpha_k = \prod_{i=0,\, i\neq k}^{N-1} (i - e^{j2\pi(\theta_i-\theta_k)N}) \tag{29}$$

and the internal summation in (28) comes from expanding the product in the numerator of (25). The realization of (28) is shown in Fig. 31. The internal summation is realized as a nonrecursive filter, whose output is fed into N parallel channels, each consisting of a complex resonator followed by a complex multiplication. The outputs of the parallel channels are summed to give the filter output.

To obtain the interpolated frequency response of networks of the form of (28), the network realization of Fig. 31 is first excited by an impulse to give the impulse response; zero-valued samples are added to the impulse response; and the entire array is transformed using the FFT. Since (28) is still linear in the coefficients, the H_k, the techniques for finding optimum values of transitions are still valid.

Two sets of nonuniform data were investigated. These data are shown in Fig. 32. The first set, shown in Fig. 32(A), consisted of 16 uniform samples with an extra sample placed between the third and fourth uniform

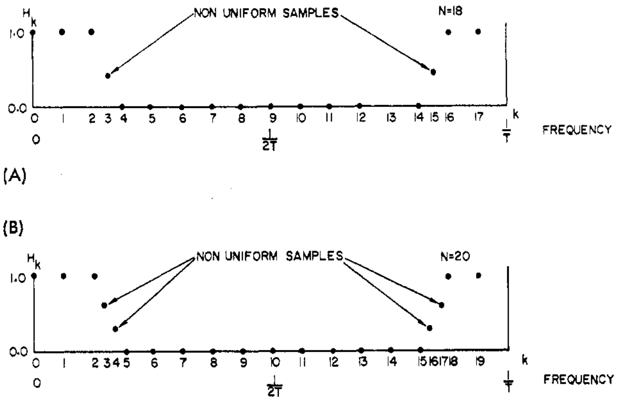


Fig. 32. Two cases of nonuniform frequency samples investigated in this paper.

samples. The design criterion was to choose an optimum position for the nonuniform sample to minimize out-of-band ripple. The optimum value turned out to be 0.468 yielding a peak out-of-band ripple of -20.3 dB, a peak in-band ripple of 1.2 dB, and a flat ripple envelope. For comparison purposes, the case of 16 uniformly spaced samples with no transitions was examined. Here the peak out-of-band ripple was -15.7 dB, the peak in-band ripple was 0.8 dB; and the ripple envelope fell to -24 dB at high frequencies.

Since the peak ripple was reduced by about 6 dB from the uniform case, a second nonuniform case, Fig. 32(B), was studied. Two additional samples were placed between the third and fourth uniform samples. The design program chose optimum values for these transitions to minimize the peak ripple. Here the results were discouraging as the peak ripple was increased to -11.3 dB.

The results obtained with nonuniformly spaced samples have not been entirely encouraging. Further work must be done before any conclusions can be arrived at as to the advantages over uniform sampling.

Conclusion

This paper has presented a technique for designing many types of finite-duration impulse-response digital filters from considerations strictly in the digital frequency domain. The ideal frequency response of the filter is approximated by placing appropriate frequency samples in the z-plane and then choosing the remaining frequency samples to satisfy an optimization criterion. This technique has been applied successfully to the design of low-pass and bandpass filters, as well as wide-band differentiators. The extension of this procedure to standard filters, such as bandstop and high-pass filters, as well as Hilbert transform filters, notch filters, double differentiators, and many others is straightforward.

The design program is sufficiently simple to implement so that it can be programmed to meet the requirements of the individual user. However, should the user merely desire a standard filter with good out-of-band characteristics, he can use the data included in the tables of this paper and proceed from there. Should the user desire a value of bandwidth which is not in the tables, a simple technique would be to interpolate linearly between the nearest values in the table. This will generally yield a suboptimum filter which is almost as good as the optimum.

The design of bandpass, bandstop, and high-pass filters can be treated as a separate design problem, using the frequency sampling technique described; or else simple frequency transformations of low-pass filters can be used to derive suboptimum designs. In many cases these suboptimum designs are nearly optimum.

The frequency sampling technique has been shown to be competitive with the standard window technique in that the number of terms needed to achieve a desired peak ripple in the stopband using this technique is about 50 percent less than the number of terms using the optimum windows described by Kaiser and Helms.

The extension of the frequency sampling technique to include nonuniform sampling points has been discussed briefly. More work must be done before the limitations and advantages of nonuniform samples are fully understood and appreciated.

Acknowledgment

The authors would like to express their appreciation for helpful technical discussions of the material in this paper with Dr. K. Jordan of M.I.T. Lincoln Lab. and Dr. R. Schafer of Bell Telephone Labs.

We would also like to thank Miss June Ley for valuable clerical assistance in the preparation of this manuscript.

References

[1] J. F. Kaiser, "Digital filters," ch. 7 in *System Analysis by Digital Computers*, F. F. Kuo and J. F. Kaiser, Eds. New York: Wiley, 1966.

[2] B. Gold and C. M. Rader, *Digital Processing of Signals.* New York: McGraw-Hill, 1969, ch. 3.

[3] H. D. Helms, "Nonrecursive digital filters: Design methods for achieving specifications on frequency response," *IEEE Trans. Audio and Electroacoustics*, vol. AU-16, pp. 336–342, September 1968.

[4] M. A. Martin, "Digital filters for data processing," Missile and Space Div., General Electric Co., Tech. Information Series Rept. 62-SD484, 1962.

[5] B. Gold and K. L. Jordan, Jr., "A direct search procedure for designing finite duration impulse response filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU-17, pp. 33–36, March 1969.

[6] T. G. Stockham, "High-speed convolution and correlation," *1966 Spring Joint Computer Conf., AFIPS Proc.*, vol. 28. Washington, D.C.: Spartan, 1965, pp. 229–233.

[7] H. D. Helms, "Fast Fourier transform method of computing difference equations and simulating filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, pp. 85–90, June 1967.

[8] C. M. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proc. IEEE*, vol. 55, pp. 149–171, February 1967.

[9] J. W. Cooley and J. W. Turkey, "An algorithm for machine computation of complex Fourier series," *Math. Computation*, vol. 19, pp. 297–301, April 1965.

[10] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp z-transform algorithm and its application," *Bell Sys. Tech. J.*, vol. 48, pp. 1249–1292, May 1969.

[11] G. Hadley, *Linear Programming.* Reading, Mass.: Addison-Wesley, 1963, ch. 2.

[12] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968, ch. 4.

[13] A. Papoulis, "On the approximation problem in filter design," *IRE Conv. Rec.*, pt. 2, pp. 175–185, 1957.

[14] D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty—I and II," *Bell Sys. Tech. J.*, vol. 40, pp. 43–84, 1961.

[15] C. J. Weinstein, "Quantization effects in digital filters," Ph.D. dissertation, Dept. of Elec. Engrg., M.I.T., Cambridge, Mass., July 1969.

[16] A. Cauchy, "Analyse mathématique-memoire sur diverses formules d'analyse," in *Oeuvres Complètes*, ser. 1, vol. 6. Paris, France: Gauthier-Villars, 1888, pp. 63–78.

Lawrence R. Rabiner (S'62–M'67) was born in Brooklyn, N. Y., on September 28, 1943. He received the S.B. and S.M. degrees simultaneously in June, 1964, and the Ph.D. degree in electrical engineering in June, 1967, all from the Massachusetts Institute of Technology, Cambridge.

From 1962 through 1964, he participated in the cooperative plan in electrical engineering at Bell Telephone Laboratories, Inc., Whippany and Murray Hill, N. J. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently he is engaged in research on speech communications and digital signal processing techniques at Bell Telephone Laboratories, Murray Hill.

Dr. Rabiner is a member of Eta Kappa Nu, Sigma Xi, Tau Beta Pi, and the Acoustical Society of America.

Bernard Gold (M'49–SM'67) was born in New York, N. Y., on March 31, 1923. He received the B.S.E.E. degree from the City College of New York, N. Y., in 1944, and the Ph.D. degree in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, N. Y., in 1948.

From 1948 to 1953 he worked at the Avion Instrument Corp. and Hughes Aircraft Company on radar and missile system electronics. He has been with the M.I.T. Lincoln Laboratory, Lexington, Mass., since 1953, working on pattern recognition, noise theory, speech bandwidth compression, and digital signal processing techniques. In 1954–1955 he was a Research Fulbright Fellow in Italy, and in 1965–1966 he was a Visiting Professor of Electrical Engineering at M.I.T., Cambridge, Mass. He is the author of about 30 papers and co-author of the book, *Digital Processing of Signals.*

Dr. Gold is a member of the Acoustical Society of America, URSI, and the American Association for the Advancement of Science.

Carol A. McGonegal was born in Plainfield, N. J., on April 2, 1947. She is currently studying for the B.A. degree in mathematics at Fairleigh Dickinson University, Rutherford, N. J.

She also works at Bell Telephone Laboratories, Inc., Murray Hill, N. J., as a computer programmer in the Acoustics, Speech and Mechanics Research Laboratory.