FILTER APPROXIMATION METHODS

This article is concerned with obtaining the transfer function of an electrical filter that meets certain specifications. These specifications include discrimination properties, time delay, or a combination of these. Depending on the complexity and severity of the requirements, one may either find solutions to these problems in closed form, or one may have to resort to iterative approximations to find solutions. Once the transfer function is computed, one must then determine an implementation of the filter, which will be treated in other articles.

The transfer function of a filter is a real, rational fractional function of the complex frequency variable $s = \sigma + j\omega$ usually given in one of the two forms:

$$\begin{aligned} \text{output/input} &= H(s) = N(s)/D(s) \\ &= \frac{n_0 + n_1 s + n_2 s^2 + \cdots + n_n s^n}{1 + d_1 s + d_2 s^2 + \cdots + d_d s^d} \\ &= H_0 \frac{\prod_{i=1}^{n}(s - z_i)}{\prod_{j=1}^{d}(s - p_j)} \end{aligned} \tag{1}$$

where the numerator polynomial $N(s)$ is of degree n and the denominator $D(s)$ is of degree d. If we express these polynomials in terms of their zeros, these zeros (z_i) and poles (p_j), if complex, occur in complex conjugate pairs. The zeros and poles are much more useful in describing the behavior of the filter than the polynomial coefficients, and all the poles [the zeros of $D(s)$] must be inside the left half of the s plane for stability.

This description is valid for analog filters i.e., those containing resistors, inductors, and capacitors (R, L, and C) or active R and C components. For infinite-impulse-response

(IIR) digital filters and microwave filters consisting of equal length open- and short-circuited as well as cascaded transmission line segments, we can still use the preceding expressions, if we replace the variable s by the expression

$$S = \tanh \frac{\pi s}{2\omega_0} \tag{2}$$

where ω_0 is half the sampling frequency in the digital case or the (common) quarter-wave frequency of the transmission line segments in the microwave case.

The significant filter performance parameters we are concerned with are the loss, defined as

$$a = 10\log_{10}|H(\mathrm{j}\omega)|^2 \tag{3}$$

and the delay

$$\tau = -\frac{d}{d\omega}\arg[H(j\omega)] = -\frac{d}{d\omega}\tan^{-1}\frac{\mathrm{Im}[H(j\omega)]}{\mathrm{Re}[H(j\omega)]} \tag{4}$$

Occasionally, we need the impulse or step responses of the filter; these can be computed as the inverse Laplace transforms of $H(s)$ and $H(s)/s$, respectively.

In addition to the restriction on the locations of the poles, the transfer function $H(s)$ must also meet the following criteria to be realizable by an R, L, and C or a microwave network:

$$|H(s)|_{s=j\omega} \leq 1 \tag{5}$$

and the polynomial $N(s)$ must be either pure even or pure odd (i.e., its zeros must be either pure imaginary or occur in complex quadruplets); furthermore, its degree may not be greater than that of $D(s)$.

While the first condition is not necessary for digital and active RC implementations, assuming that it is satisfied does not restrict the generality at all, since we can always include an amplifying stage anywhere in the structure and since the function must necessarily be bounded on the imaginary axis; hence we shall assume this bound to be unity.

For digital filters, the degree of $N(s)$ is not restricted, but again for simplicity we shall assume compliance, because otherwise difficulties arise. For IIR digital filters with numerators of degree greater than that of the denominators, please see Ref. 1.

Finally, $N(s)$ being pure even or pure odd is not strictly necessary, since zeros may occur in the right half plane and we can always pair them with zeros in the left half plane and compensate for this by having poles at the same locations. In any case, these types of zeros are found useful only in compensating for delay distortion and, as such, can and will be treated separately. However, no zeros inside the left half plane are allowed for passive RLC and microwave circuits, without matching right-half-plane zeros.

As far as IIR digital and active RC circuits are concerned, $N(s)$ is not restricted to being pure even or odd. Nevertheless, we shall assume that it is (except if the microwave filter contains unit elements), for the simple reason that it makes for a unified treatment of all filter kinds and, furthermore, there does not seem to be any advantage in assuming otherwise.

THE APPROXIMATION PROBLEM

Approximation problems in the design of filters take the following forms:

1. Requirements on the loss only. This is the most common case and has usually two forms:
   i. In the passband the loss should be low (near zero) or of a specified shape.
   ii. In the stopband the loss should be equal to or greater than some specified amount (usually as a function of frequency).
2. Requirements on the delay only. This covers the problem of delay equalization and the design of delay lines.
3. Requirements on both the loss and the delay. This is the most complex case and is usually treated by breaking it up into first dealing with the loss and subsequently handling the delay, although methods exist to handle them simultaneously.
4. Requirements on the impulse or step response. Occasionally we encounter this type of requirement placed on the time-domain response of the filter. This may even be combined with simultaneous requirements on the loss. This is rare, but we shall mention some methods of dealing with cases of this type at the end of this article.

CLOSED-FORM SOLUTIONS

The Approximation of Loss

To simplify the problem of loss approximation, we rewrite the expression for the transfer function. If the function magnitude is bounded by 1, then we can write

$$|H(s)|^2_{s=j\omega} = H(s)H(-s)|_{s=j\omega} = \frac{1}{1+\kappa(j\omega)\kappa(-j\omega)} \tag{6}$$

where the $\kappa(s)$ function is called the *characteristic* function and is of the form

$$\kappa(s) = F(s)/N(s) \tag{7}$$

Here $N(s)$ is the numerator of $H(s)$ and $F(s)$ is a completely arbitrary real polynomial of degree d [same as that of $D(s)$]. The only additional restriction is that $F(s)$ and $N(s)$ should be relative prime (i.e., have no common roots). It is easy to see that the zeros of $N(s)$ should be in or near the stopband(s), while those of $F(s)$ should be in or near the passband. The relationship between the three polynomials is concisely expressed by the celebrated Feldtkeller's equation:

$$D(s)D(-s) = F(s)F(-s) + N(s)N(-s) \tag{8}$$

Given an arbitrary $F(s)$ and an even or odd $N(s)$, one can easily find $D(s)$ such that all of its roots are in the left half of the s plane.

Butterworth Filters. Butterworth filters are one of the oldest and simplest solutions to the filter problem. The characteristic function for lowpass filters can be written as

$$\kappa(s) = \epsilon(s/\omega_\mathrm{p})^n \tag{9}$$

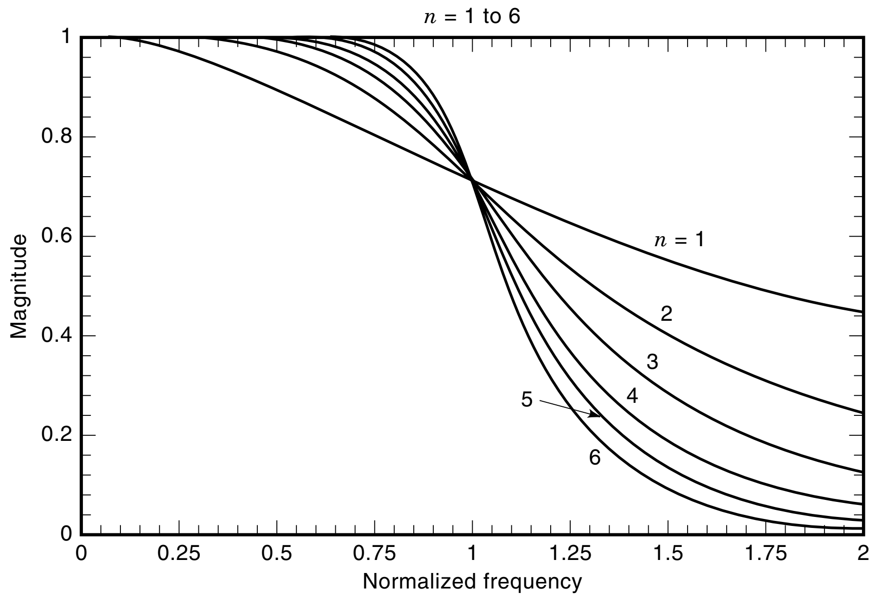


Figure 1. Butterworth transfer function.

Here ω_p is a normalization frequency, usually the passband edge. This filter type will have a maximally flat passband and a stopband loss that is monotonically increasing as we move away from the passband. The magnitude of the first few functions for $n = 1$ to 6 are shown in Fig. 1. The selection of the parameters, including the degree n, for a specific set of requirements is nearly trivial. Assuming that a filter requires not more than a_p loss (in dB) up to the frequency ω_p and a_s loss from ω_s to infinity, we compute

$$L = \frac{10^{0.1a_p} - 1}{10^{0.1a_s} - 1} \quad \text{and} \quad n \geq \frac{\ln(L)}{2\ln(\omega_p/\omega_s)}; \quad \epsilon^2 = 10^{0.1a_p} - 1 \tag{10}$$

The resulting transfer function poles can be computed in closed form, and so can the actual element values implementing this filter (although we shall not deal with that part of the design). The poles can be computed as follows:

$$1 + \kappa(s)\kappa(-s) = 1 + (-1)^n\epsilon^2(s/\omega_p)^{2n} = 0 \quad \text{which yields}$$
$$(s/\omega_p)^{2n} = (-1)^{n+1}/\epsilon^2 = e^{j\pi(n+1+2k)}/\epsilon^2$$

Hence, assuming $\epsilon = 1$,

$$\begin{aligned} s_k &= \omega_p e^{j\pi(n+1+2k)/2n} \\ &= \omega_p\left[\cos\frac{\pi(n+1+2k)}{2n} + j\sin\frac{\pi(n+1+2k)}{2n}\right] \end{aligned} \tag{11}$$

and those inside the left half plane are the poles we need.

For other than lowpass filter types, we use the well-known frequency transformation procedure by replacing the normalized frequency s/ω_p by

$$\begin{aligned} &\omega_p/s && \text{for high-pass filters} \\ &(s^2+\omega_0^2)/\delta s && \text{for bandpass filters and} \\ &\delta s/(s^2+\omega_0^2) && \text{for band-reject filters} \end{aligned} \tag{12}$$

In the latter two expressions $\omega_0 = (\omega_A\omega_B)^{1/2}$ is the center frequency of the pass (stop) band and $\delta = (\omega_B - \omega_A)$ is the pass (stop) band width. A bit more general bandpass case could be obtained using the characteristic function

$$\kappa(s) = k_0(s^2+\omega_0^2)^n/s^m \quad \text{with} \quad 0 < m < 2n \tag{13}$$

but this does not yield closed-form solutions for the transfer function poles and will be treated later under the numerical approximation methods. As an example, Fig. 2 shows a sixth-order filter with 40% bandwidth and $m = 6$ (the value we get with the preceding transformation) as well as $m = 3$. The second case, which puts three transmission zeros at zero frequency and nine zeros at infinity, yields a much more symmetrical response.

Chebyshev Filters. Chebyshev filters have the low-pass characteristic function

$$\kappa(s) = \epsilon T_n(s/\omega_p) = \epsilon\cosh[n\cosh^{-1}(s/\omega_p)] \tag{14}$$

where T_n is a polynomial that is varying between ± 1 in the passband ($s = j\omega$, $0 < \omega < \omega_p$) and ϵ determines the passband ripple a_p as before:

$$a_p = 10\log_{10}(1+\epsilon^2)$$

The stopband is monotonic, and if we need a loss a_s at frequency ω_s, then the necessary degree may be computed from

$$n \geq \frac{\cosh^{-1}L^{-1}}{\cosh^{-1}(\omega_s/\omega_p)} \quad \text{where} \quad \cosh^{-1}(x) = \ln(x+\sqrt{x^2-1}) \tag{15}$$

and L is given by Eq. (10). Fig. 3 shows the computed response of a few low-pass filters with $n = 1$ to 6 and about 1 dB passband ripple. For other filter types, the transformations of Eq. (12) are used again. In the bandpass case, the characteristic function will have nth order poles at both zero and infinite frequencies; a more general case would distribute these unequally.

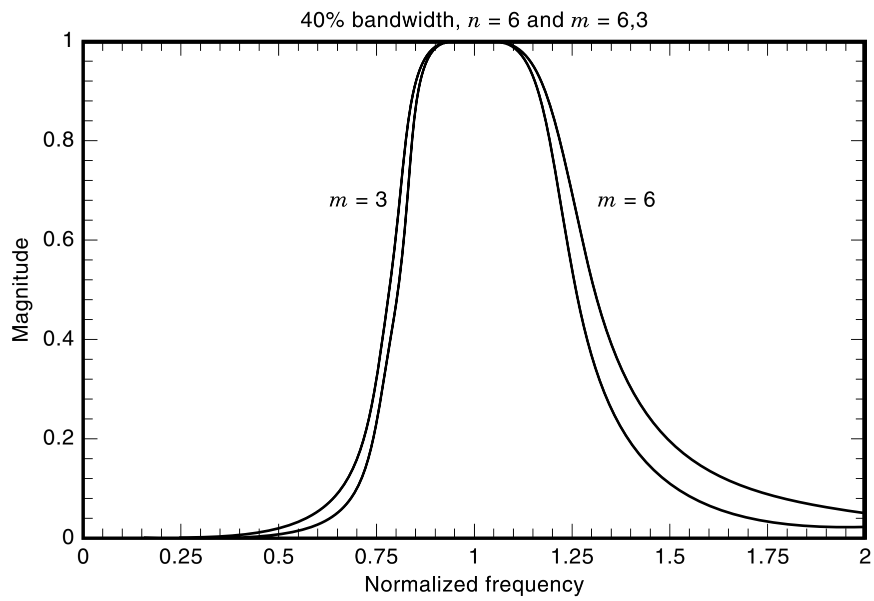


Figure 2. Butterworth bandpass function.

The resulting transfer function singularities can be found explicitly again, and so can the element values needed to implement the filter. For the poles, we can write

$$\cosh[n\cosh^{-1}(s/\omega_p)] = \pm j/\epsilon$$

and therefore

$$\begin{aligned} n\cosh^{-1}(s/\omega_p) &= \cosh^{-1}(j/\epsilon) \\ &= \ln\left(\frac{1}{\epsilon} + \sqrt{1+\frac{1}{\epsilon^2}}\right) + j\pi(1+2k)/2 \\ &= \sinh^{-1}(1/\epsilon) + j\pi(1+2k)/2 \end{aligned}$$

and consequently

$$\begin{aligned} s_k/\omega_p &= \cosh\{(1/n)[\sinh^{-1}(1/\epsilon) + j\pi(1+2k)/2]\} \\ &= \cos\frac{\pi(1+2k)}{2n}\cosh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right) \\ &\quad \pm j\sin\frac{\pi(1+2k)}{2n}\sinh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right) \end{aligned} \tag{16}$$

Inverse Chebyshev Filters. Inverse Chebyshev filters are obtained simply by using the characteristic function

$$\kappa(s) = k_0/T_n(\omega_s/s) \tag{17}$$

for lowpass filters. We note that this function will vary be-

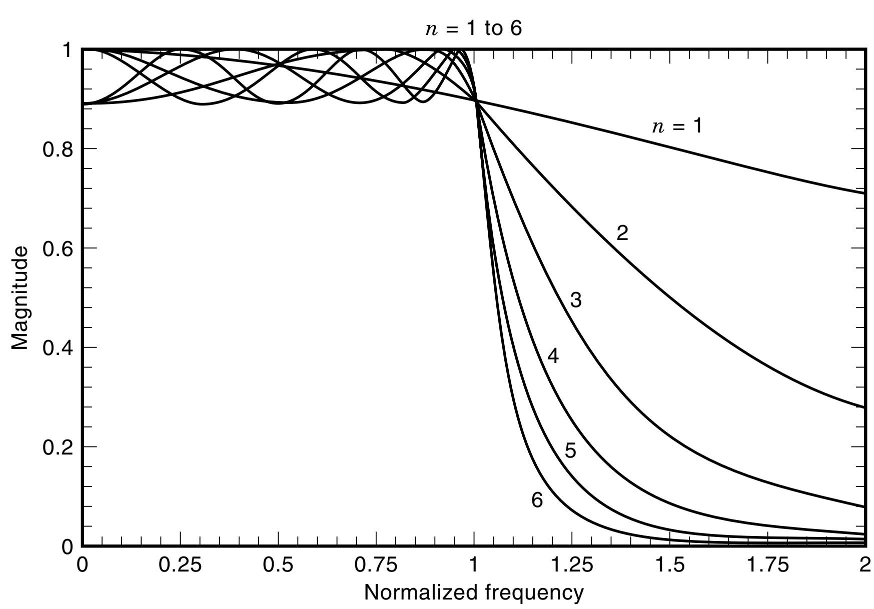


Figure 3. Chebyshev transfer function.

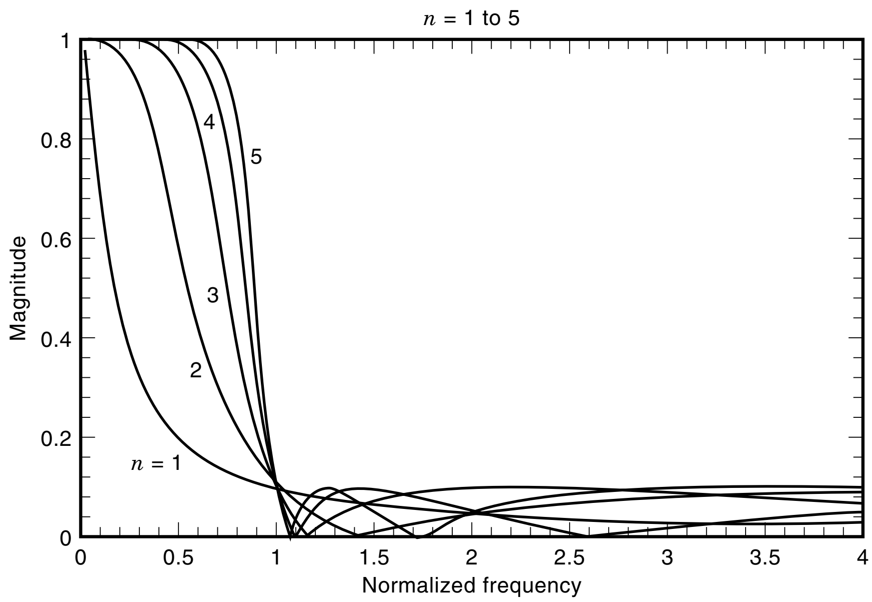


Figure 4. Inverse Chebyshev transfer function.

tween k_0 and ∞ in the frequency range $\omega_s < \omega < \infty$, which is therefore the stopband. The passband will be maximally flat. Fig. 4 displays the magnitude of a few low-pass filters with inverse Chebyshev characteristics. These were designed for a stopband loss of about 20 dB and degrees 1 through 5. If we need a passband loss not more than a_p up to frequency ω_p and a stopband loss of at least a_s, the necessary degree can be computed from exactly the same expression as in the Chebyshev case, except that k_0 is given by

$$k_0 = \sqrt{10^{0.1a_s} - 1} \tag{18}$$

The element values of the *RLC* implementation can no longer be expressed explicitly, especially since multiple implementations exist.

Elliptic (Cauer) Filters. If the filter loss requirements are uniform in both the passband and stopband(s), the most efficient design is obtained by the use of the Jacobian elliptic functions. The corresponding characteristic function for a low-pass is given by

$$\kappa(j\Omega) = \epsilon\, cd(nuK_1, k_1) \quad \text{where} \quad \Omega = cd(uK, k) \tag{19}$$

and where $cd(x, k)$ is one of the Jacobian elliptic functions (2) of parameter k. K and K_1 are the complete elliptic integrals belonging to k and k_1, respectively, while K' and K_1' are the same and belong to the parameters $k' = \sqrt{1-k^2}$ and $k_1' = \sqrt{1-k_1^2}$, respectively. These parameters are defined as follows:

$$k = (\omega_p/\omega_s) \quad \text{and} \quad k_1^2 = \frac{10^{a_p/10} - 1}{10^{a_s/10} - 1} = L \tag{20}$$

and, furthermore, the following condition must be satisfied:

$$\frac{nK'}{K} = \frac{K_1'}{K_1} \tag{21}$$

which can be used to determine the necessary degree n of the filter. The complete elliptic integrals may be easily computed using the method of arithmetic-geometric mean (2). These expressions correspond to our usual normalization $\Omega = \omega/\omega_p$; other normalizations yield slightly different expressions.

The function in Eq. (19) yields a normalized rational fraction of the form

$$\begin{aligned} \kappa(s) &= \epsilon \prod_{j=1}^{n/2} \frac{s^2 + \Omega_{zj}^2}{1 + \Omega_{pj}^2 s^2} \quad \text{where} \\ \Omega_{zj} &= cd\left(\frac{(2j-1)K}{n}, k\right) \quad \text{and} \quad \Omega_{pj} = k\Omega_{zj} \end{aligned} \tag{22}$$

The poles and zeros are at inverse locations with respect to the halfway point in the transition band. The preceding expression is for the even degree case; for the odd case, the upper limit on the product is only $(n-1)/2$ and there is an extra s multiplier in front. The odd degree case is directly usable, but in the even degree case the loss will be finite and nonzero at both zero and infinite frequencies. If that is not acceptable, a simple frequency transformation

$$s^2 \to \frac{s^2}{1 - \Omega_{p1}^2 s^2} \quad \text{or} \quad s^2 \to s^2 - \Omega_{z1}^2$$

or a combination, where Ω_{p1} and Ω_{z1} are the lowest of the values, may be used to shift the highest pole to infinity or the lowest zero to zero, respectively, but with an attendant increase in the transition bandwidth (3). Note also that the natural modes can again be calculated in closed form, but this is usually ignored, since the computation of these poles will need extensive numerical computation in any case and therefore direct root extraction methods are just as convenient. The Ω_j values of Eq. (22), can be readily computed by using the ascending Landen transformation (2), which converts the elliptic functions into hyperbolic functions, or the descending one, which converts the elliptic functions into circular ones. A particularly detailed description of elliptic functions in the design of filters is available in Ref. 4. Fig. 5 shows the magnitude of an elliptic low-pass transfer function of degree 7, with 10% transition bandwidth. These functions are not easy to

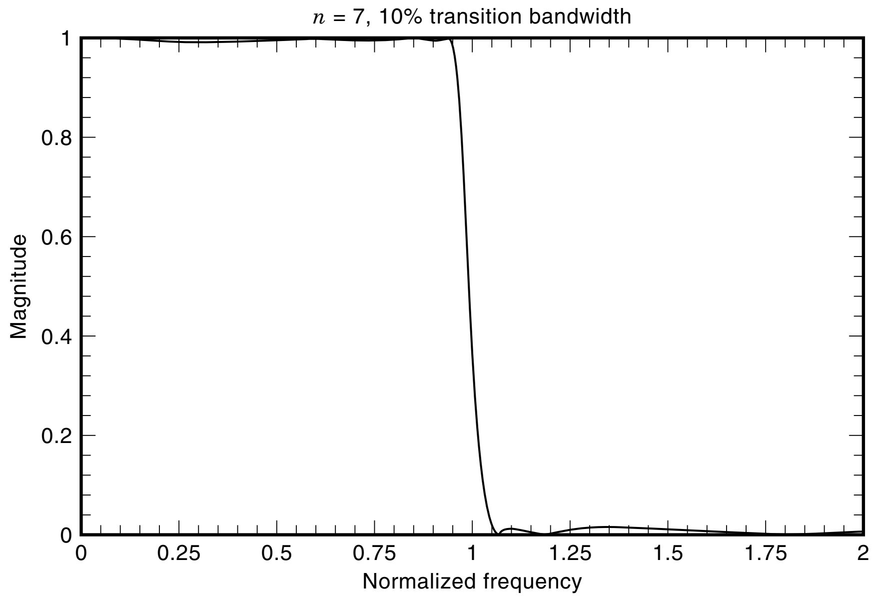


Figure 5. Elliptic transfer function.

compute, and if one has no access to some filter design software (5), then many tables of Butterworth, Chebyshev, and elliptic transfer functions (and element values) can be found, the most extensive being that in Ref. 6, followed closely by those in Refs. 7 and 8. Inverse Chebyshev functions are tabulated in Ref. 8.

Rather than using Eq. (21), we may calculate the necessary degree for a set of filter specifications by the (approximate, but very accurate) closed-form expressions

$$\begin{aligned} &\epsilon_1 \cong \frac{L}{16}\left(1+\frac{L}{2}\right); \quad \epsilon_2 = \frac{1}{2}\frac{1-k^{1/2}}{1+k^{1/2}} \quad \text{and} \\ &n \geq f(\epsilon_1)f(\epsilon_2) \quad \text{where} \\ &f(\epsilon) \cong (1/\pi)\ln(\epsilon + 2\epsilon^5 + 15\epsilon^9 + 150\epsilon^{13}) \end{aligned} \tag{23}$$

Since ϵ_2, L, and consequently ϵ_1 are usually very small, and therefore we hardly ever need more than the first term in the expansion of either ϵ_1 or $f(\epsilon)$, these expressions can be rearranged easily in several ways to be able to compute any of the four quantities a_p, a_s, $\omega_p/\omega_s = k$ and n, if the other three are specified.

Functions for other filter types may be easily generated by the familiar frequency transformation method. Note, however, that in the bandpass case, this approximation usually does not yield optimal performance. For that, the iterative method described later is preferable.

The Approximation of Delay

Bessel Filters (Maximally Flat Delay). The nth degree Bessel polynomial is defined by the recursion formula

$$\begin{aligned} B_n(s) &= (2n-1)B_{n-1}(s) + s^2 B_{n-2}(s) \\ &= b_0 + b_1 s + b_2 s^2 + \cdots + b_n s^n \end{aligned} \tag{24}$$

with starting points $B_0(s) = 1$ and $B_1(s) = 1 + s$.

The transfer function $H(s) = b_0/B_n(s)$ can be shown to provide a delay function that is maximally flat at zero frequency (i.e., the first n derivatives of the delay with respect to the frequency are all zero; see Ref. 9). The value of the delay at zero frequency is $\tau_0 = n/\omega_0$, where ω_0 is the normalization frequency. Fig. 6 shows the magnitude of the Bessel transfer function for degrees 1 through 6, and the corresponding delay curves are shown in Fig. 7. These functions were all normalized to $\tau_0 = 1$. Tables of Bessel polynomials can be found in many texts (see Ref. 10, for instance).

These characteristics can be combined with an equal-minima type stopband, using the technique of Temes and Gyi (11) (see also Ref. 12) described in detail in Appendix A.

As shown in Fig. 6, the resulting filters have an increasing loss in the passband; therefore it would be desirable to combine this delay with a flat passband of specified flatness. Consider the general low-pass transfer function $H(s) = N(s)/D(s)$, where $D(s)$ is given and we wish to select an even $N(s)$ such that the passband (i.e., the region around $\omega = 0$) is flat. If $D(s)$ is of the form

$$D(s) = d_0 + d_1 s + d_2 s^2 + d_3 s^3 + \cdots + d_n s^n \tag{25}$$

then we can generate the polynomial

$$G(s) = D(s)D(-s) = g_0 + g_1 s^2 + g_2 s^4 + \cdots + g_n s^{2n} \tag{26}$$

where the coefficients can be computed using

$$g_j = \sum_{k=0}^{2j} (-1)^k d_k d_{2j-k} \tag{27}$$

Next we compute the square root of this function:

$$M(s) = (G(s))^{1/2} = m_0 + m_1 s^2 + m_2 s^4 + \cdots + m_i s^{2i} + \cdots \tag{28}$$

where the m_i coefficients can be computed recursively as

$$m_0 = (g_0)^{1/2} \quad \text{and} \quad m_i = \frac{1}{2}\left(g_i - \sum_{k=1}^{i-1} m_k m_{i-k}\right) \tag{29}$$

Truncating this infinite series to a polynomial of degree less than that of $D(s)$ will yield the required numerator $N(s)$. Fig. 8 shows both the delay and the loss characteristics of a sev-

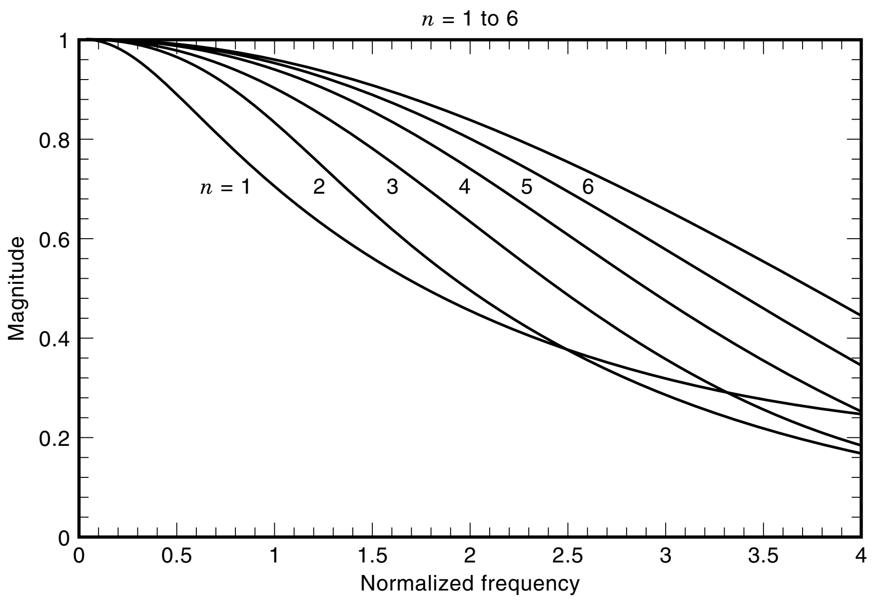


Figure 6. Bessel (linear phase) transfer function.

enth-order Bessel (maximally flat delay) transfer function. Curve a shows the loss when the numerator is a constant. Curve b illustrates the case when we introduce a fourth-order numerator to flatten the passband using the aforementioned procedure. Finally, curve c is what we obtain by the use of the Temes–Gyi procedure, when the stopband starts at the normalized frequency of 0.5.

Rhodes (13) has provided another way of combining flat delay and flat magnitude in a low-pass filter. His expression for the overall transfer function for odd degrees is as follows:

$$H(s) = \frac{Ev\{B_n(-s)[2B_{n+1}(s) - B_n(s)]\}}{B_n(s)[2B_{n+1}(s) - B_n(s)]} \tag{30}$$

where $B_n(s)$ is the nth order Bessel polynomial and the overall degree will be $2n + 1$ and $Ev\{. \ . \ .\}$ designates the even part of the polynomial inside the curly brackets. For the derivation and the even degree case, refer to the literature. As a comparison, Fig. 9 shows a ninth-degree filter designed by the preceding equation, compared to the ninth-degree case obtained by the method described previously, both having an eighth-order numerator. The Rhodes design has a somewhat steeper stopband but cannot exchange passband flatness for stopband selectivity.

Maximally Flat Delay for Digital and Microwave Filters. One cannot use the Bessel polynomials for the design of digital or microwave filters because of the frequency transformation of Eq. (2), which will negate the flat delay. However, Thiran (14) has developed a set of polynomials for generating the equivalent behavior in digital filters (see also Ref. 15). He derived the transfer function in terms of the variable $z = e^{j\omega}$ as

$$\begin{aligned} H(z) &= \frac{H_0}{\sum_{k=0}^{n} b_k z^{-k}} \quad \text{where} \\ b_k &= (-1)^k \frac{n!}{k!(n-k)!} \prod_{i=0}^{n} \frac{2\tau + i}{2\tau + k + i} \end{aligned} \tag{31}$$

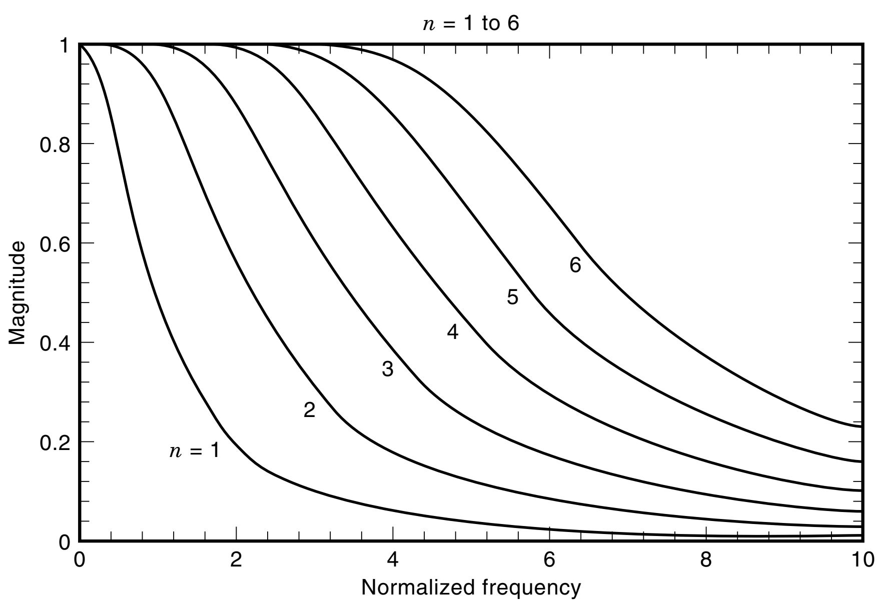


Figure 7. Bessel (linear phase) transfer function.

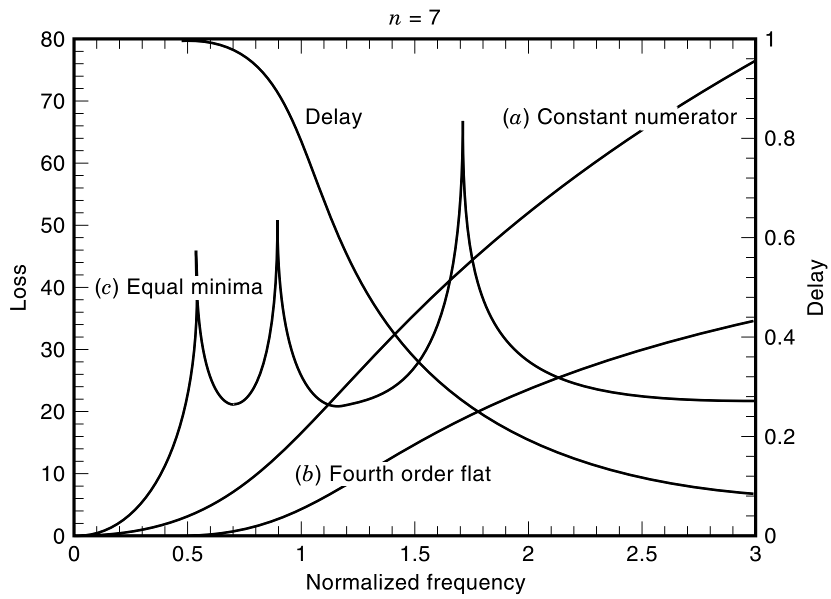


Figure 8. Bessel transfer function with various numerators.

and where the delay at zero frequency is $\tau_0 = \tau t_0$, τ being an integer and t_0 the sampling time. The disadvantage of this procedure is that the delay can only be set to discrete values. The value of H_0 is selected to set the loss at zero frequency to zero, yielding

$$H_0 = \sum_{k=0}^{n} b_k = \frac{(2n)!}{n!} \frac{1}{\prod_{i=n+1}^{2n}(2\tau + i)} \tag{32}$$

An example of this transfer function is shown in Fig. 10, which displays the loss and the delay of a ninth-order function with a delay of five samples. Note that this filter will have a finite loss at half the sampling frequency due to the constant numerator. We can, of course, introduce an arbitrary numerator as long as it is a symmetric or antimetric polynomial in z, without affecting the maximally flat delay property of the filter, except that this adds another $t_0 n/2$ flat delay, where n is the degree of the selected numerator. We may select this polynomial to provide either an equal-minima type stopband using the Temes–Gyi procedure, or a flat passband using the procedure outlined previously for the Bessel polynomial case. The way to do this is to apply the *inverse* bilinear z transform first:

$$z = \frac{1 - st_0/2}{1 + st_0/2} \tag{33}$$

where t_0 is again the sampling time, and use the resulting numerator as the starting polynomial $D(s)$ in Eq. (25) or in the Temes–Gyi procedure. Once we have the proper numera-

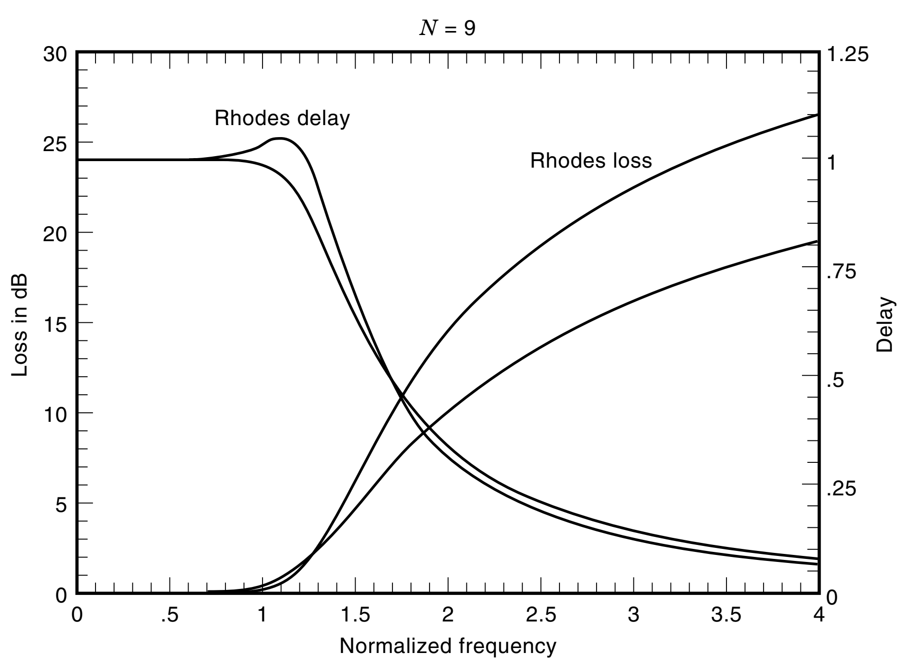


Figure 9. Low-pass with both flat loss and flat delay.

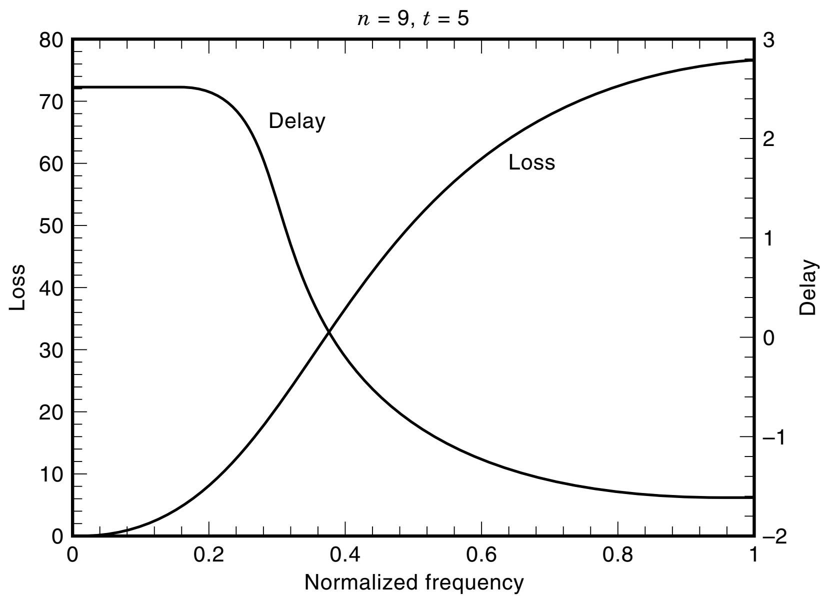


Figure 10. Maximally flat delay digital filter.

tor, we can return to the z domain using the standard bilinear z transform. Fig. 11 shows the same ninth-order denominator, combined with three different numerators. One (curve a) has a numerator with all zeros at $z = -1$ (the Nyquist rate), the next (curve b) with only five zeros there and four zeros computed to make the passband flat, and finally the third (curve c) with a numerator to provide an equal-minima type stopband from 0.15 in normalized frequency. Note that the delay (also shown) is now 9.5 (=5 + 9/2) times the sampling time.

This procedure applies equally well for the design of microwave filters with maximally flat delay, except that t_0 here is one-quarter of the inverse of the quarter-wave frequency. Furthermore, this flat delay may be combined with an equal-minima type stopband or a flat passband, exactly the same way as in the digital filter case; the only difference is that the delay will now be *independent* of the numerator. As opposed to analog filters, in the digital or microwave case, one may also obtain flat delay for *high-pass* filters, which in the microwave case are also bandpasses. The way to do this is to invert the singularities of a low-pass filter by changing the signs of the real parts of the poles and zeros in the z domain. For instance, doing that to the basic filter displayed in Fig. 11, we get the high-pass shown in Fig. 12.

Thiran has also formulated the problem for obtaining *equal-ripple* type delay in digital low-pass filters (16), but the equations presented have to be solved iteratively since no closed-form solution is known.

Mainly as a curiosity, we must also mention that there is a class of microwave filters with *exactly* linear phase. This is true for a transfer function of the form (17):

$$H(s) = \frac{N(s)}{(1+s)^d} \tag{34}$$

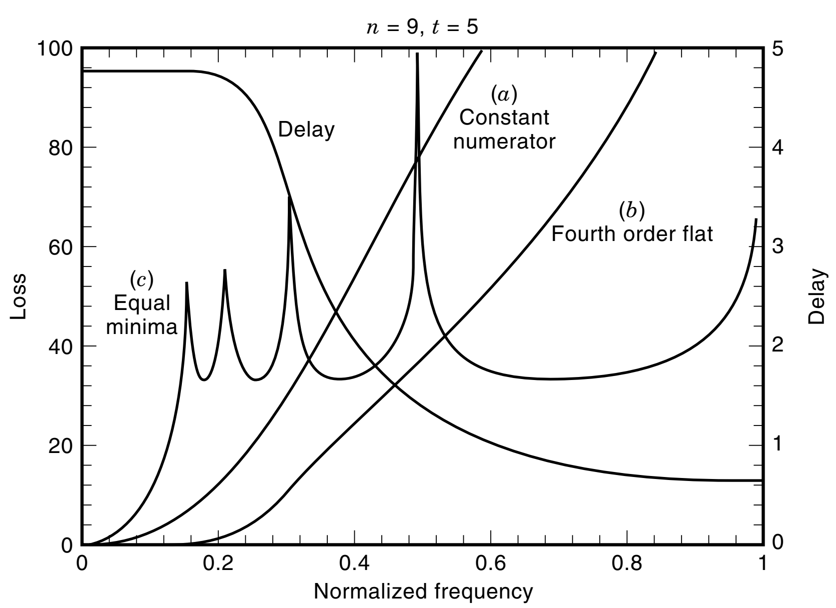


Figure 11. Maximally-flat delay digital filter with various numerators.

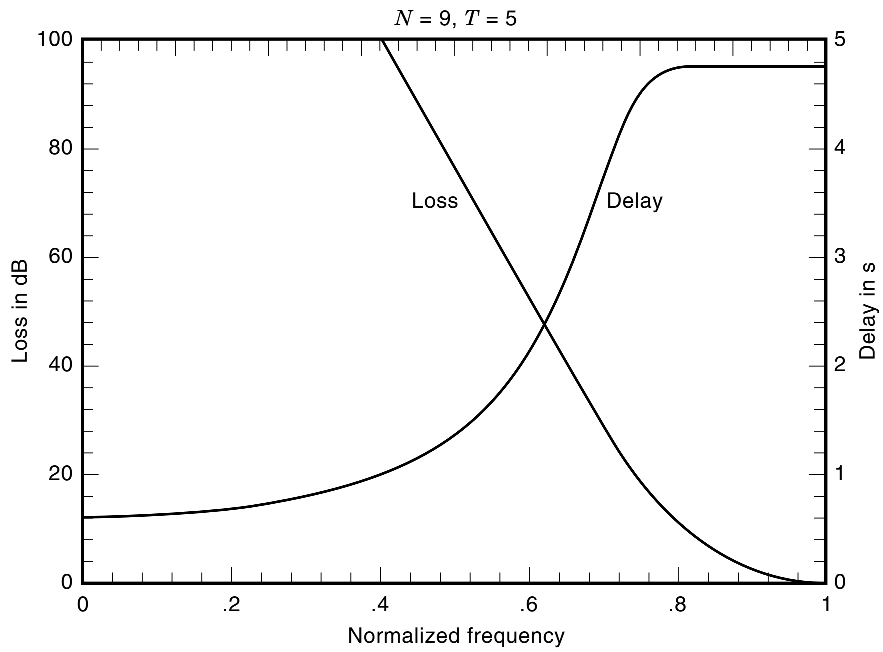


Figure 12. Maximally flat delay digital high-pass.

where $N(s)$ is an arbitrary even or odd polynomial of degree not more than d and s is as in Eq. (2). The value of the delay will be $\tau = dt_0/4$, where t_0 is the inverse of the quarter-wave frequency in Hz, as before. Again, $N(s)$ may be selected to provide either a flat passband or an equal-minima type stopband. If the circuit is selected to consist of d unit elements, then $N(s) = [\sqrt{(1-s^2)}]^d$, which will yield $|H(\omega)| \equiv 1$ and the circuit will consist of d unit elements, all of the same characteristic impedance, in cascade (which, of course, has constant delay). Other numerators can be used to provide flatter passband or equal-minima stopband. Also available are high-pass (actually bandpass) filters of various kinds. Fig. 13 shows the loss of four versions of a seventh-order filter with a flat delay of 1.75 s. The discrimination properties of these filters leave a lot to be desired. One interesting feature of this group of transfer functions is that converting them into digital form, using the bilinear z-transform method, they will become finite impulse response (FIR) filters, which helps to explain their constant delay property.

The Bessel polynomials as well as those developed by Thiran may also be used for the approximation of *delay lines* with maximally flat delay. This may be simply obtained by using a transfer function of the form

$$H(s) = \frac{B_n(-s)}{B_n(s)} \quad \text{or} \quad H(z) = \frac{z^{-n}B_n(z^{-1})}{B_n(z)} \tag{35}$$

where $B_n(s)$ is the nth degree Bessel polynomial and $B_n(z)$ the equivalent Thiran polynomial. The resulting delay at zero frequency will be twice that calculated previously, and the magnitude of $H(s)$ and $H(z)$ will be unity, of course, at real frequencies.

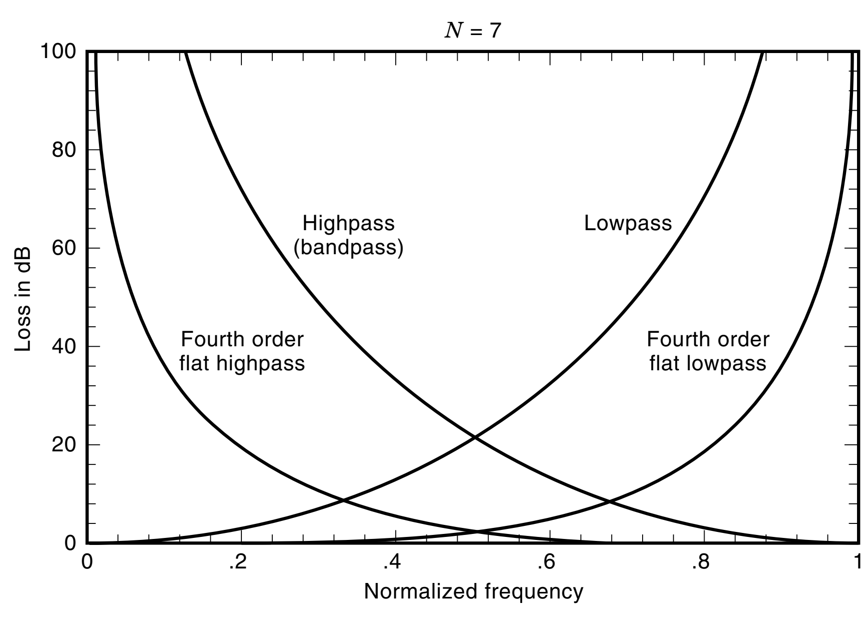


Figure 13. Constant delay microwave filter transfer functions.

ITERATIVE SOLUTIONS

All of the results presented so far are closed-form solutions (i.e., solutions that can be computed exactly in a finite number of steps). In many situations, we do not have closed-form solutions and must rely on iterative optimization procedures. We will find many different procedures useful in different circumstances. No general-purpose procedure has been found yet that can be applied to all problems with guaranteed success.

Flat Passband Loss

The most common requirement in the passband is a flat loss, and this may be approximated either in the maximally flat or the equal-ripple sense. This can be combined with the following types of approximation in the stopband(s):

1. Monotonically increasing loss as we move away from the passband. This combined with maximally flat passband is the familiar Butterworth characteristic. If we combine it with equal-ripple type passband, we have the equally familiar Chebyshev type of filter.
2. Equal-minima type stopband(s). This again can be combined with the maximally flat passband, which yields the inverse Chebyshev type filter; while combining it with the equal-ripple type passband leads to the elliptic (or Cauer) filter type.

All of these filter types have been treated previously.

3. A more general stopband type is the piecewise-constant loss specification. Again, this may be combined with either of the aforementioned passband characteristics, and a very rugged and fast converging approximation procedure is available for handling both cases (18).

To explain this procedure, we will use a change of variable to place the passband in evidence:

$$z^2 = (s^2 + \omega_A^2)/(s^2 + \omega_B^2) \tag{36}$$

where ω_A is the lower edge and ω_B is the upper edge of the passband, assuming a bandpass filter for generality. (This variable z should *not* be confused with the variable used in the digital filter design procedure. Unfortunately, the literature uses the same letter for both.) Low- and high-pass filters can be handled in an obvious manner; furthermore, in the case of digital or microwave filters, ω_A and ω_B will be replaced by $\Omega_A = \tan(\pi\omega_A/2\omega_0)$ and $\Omega_B = \tan(\pi\omega_B/2\omega_0)$, respectively.

From the preceding expression we can see that the variable z is pure imaginary in the passband and will vary between 0 and ∞, while it will be real for both the lower and the upper stopbands. In particular, it will vary from $\omega_A/\omega_B = \beta$ to 0 in the lower stopband and from ∞ to 1 in the upper stopband.

Now we are ready to form a function, first for the maximally flat passband:

$$\kappa(z) = \kappa_0 \frac{(z^2 + z_0^2)^{d/2}}{\prod_{j=1}^{d/2}(z^2 - z_j^2)} \tag{37}$$

This is an even rational fractional function in z and hence it is also an even rational one in terms of s when we substitute the expression for z to obtain the $\kappa(s)$ characteristic function. For a real z_0, it has a multiple zero inside the passband, and if all z_j values are also real, it has poles in the stopbands. The range $\beta < z < 1$ is off limits for the poles. We can select z_0 in such a manner that the function values at the ends of the passband (0 and ∞, respectively, in terms of z) are equal:

$$z_0 = \left\{ \prod_{j=1}^{\frac{d}{2}} z_j \right\}^{\frac{2}{d}} \tag{38}$$

Note that for low-pass or high-pass filters, z_0 may be selected to be zero or infinity, respectively, but need not. A finite z_0 will then provide a maximally flat point inside the passband; and the loss will be nonzero at zero or infinite frequencies respectively, yielding nonequal terminations ("matching filters"). Returning to the general bandpass case, the passband will be maximally flat and the stopband will have transmission zeros at the values specified by z_j:

$$\omega_j^2 = (\omega_A^2 - z_j^2\omega_B^2)/(1 - z_j^2) \tag{39}$$

Now it is very simple to modify the z_j values and the multiplier κ_0 to obtain the required stopband behavior.

This procedure yields an even degree $N(s)$; for the odd degree case we need to modify the $\kappa(z)$ function slightly. We have to replace one of the factors in the denominator by

$$\sqrt{(1 - z^2)(z^2 - \beta^2)} \tag{40}$$

and modify the value of z_0 accordingly:

$$z_0 = \beta^{\frac{1}{d}} \left\{ \prod_{j=1}^{\frac{d}{2}-1} z_j \right\}^{\frac{2}{d}} \tag{41}$$

This will yield the same overall even degree d, but the numerator polynomial $N(s)$ will be odd and of degree $d - 1$. Odd overall degree is also possible by the use of what is called *parametric* design and will be considered under that heading later.

Equal-Ripple Passband Loss

Let us now consider the equal-ripple type passband. We first recognize that the variable z is pure imaginary in the passband; hence the function

$$(z_j + z)/(z_j - z) \tag{42}$$

is of magnitude 1 (if z_j is real) and its phase varies from 0 to π as ω varies from ω_A to ω_B. Consequently, the function:

$$e^{j\varphi} = \prod_{j=1}^{d} \frac{(z_j + z)}{(z_j - z)} \tag{43}$$

will also be of unity magnitude and φ will vary from 0 to $d\pi$ in the passband. We can therefore form the function

$$\cos\varphi = \frac{1}{2}(e^{j\varphi} + e^{-j\varphi}) = \frac{\prod_{j=1}^{d}(z_j + z)^2 + \prod_{j=1}^{d}(z_j - z)^2}{\prod_{j=1}^{d}(z_j^2 - z^2)} \tag{44}$$

This function is going to vary between +1 and −1 in the passband, and it is an even rational function of z (the odd terms in the numerator cancel) and therefore will be an even rational function of s after substitution. The proper characteristic function therefore is

$$\kappa(z) = \epsilon \cos\varphi \tag{45}$$

where ϵ is a constant, determining the passband loss ripple. Again, the degrees d and n are equal and even; for odd degree n, we must modify $e^{j\varphi}$ by replacing one of the factors by

$$\sqrt{\frac{(1+z)(z+\beta)}{(1-z)(z-\beta)}} \tag{46}$$

It is clear that φ will still vary between 0 and $d\pi$ in the passband, and once we substitute z as per Eq. (36) above, the resulting $\kappa(\omega)$ will still have an even numerator of degree d and now an odd denominator of degree $n = d - 1$. For a microwave filter that contains u unit elements, we must further include the factor

$$\left(\frac{z_u + z}{z_u - z}\right)^{\frac{u}{2}} \tag{47}$$

where z_u is given by

$$z_u^2 = \frac{\Omega_A^2 + 1}{\Omega_B^2 + 1} \tag{48}$$

in order to have a factor $(\sqrt{1-s^2})^u$ in $N(s)$, which is necessary for the implementation of unit elements. Note that the value of z_u is between β and 1 [i.e., in the previously forbidden region and that for the purpose of the computation of the loss, we can replace $\sqrt{1-s^2}$ by $(1 + s)$]. Since in the stopband(s) the variable z is real, we introduce the new variable for the purpose of computing the loss:

$$\gamma = \ln z = \ln[(\omega^2 - \omega_A^2)/(\omega^2 - \omega_B^2)] \tag{49}$$

and since the loss is given by

$$a = 10\log_{10}(1 + \kappa^2(\omega)) = 4.343\ln(1 + \kappa^2(\omega)) \tag{50}$$

and

$$e^{\psi} = \prod_{i=1}^{d} \coth\frac{\gamma_i - \gamma}{2} \tag{51}$$

where $\psi = j\varphi$, therefore we obtain

$$\kappa(\omega) = \epsilon\cosh\psi = \frac{\epsilon}{2}\left(\prod_{i=1}^{d}\coth\frac{\gamma_i - \gamma}{2} + \prod_{i=1}^{d}\tanh\frac{\gamma_i - \gamma}{2}\right) \tag{52}$$

and since we can usually neglect the 1 next to the characteristic function, we can have

$$a \cong 8.686\ln\epsilon + 8.686\ln\cosh\psi \tag{53}$$

Next we realize that in the stopband ψ is usually large, and we can approximate $\cosh(\psi)$ by $e^{|\psi|}/2$, and consequently to a very good approximation [this approximation is not necessary; we can simply solve Eq. (53) for ψ]

$$\begin{aligned} a &\cong 8.686\ln(\epsilon/2) + 8.686\psi \\ &\cong 8.686\ln(\epsilon/2) + 8.686\sum_{i=1}^{d}\ln\left(\coth\left|\frac{\gamma_i - \gamma}{2}\right|\right) \end{aligned} \tag{54}$$

The loss is therefore given as the sum of a number of identical functions that are just shifted along the stopband (γ) axis. The resulting characteristic function will be of an even degree for general bandpass filters. Odd overall degree is also available by the use of *parametric* design techniques and will be considered later under that heading.

The z_i values represent the variable (free) transmission zero locations, although any number of them may be fixed by the designer. Our job is to determine how many of these variable zeros we need and where to put them. We first plot the stopband requirements as a function of the variable γ. The first objective is reached by replacing these frequency-dependent stopband requirements by an averaged constant requirement, which can be satisfied by an elliptic type design. From the closed-form solution of this problem we can get an estimate of the required number of zeros, from which we subtract the number of fixed zeros and then distribute the additional ones as uniformly as possible over the stopband(s).

Next we locate the loss minima between any two consecutive transmission zeros (including fixed zeros) and evaluate the loss at all breakpoints (where the requirement changes) and subtract the required loss values from all of these. This yields a short list of frequencies and excess loss values. If there is a minimum as well as a breakpoint(s) between two zeros, we discard the pair with the larger excess loss value until we have only one pair between the zeros. Next we consider the fixed poles. If there are minima on both sides of the fixed pole, we discard the one with the higher excess loss. Note that those zeros we explicitly put to zero or infinite frequencies ($z_i = \beta$ or 1, respectively) are fixed by definition, but they are not counted here, since the region between them is of no interest. This way, the number of items in this list of excess loss minima is reduced to the number of movable zeros (plus one in the bandpass case).

If the remaining excess loss values (all but one in the bandpass case) are all equal and positive (an error of 0.5 dB is usually acceptable), we are done and the approximation converged. Otherwise, we average the excess loss values and compute the deviation from this average and denote it by Δa_k. The actual iteration is performed by first computing the derivative of the loss at each of these frequencies ($\partial a_k/\partial\gamma_i$), with respect to the parameters of the variable zeros z_i. Finally, we solve the approximate equations for the necessary changes $\Delta\gamma_i$ in these parameters as follows:

$$\sum_i \frac{\partial a_k}{\partial \gamma_i}\Delta\gamma_i = -\Delta a_k \quad k = 1, 2, \ldots \tag{55}$$

Such a routine has been described by Smith and Temes in their classic paper (18) and in a slightly modified form to handle piecewise linear requirements by Bell (19). Note that the simplicity of the expression makes it easy to compute the derivatives that are necessary for the optimization. This procedure has been found to be fast and accurate, hardly ever needing more than 10 iterations to converge, and, of course, has been further generalized to handle multiple zeros, the digital, microwave (perhaps containing unit elements), and parametric filter cases and their combinations as well. Another extension, described in the literature (20), permits the program to exchange excess loss for wider passband automatically.

Returning to the maximally flat passband case [Eq. (37)], that expression is simple enough to be handled directly, although for uniformity, the new variable $\gamma = \ln z$ is usually introduced there also. For the details, we refer to the book by Daniels (21).

Parametric Bandpass Filters. As mentioned previously, bandpass filters designed by the methods outlined always turn out to be of even degree. In some instances it would be desirable to have an odd degree filter, which means a characteristic function with an odd degree numerator (i.e., a root on the real s axis). Also, for generating an LC structure with the absolute minimum number of inductors, we often need a characteristic function numerator with *two* real axis zeros. The explanation of this fact will have to wait until the article on the LC implementation of bandpass filters.

In any case, in the equal-ripple type passband approximation procedure, both of these objectives can be achieved (22) by the introduction of another factor

$$\sqrt{\frac{\alpha - z}{\alpha + z}} \quad \text{or} \quad \frac{\alpha - z}{\alpha + z} \tag{56}$$

in Eq. (43), where α is again in the forbidden zone:

$$\beta < \alpha < 1$$

Since its exact value is of minor importance, we usually select $\alpha = \beta^{1/2}$. Let us consider now the effect of this additional factor. First note that these factors have the difference terms in their numerator, not in the denominator, as all the others. Let us consider the second case first, where we must also have an odd multiplicity of transmission zeros at both zero and infinite frequencies [i.e., we have the factor of Eq. (46) in the definition of the characteristic function]. This characteristic function will now have a factor $(\alpha^2 - z^2)$ in the denominator, while the numerator can be written in the form

$$\begin{aligned}(\alpha - z)^2(1+z)(\beta+z)\prod_j(z_j+z)^2 \\ + (\alpha+z)^2(1-z)(\beta-z)\prod_j(z_j-z)^2\end{aligned} \tag{57}$$

We can see that the second term will be negative between $\beta < z < 1$, while the first term is nonnegative there but has a double zero at $z = \alpha$ inside this range. Consequently, the function is negative at $z = \alpha$ but positive at $z = \beta$ and $z = 1$; that is, it must have two zeros, a_1 and a_2 such that

$$\beta < a_1 < \alpha < a_2 < 1$$

Furthermore, since the complete numerator will be an even function of z, it must, in fact, contain the factors: $(a_1^2 - z^2)(a_2^2 - z^2)$. At this stage, we can simplify the factors in the transfer function by writing

$$\frac{(a_1^2 - z^2)(a_2^2 - z^2)}{\alpha^2 - z^2} \cong \left(\frac{a_1^2 a_2^2}{\alpha^2} - z^2\right) \tag{58}$$

which gives us the two (in fact, a double) real roots in the s domain. The resulting characteristic function yields a filter that is called *even parametric.* To find out how much this approximation is going to affect the equal-ripple property of the transfer function, let us express the relative error in the passband, where z is pure imaginary ($z = jy$):

$$\begin{aligned}\text{error} &= \frac{(\alpha^2 + y^2)\left(\dfrac{a_1^2 a_2^2}{\alpha^2} + y^2\right)}{(a_1^2 + y^2)(a_2^2 + y^2)} - 1 \\ &= \frac{\left(\alpha^2 + \dfrac{a_1^2 a_2^2}{\alpha^2} - a_1^2 - a_2^2\right) y^2}{(a_1^2 + y^2)(a_2^2 + y^2)}\end{aligned} \tag{59}$$

This function has a maximum at $y^2 = a_1 a_2$ and therefore

$$\text{max error} = \left[\frac{\alpha^2 + a_1 a_2}{\alpha(a_1 + a_2)}\right]^2 - 1 \tag{60}$$

Let us consider a very wide passband ($\beta = 0.25$) filter with $\alpha = 0.5$ and three transmission zeros at both zero and infinite frequencies and no finite zeros. For this simple filter we calculate

$$a_1 = 0.4754975 \quad \text{and} \quad a_2 = 0.5225556$$

which gives

$$\text{max error} = 0.002084$$

Assuming a passband ripple of 0.5 dB, this introduces an error that is less than 0.001 dB. This filter can be implemented using only two inductors and four capacitors, instead of the three inductors and three capacitors needed for the nonparametric case. For more complex filters and narrower passbands, the error will further decrease rapidly, because the values of a_1, a_2 and α get closer and closer.

To get the odd parametric case, we introduce the square-root factor specified previously and note that one of the multiplicities of the transmission zeros at zero and infinity must be odd, the other even. Again expanding the characteristic function, we see that the denominator will contain the factor $\sqrt{\alpha^2 - z^2}$, while the numerator can be written in the form

$$\begin{aligned}(\alpha - z)(\beta+z)^{n_z}(1+z)^{n_i}\prod_j(z_j+z)^2 \\ + (\alpha+z)(\beta-z)^{n_z}(1-z)^{n_i}\prod_j(z_j-z)^2\end{aligned} \tag{61}$$

where one of n_i and n_z is odd, the other even. If n_z is odd, then the second term will be negative in the range $\beta < z < 1$ and zero at the boundaries, while the first term changes sign at $z = \alpha$. Consequently, the sum is positive at $z = \beta$ and negative at $z = 1$; hence it must have a zero in between, say at a. Furthermore, since the numerator will be an even function of z, it must therefore have a factor $(a^2 - z^2)$. A similar result can be observed if n_i is odd. Now if we combine this numerator factor with the irrational parts of the denominator, we get, if n_z is odd, the factor

$$\frac{(a^2 - z^2)}{\sqrt{(\alpha^2 - z^2)(\beta^2 - z^2)}} \tag{62}$$

and, if n_i is odd, the factor

$$\frac{(a^2 - z^2)}{\sqrt{(\alpha^2 - z^2)(1 - z^2)}} \tag{63}$$

Substituting Eq. (36), we get, in terms of the variable s, after some rearrangement and saving for a multiplier,

$$\frac{A - s^2}{s\sqrt{B^2 - s^2}}$$

in the first case and

$$\frac{A' - s^2}{\sqrt{B'^2 - s^2}}$$

in the second case, where A and B are both positive and very close and so are A' and B'. Now we can again simplify:

$$\frac{A - s^2}{\sqrt{B^2 - s^2}} \cong \frac{A}{B} - s \tag{64}$$

similarly for the other case, yielding the real root we need in the characteristic function. Again the approximation is very accurate and gets better with increasing degree.

For the maximally flat case, the situation is much simpler; we may just modify the function to read

$$\kappa(z) = \kappa_0 \frac{(z^2 + z_0^2)^{d/2-1}(z^2 - a^2)}{\prod_{j=1}^{d/2}(z^2 - z_j^2)} \tag{65}$$

and get an even parametric case, where a is again in the forbidden region, and just introduce a linear factor in terms of s for the odd parametric case.

The algorithm to locate the movable transmission zeros can readily be modified to handle these parametric cases as well.

Finally, for completeness we may mention that an even parametric approximation is possible if the multiplicities of the zeros at zero and infinite frequencies are both even, but the resulting transfer function turns out to be useless.

Low-*Q* Approximations. All the standard approximating functions (Butterworth, Chebyshev, or elliptic) have some of their poles (those closest to the passband edges) too close to the imaginary ($j\omega$) axis, which may cause difficulties in active RC implementations. This can be alleviated by replacing the two or three poles closest to the imaginary axis by a multiple pole with multiplicity two or three, which will not be quite so close to the axis. To maintain the nature of the approximation, the other parameters must, naturally, be readjusted also. This needs an iterative approach, which is of no particular interest to us here except to mention that the resulting functions have been extensively tabulated in Refs. 23–25.

Arbitrary Loss Shape

When we come to the question of completely arbitrary loss shapes, we have fewer tools to simplify the problem. We must go back to the original equation [Eq. (1)] and deal with that, preferably in factored form. This will provide us with substantially better control over numerical accuracy and a direct control over the question of stability. Here we are basically reduced to the two possibilities of the least pth, or the minimax optimization procedures of classical optimization theory. There are some other methods (the Pade method comes to mind as an example), but none of them has been found to be of general use.

The basic options we have here depend on our choice of error function. The most often cited such error function we must minimize is of the form

$$E = (1 - \lambda)\sum_{\omega_i} w(\omega_i)(H_\mathrm{r}(\omega_i) - H(\omega_i))^p + \lambda \sum_{\omega_i} v(\omega_i)(\tau_\mathrm{r}(\omega_i) - \tau(\omega_i))^p \tag{66}$$

where $H_\mathrm{r}(\omega)$ is the required transfer function magnitude and $H(\omega)$ is the actual magnitude, $\tau_\mathrm{r}(\omega)$ is the required delay (to make the approximation more general), $\tau(\omega)$ is the actual delay, $w(\omega)$ and $v(\omega)$ are the (user-specified) loss and delay weights, respectively, and λ is a parameter allocating the error of the loss and the delay. The (even) parameter p is also a user-selected parameter, controlling the approximation. Some people prefer to work with loss, rather than transfer function magnitude, but then the stopband causes problems since if the actual loss is much higher than the required one, that adds to the error, while in fact it is quite acceptable. Consequently, if the loss is greater than the required value, we usually set the corresponding $w(\omega)$ weight to be zero. The problem with this choice is that it makes the error a nonanalytic function, causing problems with the iteration methods one uses.

The other, minimax, method can only handle one quantity at a time (i.e., either the loss or the delay). Here we must minimize the function

$$\max w(\omega_i)|H_\mathrm{r}(\omega_i) - H(\omega_i)| \tag{67}$$

or a similar one for the delay, over the whole frequency range. See Appendix B for a brief description of several of the more useful optimization strategies in use today for this problem.

Constant Delay Approximation

Consider first the problem of constant delay. The maximally flat approximation has a closed-form solution, which was described previously. Equal-ripple approximation of a constant delay is possible, using the following approximation technique.

There is a closed-form solution to the problem of interpolating a linear phase function at equidistant points (13). Assuming that the points are multiples of the step ϵ and the phase is required to be proportional to ω, the interpolating polynomials can be obtained by the recursion formula:

$$P_{n+1}(s,\epsilon) = P_n(s,\epsilon) + \left(\frac{\tan\epsilon}{\epsilon}\right)\frac{(s^2 + (\epsilon n)^2)}{(2n+1)(2n-1)}P_{n-1}(s,\epsilon) \tag{68}$$

with initial conditions $P_0(s, \epsilon) = 1$ and $P_1(s, \epsilon) = 1 + (\tan \epsilon/\epsilon)s$. The value of ϵ is restricted to $\epsilon < \pi/2$ by stability considerations. If one plots this type of approximation, it is clear that while the delay will not be equal ripple, it will definitely have the correct number of extrema, and these are going to be close to the interpolation points. Consequently, it is a relatively easy matter of locating these extrema and then using an iterative procedure to make them all equal. Such a procedure has been implemented and takes very few (three to five) iterations to converge. The resulting polynomial will be the denominator $D(s)$ of the transfer function, and we can still select the numerator to shape the loss in the pass- or stopband. In particular, both the equal minima solution of Temes-Gyi as well as the flat passband solution, outlined for the maximally flat case, are available. Fig. 14 shows a seventh-degree equal-ripple delay function over the frequency range from 0 to 7.5, normalized to unity average passband delay. This can be combined with a constant numerator, or a fourth-degree numerator yielding flat passband loss, obtained by the procedure outlined previously. Finally, the third possibility is a sixth-degree numerator yielding an equal-minima type stopband from normalized frequency 6.5 calculated using the Temes–Gyi procedure and providing a minimum loss of about 48.9 dB, all shown in Fig. 15.

The polynomials of Eq. (68) can, of course, be used directly, since the delay deviation from a constant, while not exactly equal ripple, will be found satisfactory in most cases. Everything that we have said about equal-ripple delay functions will work equally well with these polynomials. Additional methods of simultaneous approximation of linear (or arbitrary) phase and flat magnitude may be found in Ref. 13.

Naturally, this procedure works for low-pass filters, but for bandpasses, there is no closed-form interpolating polynomial available, and due to the arbitrary intercept point of the linear phase line, we have another variable to be concerned about. An approximate procedure for bandwidths of about 25% or less is to shift the low-pass poles and zeros by the amount $\pm\omega_0$ parallel to the $j\omega$ axis, where ω_0 is the center of the new bandpass filter. Clearly, all finite singularities of the lowpass must be smaller than $2\omega_0$. The number of zeros of the lowpass at infinity will be doubled and (if there is more than one) are split up such that the bandpass has about three times as many zeros at infinity as at zero. The resulting filter will have the same delay as the low-pass and, in addition, will be close to having an arithmetically symmetrical frequency response (17). A direct iterative approximation method has been described for equal-ripple delay bandpass design by Ulbrich and Piloty (see Ref. 26, which also contains tabulated results for low-pass and bandpass filters).

Delay Lines. As explained previously in the discussion of Bessel polynomials, these results may also be used to obtain a delay line (i.e., a transfer function with $|H| \equiv 1$) by simply using a numerator polynomial $N(s) = D(-s)$ yielding twice the delay.

Delay Equalization. Transfer functions of the form of Eq. (1) will have a phase characteristics that will usually be far from linear. This can be seen from the fact that *minimum phase* transfer functions (those that have no zeros in the right half of the s plane) have a unique relationship between loss and phase (27). This can be expressed in the form

$$\theta(\omega) = \frac{2\omega}{\pi} \int_0^\infty \frac{a(x)}{x^2 - \omega^2}\, dx \tag{69}$$

where $\theta(\omega)$ is the phase and $a(\omega)$ is the loss in nepers (named after Napier, the discoverer of the natural logarithm). That is, $a(\omega) = -\ln|H(j\omega)|$ and we take the principal value of the integral at the pole $x = \pm\omega$. For instance, if $a(\omega) = 0$ for $\omega < \omega_0$ and $a(\omega) = A$ elsewhere (an ideal low-pass filter that cannot be realized but may be approximated as closely as required), the corresponding minimum phase will be

$$\theta(\omega) = \frac{A}{\pi} \ln \frac{\omega + \omega_0}{|\omega - \omega_0|} \tag{70}$$

and, consequently, the delay is

$$\tau(\omega) = \frac{2A\omega_0}{\pi} \frac{1}{(\omega - \omega_0)^2} \tag{71}$$

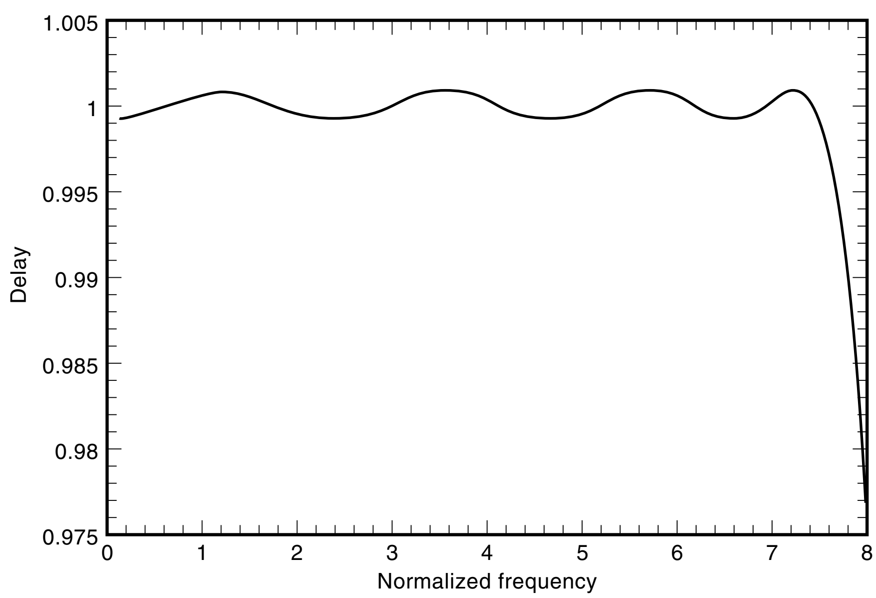


Figure 14. Equal-ripple approximation of constant delay.

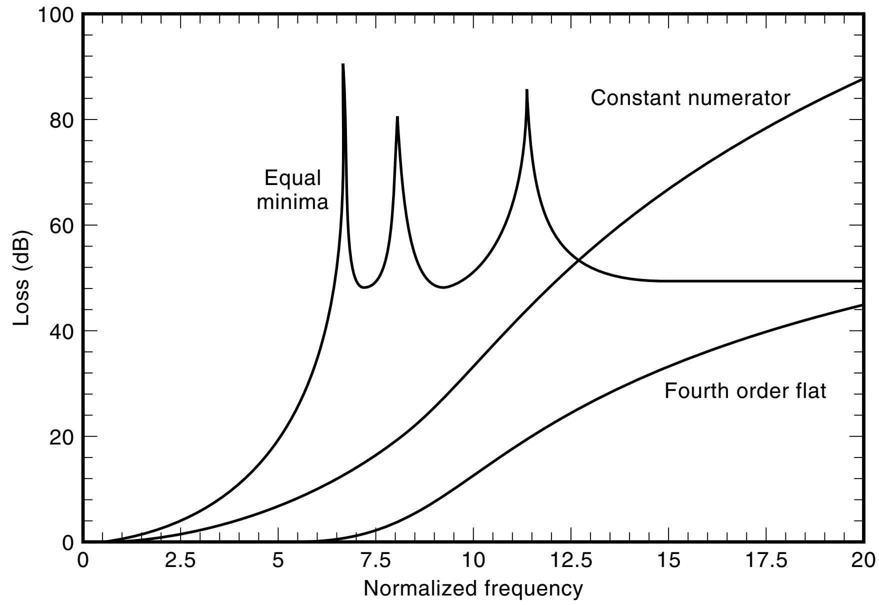


Figure 15. Equal-ripple delay transfer function with various numerators.

which is far from constant. Consequently, once the required loss characteristics have been achieved, the minimum delay is determined and our only choice left is to add additional circuitry to approximate the required delay (if, in fact, the delay must also meet certain criteria). When the need arises for the equalization of a computed or measured delay curve, we basically use the second half of Eq. (66) to define our error function:

$$E = \sum_{\omega_i} v(\omega_i)(\tau_r(\omega_i) - \tau(\omega_i))^p \tag{72}$$

where τ_r is the required delay and the summation goes over the frequency range of interest. We can usually select these at our convenience, but if we are dealing with measured delay, the measurements usually also specify the frequencies. The exponent p is going to control the nature of the approximation, and the weight function $v(\omega)$ is also ours to choose. This is a relatively simple procedure due to the relatively simple dependence of the delay curves on the quadratic coefficients, and the solution is routine (see, e.g., Ref. 5). Namely, the transfer function of a delay equalizer (allpass network) is most often written in the form

$$H(s) = \prod_k \frac{1 - a_k s + b_k s^2}{1 + a_k s + b_k s^2} \tag{73}$$

where all coefficients are positive and sometimes (in the case of a lowpass function) we may have a linear factor $(1 - a_0 s)/(1 + a_0 s)$ as well. The magnitude of H is unity for all frequencies, while the delay can be written as

$$\tau(\omega) = \left\langle \frac{2}{1 + (\omega/\omega_0)^2} \right\rangle + \sum_k \frac{2Q_k[1 + (\omega/\omega_k)^2]}{[1 - (\omega/\omega_k)^2]^2 + Q_k^2(\omega/\omega_k)^2} \tag{74}$$

where we have used the notation $a_0 = 1/\omega_0$, $a_k = Q_k/\omega_k$, and $b_k = 1/\omega_k^2$. We have plotted a few cases with various Q_k values in Fig. 16 to show their general behavior. The problem is to select the number of sections k to be used and the parameters Q_k and ω_k such that the delay of the equalizers added to the delay to be equalized is flat. A simple example shows the equalization of the delay of the seventh-order elliptic lowpass, with magnitude shown in Fig. 5. We have selected three second-order sections to equalize the delay over 90% of the passband, and the results are shown in Fig. 17. The curves shown are the original delay [fairly close to the one we would get from Eq. (71)], the delay of each of the three equalizer sections, and finally their sum as the equalized delay. The approximation was performed in the least squares sense, using Eq. (72) with $p = 2$ and unity weighting.

For digital filters, the exact form of the equations is somewhat different, since the general allpass function is of the form

$$H(z) = \left\langle \frac{\beta_0 + z^{-1}}{1 + \beta_0 z^{-1}} \right\rangle \prod_k \frac{\beta_{2k} + \beta_{1k} z^{-1} + z^{-2}}{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}} \tag{75}$$

If we designate the poles of a quadratic factor as $p_{1,2} = re^{\pm j\varphi}$ and the zeros as $z_{1,2} = r^{-1}e^{\pm j\varphi}$, where $0 < r < 1$, then the delay can be written as

$$\tau = \frac{r^2 - 1}{1 + 2r\cos(\omega - \varphi) + r^2} + \frac{r^2 - 1}{1 + 2r\cos(\omega + \varphi) + r^2} \tag{76}$$

for each of the second-order factors. For the linear factor, we just have one of the terms with $\varphi = 0$. While the specific equation is different from that of the analog system, the shapes of these delay curves, shown in Fig. 18 for $\varphi = 90°$ and various values of r, are quite similar, and so is the iterative procedure.

The minimax approximation to an arbitrary delay shape is also possible, since it is easy to generate a starting approximation that has the requisite number of extrema by selecting high enough values for the starting Q_k's. One approach has been described by Deczky in Ref. 28.

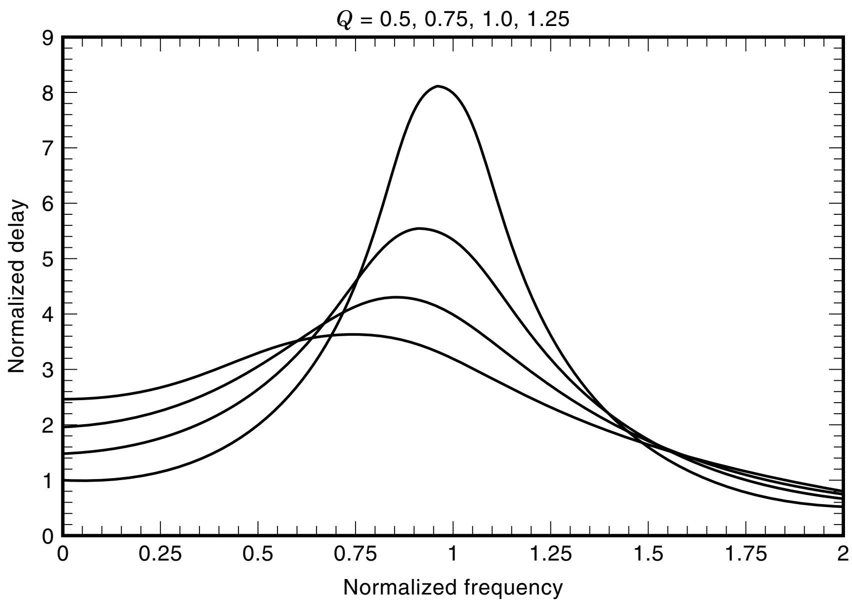


Figure 16. Delay of second-order delay sections.

APPROXIMATION OF FIR DIGITAL FILTERS

Formulation

FIR digital filters have to be treated differently than IIR filters. IIR filters have a rational transfer function and, as such, can be obtained from analog filter functions, as we have done previously. FIR filters, on the other hand, are represented by a polynomial in $z = e^{j\omega T}$ and have no analog equivalent (except the special case mentioned previously). The significant advantage of FIR filters over their IIR counterparts is that FIR filters may have exactly linear phase. This is easily observable if the coefficients of the polynomial have even or odd symmetry:

$$H(z) = \sum_{k=0}^{N} a_k z^{-k} \quad \text{with} \quad a_{N-k} = a_k \quad \text{or} \quad a_{N-k} = -a_k \tag{77}$$

Depending on the parity of N we have the following precise forms. For even N and symmetrical coefficients,

$$H(\omega) = e^{-j\omega NT/2}\left[a_{N/2} + \sum_{k=1}^{N/2} 2a_{k-1}\cos\omega T(k - N/2)\right] \tag{78a}$$

and for antimetrical coefficients

$$H(\omega) = e^{-j(\omega NT/2-\pi/2)}\sum_{k=1}^{N/2} 2a_{k-1}\sin\omega T(k - N/2) \tag{78b}$$

while if N is odd, we have

$$H(\omega) = e^{-j\omega NT/2}\sum_{k=0}^{(N-1)/2} 2a_k\cos\omega T(k - N/2) \tag{78c}$$

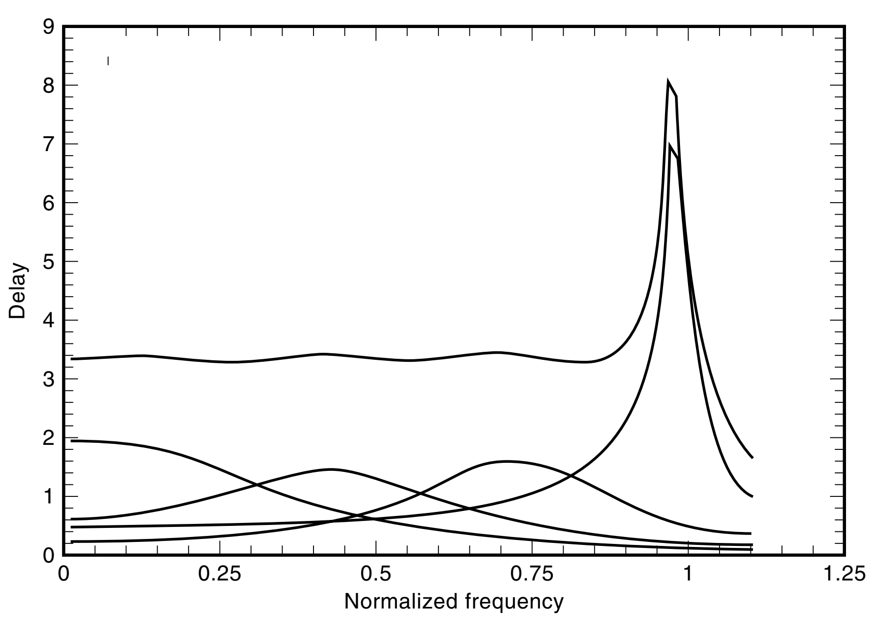


Figure 17. Delay equalization of low-pass using a three-section equalizer.

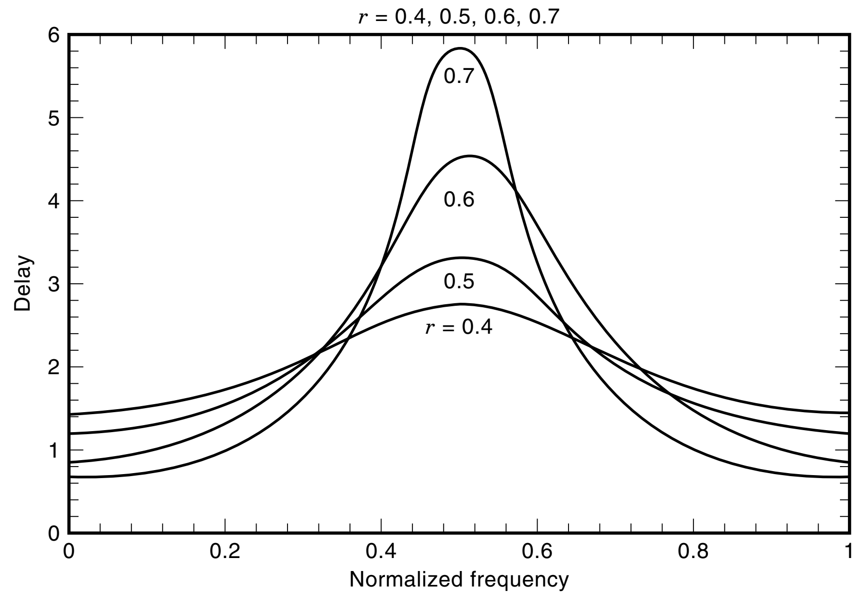


Figure 18. Delay of digital second-order delay sections.

for symmetrical coefficients and

$$H(\omega) = e^{-j(\omega NT/2-\pi/2)} \sum_{k=0}^{(N-1)/2} 2a_k \sin \omega T(k - N/2) \qquad (78d)$$

for antimetrical ones. Here T is the inverse of the sampling frequency, which we can simply set to unity as normalization. All of these have exactly linear phase and a delay of $NT/2$. Ignoring the phase terms for the time being, we see that all of these expressions are trigonometric series, and the last two of these contain terms of the form $\sin(n + \frac{1}{2})\omega$ or $\cos(n + \frac{1}{2})\omega$.

Using the following trigonometric identities recursively,

$$\begin{aligned} \sin(n\omega) &= \sin(\omega)\cos((n-1)\omega) + \cos(\omega)\sin((n-1)\omega) \\ \cos(n\omega + \omega/2) &= 2\cos(\omega/2)\cos(n\omega) - \cos(n\omega - \omega/2) \\ \sin(n\omega + \omega/2) &= 2\sin(\omega/2)\cos(n\omega) + \sin(n\omega - \omega/2) \end{aligned} \qquad (79)$$

in the last three of the preceding equations, we see that all four expressions can be represented in the general form (where we have ignored the phase factor):

$$H(\omega) = Q(\omega)P(\omega) \qquad (80)$$

where

$$P(\omega) = \sum_{k=0}^{M} \alpha_k \cos(k\omega) = \sum_{k=0}^{M} \beta_k \cos^k \omega \qquad (81)$$

and where $Q(\omega)$ is one of the four functions:

Case 1:	1	$M = N/2$
Case 2:	$\sin(\omega)$	$M = N/2 - 1$
Case 3:	$\cos(\omega/2)$	$M = (N-1)/2$
Case 4:	$\sin(\omega/2)$	$M = (N-3)/2$

Approximation

There are basically two methods of FIR filter design in use at the present time. One is the windowed design, and the other is the equal-ripple approximation method. The windowed method is not a true approximation technique; it is more a trial-and-error procedure and will be treated only briefly here.

Windowed Design. The ideal low-pass filter transfer function is of the form

$$\begin{aligned} H(s) &= 1 \quad \text{for} \quad 0 \le \omega \le \omega_c \\ &= 0 \quad \text{for} \quad \omega_c < \omega \le \pi \end{aligned} \qquad (82)$$

and has a corresponding impulse response

$$h_d(n) = \frac{\omega_c}{\pi}\left(\frac{\sin \omega_c n}{\omega_c n}\right) \quad \text{with} \quad h_d(0) = \frac{\omega_c}{\pi} \qquad (83)$$

which is, of course, of infinite length. We can, however, truncate it to a symmetrical set of finite length:

$$\begin{aligned} h(n) &= h_d(n) \quad \text{for} \quad |n| \le (N-1)/2 \\ &= 0 \quad \text{elsewhere} \end{aligned} \qquad (84)$$

which may be the coefficients of a (linear phase) FIR filter. One can, of course, determine the corresponding frequency response, but that is not our direct concern here. Suffice it to say that the resulting filter will have very limited stopband suppression, and increasing N will not help here due to the Gibbs phenomenon familiar from Fourier series theory. Many people have come up with ideas for shaping these coefficients in one way or another to alleviate this problem. This simply means using the modified coefficients

$$\tilde{h}(n) = h(n)w(n) \qquad (85)$$

where $w(n)$ is a "window" function and $h(n)$ are the coefficients specified previously. We have seen more than 30 different window functions proposed (29) and will mention here only a few. All equations are valid for odd N values; for even N they must be modified slightly.

The window function that does nothing [i.e., $w(n) \equiv 1$] is called the *rectangular* window. The *triangular* (or Bartlett) window is defined as

$$w(n) = 1 - \frac{|2n|}{N+1} \tag{86}$$

The *Hamming* window is

$$w(n) = 0.54 + 0.46 \cos \frac{2\pi n}{N-1} \tag{87}$$

The *Blackmann* window is

$$w(n) = 0.42 + 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{4\pi n}{N-1} \tag{88}$$

Finally, the *Hann* (raised cosine) window is

$$w(n) = \cos^2 \frac{\pi n}{N+1} \tag{89}$$

The last three are all examples of a large family of windows, all in the form of a cosine series. Many more are described in Ref. 29.

The *Kaiser* window is defined as

$$w(n) = \frac{I_0\left[\beta\sqrt{1-\left(\dfrac{2n}{N-1}\right)^2}\right]}{I_0(\beta)} \tag{90}$$

The value of β is related to the desired minimum stopband loss through the following empirical relationship. If the required stopband loss in dB is a_s, then

$$\beta = \begin{cases} 0.0 & \text{for } a_s < 21\,\text{dB} \\ 0.5824(a_s - 21)^{0.4} + 0.07886(a_s - 21) & \\ & \text{for } 21\,\text{dB} < a_s < 50\,\text{dB} \\ 0.1102(a_s - 8.7) & \text{for } a_s > 50\,\text{dB} \end{cases} \tag{91}$$

where $I_0(x)$ is the modified zeroth-order Bessel function and β is a selectable parameter.

The *Gaussian* window is

$$w(n) = \exp[-2(an/(N-1))^2] \tag{92}$$

where a is a selectable parameter.

For the *Chebyshev* (also called Dolph–Chebyshev) and *Taylor* windows, $w(n)$ will also be a cosine series, where the coefficients are calculated by evaluating the Chebyshev polynomial at N equidistant points along the unit circle and subsequently calculating its inverse discrete Fourier transform. This attempts to make all sidelobes to be about equal and of specified height. The equation that defines the weights is as follows:

$$\sum_{n=-M}^{M} \overline{w}(n)e^{-j\omega} = T_M[\gamma \cos \omega + (\gamma - 1)] \tag{93}$$

where $M = (N-1)/2$, $T_k(x)$ is the Chebyshev polynomial of order k, and

$$\gamma = \left(1 + \cos \frac{2\pi}{2M+1}\right) \Big/ \left(1 + \cos \frac{2\beta\pi}{2M+1}\right) \tag{94}$$

where β is an adjustable parameter. The $\overline{w}(n)$ coefficients are unscaled; they should be scaled by dividing them by $\overline{w}(0)$. The empirical relationship between the parameter β and the stopband loss a_s is given by

$$\begin{aligned} \beta &= 0.0000769(a_s)^2 + 0.0248a_s + 0.330 \quad \text{for } a_s \le 60\,\text{dB} \\ &= 0.0000104(a_s)^2 + 0.0328a_s + 0.079 \quad \text{for } a_s > 60\,\text{dB} \end{aligned} \tag{95}$$

The Taylor window is a simplified version of this that attempts to hold a subset of the sidelobes constant and permits the rest to decrease at 6 dB per octave. For the exact formulation of these and many other windows, please see the references.

For other than low-pass filters, one must appropriately modify the $h(n)$ function before applying the windowing. For instance, for a bandpass filter with passband from ω_A to ω_B, the ideal impulse response is

$$h_d(n) = \frac{1}{\pi n}[\sin(\omega_B n) - \sin(\omega_A n)] \quad \text{with} \quad h_d(0) = \frac{\omega_B - \omega_A}{\pi} \tag{96}$$

Figures 19(a) through 19(h) illustrate some of these windows. All the filters are 51 taps long low-pass filters with passband up to 0.4 times the Nyquist rate and, when possible, requesting 50 dB stopband rejection. For the Gaussian window, we selected $a = 3$. We may conclude from these figures and other studies that precise control of pass- *and* stopband properties is not possible with this method. Its major advantage is that any filter length may be easily obtained without computational problems.

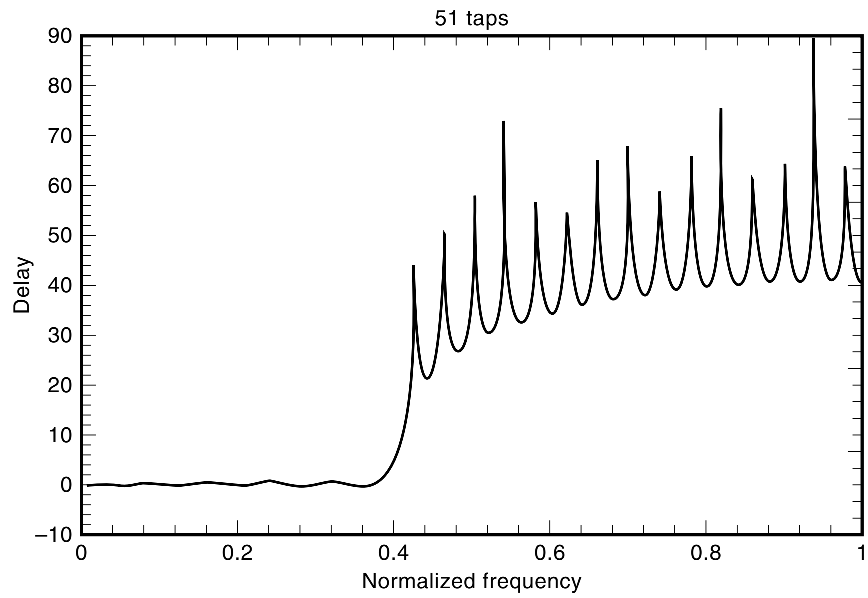
Remez Algorithm (Equal-Ripple Design). Returning to Eqs. (80) and (81) for the equal-ripple design, we have the unknown α_k coefficients of the trigonometric polynomial $P(\omega)$ to determine, and the best procedure for this purpose is the Remez exchange algorithm. The formulation of the problem is based on the error function

$$\begin{aligned} E(\omega) &= W(\omega)[H_r(\omega) - H(\omega)] = W(\omega)[H_r(\omega) - Q(\omega)P(\omega)] \\ &= W(\omega)Q(\omega)[H_r(\omega)/Q(\omega) - P(\omega)] \end{aligned} \tag{97}$$

where W is the usual weight function and P is the only unknown. Naturally, we must also select the case and therewith the Q function, especially since some filter types can only be implemented with some of the cases. For instance, lowpass filters may not be implemented in a case where Q is a sine function. Using the Remez algorithm, also described in Appendix B, the first step is to select a set of frequencies ω_i, one more than the number of free parameters in $P(\omega)$, and set the value of $E(\omega_k) = \pm\delta$ in an alternating manner. Solving for the trigonometric polynomial P, we obtain the expression

$$P(\omega_i) = \frac{H_r(\omega_i)}{Q(\omega_i)} \pm \frac{\delta}{W(\omega_i)Q(\omega_i)} = A_i \pm \delta B_i = C_i \tag{98}$$

at the selected frequencies, where the A_i and B_i values are known. Including the unknown deviation δ, we have the right number of equations for the right number of unknowns. In



(a)

Figure 19(a). Characteristics of a rectangular window.

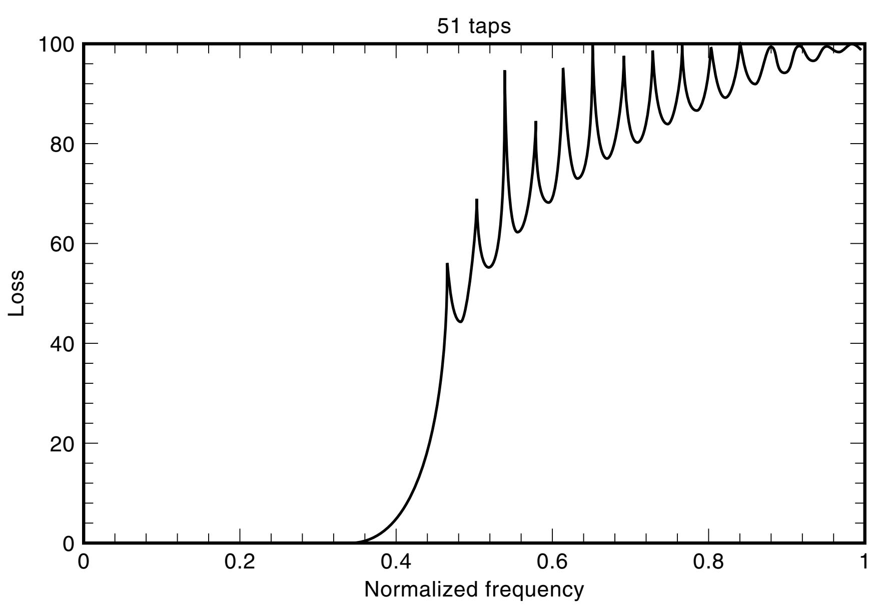
particular, these equations can be written in matrix form, using Eq. (81):

$$\begin{bmatrix} 1 & \cos\omega_1 & \cos^2\omega_1 & \cos^3\omega_1 \ldots & \cos^M\omega_1 & -B_1 \\ 1 & \cos\omega_2 & \cos^2\omega_2 & \cos^3\omega_2 \ldots & \cos^M\omega_2 & B_2 \\ \cdot & & & & & \\ \cdot & & & & & \\ 1 & \cos\omega_{M+1} & \cos^2\omega_{M+1} & \cos^3\omega_{M+1} \ldots & \cos^M\omega_{M+1} & (-1)^{M+1}B_{M+1} \\ 1 & \cos\omega_{M+2} & \cos^2\omega_{M+2} & \cos^3\omega_{M+2} \ldots & \cos^M\omega_{M+2} & (-1)^{M+2}B_{M+2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \\ \\ \beta_M \\ \delta \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \\ \\ A_{M+1} \\ A_{M+2} \end{bmatrix} \quad (99)$$

A similar matrix equation may be written for the α_k coefficients, if we replace the powers of cosine by the multiple angle forms of the cosine function. This linear set of $M + 2$ equations in $M + 2$ unknowns is not solved directly, because that would be time consuming. Instead, we first calculate δ, for which we can find a closed-form expression:

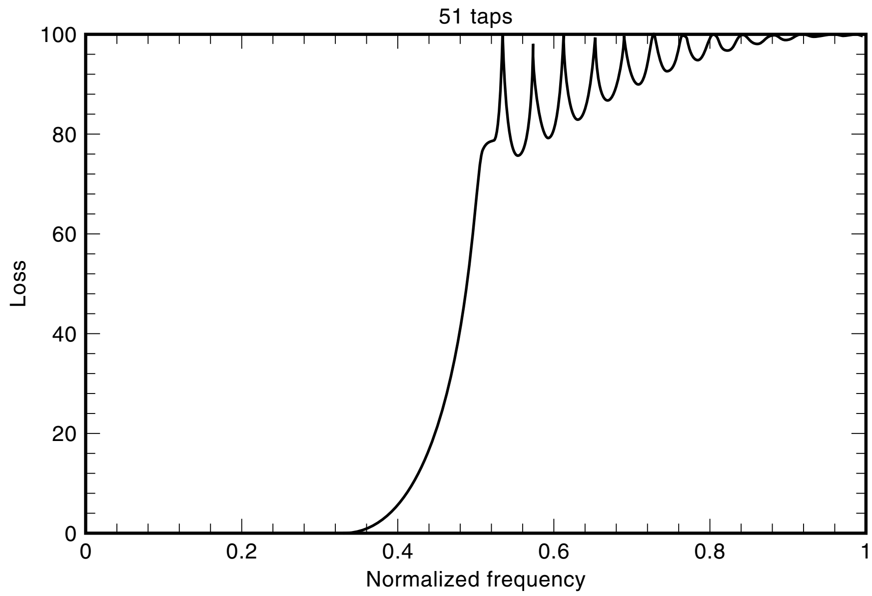
$$\delta = \frac{\sum_{k=0}^{M+2} q_k A_k}{\sum_{k=0}^{M+2} (-1)^k q_k B_k} \quad \text{where} \quad q_k = \prod_{i=0,\neq k}^{M+2} \frac{1}{\cos\,\omega_k - \cos\,\omega_i} \quad (100)$$

Once this is computed, the remaining equations can be obtained by deleting the last row and the last column from the preceding matrix equation and replacing the right side by the column containing $C_i = A_i \pm \delta B_i$. This forms an interpolation



(b)

Figure 19(b). Properties of a Hann window.



(c)

Figure 19(c). Behavior of a Blackman window.

problem, which can be solved again explicitly by Lagrange's method in an effective manner (30):

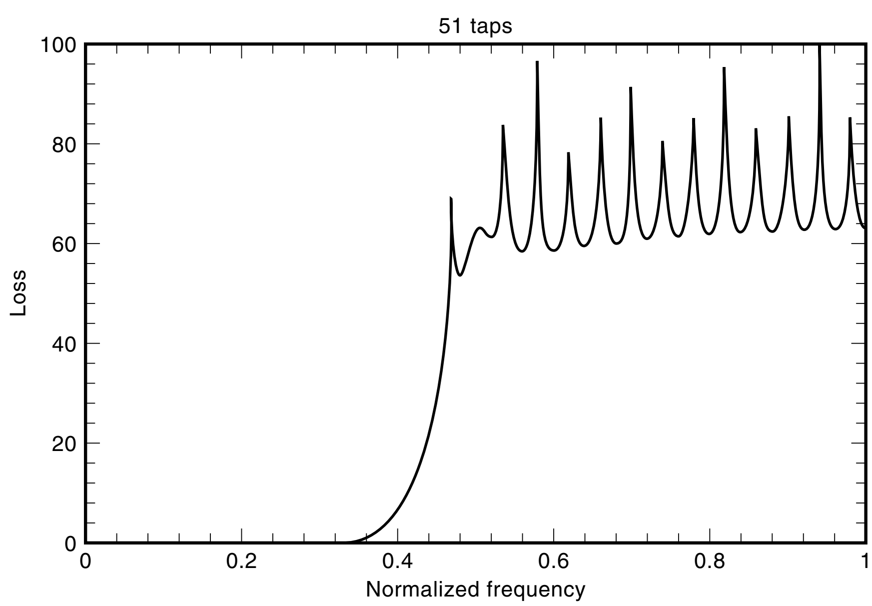
$$P(\omega) = \sum_{k=0}^{M+1} l_k(\omega) C_k$$

where $$l_k(\omega) = \frac{\prod_{i=0,\neq k}^{M+1}(\cos\,\omega - \cos\,\omega_i)}{\prod_{i=0,\neq k}^{M=1}(\cos\,\omega_k - \cos\,\omega_i)} \qquad (101)$$

Next we evaluate the function $E(\omega)$ on a dense set of frequencies to locate the true extrema ω_i' and replace the previous frequencies by these new ω_i' values. Repeating the process leads to the true minimax approximation in a very few steps.

Naturally, the procedure needs additional safeguards, especially concerning the treatment of extra ripples that may occur and, of course, the convergence criteria and numerical problems, if any. Nevertheless, a program has been available in the public domain for some time now (31) and produces excellent results. In this method there is no need to distinguish between low-pass, high-pass, or bandpass filters. Indeed, the procedure works for any number of pass- and stopbands. Also note that the requirements need *not* be flat; any specified shape can be accommodated.

Fig. 20 shows a 51 tap long low-pass filter designed by this method and requesting a passband up to 0.4 and a stopband from 0.475 to 1.0, the Nyquist rate. The filter has less than 1 dB passband loss ripple, and the minimum stopband loss is



(d)

Figure 19(d). Loss of a Hamming window.

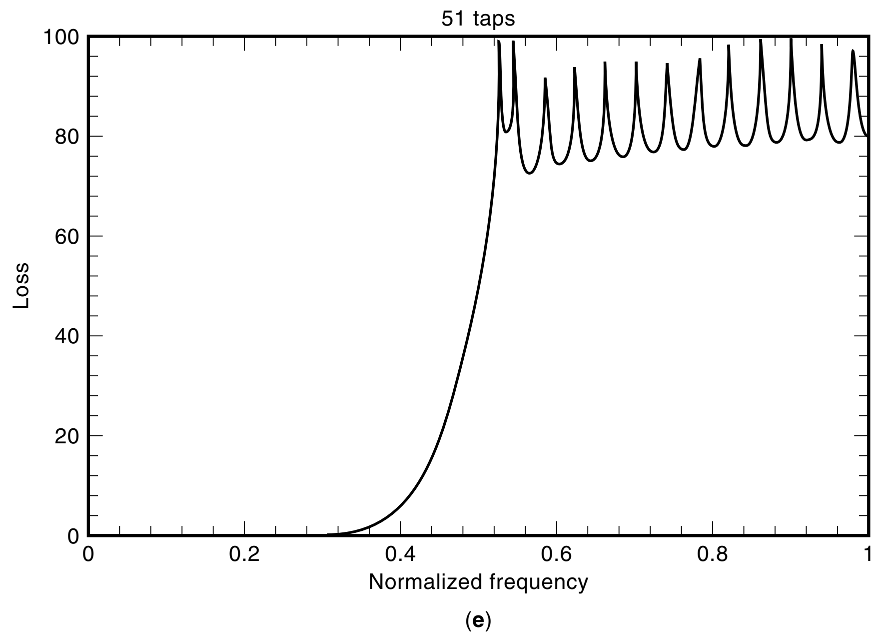


Figure 19(e). Loss shape of a Gaussian window.

about 51 dB. This design, of course, compares favorably with any of the windowed designs demonstrated before.

Least Squares Design. If we go back to Eq. (66), consider the loss only (since the phase *is* linear), and use the special case $p = 2$, we get

$$\begin{aligned} E &= \sum_{i=1}^{L} w(\omega_i)[H_r(\omega_i) - H(\omega_i)]^2 \\ &= \sum_{i=1}^{L} w(\omega_i)[H_r(\omega_i) - Q(\omega_i)P(\omega_i)]^2 \end{aligned} \tag{102}$$

where $H(\omega)$ is now given by Eqs. (80) and (81), we can see that E is a quadratic function of all the unknown coefficients α_k or β_k. The number of frequencies used in the summation must be $L \geq M + 1$ (i.e., the number of available free parameters).

To clarify the formulation of the problem, let us introduce the following vector-matrix notation. Let the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots \beta_M)^T$ be the unknown coefficient vector, F the $M + 1$ by L matrix:

$$\begin{bmatrix} 1 & \cos\omega_1 & \cos^2\omega_1 & \ldots & \cos^M\omega_1 \\ 1 & \cos\omega_2 & \cos^2\omega_2 & \ldots & \cos^M\omega_2 \\ \cdot & & & & \\ 1 & \cos\omega_L & \cos^2\omega_L & \ldots & \cos^M\omega_L \end{bmatrix} \tag{103}$$

H_d is the requirement vector, $H_d = (H_d(\omega_1), H_d(\omega_2), \ldots H_d(\omega_L))^T$, and finally Q and V are L by L diagonal matrices,

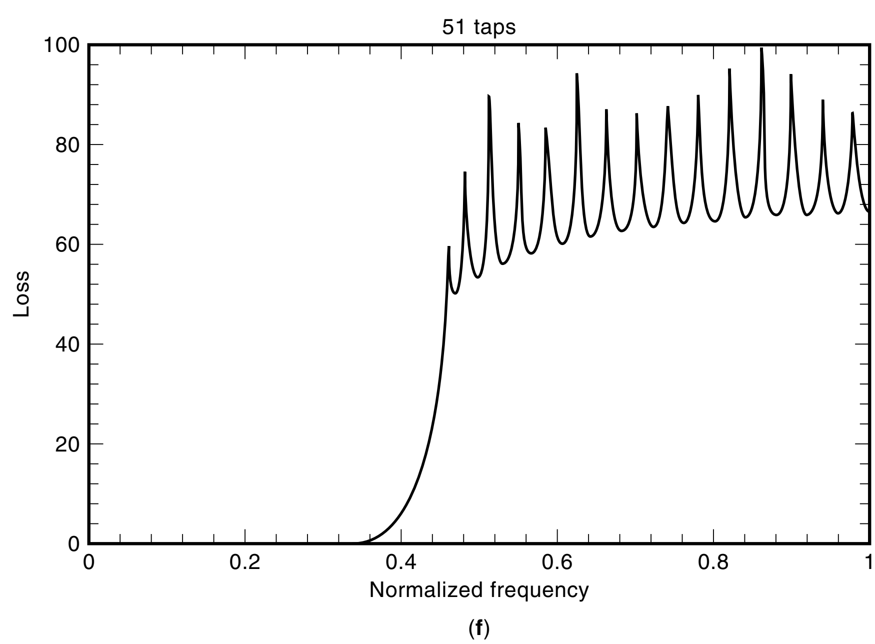
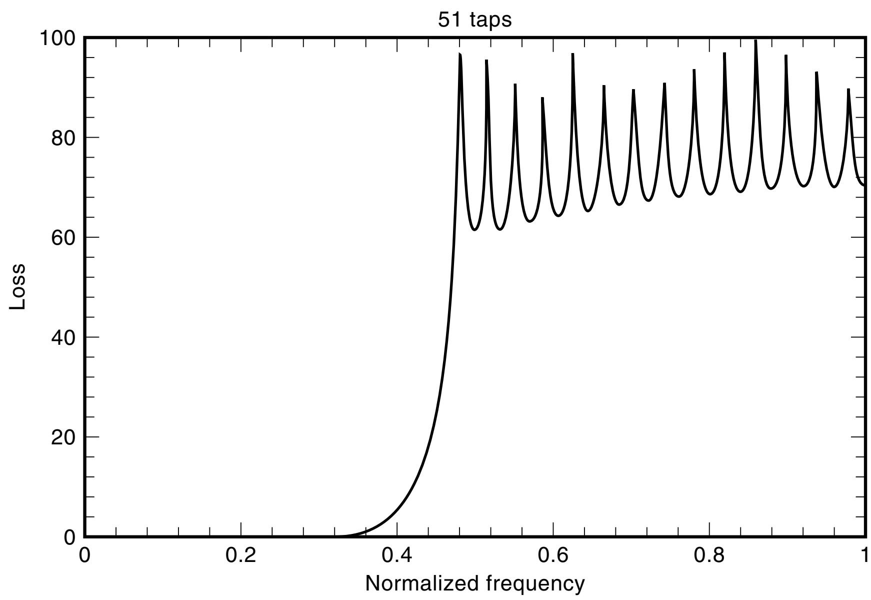


Figure 19(f). Behavior of a Kaiser window.



(g)

Figure 19(g). Properties of a Chebyshev window.

where the diagonal values are $Q(\omega_i)$ and $\sqrt{W(\omega_i)}$, respectively. With this notation, we can formulate an error vector of length L as follows:

$$\boldsymbol{e} = V(H_d - QF\beta) \tag{104}$$

and the total error is now given as $E = e^Te$.

If $L = M + 1$, then all matrices are square and the vector $\boldsymbol{e}$ can be set to zero and the unknown vector computed as (the weights are now immaterial)

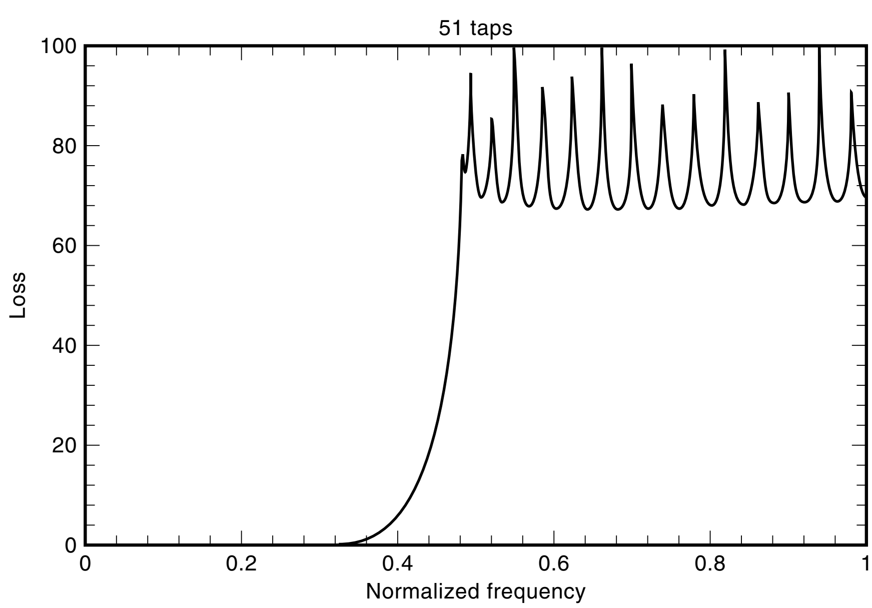
$$\boldsymbol{\beta} = (QF)^{-1}H_d \tag{105}$$

This is indeed a slight generalization of the method of frequency sampling and can be used for FIR filter design. Naturally, the inverse matrix is not computed, but the equations are solved by some other, numerically preferable, method.

If, however, $L > M + 1$ or even $L \gg M + 1$, then $\boldsymbol{e}$ has many more elements than $\boldsymbol{\beta}$ and consequently cannot be made to disappear; we can only attempt to minimize its norm (that is, e^Te). This can be done by the use of the "pseudoinverse" of a rectangular matrix. We obtain this by premultiplying the error equation by $(QF)^T$, obtaining

$$(QF)^T(QF)\boldsymbol{\beta} = (QF)^TVH_d - (QF)^Te \tag{106}$$

We can now set the last error term to zero and solve this equation, because the matrix on the left $(QF)^T(QF)$ is an $M + 1$ by $M + 1$ square matrix. We leave the details for the literature (32,33). We must be careful about using this algorithm,



(h)

Figure 19(h). Characteristics of a Taylor window.

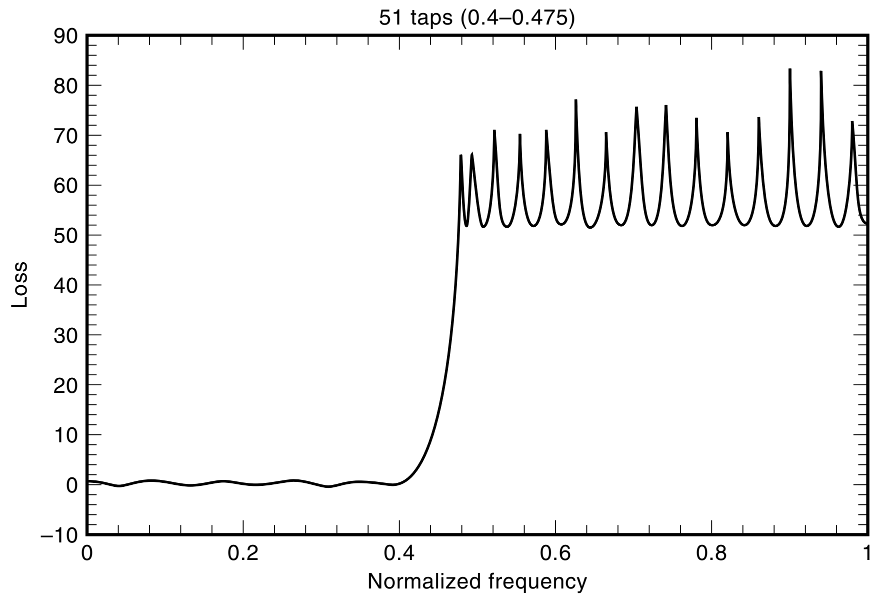


Figure 20. Equal-ripple (Remez) FIR filter characteristics.

since the procedure can get numerically ill conditioned. Instead, we recommend the use of the methods in the LINPACK program package (34) or the method of singular-value decomposition (33).

This method can also be applied to the case of nonlinear phases, and it is one of the methods most often used in that case.

Closed-Form Solutions. We may mention two special cases, in which we can obtain closed-form expressions for the FIR filter. Both use the Chebyshev polynomials $T_n(x)$ we have already used (35). Since $T_n(x)$ varies between ± 1 if x is in the range $-1 < x < +1$, we can simply replace x by an expression in terms of $\cos(\omega)$. If we need a low-pass with an equal-ripple passband, we select

$$x = \frac{(1+\cos\omega_\mathrm{p}) - 2\cos\omega}{1-\cos\omega_\mathrm{p}} \tag{107}$$

and use the transfer function

$$H(\omega) = 1 - \delta_\mathrm{p} T_n(x) \tag{108}$$

The stopband will be monotonic, and to make the magnitude of the transfer function at the Nyquist frequency zero we need to select

$$\delta_\mathrm{p} = 1/T_n[(3-\cos\omega_\mathrm{p})/(1+\cos\omega_\mathrm{p})] \tag{109}$$

The other case is when we need an equal-ripple stopband; in which case we use

$$x = \frac{2\cos\omega + 1 - \cos\omega_\mathrm{s}}{1+\cos\omega_\mathrm{s}} \tag{110}$$

and the transfer function will be given by

$$H(\omega) = \delta_\mathrm{s} T_n(x) \quad \text{where} \quad \delta_\mathrm{s} = 1/T_n[(3-\cos\omega_\mathrm{s})/(1+\cos\omega_\mathrm{s})] \tag{111}$$

if we wish $|H|$ to be unity at $\omega = 0$. High-pass filters with similar behavior are easily obtainable through a change of the expression for x, but bandpass filters are more difficult since we must make both $H(0)$ and $H(\pi)$ disappear.

In any case, we have very little control over the band that is *not* equal ripple. Fig. 21 shows a pair of filters with 21 taps; one has an equal ripple passband from 0 to 0.5, the other an equal ripple stopband from 0.5 to 1.0. The ripple values in both cases are extremely small.

A much more useful closed-form approximation (36,37) exists for maximally flat pass- *and* stopband lowpass filters. Using the case 1 formulation (symmetrical coefficients and N even), we can find an $H(\omega)$ such that it has $2L$ zeros at $\omega = \pi$ and $H(\omega) - 1$ has $2K$ zeros at $\omega = 0$, where $M = L + K - 1$. Ignoring the phase factor, we can then write this transfer function in two equivalent forms:

$$\begin{aligned} H(\omega) &= \left[\frac{1+\cos\omega}{2}\right]^K \sum_{n=0}^{L-1} d_n \left[\frac{1-\cos\omega}{2}\right]^n \\ &\equiv 1 - \left[\frac{1-\cos\omega}{2}\right]^L \sum_{n=0}^{K-1} \overline{d}_n \left[\frac{1+\cos\omega}{2}\right]^n \end{aligned} \tag{112}$$

which is satisfied if (37)

$$d_n = \frac{(K-1+n)!}{(K-1)!n!} \quad \text{or} \quad \overline{d}_n = \frac{(L-1+n)!}{(L-1)!n!} \tag{113}$$

The design has only the powers K and L as free parameters, and the way to satisfy specific requirements is also outlined in Ref. 37. The parameters usually specified are the $H(\omega) = 0.5$ point and the transition bandwidth, usually defined as the distance between the 95% and the 5% transmission points. Figure 22 shows an example, with $K = 11$ and $L = 8$, yielding a (normalized) transition bandwidth of 0.24 and a half-power point at $\omega = 0.448$. High-pass filters can easily be obtained by using $1 - H(\omega)$, but there is no way to design bandpass filters with similar characteristics. As pointed out by Kaiser,

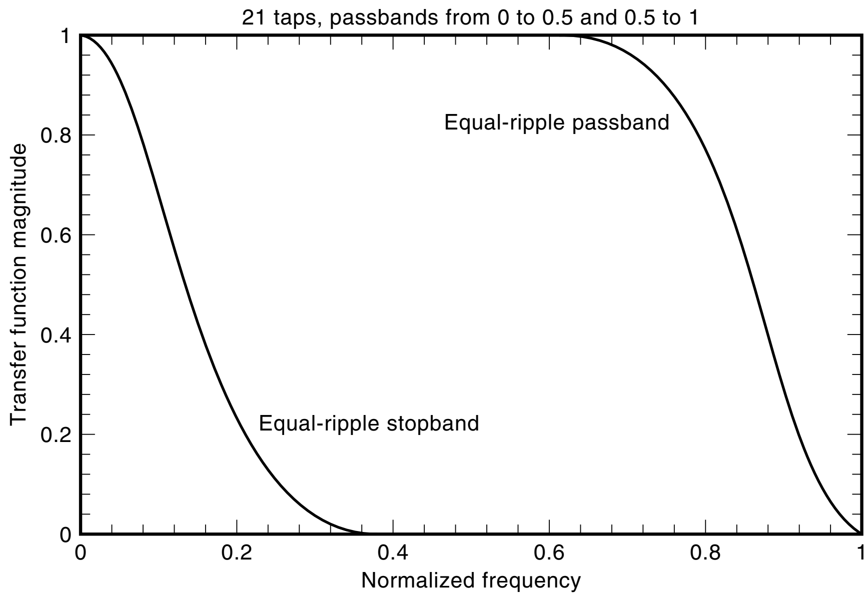


Figure 21. FIR filters with equal ripple pass- or stopband.

high-order filters designed by this method will have a number of coefficients at the end with very small values. Consequently, these filters are practical for medium complexity ($N \leq 50$) only.

Attempts have been made for developing algorithms for the design of FIR filters with flat passband and equal-ripple stopband or vice versa. Today, few of these methods are in general use.

Additional algorithms have been developed for cascading two or more functions to generate more selective filters and their design, for which we refer the reader to the literature (38).

TIME-DOMAIN APPROXIMATION

Returning to Eq. (1) for the overall transfer function and assuming that we are interested in the step response of the filter, the first step is to expand the function divided by s into partial fraction form:

$$\frac{H(s)}{s} = \sum_{j=1}^{d+1} \frac{A_j}{s - p_j} \tag{114}$$

where A_j is the residue at the pole p_j. Incidentally, while we may make allowances for multiple poles, we have never encountered them in practical situations. The residues are, of course, functions of the poles and zeros:

$$A_j = H_0 \frac{\prod_{i=1}^{n}(p_j - z_i)}{\prod_{k=1,\neq j}^{d+1}(p_j - p_k)} \tag{115}$$

The step response can now be expressed as

$$a(t) = \sum_{j=1}^{d+1} A_j e^{p_j t} \tag{116}$$

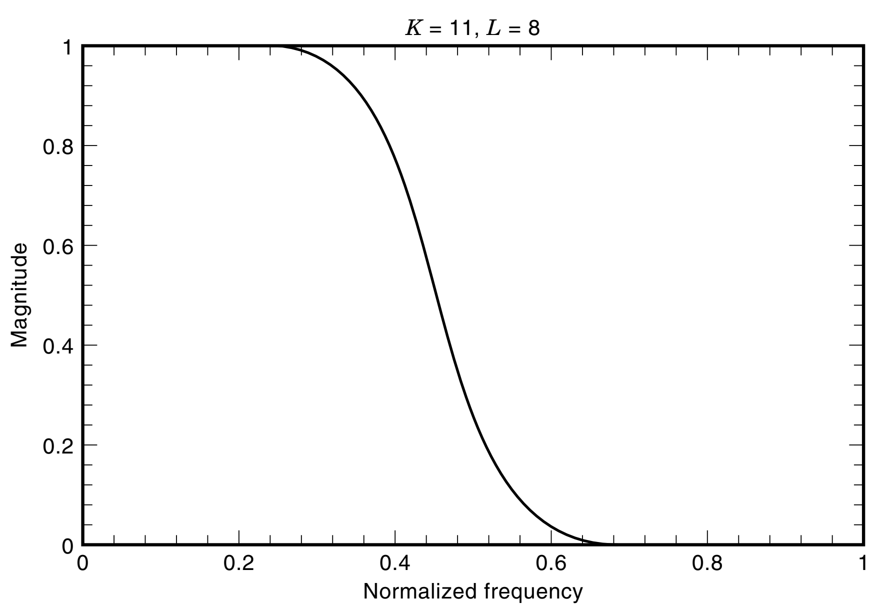


Figure 22. Maximally flat FIR filter characteristics.

Some of the poles will be complex, but they appear in complex conjugate pairs, and the corresponding residues will also be complex conjugates, yielding a real time function. A number of papers have been published about approximating either the step or the impulse response to have specified magnitude of ringing and simultaneously the filter to have equal minima type stopband with specified loss (39–41). The approximation was performed in the minimax sense, and extensive tabulated results are available.

FIR filter design is basically a time-domain approach and therefore need not be discussed. However, if the impulse response of IIR filters is specified, Prony's method may be used to obtain the corresponding transfer function. This method is based on the relationship

$$H(z) = \frac{N(z)}{D(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \sum_{n=0}^{\infty} h(n) z^{-n} \tag{117}$$

where the $h(n)$ values are given and the a_i and b_i coefficients are to be determined. If we truncate the right side to $K = M + N + 1$ terms and cross multiply, we can compare coefficients of z^{-k} and obtain the following set of linear equations [denoting $h(n)$ by h_n for simplicity]:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_M \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & \ldots & 0 \\ h_1 & h_0 & 0 & \ldots & 0 \\ h_2 & h_1 & h_0 & \ldots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ h_M & h_{M-1} & h_{M-2} & \ldots & h_{M-N} \\ h_{M+1} & h_M & h_{M-1} & \ldots & h_{M-N+1} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ h_{M+N} & h_{M+N-1} & h_{M+N-2} & \ldots & h_M \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \\ \\ a_N \end{bmatrix} \tag{118}$$

Ignoring the first $M + 1$ equations for the moment, the rest can be rewritten as

$$\begin{aligned} &h_M a_1 + h_{M-1} a_2 + h_{M-2} a_3 + \cdots + h_{M-N+1} a_N = -h_{M+1} \\ &h_{M+1} a_1 + h_M a_2 + h_{M-1} a_3 + \cdots + h_{M-N+2} a_N = -h_{M+2} \\ &\vdots \\ &h_{M+N-1} a_1 + h_{M+N-2} a_2 + h_{M+N-3} a_3 + \cdots + h_M a_N = -h_{M+N} \end{aligned} \tag{119}$$

This set of N equations in the N unknown a_i denominator coefficients can be solved if the (square) matrix on the left is nonsingular. Once this is done, we can go back to the first $M + 1$ equations and solve them for the numerator coefficients. If the matrix is singular, this indicates that the problem may be solved by a lower-degree $H(z)$ function.

The problem with this method is that we have no control over the values of $h(n)$ beyond $n = M + N + 1$ and, more significantly, we have no idea if the resulting transfer function will turn out to be stable. The first of these can be somewhat alleviated by adding additional equations to those in Eq. (119) and solve this (overdetermined) set of equations using least squares techniques.

APPENDIX A: TEMES-GYI PROCEDURE

To generate a low-pass transfer function with an equal-minima type stopband behavior with a given denominator $D(s)$, we shall start by writing the transfer function denominator in factored form:

$$D(s) = \prod_{k=1}^{d} (s - p_k) \tag{A.1}$$

where the multiplier is immaterial and p_k, if complex, appears in complex conjugate pairs. Next we introduce a new variable z:

$$z = \sqrt{1 + (s/\omega_s)^2} \tag{A.2}$$

which will be pure imaginary in the stopband from ω_s to infinity. If we compute the transformed values of the p_k poles as

$$z_k = \sqrt{1 + (p_k/\omega_s)^2} \tag{A.3}$$

then we can observe that since the z_k values, if complex, also occur in complex conjugate pairs, hence for pure imaginary z values the function

$$\prod_{k=1}^{d} \frac{z_k - z}{z_k + z} \tag{A.4}$$

will have unit magnitude in the stopband and can therefore be written as $e^{j\varphi}$, that is,

$$\begin{aligned} \cos \varphi/2 &= \frac{1}{2}(e^{j\varphi/2} + e^{-j\varphi/2}) = \frac{1}{2}\left\{ \prod_{k=1}^{d} \sqrt{\frac{z_k - z}{z_k + z}} + \prod_{k=1}^{d} \sqrt{\frac{z_k + z}{z_k - z}} \right\} \\ &= \frac{Ev\left\{\prod_{k=1}^{d}(z_k + z)\right\}}{\prod_{k=1}^{d}\sqrt{(z_k^2 - z^2)}} \end{aligned} \tag{A.5}$$

is going to vary between zero and one in the stopband. Ev designates the even part of the polynomial. The square of this quantity is therefore

$$\cos^2(\varphi/2) = \frac{\left\{Ev \prod_{k=1}^{d}(z_k + z)\right\}^2}{\prod_{k=1}^{d}(z_k^2 - z^2)} \tag{A.6}$$

where the denominator corresponds to the polynomial $D(s)\,D(-s)$, while the numerator is $[N(s)]^2$, where $N(s)$ is an even polynomial. This is therefore the magnitude function (save for a multiplier) we are looking for, and the required transmission zeros are obtained by calculating the roots of the polynomial

$$Ev \prod_{k=1}^{d} (z_k + z)$$

and converting them back to s. The available minimum stopband loss can be computed simply by calculating the magnitude of $N(s)/D(s)$ at ω_s, assuming that the magnitude at $s = 0$ is set to unity. Alternatively, we can evaluate the expression above for $\cos^2(\varphi/2)$ at $z = 1$ ($\omega = 0$) and the minimum loss will be

$$a_{\min} = 10\log_{10}(1/\cos^2(\varphi/2)) \text{ at } z = 1$$

This works fine if the degree d is odd. If it is even, the resulting loss will be finite at infinity, since the degree of $N(s)$ will be the same as that of $D(s)$. This is acceptable for active RC or digital implementations. For RLC realization, we could apply a simple shift to the zeros and poles, as we have done in the preceding elliptic case above, but that would also shift ω_s and, more significantly, all the zeros of $D(s)$ as well. The solution is to apply a reverse shift to these zeros and to ω_s, followed by the preceding computation, followed by the forward shift to cancel the reverse one. If this shifts the last zero to infinity, we are done. If not, we modify the amount of reverse shift we used and repeat. This iterative procedure converges very fast, hardly ever needing more than two or three steps. A somewhat different procedure is described in Ref. 12.

APPENDIX B: OPTIMIZATION STRATEGIES

The general optimization problem can be formulated as follows. The overall error function is a general, nonlinear function of the transfer function poles, zeros, and possibly a multiplier:

$$E = f(x_1, x_2, x_3, \ldots x_n) = f(\boldsymbol{x}) \tag{B.1}$$

The variables x_i are usually combined into a single vector $\boldsymbol{x}$. If we wish to reduce the problem to real variables, we may use the quadratic coefficients in a factored form, instead of the roots of these quadratics. We start from a set of initial values $\boldsymbol{x}_0$ and wish to determine $\boldsymbol{x}$ such that E is minimized. The method to be used is dependent on the exact form of the error function $f(\boldsymbol{x})$.

The Least *p*th Approximation

Consider first the least pth error definition of Eq. (66). The currently favored methods can be classified according to whether they need derivatives or not.

The Simplex Method. Nelder and Mead (42) introduced the simplex method, which needs only function evaluations. It starts by evaluating the function at the starting set of parameter values $\boldsymbol{x}_0$ and n additional points, which we get by changing the x_i parameters by a fixed amount, one at a time (the corner points of a polyhedron). Out of these $n + 1$ points, we select the one where $f(\boldsymbol{x})$ is the largest and *reflect* this point through the center of gravity of the remaining n points. At this juncture, we again have $n + 1$ points and function values and we can repeat the procedure. Many refinements are possible, indeed, necessary. One is that if the function value at the reflected point is better than at any other, we move further in the same direction, by a factor usually selected to be about two. This is called *expansion,* and if it works, we accept the new point; if not, we back off. If, on the other hand, the new point has a value $f(\boldsymbol{x})$, which is better than the worst point but worse than all others, we *contract* the step (i.e., move a shorter distance in the indicated direction). Finally, if the new point yields an evaluation that is still the worst, we *reduce* the size of the polyhedron by a factor of two, starting from the best point. The test for convergence is usually

$$\left[\frac{1}{n+1}\sum_{i=1}^{n+1}(f(x_i) - f_{\text{ave}})^2\right]^{1/2} \le eps \tag{B.2}$$

where f_{ave} is the average of all the function values and *eps* is the specified tolerance. This simply means that the function values are now so close as to make any distinction between them meaningless.

This procedure is fast and cheap in terms of computational expenses. Since the polyhedron is changing its shape and size during the iteration, it is able to follow the terrain fairly well and has been found to be effective in starting the optimization.

The Gradient Method. The gradient method needs the computation of the first set of partial derivatives:

$$\nabla f(x) = \left\{\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots \frac{\partial f}{\partial x_n}\right\}^{\mathrm{T}} \tag{B.3}$$

either analytically or approximately (numerically). The superscript T indicates transposition. Since the direction ∇f is where the value of f would increase the fastest, we go in the opposite direction and search for the minimum along

$$\boldsymbol{x} = \boldsymbol{x}_0 - \lambda\nabla f \tag{B.4}$$

where λ is a scalar. There are again many ways to perform this one-dimensional search that can be done with or without calculating further derivatives. Some of the simplest ones are the golden section and the Fibonacci searches. Here we compute the function values for two values of λ that are sure to bracket the minimum and subdivide this range by either the golden section or the Fibonacci series ratio. Once the new function value is computed, we can do further subdivisions and arrive at the location of the minimum in optimal time. Another could be to calculate the function values for three values of λ, fit a quadratic function to these points, and calculate its minimum. Repeating this procedure can locate the minimum reasonably accurately.

Once the line search has located the minimum along the variable λ, we recalculate the gradient and repeat the process. It can readily be shown that if we locate the minimum along this direction exactly, the new direction will be orthogonal to the previous one, which may not help if we need to go along a narrow valley. Refinements can come in the form of averaging the directions of several consecutive derivative calculations and many others.

Newton-Raphson Method. Newton's method goes one step further along in expanding the function $f(\boldsymbol{x})$ around $\boldsymbol{x}_0$ into a Taylor series:

$$f(x) \cong f(x_0) + \nabla f^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^T H \Delta x \tag{B.5}$$

where H is the matrix of second derivatives (also called Hessian matrix):

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{B.6}$$

To find the point where $f(\boldsymbol{x})$ is optimum (minimum), we must find the value of $\Delta\boldsymbol{x}$ that will make all components of the vector ∇f disappear:

$$\nabla f(x) \cong \nabla f(x_0) + H(x_0)\Delta x = 0$$

or

$$\Delta x = (x - x_0) = -H^{-1}(x_0)\nabla f(x_0) \tag{B.7}$$

This method only works if H is positive definite, but when we are close to the optimum, it converges fast. The main problem is the cost of evaluating (analytically or numerically) the Hessian matrix and inverting it. This method is hardly ever used in its original form; it is useful mainly to introduce the next method, the Davidon–Fletcher–Powell method.

The Davidon–Fletcher–Powell Method. This is one of a family of methods called the *variable metric* algorithms (43). The idea for this method comes from the realization that the gradient method can be written in the form

$$\boldsymbol{x} = \boldsymbol{x}_0 - \lambda\nabla f(\boldsymbol{x}_0) = \boldsymbol{x}_0 - \lambda G\nabla f(\boldsymbol{x}_0) \tag{B.8}$$

where G is the unit matrix, while the Newton–Raphson method has the same form, except that G is then proportional to the inverse Hessian. Davidon had the idea that we can start the approximation with G being the unit matrix but then, as the iterations continue, build it up to approximate the inverse Hessian numerically, without actually having to calculate the Hessian and invert it. The reason for this being a whole family of methods is that there is no unique way of doing this, but many different ways instead. At any particular iteration, we locate the minimum along the current direction and determine the corresponding λ_{k+1} and from that $\boldsymbol{x}_{k+1}$ and $\nabla f(\boldsymbol{x}_{k+1})$, which give us

$$\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \quad \text{and} \quad \boldsymbol{y}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)$$

$$G_{k+1} = G_k - \frac{(G_k y_k)(G_k y_k)^{\mathrm{T}}}{y_k^{\mathrm{T}} G_k y_k} + \frac{p_k p_k^{\mathrm{T}}}{y_k^{\mathrm{T}} p_k} \tag{B.9}$$

as one of the possible update expressions. (Note that if $\boldsymbol{v}$ and $\boldsymbol{w}$ are two vectors, then $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}$ is a scalar, but $\boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$ is a matrix.)

It can be shown that if $f(\boldsymbol{x})$ is a true quadratic function [i.e., the Taylor series expansion of Eq. (B.5) is exact], then this G_k converges to the inverse Hessian in exactly n steps. Also, if the original Hessian H is positive definite, the sequence of matrices G_k will also be positive definite.

Naturally, one equation an algorithm does not make; we need convergence criteria, ways of handling special cases, numerical instability, and a host of other issues besides using different expressions. For all of these as well as for finding computer programs implementing the foregoing, we refer to the extensive literature.

Thoroughly tested and highly efficient routines are available for these and other optimization techniques either commercially or in the public domain. All methods considered previously were of the unconstrained variety (that is, there were no limits placed on the possible values of the variables). This is no restriction if we consider losses only, since dealing with poles and zeros, putting all the poles back into the left half of the s plane leaves the loss unchanged, and that is the only restriction we need to satisfy. For delay requirements, however, the poles may sometimes wander over to the right-half plane, which is not permitted. We must then increase the additional flat delay required to force these poles back into the left half of the s plane. More complex *constrained* optimization techniques exist, but if we restrict our techniques to optimizing the transfer function itself, these are usually not necessary.

Minimax Approximation

All of the preceding methods are applicable if the error function is of the form of Eq. (66). For the minimax formulation of Eq. (67), we have basically two options. One is based on the fact that if the value of the exponent p in Eq. (66) tends to very large values, the approximation in fact approaches the minimax criteria. The other option is the application of the Remez algorithm.

Remez Algorithm. The idea behind this algorithm is very simple (44), and it is based on the *alternation theorem:* If $P(\omega)$ is a linear combination of M cosine functions,

$$P(\omega) = \sum_{k=0}^{M} \alpha_k \cos(k\omega) \tag{B.10}$$

then a necessary and sufficient condition that $P(\omega)$ be the unique best weighted Chebyshev approximation to a continuous function $H_{\mathrm{r}}(\omega)$ is that the weighted error function

$$E(\omega) = W(\omega)[P(\omega) - H_{\mathrm{r}}(\omega)] \tag{B.11}$$

exhibit at least $M + 1$ extremal frequencies in the range of interest in ω.

We select $M + 2$ frequency points ω_k that is one more than the number of free parameters and use the (weighted) approximating function to interpolate the required function $H_{\mathrm{r}}(\omega_k) \pm\delta$, where the sign alternates at consecutive frequencies. Since we have $M + 2$ frequencies, where $M + 1$ is the number of free parameters and $M + 2$ parameters (δ is also unknown), this should be a well-defined problem. Next we find all the extrema of this approximating function and replace the ω_k values by the locations of these extrema. Repeating the process will lead to the required minimax result. The problem is the interpolation step, especially if the approximating function is highly nonlinear, when solving the interpolation problem is itself equivalent to an approximation procedure. In a few special cases, we can obtain an appropriate interpolation relatively easily; one is the procedure of approximating a constant delay, and the other is the design of FIR digital filters.

APENDIX C: SPECIAL FUNCTIONS

A number of classical polynomials have been tried to generate characteristic functions, including Jacobi, Laguerre, Legendre, and various Chebyshev polynomials [other than the $T_n(x)$ we have used previously], but they have not been found useful in practice. A few exceptions are as follows.

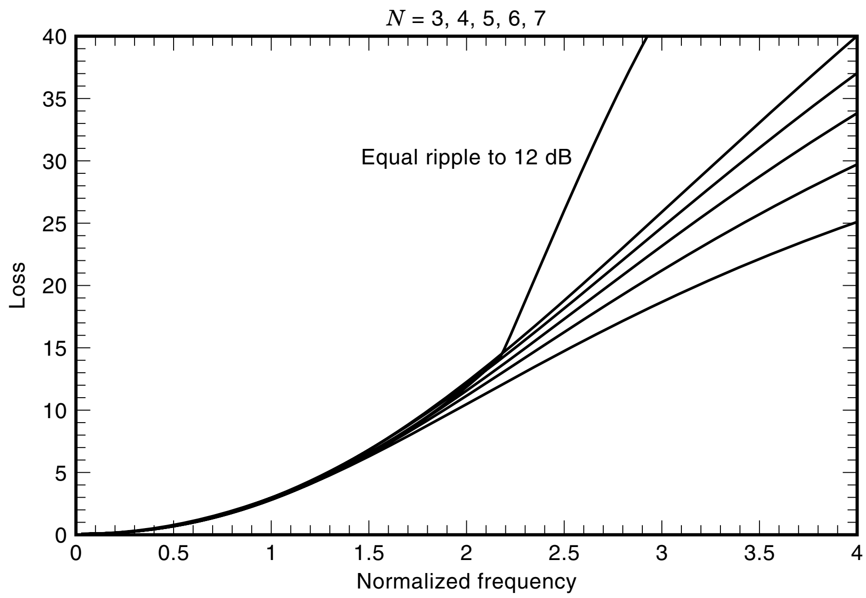


Figure 23. Loss characteristics of Gaussian filters.

Gaussian Filter

In certain situations, one would like to have a filter characteristic that approximates the Gaussian shape: $|H(\omega)|^2 \cong \exp[-(\omega/\omega_0)^2]$. One can again do this in various ways; the simplest one approximating this shape in the maximally flat sense is to use the characteristic function

$$\kappa(s)\kappa(-s) = \exp(s/\omega_0)^2 - 1 = \sum_{n=1}^{\infty} \frac{\left(\dfrac{s}{\omega_0}\right)^{2n}}{n!} \qquad \text{(C.1)}$$

Truncating this expansion to a finite number of terms will yield a number of (equivalent) $\kappa(s)$ functions depending on how we allocate the zeros to $\kappa(s)$ and $\kappa(-s)$. Equal ripple-type approximation has also been tried, but the results are simply tabulated natural modes for degrees from 3 to 10 and approximation errors of 0.05 dB up to either the 6 dB or 12 dB points. Fig. 23 shows the loss characteristics of the first few maximally flat approximations and the seventh-order equal ripple to 12 dB approximation. This last one has a much steeper rise of the loss beyond the 12 dB point. Tabulated functions are available (for instance, in Refs. 10 and 45).

Papoulis Filter

Papoulis (46) has found the function that provides a loss that rises the fastest among all the monotonically increasing transfer functions with a constant numerator function. One can derive the denominator polynomial of such a function as follows (see Refs. 46 and 47). For n odd

$$\kappa(\omega)\kappa(-\omega) = \int_{-1}^{2\omega^2-1} [a_0 + a_1 P_1(x) + \cdots + a_{(n-1)/2} P_{(n-1)/2}(x)]^2\, dx \qquad \text{(C.2)}$$

where the $P_k(x)$ are the Legendre polynomials defined by

$$P_0(x) = 1; \quad P_1(x) = x; \quad \text{and}$$
$$P_{k+1}(x) = \frac{2k+1}{k+1} x P_k(x) - \frac{k}{k+1} P_{k-1}(x) \qquad \text{(C.3)}$$

and

$$a_0 = a_2/5 = \cdots = a_{(n-2)/2}/(n-1) = \frac{2}{\sqrt{n(n+2)}}$$

For n even, on the other hand, we have

$$\kappa(\omega)\kappa(-\omega) = \int_{-1}^{2\omega^2-1} (x+1)[a_0 + a_1 P_1(x) + \cdots + a_{(n-1)/2} P_{(n-1)/2}(x)]^2\, dx \qquad \text{(C.4)}$$

where we have two subcases:

$$\begin{aligned} &\text{Case 1: } [(n-2)/2] \text{ is even:}\\ &a_0 = a_2/5 = \cdots = a_{(n-2)/2}/(n-1) = \frac{2}{\sqrt{n(n+2)}}\\ &a_1 = a_3 = \cdots = a_{(n-4)/2} = 0\\ &\text{Case 2: } [(n-2)/2] \text{ is odd:}\\ &a_1/3 = a_3/7 = \cdots = a_{(n-2)/2}/(n-1) = \frac{2}{\sqrt{n(n+2)}}\\ &a_0 = a_2 = \cdots = a_{(n-4)/2} = 0 \end{aligned} \qquad \text{(C.5)}$$

Reference 45 contains tables of these polynomials. Fig. 24 shows the fifth-order Butterworth and Papoulis filter characteristics. For comparison, we also included the fifth-order Chebyshev function with 1 dB ripple, but scaled to the same 3 dB point as the others. The Chebyshev is, of course, the fastest rising but it is *not* monotonic.

Butterworth–Thomson Filter

Filter designers have found that the Butterworth characteristics are desirable from the loss point of view but have undesirable delay performance. The Bessel functions, on the other hand, have the opposite behavior. It follows naturally that someone would try to combine the two, yielding the Butterworth–Thomson filter. (In this context, Thomson's name is be-

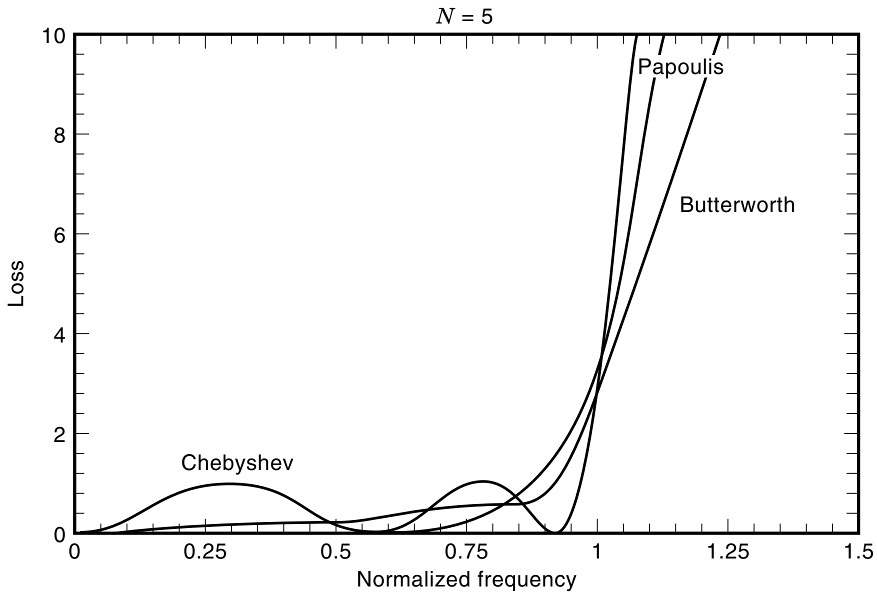


Figure 24. Comparison of polynomial low-pass transfer functions.

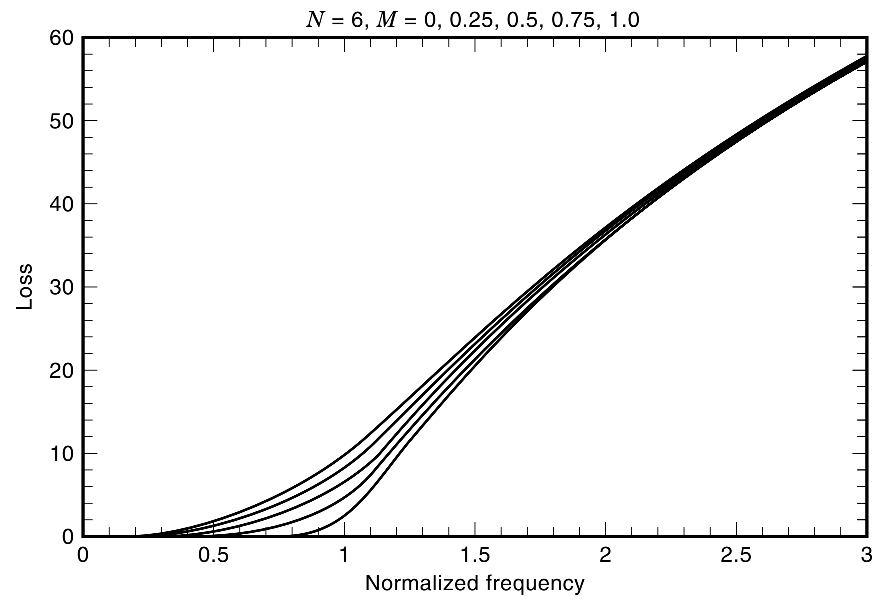


Figure 25. Loss characteristics of Butterworth–Thomson filters.

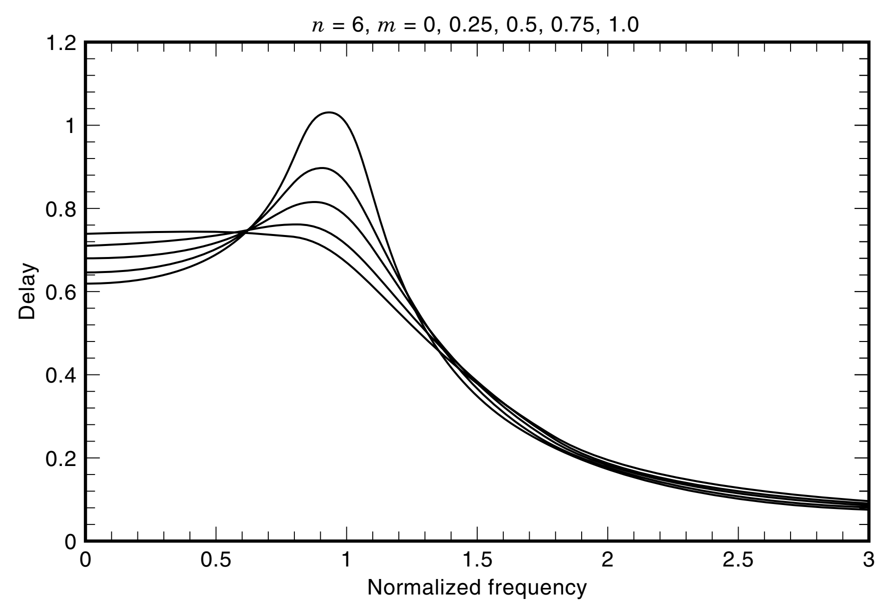


Figure 26. Delay characteristics of Butterworth–Thomson filters.

ing used instead of Bessel.) The idea is simply to take the natural modes of the nth order Butterworth filter

$$z_k^{\mathrm{B}} = \exp(j\varphi_k^{\mathrm{B}}) \qquad k = 1, 2, \ldots n$$

and the zeros of the (same degree) Bessel polynomial [Eq. (21)]

$$z_k^{\mathrm{T}} = r_k^{\mathrm{T}} \exp(j\varphi_k^{\mathrm{T}}) \qquad k = 1, 2, \ldots n$$

The transitional Butterworth–Thomson filter will have natural modes given by

$$z_k = r_k \exp(j\varphi_k) \tag{C.6}$$

where

$$r_k = (r_k^{\mathrm{T}})^m \quad \text{and} \quad \varphi_k = \varphi_k^{\mathrm{B}} - m(\varphi_k^{\mathrm{B}} - \varphi_k^{\mathrm{T}}) \tag{C.7}$$

Here m is a parameter between zero and one; $m = 0$ yields the pure Butterworth solution, while $m = 1$ is the Bessel filter (see Ref. 48). The roots of the Bessel polynomial r_k^{T} are usually scaled first by dividing them by the factor

$$\frac{2n}{e} 2^{\frac{1}{2n}}$$

to bring their magnitude close to unity. This quantity is an approximation to the nth root of the constant term in the polynomial and is based on Stirling's approximation of $n!$, but it is very good even for low degrees. Also, this renormalization will change the normalized delay at zero frequency from unity to this value. Figures 25 and 26 show the loss and delay of the sixth-order Butterworth–Thomson filters with m values in the range 0.0 (0.25) 1.0.

BIBLIOGRAPHY

1. T. Saramaki, Design of optimum recursive digital filters with zeros on the unit circle, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-31**: 450–458, 1983.
2. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions,* Washington, DC: National Bureau of Standards, 1964.
3. S. Darlington, Synthesis of reactance 4-poles which produce prescribed insertion loss characteristics, *J. Math. Physics,* **18**: 257–353, 1939.
4. H. J. Orchard and A. N. Willson, Jr., Elliptic functions for filter design, *IEEE Trans. Circuits Syst. I,* **CAS-44**: 273–287, 1997.
5. G. Szentirmai, Computer-aided design methods in filter design, in J. T. Taylor and Q. Huang (Eds.), *CRC Handbook of Electrical Filters,* Boca Raton, FL: CRC Press, 1996.
6. R. Saal, *Handbook of Filter Design,* Berlin, Germany: AEG-Telefunken, 1979.
7. J. K. Skwirzynski, *Design Theory and Data for Electrical Filters,* New York: Van Nostrand-Reinhold, 1965.
8. E. Christian and E. Eisenmann, *Filter Design Tables and Graphs,* New York: Wiley, 1966.
9. L. Storch, Synthesis of constant-time-delay ladder networks using Bessel polynomials, *Proc. IRE,* **42**: 1666–1675, 1954.
10. A. I. Zverev, *Handbook of Filter Synthesis,* New York: Wiley, 1967.
11. G. C. Temes and M. Gyi, Design of filters with arbitrary passband and Chebyshev stopband attenuation, *1967 IEEE Int. Conf. Rec.,* pp. 2–12.
12. R. Unbehauen, Low-pass filters with predetermined phase or delay and Chebyshev stopband attenuation, *IEEE Trans. Circuit Theory,* **CT-15**: 337–341, 1968.
13. J. D. Rhodes, *Theory of Electrical Filters,* New York: Wiley, 1976.
14. J. P. Thiran, Recursive digital filters with maximally flat group delay, *IEEE Trans. Circuit Theory,* **CT-18**: 659–664, 1971.
15. A. Fettweis, A simple design of maximally flat delay digital filters, *IEEE Trans. Audio Electroacoust.,* **AU-20**: 112–114, 1972.
16. J. P. Thiran, Equal-ripple delay recursive digital filters, *IEEE Trans. Circuit Theory,* **CT-18**: 664–669, 1971.
17. G. Szentirmai, The design of arithmetically symmetrical bandpass filters, *IEEE Trans. Circuit Theory,* **CT-10**: 367–375, 1963.
18. B. R. Smith and G. C. Temes, An iterative approximation procedure for automatic filter synthesis, *IEEE Trans. Circuit Theory,* **CT-12**: 107–112, 1965.
19. H. C. Bell, private communication.
20. H. C. Bell, Bandwidth adjustment in iterative approximation procedures, *IEEE Trans. Circuits Syst.,* **CAS-25**: 951–954, 1978.
21. R. W. Daniels, *Approximation Methods for Electronic Filter Design,* New York: McGraw-Hill, 1974.
22. H. Watanabe, Approximation theory for filter networks, *IRE Trans. Circuit Theory,* **CT-9**: 341–356, 1961.
23. A. Premoli, The MUCROMAF polynomials: An approach to the maximally-flat approximation of RC-active filters with low sensitivity, *IEEE Trans. Circuit Theory,* **CT-20**: 77–80, 1973.
24. A. Premoli, A new class of equal-ripple filtering functions with low Q factors, *IEEE Trans. Circuits Syst.,* **CAS-21**: 609–613, 1974.
25. M. Biey and A. Premoli, *Tables for Active Filter Design,* Norwood, MA: Artech House, 1985.
26. E. Ulbrich and R. Piloty, Über den Entwurf von Allpässen, Tiefpässen und Bandpässen mit einer in Tschebyscheffschen Sinne approximiert konstanten Gruppenlaufzeit, *AEU,* **14**: 451–467, 1960.
27. H. W. Bode, *Network Analysis and Feedback Amplifier Design,* New York: Van Nostrand-Reinhold, 1945.
28. A. G. Deczky, Equiripple and minimum (Chebyshev) approximation for recursive digital filters, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-22**: 98–111, 1974.
29. D. F. Elliott (Ed.), *Handbook of Digital Signal Processing,* New York: Academic Press, 1987.
30. F. B. Hildebrand, *Introduction to Numerical Analysis,* New York: McGraw-Hill, 1956.
31. J. H. MacClellan, T. W. Parks, and L. R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust.,* **AU-21**: 506–526, 1973.
32. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Philadelphia: SIAM, 1996.
33. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems,* Englewood Cliffs, NJ: Prentice-Hall, 1974.
34. J. J. Dongarra et al., *LINPACK User's Guide,* Philadelphia: SIAM, 1979.
35. O. Herrmann, L. R. Rabiner, and D. S. K. Chan, Practical design rules for optimum finite impulse response lowpass digital filters, *Bell Sys. Tech. J.,* **52**: 769–799, 1973.
36. O. Herrmann, On the approximation problem in nonrecursive digital filter design, *IEEE Trans. Circuit Theory,* **CT-18**: 411–413, 1971.
37. J. F. Kaiser, Design subroutine (MXFLAT) for symmetric FIR low-pass digital filters with maximally-flat pass and stop bands, in Digital Signal Processing Committee, *Programs for Digital Signal Processing,* New York: IEEE Press, 1979, pp. 5.3-1–5.3-6.

38. J. F. Kaiser and R. W. Hamming, Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-25**: 415–422, 1977.
39. J. Jess, Uber Impulsfilter mit Tschebyscheffverhalten in Zeit- und Frequenzbereich, *AEU,* **17**: 391–401, 1963.
40. J. Jess, On the design of pulse-forming networks, *IEEE Trans. Circuit Theory,* **CT-12**: 393–400, 1965.
41. K. L. Su, *Time-Domain Synthesis of Linear Networks,* Englewood Cliffs, NJ: Prentice-Hall, 1971.
42. D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software,* Englewood Cliffs, NJ: Prentice-Hall, 1989.
43. M. Avriel, *Nonlinear Programming,* Englewood Cliffs, NJ: Prentice-Hall, 1976.
44. E. Y. Remez, General computational methods of Chebyshev approximation, Atomic Energy Translation 4491, Kiev, USSR, 1957.
45. D. S. Humpherys, *The Analysis, Design and Synthesis of Electrical Filters,* Englewood Cliffs, NJ: Prentice-Hall, 1970.
46. A. Papoulis, A new class of filters, *Proc. IRE,* **46**: 649–653, 1958.
47. M. Fukado, Optimum filters of even order with monotonic responses, *IRE Trans. Circuit Theory,* **CT-6**: 277–281, 1959.
48. Y. Peless and T. Murakami, Analysis and synthesis of transitional Butterworth-Thomson filters and bandpass amplifiers, *RCA Rev.,* **18**: 60–94, 1957.

George Szentirmai
DGS Associates Inc.

FILTER BANKS. See Wavelets.

FILTERING. See Image enhancement.