would require less hardware than the other possibilities.

Another means of evaluating these different realizations is to examine the effect of coefficient roundoff error and multiplier truncation noise. These have not yet been considered.

The techniques discussed in this paper can be easily adapted for use in realizing transfer functions in the s-domain. All the results discussed are equally valid if we replace z^{-1} by s^{-1} in the mathematical expressions, and replace delays by integrators, adders by summing amplifiers, and multipliers by amplifiers in the diagrams.

Conclusion

A technique similar to that of [6] for realizing a digital filter in a ladder like structure is proposed. It is especially suitable for realizing a transfer function having all its zeros at the origin although it can easily be used to realize any general transfer function. With a slight modification this technique becomes suitable for realizing transfer functions obtained via the bilinear transform from all pole s-domain transfer functions. Two other forms are obtained, one of which is topologically similar to the coupled form [11].

References

[1] F. F. Kuo and J. F. Kaiser, Eds., *System Analysis by Digital Computer.* New York: Wiley, 1967, ch. 7.
[2] C. Rader and B. Gold, "Digital filter design techniques in the frequency domain," *Proc. IEEE*, vol. 55, pp. 149-171, Feb. 1967.
[3] R. M. Golden, "Digital filter synthesis by sampled data transformation," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 321-329, Sept. 1968.
[4] A. Fettweis, "Some principles of designing digital filters imitating classical filter structures," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 314-316, Mar. 1971.
[5] S. K. Mitra and R. Sherwood, "Canonic realizations of digital filters using the continued fraction expansion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 185-194, Aug. 1972.
[6] —, "Digital ladder networks," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 30-36, Feb. 1973.
[7] R. Edwards, J. Bradley, and J. B. Knowles, "Comparison of noise performances of programming methods in the realization of digital filters," in *Proc. Symp. Comput. Processing Commun.*, PIB-MRI Symp. Ser., vol. 19. Brooklyn, N.Y.: Polytechnic Press, 1969, pp. 295-311.
[8] O. Hermann and W. Schussler, "On the accuracy problem in the design of nonrecursive digital filters," *Arch. Elek. Übertragung*, vol. 24, pp. 525-526, 1970.
[9] R. E. Crochiere, "Digital ladder structures and coefficient sensitivity," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 240-246, Oct. 1972.
[10] J. Tow and Y. L. Kuo, "Coupled biquad active filters," in *1972 Proc. IEEE Int. Symp. Circuit Theory*, pp. 164-167.
[11] B. Gold and C. Rader, *Digital Processing of Signals.* New York: McGraw-Hill, 1969, pp. 45-46, 101-103.
[12] M. E. Van Valkenburg, *Introduction to Modern Network Synthesis.* New York: Wiley, 1960, pp. 86-93.

Digital Lattice and Ladder Filter Synthesis

A. H. GRAY, JR., Member, IEEE, and
JOHN D. MARKEL, Member, IEEE

Abstract—There is evidence that in addition to standard digital filter forms such as the direct, parallel, and cascade forms, digital lattice and ladder filters may play an important role in finite word length implementation problems.

In this paper, techniques are developed in detail for efficiently synthesizing digital lattice and ladder filters from any stable direct form. In one form, a lattice filter canonic in terms of multiplies and delays is obtained. An internal scaling procedure is also introduced that will be of importance for optimizing one of the lattice forms for finite word length implementation.

Manuscript received May 15, 1973; revised August 9, 1973. This work was supported under ONR Contract N00014-67-C-0118.

A. H. Gray, Jr. is with the Department of Electrical Engineering and Computer Science, University of California, Santa Barbara, Calif. 93106.

J. D. Markel is with Speech Communications Research Laboratory, Santa Barbara, Calif. 93109.

Introduction

Techniques for synthesizing digital ladder structures [1], [2] have recently been studied because of their apparent frequency response insensitivity with respect to the more standard direct, parallel, and cascade forms [3]. Recent studies by Fettweis [4], [5] show that digital ladder filters can be patterned after classical analog structures. Crochiere has demonstrated that, at least for one filter, the coefficient sensitivity of the digital ladder filter is at least several bits less than a cascade implementation of the same filter [6]. It is certainly desirable to know how to directly transform a direct form into a corresponding ladder or lattice structure. Notable steps in this direction have been made by Mitra and Sherwood [1], [2]. Unfortunately, their results are restrictive in that only certain classes of stable filters can be synthesized. In addition, their procedure requires the solving of a set of linear simultaneous equations in order to obtain the tap gains (gain terms for implementing the numerator without additional delays).

In this paper, a general solution to the synthesis of digital lattice and ladder structures from direct forms is presented. In one form, the resultant filter is canonical in the number of multipliers and delays. The procedure follows from an application of an inner product formulation first developed by Itakura and Saito [7] and later expanded upon by Markel and Gray [8].

The solution is easily developed and programmed as a recursive procedure for the filter design. Furthermore, the procedure is extremely efficient. No simultaneous equation solution for tap gains is necessary. What "falls out" of the procedure is a "two-multiplier" lattice form. A trivial transformation results in a "three-multiplier" ladder form. From the two-multiplier form a new generalized "one-multiplier" form, canonic in both multipliers and delays, is developed. With the generalized one-multiplier model, a new technique is presented for optimally scaling the filter structure with respect to finite word length implementation.

Lattice Filter Formulation

We wish to design a digital filter that will implement the direct form transfer function

$$G = G(z) = P_M(z)/A_M(z) \tag{1}$$

where $P_M(z)$ and $A_M(z)$ are Mth order polynomials in z^{-1}, of the form

$$P_m = P_m(z) = \sum_{n=0}^{m} p_{m,n} z^{-n} \tag{2}$$

and

$$A_m = A_m(z) = \sum_{n=0}^{m} a_{m,n} z^{-n} \tag{3}$$

where $m = M$. Without loss of generality, it is assumed that the leading coefficient of $A_M(z)$ is one, $a_{M,0} = 1$.[1] If the filter is implemented in the direct form, then the parameters representing the filter are $2M + 1$ in general, $p_{M,0}, p_{M,1}, \cdots, p_{M,M}, a_{M,1}, a_{M,2}, \cdots, a_{M,M}$.

The $2M + 1$ parameters of the lattice filter that will be designed are the M k-parameters $k_0, k, \cdots, k_{M-1}$ and the $M + 1$ tap parameters $\nu_0, \nu_1, \cdots, \nu_M$. These parameters are recursively obtained by starting from $A_M(z)$ and $P_M(z)$ as

$$zB_m(z) = A_m(1/z)z^{-m} \tag{4}$$

$$k_{m-1} = a_{m,m} \tag{5}$$

$$A_{m-1}(z) = [A_m(z) - k_{m-1}zB_m(z)]/(1 - k_{m-1}^2) \tag{6}$$

$$\nu_m = p_{m,m} \tag{7}$$

$$P_{m-1}(z) = P_m(z) - zB_m(z)\nu_m \tag{8}$$

for $m = M, M - 1, \cdots, 1$ with $\nu_0 = p_{0,0}$.

The only division that arises is in (6). This will never be a division by zero, for it is known that the k-parameters will always have a magnitude less than one for a stable filter [7], [8]. Should any k_m equal plus or minus one, or have a magnitude larger than one, then $A_M(z)$ will not have all its roots within the unit circle and the filter will be unstable. In terms of the new variables, $P_M(z)$ is equivalent to

$$P_M(z) = \sum_{m=0}^{M} \nu_m zB_m(z). \tag{9}$$

Therefore, an equivalent representation of $G(z)$ is given as

$$G(z) = \sum_{m=0}^{M} \nu_m \frac{zB_m(z)}{A_M(z)}. \tag{10}$$

It remains to be shown that this formulation results in a digital lattice structure. First, the general proof of these equations will be presented.

Proof of Formulation

The results stated here are essentially applications of the inner product formulation published elsewhere [7], [8].

First, we define

$$R(z) = [A_M(z)A_M(z^{-1})]^{-1}, \tag{11}$$

which will be positive on the unit circle. An inner, or scalar, product of functions of z can be then defined by

$$\langle F(z), G(z)\rangle = \int_{-\pi}^{\pi} R(e^{j\theta})F^*(e^{j\theta})\,G(e^{j\theta})\,\frac{d\theta}{2\pi} \tag{12}$$

where the asterisk denotes a complex conjugate. Using this notation, it can be shown [8] that by defining $zB_m(z) = A_m(1/z)z^{-m}$, both the polynomials $A_m(z)$ and $B_m(z)$ are orthogonal to the powers of z^{-1}, from the 1st through the mth; that the polynomials $zB_m(z)$ form an orthogonal set; and that the polynomials are related by the recursion matrix

$$\begin{bmatrix} A_{m+1}(z) \\ B_{m+1}(z) \end{bmatrix} = \begin{bmatrix} 1 & k_m \\ k_m z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} A_m \\ B_m \end{bmatrix} \tag{13}$$

for $m = 0, 1, \cdots, M - 1$ with $A_0(z) = 1$ and $B_0(z) = z^{-1}$.

In addition, the coefficients α_m can be found from

$$\alpha_m = \langle A_m(z), A_m(z)\rangle = \langle B_m(z), B_m(z)\rangle = \langle 1, A_m(z)\rangle, \tag{14}$$

or through a recursion relation

[1] Throughout the paper, for notational simplicity, capital letters will be used to refer to the corresponding z transforms.

$$\alpha_{m+1} = \alpha_m(1 - k_m^2) \qquad \text{for } m = 0, 1, \cdots, M-1 \quad (15)$$

with $\alpha_M = 1$. The k-parameters can also be expressed in terms of scalar products as $k_m = -\beta_m/\alpha_m$ where

$$\beta_m = \langle A_m(z), B_m(z)\rangle = \langle 1, B_m(z)\rangle. \quad (16)$$

Equation (6) is obtained by simply computing the matrix inverse from (13) and replacing m by $m - 1$. Equations (7) and (8) are obtained by noting that if ν_m is defined as $P_{m,m}$, then $P_m - zB_m\nu_m$ results in a polynomial reduced in order by 1 since zB_m has a final coefficient of unity for all m (from (4) and the fact that $a_{m,0} = 1$).

From the recurrence relations, it can be seen that each $P_m(z)$ will be of the form of (2) and that each $A_m(z)$ will be of the form of (3) with $a_{m,0} = 1$. Equation (9) is proven from (8) by simply summing $P_m - P_{m-1}$ from $m = 0$ to M with $P_{-1} = 0$.

A "Two-Multiplier" Lattice Filter

In this section, the general form of the filter as a block diagram is first developed. If $X = X(z)$ is the input to $G = G(z)$, and $Y = Y(z)$ is the output, from (10),

$$Y = \sum_{m=0}^{M} \nu_m \frac{zB_mX}{A_M}. \quad (17)$$

The recursions specified by (13) and (14) can be rewritten in the form

$$\frac{A_mX}{A_M} = \frac{A_{m+1}X}{A_M} - k_m \frac{B_mX}{A_M} \quad (18)$$

$$\frac{zB_{m+1}X}{A_M} = k_m \frac{A_mX}{A_M} + \frac{B_mX}{A_M}. \quad (19)$$

Noting that $A_MX/A_M = X$ and $A_0X/A_M = zB_0X/A_M$, a block diagram for the filter is shown in Fig. 1. The details of a prototype section G_m are obtained from (18) and (19) as shown in Fig. 2. The overall implementation of $G(z)$ with these particular sections is referred to as a "two-multiplier model" since each section contains two multiplies. By making proper analogies to classical network theory [9], this filter would be defined as a lattice type of filter. It is trivial, however, to obtain a ladder representation from the lattice. Substituting (18) into (19) results in a "three-multiplier" form as shown in Fig. 3. By inspection of the figure it is possible to easily reduce it to a two-multiplier ladder form.

As can be noted from Fig. 1, each of the terms multiplying the tap parameters in the summation of (10) appear explicitly; direct multiplication by these tap parameters and a single summation will fully implement the filter in the form of (10).

The forms presented are canonical with respect to delay only. By further manipulation it is possible to obtain a form that is also canonical with respect to multiplications. It was evidently first shown by Itakura and Saito that (18) and (19) can be transformed into a one-multiplier model. In the next section the one-multiplier model is derived with a generalization that includes a sign parameter definition. The potential value of this generalization will be discussed in a later section.

One-Multiplier Implementation

Here we define a new set of sign parameters, $\epsilon_0, \epsilon_1, \cdots, \epsilon_{M-1}$. Each of these will be equal to plus one or minus one. How the sign parameters are chosen will be the topic of a later section. These are then used to obtain a set of modified tap parameters, $\hat{\nu}_0, \hat{\nu}_1, \cdots, \hat{\nu}_M$, and a set of modified polynomials by the relations

$$\hat{\nu}_m = \nu_m/\pi_m \quad (20)$$

$$\hat{A}_m(z) = \pi_m A_m(z), \quad \hat{B}_m(z) = \pi_m B_m(z) \quad (21)$$

where

$$\pi_m = \begin{cases} 1 & \text{for } m = M \\ \prod_{n=m}^{M-1} (1 + \epsilon_n k_n) & \text{for } m = 0, 1, \cdots, M-1. \end{cases} \quad (22)$$

The recursion relations of (13) then become

$$(1 + \epsilon_m k_m)\hat{A}_{m+1}(z) = \hat{A}_m(z) + k_m \hat{B}_m(z) \quad (23)$$

$$z(1 + \epsilon_m k_m)\hat{B}_{m+1}(z) = k_m \hat{A}_m(z) + \hat{B}_m(z). \quad (24)$$

From (23), $\hat{A}_m(z)$ is obtained by inspection as

$$\hat{A}_m(z) = \hat{A}_{m+1}(z) - k_m[\hat{B}_m(z) - \epsilon_m \hat{A}_{m+1}(z)]. \quad (25)$$

Substituting (25) into (24) and noting that $(1 - k_m^2) = (1 + \epsilon_m k_m)(1 - \epsilon_m k_m)$ and $\epsilon_m^2 = 1$ results in

$$z\hat{B}_{m+1}(z) = \hat{B}_m(z) - \epsilon_m k_m [\hat{B}_m(z) - \epsilon_m \hat{A}_{m+1}(z)]. \quad (26)$$

Equations (25) and (26) can be implemented as indicated in Fig. 4, where Fig. 4(a) shows the blocks in detail for $\epsilon_m = +1$ and Fig. 4(b) shows the blocks in detail for $\epsilon_m = -1$. Note that the general form of Fig. 1 is still valid with the A_m, B_m, and ν_m terms hatted since by substituting (20) through (22) into (10), $G(z)$ is equivalent to

$$G(z) = \sum_{m=0}^{M} \hat{\nu}_m \frac{z\hat{B}_m(z)}{\hat{A}_M(z)}. \quad (27)$$

From Fig. 1 (with the replacement of hatted terms), it is seen that each term needed to multiply the modified tap parameters, as indicated in (27), appears explicitly in the model; all that needs to be added are the multiplications by the tap parameters and the summation indicated.

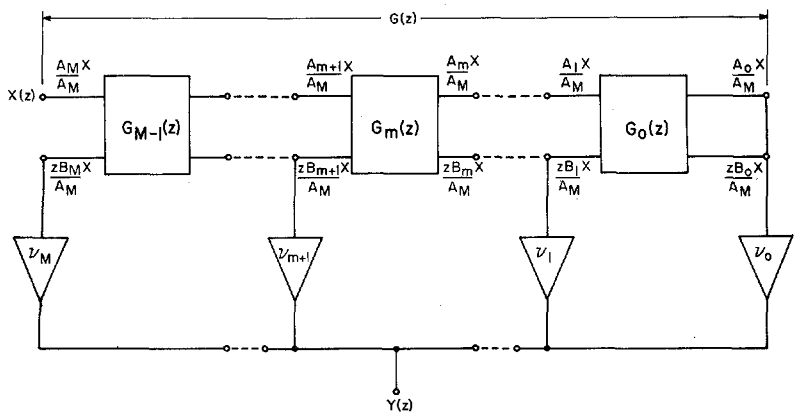

Fig. 1. Ladder and lattice form implementation.

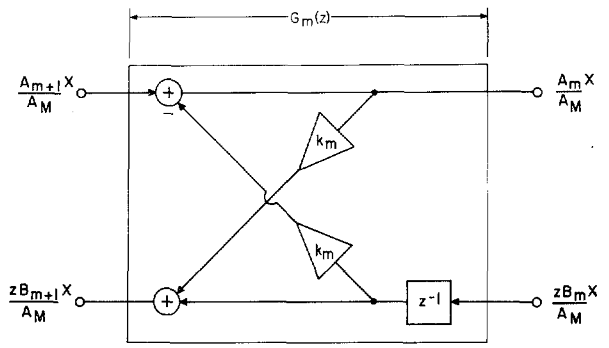

Fig. 2. Prototype filter for two-multiplier lattice model.

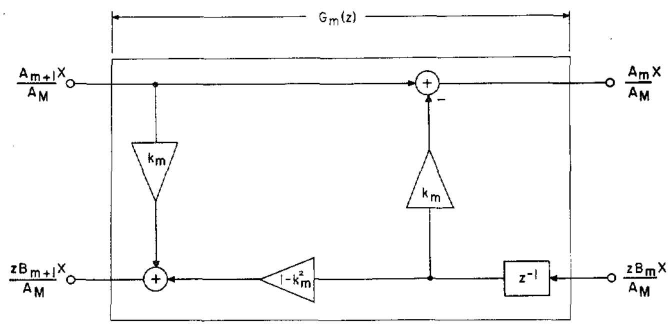

Fig. 3. Prototype filter for three multiplier ladder model.

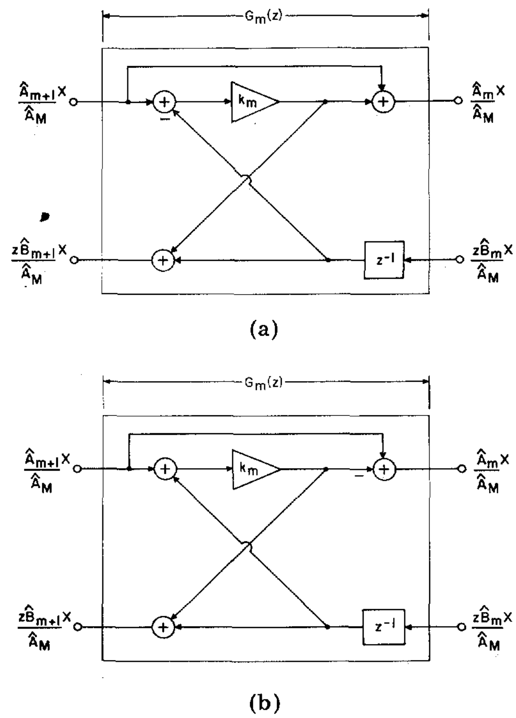

Fig. 4. Prototype filters for generalized one-multiplier lattice model. (a) $\epsilon_m = +1$. (b) $\epsilon_m = -1$.

Sensitivity

It has been noted that when poles of a filter lie at low frequencies and are near the unit circle, direct implementation of a digital filter by using a recursive approach is much worse than factoring the filter into a complex pole pair and using a cascade approach [3]. The meaning of the term "much worse" is related to how sensitive the angular location of the complex pole is to small percentage changes in the filter parameters.

Due to the complexity of the problem, no attempt is made here to derive relationships for sensitivity of the pole or zero locations with respect to small percentage variations in the filter parameters. Rather, we shall determine which calculations are the most sensitive in changing the gross behavior of the filter and, then, show how to choose the sign parameters so as to make those the calculations that are performed with the numbers having the largest magnitudes.

In linear prediction synthesis of speech, it has been observed that it is those k-parameters having the largest magnitude that are the most important. In dealing with voiced speech signals it usually is true that k_0 and k_1 have the largest effect on the end results.

Filter sensitivity to k-parameters near unit magnitude can be noted by considering the case of two separate filters being implemented, whose only difference lies in the lth k-parameter. One of the filters uses k_l and the other filter uses $k_l' = k_l + \Delta k_l$. The polynomials for the filter might be denoted by $A_n(z)$ and $A'_n(z)$ so that the polynomials are equal for $n = 0, 1, \cdots, l$. For n larger than l they will differ, and the effect of this difference will first show up in the $n = l + 1$ polynomial. It is shown in Appendix I that the difference of the log magnitudes of $A'_{l+1}(e^{j\theta})$ and $A_{l+1}(e^{j\theta})$ will oscillate between the values $\log [1 + \Delta k_l/(1 + k_l)]$ and $\log [1 - \Delta k_l/(1 - k_l)]$. Thus, the closer that the magnitude of k_l is to one, the more significant the effects of nonzero Δk_l.

As a result of this, it can be noted that the k-parameters are inherently scaled in a desirable fashion, that is, the ones that affect the filter most are the ones with the largest magnitudes. In the following sections the problem of scaling is studied in more detail.

Relative Sizes of Numerical Values

For purposes of normalization we shall define a set of z transforms $U_m(z)$, $V_m(z)$, and $W_m(z)$ by the relations

$$U_m(z) = A_m(z)X(z)/[A_M(z)\sqrt{\alpha_m}], \tag{28}$$

$$V_m(z) = B_m(z)X(z)/[A_M(z)\sqrt{\alpha_m}], \tag{29}$$

and

$$W_m(z) = [V_m(z) - \epsilon_m U_m(z)]/\sqrt{2(1 + \epsilon_m k_m)}, \tag{30}$$

and $X(z)$ represents the z transform of the input to the filter. The particular definitions correspond to normalized node values and multiplier input values for the various filter forms. The motivation for these particular normalization constant choices will be presented shortly. While it is impossible to judge the sizes of the time sequences associated with $U_m(z)$, $V_m(z)$, and $W_m(z)$ for all possible inputs, some statements can be made about their sizes based upon their spectra, their log spectra, and their rms value when the input is zero mean uncorrelated noise.

First, in terms of their spectra, it is shown in Appendix II that each of the three has the unit energy or total squared integral when $X(z) = 1$. Second, when the input time sequence is a random, stationary, uncorrelated, and zero mean process having a variance or mean square equal to σ_x^2, then it is shown in Appendix II that the time sequences associated with $U_m(z)$, $V_m(z)$, and $W_m(z)$ will all be random, stationary, zero mean process each having a variance or mean square equal to σ_x^2. Thus, in a qualitative sense, one can say that these time sequences, on the average, will have identical rms values. Third, the rather surprising property also shown in Appendix III is that the $U_m(z)$ and $V_m(z)$ form two separate sequences of mini-max approximations in terms of their log spectra.

In the two-multiplier model of Fig. 2, the node values used in the mth block are

$$\frac{A_m(z)}{A_M(z)} X(z) = U_m(z)\sqrt{\alpha_m} \tag{31}$$

and

$$\frac{B_m(z)}{A_M(z)} X(z) = V_m(z)\sqrt{\alpha_m}. \tag{32}$$

These also represent the inputs to the multipliers as seen from Fig. 2. From (15), one should note that the α_m form a sequence that decreases with increasing m. Thus the largest multiplier inputs will be in the blocks with the smallest m, those furthest from the input. This is useful in speech synthesis where k_0 and k_1 are the k-parameters closest to unit magnitude. However if the α_m cover a wide range $[\alpha_0 >> \alpha_M = 1]$, then the multiplier inputs in the blocks with large m will be a great deal smaller in size than those with small m.

In the one-multiplier models of Fig. 4, the node values are specified by

$$\frac{\hat{A}_m(z)}{\hat{A}_M(z)} X(z) = \pi_m\sqrt{\alpha_m}\, U_m(z) \tag{33}$$

and

$$\frac{\hat{B}_m(z)}{\hat{A}_m(z)} X(z) = \pi_m\sqrt{\alpha_m}\, V_m(z). \tag{34}$$

From (26) and Fig. 4, the multiplier input in the mth block is easily seen to be

$$\begin{aligned}\frac{\hat{B}_m(z) - \epsilon_m \hat{A}_{m+1}(z)}{\hat{A}_M(z)} X(z) &= \frac{\pi_{m+1}[B_m(z) - \epsilon_m A_m(z)]}{A_M(z)} X(z) \\ &= \pi_m\sqrt{\alpha_m}\sqrt{2/(1 + \epsilon_m k_m)}\, W_m(z).\end{aligned} \tag{35}$$

The product $\pi_m\sqrt{\alpha_m}$ appears in each of these expressions and is obtained by recursion from (15) and (22) as

$$\pi_m\sqrt{\alpha_m} = \prod_{n=m}^{M-1} \sqrt{(1 + \epsilon_n k_n)/(1 - \epsilon_n k_n)}. \tag{36}$$

As will be seen in the next section, appropriate choices of the sign parameters can be utilized to make this product largest when k-parameters are near one in magnitude, without requiring a monotonically decreasing sequence with increasing m, as was the case with the $\sqrt{\alpha_m}$ terms in the two-multiplier model.

As in the case of showing that $U_m(z)$, $V_m(z)$, and $W_m(z)$ represent z transforms of unit energy signals (when $X(z) = 1$) and that their time sequences will have variances equal to σ_x^2 for random inputs with variance equal to σ_x^2, it is also shown in Appendix II that the impulse response of the filter contains a total amount of energy given by

$$\int_{-\pi}^{\pi} |G(e^{j\theta})|^2 \frac{d\theta}{2\pi} = \sum_{n=0}^{M} \nu_n^2 \alpha_n, \tag{37}$$

and that when the input is the random, zero mean, uncorrelated stationary sequence with variance σ_x^2, the filter output y_n will be random and zero mean, with a variance given by

$$\sigma_y^2 = \sigma_x^2 \sum_{n=0}^{M} \nu_n^2 \alpha_n. \tag{38}$$

Sign Parameter Choices

Numerous criteria could be utilized to pick optimal values of the sign parameters, based upon the user's definition of optimal. In order to obtain an effective filter over differing types of input, we shall choose our criteria in terms of rms values for the case where the input signal is random, uncorrelated, stationary, and zero mean. If the variance of the input signal is σ_x^2, so that its rms value is σ_x; then the results of Appendix II show that the rms values of the time sequences associated with $\hat{A}_m(z)X(z)/\hat{A}_M(z)$ and $\hat{B}_m(z)X(z)/\hat{A}_M(z)$ will both be given by $\pi_m\sqrt{\alpha_m}$, the rms value of the multiplier input in the mth section will be given by $\pi_m\sqrt{\alpha_m}\,\sigma_x\sqrt{2/(1+\epsilon_m k_m)}$, and that the rms value of the output of the overall filter from (38) is

$$\sigma_y = \sigma_x \left[\sum_{n=0}^{M} \nu_n^2 \alpha_n \right]^{1/2}. \tag{39}$$

One optimization criterion based upon rms values is to require that $\pi_m\sqrt{\alpha_m}$ be largest for the k-parameter nearest unity in magnitude. Then the sign parameters are recursively found by requiring that $\pi_m\sqrt{\alpha_m}$ be as large as possible without exceeding the maximum value. If that maximum occurs for $m = l$, then the recursion relation proceeds for $m = l-1, l-2, \cdots, 0$ and again for $m = l, l+2, \cdots, M-1$. This relation is simple to effect because of the fact that

$$\frac{\pi_m\sqrt{\alpha_m}}{\pi_{m+1}\sqrt{\alpha_{m+1}}} = \sqrt{(1+\epsilon_m k_m)/(1-\epsilon_m k_m)}. \tag{40}$$

By simply changing the sign parameter, this ratio can always be made smaller than or larger than one.

Once the sign parameters are chosen, from the aforementioned or some other rule, the rms values of the multiplier inputs are simply evaluated, in addition to the rms value of the output from (39). Scaling of the input signal can be carried out by observing the maximum rms value appearing in the filter and applying some sort of rule to relate rms value to probability of this peak value exceeding one. For example, if the input is Gaussian and scaled so that the peak rms value appearing in the filter is $\frac{1}{3}$, then normal distribution tables show that the probability of that signal's magnitude exceeding unity will be given by 0.0026.

To implement this approach, the following algorithm is applied. Assume that k_l has the largest magnitude among the k_m, $m = 0, 1, \cdots, M-1$. Define the quantities

$$Q_m = \frac{\pi_m\sqrt{\alpha_m}}{\pi_l\sqrt{\alpha_l}} \tag{41}$$

and

$$q_m = \sqrt{(1+|k_m|)/(1-|k_m|)}. \tag{42}$$

Then $Q_l = 1$. Each Q_m should be as large as possible without exceeding Q_l.

By combining (39) through (42),

$$Q_m/Q_{m+1} = \begin{cases} q_m & \text{if } \epsilon_m = \operatorname{sgn}(k_m) \\ 1/q_m & \text{if } \epsilon_m = -\operatorname{sgn}(k_m). \end{cases} \tag{43}$$

For $m = l-1, l-2, \cdots, 0$, choose $\epsilon_m = \operatorname{sgn}(k_m)$ if $Q_{m+1} < 1/q_m$. Otherwise choose $\epsilon_m = -\operatorname{sgn}(k_m)$. For $m = l, l+1, \cdots, M-1$, choose $\epsilon_m = -\operatorname{sgn}(k_m)$ if $Q_m < 1/q_m$, otherwise choose $\epsilon_m = \operatorname{sgn}(k_m)$. The effect of these choices is to maximize Q_m at each node without exceeding unity.

An Example

As an illustrative example, the third-order Chebyshev low-pass filter used by Mitra and Sherwood [2] is synthesized in a lattice form. In this case, $M = 3$ and

$$P_3(z) = 0.0154 + 0.0462\,z^{-1} + 0.0462\,z^{-2} + 0.0154\,z^{-3}, \tag{44}$$

and

$$A_3(z) = 1 - 1.990\,z^{-1} + 1.572\,z^{-2} - 0.4583\,z^{-3}. \tag{45}$$

In order to illustrate the simplicity of the approach, and also to provide enough details so that a user could check a computer program, all calculations were carried out using only paper, pencil and a calculator (HP-35).

First, from (5) and (7),

$$k_2 = -0.4583 \text{ and } \nu_3 = 0.0154.$$

From (4),

$$zB_3(z) = -0.4583 + 1.572\,z^{-1} - 1.990\,z^{-2} + z^{-3}.$$

From (6) and (8) with $m = 3$,

$$P_2(z) = 0.02245782 + 0.0219912\,z^{-1} + 0.076846\,z^{-2}$$

and

$$A_2(z) = 1 - 1.607107469\,z^{-1} + 0.835462647\,z^{-2}.$$

Repeating the procedure from (4) through (8) with $m = 2$,

$$k_1 = 0.835462647$$

and

$$\nu_2 = 0.076846$$

with the reduced order polynomials

$$P_1(z) = -0.0417441466 + 0.1454909806\, z^{-1}$$

and

$$A_1(z) = 1 - 0.8755871285\, z^{-1}.$$

Using $m = 1$ in the recursion relations gives

$$k_0 = -0.8755871285$$

and

$$\nu_1 = 0.1454909806$$

with the final polynomials

$$P_0(z) = 0.0856458833$$

$$A_0(z) = 1$$

so that $\nu_0 = 0.0856458833$. This completes the evaluation of the k-parameters and the tap parameters. Since all the k-parameters have magnitudes less than one, the filter is stable. These results define a two-multiplier lattice form. Next, the one-multiplier model with optimal sign choice is presented.

To obtain the sign parameters, the algorithm of the preceding section is applied. Since k_0 has the largest magnitude, (41) and (42) result in $l = 0$,

$$Q_0 = 1$$

and

$$q_0 = 3.882719036.$$

As Q_0 is more than $1/q_0$, choose $\epsilon_0 = \text{sgn}(k_0) = -1$. Therefore

$$Q_1 = Q_0/q_0 = 0.2575514712.$$

From (42),

$$q_1 = 3.339954283.$$

As Q_1 is less than $1/q_1$, choose $\epsilon_1 = -\text{sgn}(k_1) = -1$. Therefore

$$Q_2 = Q_1 q_1 = 0.8602101393.$$

From (42),

$$q_2 = 1.640756072.$$

As Q_2 is more than $1/q_2$, choose $\epsilon_2 = \text{sgn}(k_2) = -1$.

Having found the sign parameters, the multipliers π_m can be obtained from (22):

$$\begin{aligned}\pi_3 &= 1\\ \pi_2 &= 1.4583\\ \pi_1 &= 0.2399448219\\ \pi_0 &= 0.4500374194.\end{aligned}$$

Using these with (20), the modified tap parameters are obtained as

$$\begin{aligned}\hat{\nu}_3 &= 0.0154\\ \hat{\nu}_2 &= 0.0526956044\\ \hat{\nu}_1 &= 0.6063518248\\ \hat{\nu}_0 &= 0.1903083602.\end{aligned}$$

This completes the design procedure for the one-multiplier model. The implementation is shown in Fig. 5.

In order to obtain an idea about relative sizes of the numbers being utilized, one can utilize results of the previous section and Appendix II. In particular, at stage m the nodes $X\hat{A}_m/\hat{A}_M$, $X\hat{B}_m/\hat{A}_M$, and $Xz\hat{B}_m/\hat{A}_M$ will all have rms values given by $\pi_m\sqrt{\alpha_m}\,\sigma_x$ for a random input with variance equal to σ_x^2.

The values of α_m are most easily evaluated from (15). Starting with $\alpha_M = 1$,

$$\begin{aligned}\alpha_3 &= 1\\ \alpha_2 &= 1.265885102\\ \alpha_1 &= 4.191642467\\ \alpha_0 &= 17.96311599.\end{aligned}$$

This then gives the results for the node values

$$\begin{aligned}\pi_3\sqrt{\alpha_3}\,\sigma_x &= \sigma_x\\ \pi_2\sqrt{\alpha_2}\,\sigma_x &= 1.640563849\,\sigma_x\\ \pi_1\sqrt{\alpha_1}\,\sigma_x &= 0.4912510573\,\sigma_x\\ \pi_0\sqrt{\alpha_0}\,\sigma_x &= 1.907389831\,\sigma_x.\end{aligned}$$

The multiplier inputs were shown to have rms values given by $\pi_m\sqrt{\alpha_m}\,\sigma_x\sqrt{2/(1+\epsilon_m k_m)}$. Therefore,

$$\begin{aligned}\pi_0\sqrt{\alpha_0}\,\sigma_x\sqrt{2/(1+\epsilon_0 k_0)} &= 1.969635389\,\sigma_x\\ \pi_1\sqrt{\alpha_1}\,\sigma_x\sqrt{2/(1+\epsilon_1 k_1)} &= 1.712719502\,\sigma_x\\ \pi_2\sqrt{\alpha_2}\,\sigma_x\sqrt{2/(1+\epsilon_2 k_2)} &= 1.921253614\,\sigma_x.\end{aligned}$$

The rms value of the output in this case is found simply from (39) to be given by

$$\sigma_y = 0.4777060586\,\sigma_x.$$

Conclusions

Several different digital lattice and ladder forms have been developed starting from the standard direct form. The procedure was shown to have several desirable features as follows.

1) All stable recursive filters can be transformed into the forms presented. Other procedures developed thus far have undesirable restrictions such as all poles being required to be in the left half of the z-plane.

2) A built-in stability test exists within the synthesis process. If any k-parameter magnitude exceeds or equals unity, the filter is unstable. Otherwise, it is stable.

3) The synthesis procedure is very efficient. Only on the order of M^2 operations are necessary for the synthesis. No polynomial root solvers or explicit simultaneous equation solvers are necessary.

A new procedure was also introduced for scaling nodes of a particular one-multiplier lattice form.

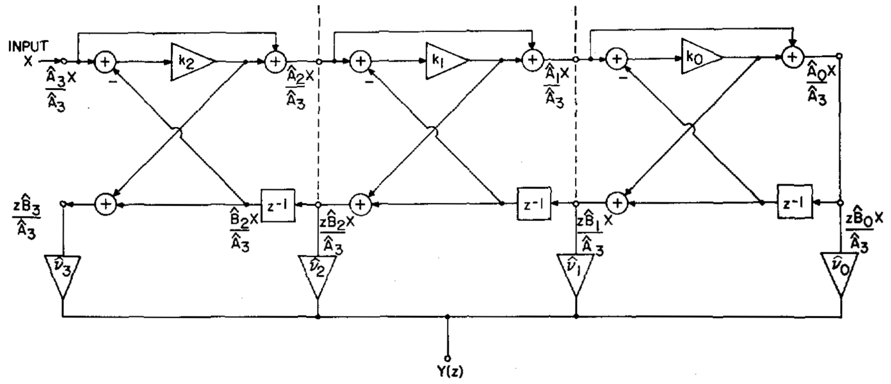

Fig. 5. One-multiplier implementation of the example.

Presently, the application of this scaling procedure is being studied for finite word length digital filter implementation.

Appendix I

Let $A_m(z)$ satisfy the recursion relation (13) where $B_m(z)$ is from (4) given by

$$B_m(z) = z^{-(m+1)} A_m(1/z), \tag{I-1}$$

$A_0(z) = 1$, and $m = 0, 1, \cdots, M-1$. Let $A'_m(z)$ satisfy a similar set of recursion relations whose only difference lies in the fact that k_l is replaced by $k'_l = k_l + \Delta k_l$. Thus $A'_m(z) = A_m(z)$ for $m = 0, 1, \cdots, l$. Combining (13) and (I-1) with $z = e^{j\theta}$, one finds that on the unit circle

$$\frac{A'_{l+1}(e^{j\theta})}{A_{l+1}(e^{j\theta})} = \frac{A_l(e^{j\theta}) + k'_l e^{-(l+1)j\theta} A_l(e^{-j\theta})}{A_l(e^{j\theta}) + k_l e^{-(l+1)j\theta} A_l(e^{-j\theta})}.$$

This can be rewritten as

$$\frac{A'_{l+1}(e^{j\theta})}{A_{l+1}(e^{j\theta})} = \frac{1 + k'_l e^{-j\Psi}}{1 + k_l e^{-j\Psi}} \tag{I-2}$$

where Ψ is given by

$$\Psi = \Psi(\theta) = (l+1)\theta + 2 \arg [A_l(e^{j\theta})]. \tag{I-3}$$

By using the form of $A_l(z)$ from (3) and the fact that it has all of its zeros within the unit circle, one can show that

$$\Psi(0) = 0 \tag{I-4}$$

$$\Psi(\pi) = (l+1)\pi \tag{I-5}$$

$$\frac{d\Psi}{d\theta} > 1. \tag{I-6}$$

The ratio of (I-2) has a magnitude that oscillates between the values $(1 + k'_l)/(1 + k_l)$ when Ψ is an even multiple of π, and $(1 - k'_l)/(1 - k_l)$ when Ψ is an odd multiple of π. As Ψ is a monotonically increasing function of θ, these extreme values will all be met.

In terms of the log spectra, this yields the result that

$$\log |A'_{l+1}(e^{j\theta})| - \log |A_{l+1}(e^{j\theta})|$$

will oscillate between the values

$$\log [(1 + k'_l)/(1 + k_l)] = \log [1 + \Delta k_l/(1 + k_l)]$$

and

$$\log [(1 - k'_l)/(1 - k_l)] = \log [1 - \Delta k_l/(1 - k_l)].$$

Appendix II

Let $U_m(z)$, $V_m(z)$, and $W_m(z)$ be as given in (28)–(30). When $X(z) = 1$, one can apply (11) and (12) to obtain their energies in terms of inner products. In this way we find that

$$\int_{-\pi}^{\pi} |U_m(e^{j\theta})|^2 \frac{d\theta}{2\pi} = \langle A_m(z)/\sqrt{\alpha_m}, A_m(z)/\sqrt{\alpha_m} \rangle$$

$$= \frac{\langle A_m(z), A_m(z) \rangle}{\alpha_m}.$$

From (14) one should note that this is simply one. In exactly the same manner, one can show that

$$\int_{-\pi}^{\pi} |V_m(e^{j\theta})|^2 \frac{d\theta}{2\pi} = 1.$$

One more step is needed in the handling of $W_m(z)$, for by the same applications of scalar products one finds that

$$\int_{-\pi}^{\pi} |W_m(e^{j\theta})|^2 \frac{d\theta}{2\pi}$$

$$= \frac{\langle B_m(z) - \epsilon_m A_m(z), B_m(z) - \epsilon_m A_m(z) \rangle}{2\alpha_m (1 + \epsilon_m k_m)}.$$

If the numerator terms are multiplied out and use is made of (14) and (16), one finds that this quantity also equals unity. As a result,

$$\int_{-\pi}^{\pi} |U_m(e^{j\theta})|^2 \frac{d\theta}{2\pi} = \int_{-\pi}^{\pi} |V_m(e^{j\theta})|^2 \frac{d\theta}{2\pi}$$

$$= \int_{-\pi}^{\pi} |W_m(e^{j\theta})|^2 \frac{d\theta}{2\pi} = 1. \tag{II-1}$$

In the same manner, one can obtain the energy of the overall filter output when $X(z) = 1$, by expressing the energy of $G(z)$ as a scalar product. From (1), (11), and (12)

$$I = \int_{-\pi}^{\pi} |G(e^{j\theta})|^2 \frac{d\theta}{2\pi} = \langle P_M(z), P_M(z) \rangle.$$

If $P_M(z)$ is expressed as in (9), the orthogonality of the polynomials $zB_m(z)$ can combine with (14) to give

$$I = \sum_{m=0}^{M} \nu_m^2 \alpha_m. \tag{II-2}$$

The symbolic notation $H(z)X(z)$ represents the z transform of the output of a filter whose input time sequence has a z transform and whose transfer function is $H(z)$. When we speak of the time sequence associated with $H(z)X(z)$ we imply the convolution

$$s_k = \sum_{n=-\infty}^{\infty} h_n x_{k-n} \tag{II-3}$$

where h_k represents the unit sample response to the filter and its z transform is $H(z)$. Equation (II-3) is considered to apply, even in cases where the input sequence, x_k, does not have a convergent z transform. In such cases, referring to a time sequence associated with $H(z)X(z)$ will be understood to imply (II-3), even when $X(z)$ is not convergent.

If the input sequence is random and uncorrelated, then the output sequence will also be random. Using the symbol E to denote the expected value (expectation or mean) of a random variable, one finds that the mean square of the sequence s_k will be given by

$$\sigma_s^2 = E[s_k^2] = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} h_n h_m E[x_{k-n} x_{k-m}]. \tag{II-4}$$

If the input is stationary and uncorrelated,

$$E[x_{k-n} x_{k-m}] = \sigma_x^2 \delta_{nm}$$

where σ_x^2 is the input mean square and δ_{nm} is the Kronecker delta. In this case, (II-4) can be reduced to a single summation, and by applying z-transform properties one finds

$$\sigma_s^2 = \sigma_x^2 \sum_{n=-\infty}^{\infty} h_n^2 = \sigma_x^2 \int_{-\pi}^{\pi} |H(e^{j\theta})|^2 \frac{d\theta}{2\pi}. \tag{II-5}$$

By taking the definitions of $U_m(z)$, $V_m(z)$, and $W_m(z)$, along with the unit energy results of (II-1), one can then use (II-5) to note that when the input time sequence is random, stationary, and uncorrelated, the time sequences associated with $U_m(z)$, $V_m(z)$, and $W_m(z)$ will all have a mean square equal to the input mean square and, thus, rms values equal to the input rms value, σ_x. In addition, the rms output value of the overall filter is equal to $\sigma_x \sqrt{I}$ where I is found from (II-2).

These results can be utilized with the various expressions for the z transforms of the node values and multiplier inputs by applying (31) through (35). These are summarized in Table I.

TABLE I
The rms Values at Nodes and Multiplier Inputs for One- and Two-Multiplier Models

Model	Notation	rms Value
Two multiplier Node values and multiplier inputs	$A_m X/A_M$ $B_m X/A_M$ $zB_m X/A_M$	$(\alpha_m \sigma_x^2)^{1/2}$
One multiplier Node values	$\hat{A}_m X/\hat{A}_M$ $\hat{B}_m X/\hat{A}_M$ $z\hat{B}_m X/\hat{A}_M$	$\Pi_m (\alpha_m \sigma_x^2)^{1/2}$
One multiplier multiplier input	$(\hat{B}_m - \epsilon_m \hat{A}_{m+1})X/\hat{A}_m$	$\Pi_m \left(\dfrac{2\alpha_m \sigma_x^2}{1+\epsilon_m k_m}\right)^{1/2}$

Appendix III

We shall consider, here, only the $U_m(z)$, given by (28), for in the frequency domain, the only difference between $U_m(e^{j\theta})$ and $V_m(e^{j\theta})$ lies in the phase since

$$B_m(e^{j\theta}) = e^{-j(m+1)\theta} A_m^*(e^{j\theta}). \tag{III-1}$$

The ratio of any two successive $U_m(z)$ terms can be simply expressed as

$$\frac{U_{m+1}(z)}{U_m(z)} = \sqrt{\alpha_m/\alpha_{m+1}} \, \frac{A_{m+1}(z)}{A_m(z)}.$$

This can be expressed in the frequency domain by applying (I-2) with $l = m$, $k_l' = 0$, so that $A_{l+1}'(e^{j\theta}) = A_m(e^{j\theta})$. Combining this result with the recursion relation of (18), one obtains

$$\frac{U_{m+1}(e^{j\theta})}{U_m(e^{j\theta})} = \frac{1 + k_m e^{-j\Psi}}{[1 - k_m^2]^{1/2}} \tag{III-2}$$

where Ψ is given by (I-3) with $l = m$.

The magnitude of the ratio of (III-2) oscillates between $\sqrt{(1+k_m)/(1-k_m)}$ when Ψ is an even multiple of π, and $\sqrt{(1-k_m)/(1+k_m)}$ when Ψ is an odd multiple of π. From Appendix I, it should be noted that Ψ is a monotonically increasing function of θ, going from $\Psi(0) = 0$ to $\Psi(\pi) = (m+1)\pi$. Thus, as θ goes from zero to π, the ratio of (III-2) will hit its extreme values at exactly $m + 2$ points, two of them being the end points $\theta = 0$ and $\theta = \pi$.

In terms of the log spectrum, this yields the result that the difference

$$\log |U_{m+1}(e^{j\theta})| - \log |U_m(e^{j\theta})|$$

hits the extreme values

$$\pm\tfrac{1}{2} \log [(1+k_m)/(1-k_m)]$$

exactly $m + 2$ times. Thus the surprising result is shown that $U_m(z)$, and similarly $V_m(z)$, form min-max sequences. In fact, this result can be used to show that $A_m(z)/\sqrt{\alpha_m}$ is the mth order polynomial having all its roots inside the unit circle which is the mini-max approximation in terms of the log spectrum to $A_{m+1}(z)/\sqrt{\alpha_{m+1}}$.

The phase of the ratio of (III-2) can also be studied, and it can be shown that the phase of the ratio will hit its extreme values, $\pm \sin^{-1}(k_m)$, exactly $m+1$ times as θ goes from 0 to π.

References

[1] S. K. Mitra and R. J. Sherwood, "Canonic realizations of digital filters using the continued fraction expansion," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 185–194, Aug. 1972.

[2] ——, "Digital ladder networks," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 30–36, Feb. 1973.

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

[4] A. Fettweis, "Digital filter structures related to classical filter networks," *Arch. Elek. Übertragung.*, vol. 25, pp. 79–89, Feb. 1971.

[5] ——, "Some principles of designing digital filters imitating classical filter structures," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 314–316, Mar. 1971.

[6] R. Crochiere, "Digital ladder filter structures and coefficient sensitivity," Res. Lab. Electronics Rep. 103, Mass. Inst. Tech., Cambridge, Oct. 15, 1971.

[7] F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," presented at the 7th Int. Congr. Acoust., Paper 25C-1, Budapest, 1971.

[8] J. D. Markel and A. H. Gray, Jr., "On autocorrelation equations with applications to speech analysis," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 69–79, Apr. 1973.

[9] E. A. Guillemin, *Synthesis of Passive Networks.* New York: Wiley, 1957.

Considerations of the Padé Approximant Technique in the Synthesis of Recursive Digital Filters

FRANCIS BROPHY and ANDRES C. SALAZAR, Member, IEEE

Abstract—The Padé approximant technique provides a quick design of recursive digital filters. An added advantage of the technique lies in that spectrum shaping requirements as well as linear phase constraints can be handled easily, even for higher order filters. This is important in supplying initial guesses of the filter parameters to iterative routines that would then seek a locally optimal design solution. These advantages are among those discussed in a partly tutorial presentation of the technique that relates to filter needs found in data transmission systems. In addition, the question of stability is treated and a new criterion is presented. The criterion provides sufficient conditions in establishing stability for a filter designed by using the Padé approximant technique.

I. Introduction

The design of spectrum-shaping recursive digital filters in the z-plane often requires the use of a routine that calculates the extremum of an object function of several variables. The function is generally nonlinear and positive definite and indicates the "closeness" of the designed spectrum to the desired spectrum. In some cases, depending on the complexity of the function, the number of iterations or even convergence to an extremum is dependent on the initial guess for the α and β (feedforward and feedback) parameters.

By working in the time domain the degrees of freedom available can be used to match a set of time samples exactly, thus reducing the design to the solution of a linear system of equations. While this approach, call the Padé approximant,[1] does not lead to a locally optimal solution as an iterative technique would, it nevertheless provides a viable solution in a fraction of the time.

In the following we show how the Padé approximant technique can yield a simple digital filter design for spectrum shaping networks with linear (or nonlinear, if so desired) phase constraints often required in data transmission systems. The problem of stability is discussed and sufficient conditions are given to ensure that the design procedure will not lead to an unstable filter.

II. Padé Approximate in Digital Filter Design

Let $H(\omega)$ denote in the interval $[-2\pi W, 2\pi W]$ the bounded filter amplitude characteristic that is to be synthesized. Since $H(\omega) \in L_p[-2\pi W, 2\pi W]$, $p \geqslant 1$ it has a Fourier series expansion

$$H(\omega) = \sum h_n e^{-jn\omega/2W} \tag{1}$$

Manuscript received December 28, 1972; revised June 29, 1973.

The authors are with Bell Laboratories, Holmdel, N.J. 07733.

[1] "Prony's method" is related to the Padé approximant technique through a transformation of variables (see [1] for more details).