Exact Synthesis of LDI and LDD Ladder Filters

E. S. K. LIU, L. E. TURNER, AND L. T. BRUTON

Abstract—The low sensitivity property of lossless discrete integrator (LDI) low-pass ladder filters is shown to be preserved in lossless discrete differentiator (LDD) high-pass ladder filters [1]. The exact design method for LDI ladder filters given in [2] is further developed by introducing a set of closed-form design formulas for digital all-pole Chebyshev transfer functions. A new technique for improving the numerical accuracy in the synthesis procedure is introduced. Finally, a comprehensive LDI ladder filter design program is presented.

I. Introduction

LOSSLESS discrete integrator (LDI) ladder filters [1] have become a popular filter structure in the field of signal processing [2]–[8]. The low coefficient sensitivity and the simplicity of the structure make it suitable for constructing high quality digital filters [1], [2]. The LDI concept has also been successfully applied to switched-capacitor filter networks [5]–[8]. Many monolithic switched capacitor LDI ladder filters have been fabricated [5], [6], Low sensitivity digital low-pass ladder filters can be constructed by using the lossless discrete integrator (LDI) $1/(z^{1/2}-z^{-1/2})$ as a basic building block. Similarly, low sensitivity digital high-pass filters can be constructed by using the lossless discrete differentiator (LDD) $1/(z^{1/2}+z^{-1/2})$ as a basic building block [1]. A transformation is introduced so that an LDI low-pass ladder filter can be transformed to an LDD high-pass ladder filter. The topology of the signal flow graph is preserved in this transformation so that the low sensitivity property of the LDI ladder filter is preserved in the transformed LDD ladder filter.

Most of the discussion of LDI ladder filters in the literature is based on the original approximate design technique given in [1]. By using the LDI transformation $s \rightarrow (z^{1/2}-z^{-1/2})/T$ and then carrying out $z^{1/2}$ impedance scaling throughout the digital filter signal flow graph, one ends up with ladder terminations of half unit delays [1, 3]. Since these half unit delays are not realizable, they have to be replaced by either unit delays or no delays. These half unit delay replacements introduce error to the frequency response characteristic of the filter [1], [3], [7]. A number of methods have been devised to eliminate this error yielding the exact desired frequency response [2], [7], [8], [15].

Manuscript received December 15, 1982.
The authors are with the Department of Electrical Engineering, University of Calgary, Calgary, Alta., Canada T2N 1N4.

The method reported in [2] is further developed in this paper. The first step in this design procedure requires the determination of a digital all-pole transfer function. Digital Chebyshev transfer functions designed by applying bilinear transformation to analog Chebyshev transfer functions possess a number of zeros [9], [10] so they are not suitable for LDI ladder filter implementation. On the other hand, all-pole digital transfer functions were traditionally designed by computer-aided iterative methods [9], [10]. This paper introduces a set of closed form design formulas for digital all-pole Chebyshev transfer functions.

A numerical problem has been pointed out in [2]. The synthesis procedure involves the factorization of a polynomial product $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ to the polynomials $\boldsymbol{K}(z)$ and $\boldsymbol{K}(z^{-1})$. A straightforward method for doing this is to find all the zeros of the polynomial product $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ and then arbitrarily assign the zeros inside the unit circle to $\boldsymbol{K}(z)$. Unfortunately, the closely paired or highly clustered zeros of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ renders the usual root-finding algorithms inaccurate.

In this paper, new factorization and root-squaring techniques are developed to improve the numerical accuracy of the design. Some other synthesis equations are reformulated in order to facilitate computer programming. Finally, a comprehensive LDI ladder filter design program together with two design examples are presented.

II. Theory and Design Procedure

In order to be self-contained, this section summerizes the theory and the design equations given in [2] for the synthesis of an LDI ladder filter.

An LDI ladder filter can be considered to be a doubly terminated two-pair network, as shown in Fig. 1.

Disregarding the internal structure, the two pair network can be characterized by a chain matrix equation

$$\begin{bmatrix} U_1(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} A(z) & B(z) \\ C(z) & D(z) \end{bmatrix} \begin{bmatrix} Y_2(z) \\ U_2(z) \end{bmatrix} \tag{1}$$

with the terminal nodes being constrained by

$$U_1(z) = U(z) + z^{-1/2} Y_1(z)$$

and

$$U_2(z) = -z^{-1/2} Y_2(z). \tag{2}$$

From (1) and (2), the reciprocal of the transfer function is

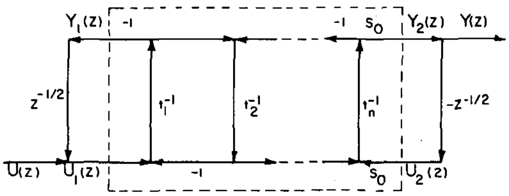


Fig. 1. LDI ladder filter as a doubly terminated two-pair network.

given by

$$[H(z)]^{-1} = H(z) = \frac{U(z)}{Y(z)} = \frac{U(z)}{Y_2(z)}$$
$$= [A(z) - z^{-1/2}B(z)]$$
$$- z^{-1/2}[C(z) - z^{-1/2}D(z)]. \qquad (3)$$

In order to determine the chain matrix elements $A(z)$, $B(z)$, $C(z)$, and $D(z)$, the following functions are introduced:

$$E(z) = A(z) - z^{-1/2}B(z) \qquad (4)$$
$$F(z) = C(z) - z^{-1/2}D(z). \qquad (5)$$

Substituting (4) and (5) in (3) yields

$$\boldsymbol{H}(z) = E(z) - z^{-1/2}F(z). \qquad (6)$$

If we define an auxiliary function

$$\boldsymbol{K}(z) = E(z) + z^{1/2}F(z) \qquad (7)$$

then $E(z)$ and $F(z)$ can be found by solving (6) and (7), which yields

$$E(z) = \frac{z^{-1/2}\boldsymbol{K}(z) + z^{1/2}\boldsymbol{H}(z)}{z^{1/2} + z^{-1/2}} \qquad (8)$$

and

$$F(z) = \frac{\boldsymbol{K}(z) - \boldsymbol{H}(z)}{z^{1/2} + z^{-1/2}}. \qquad (9)$$

The auxiliary function $\boldsymbol{K}(z)$ is to be determined in such a way that the magnitude-squared transfer function of the filter is insensitive to the multiplier coefficient perturbations. This can be accomplished by imposing the following constraints to the chain matrix elements

$$A(z) = \mp A(z^{-1})$$
$$B(z) = \pm B(z^{-1})$$
$$C(z) = \pm C(z^{-1})$$
$$D(z) = \mp D(z^{-1})$$

and

$$A(z)D(z) - B(z)C(z) = (-1)^n = \mp 1 \qquad (10)$$

where the upper signs correspond to odd order filters and the lower signs correspond to even-order filters. From (3)–(10), it follows that

$$\boldsymbol{K}(z)\boldsymbol{K}(z^{-1}) - \boldsymbol{H}(z)\boldsymbol{H}(z^{-1}) = -(z^{1/2} + z^{-1/2})^2. \qquad (11)$$

It is shown in the original paper [2] that $A(z)$ must be a polynomial in w of order n, where $w = z^{1/2} - z^{-1/2}$. Similarly, $B(z)$ and $C(z)$ are polynomials in w of order $n-1$ and $D(z)$ is a polynomial in w of order $n-2$. In addition to this, each polynomial contains either odd or even powers of w only. These polynomials are called image polynomials.

Referring to Fig. 1, the two pair network described by the chain matrix is a cascade of n ladder sections

$$\begin{bmatrix} A(z) & B(z) \\ C(z) & D(z) \end{bmatrix} \equiv \begin{bmatrix} A_n(z) & B_n(z) \\ C_n(z) & D_n(z) \end{bmatrix}$$
$$= \begin{bmatrix} t_1(z) & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} t_2(z) & -1 \\ -1 & 0 \end{bmatrix} \cdots$$
$$\begin{bmatrix} t_n(z) & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} s_0^{-1} & 0 \\ 0 & s_0 \end{bmatrix}$$

where

$$t_i(z) = s_i^{-1}w = s_i^{-1}(z^{1/2} - z^{-1/2}), \qquad 1 \leqslant i \leqslant n \qquad (12)$$

and s_i corresponds to the ith ladder section multiplier value and s_0 is the rightmost ladder branch multiplier value. The s_i's can be found by using synthetic division n times as follows:

$$\frac{A_i(z)}{C_i(z)} = -s_{n-i+1}^{-1}w + \frac{R_i(z)}{C_i(z)}$$
$$A_{i-1}(z) = -C_i(z)$$

and

$$C_{i-1}(z) = -R_i(z), \qquad i = n, n-1, \cdots, 2, 1. \qquad (13)$$

This procedure is referred to as ladder decomposition.

The above outline of the theory is actually oversimplified. For more detail, refer to the original paper [2].

Now, the design procedure for an LDI ladder filter is listed stepwise:

1) Determine the reciprocal of the transfer function $\boldsymbol{H}(z)$.

2) Calculate $\boldsymbol{H}(z)\boldsymbol{H}(z^{-1})$ from $\boldsymbol{H}(z)$, find $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ from $\boldsymbol{H}(z)\boldsymbol{H}(z^{-1})$ by using (11) and then factorize $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ to $\boldsymbol{K}(z)$ and $\boldsymbol{K}(z^{-1})$.

3) Find $E(z)$ and $F(z)$ from $\boldsymbol{H}(z)$ and $\boldsymbol{K}(z)$ using (8) and (9) and then determine the chain matrix elements $A(z)$, $B(z)$, $C(z)$, and $D(z)$ from $E(z)$ and $F(z)$.

4) Carry out the ladder decomposition given by (13) in order to determine the LDI ladder section multiplier and the rightmost ladder branch multiplier values.

These design steps will be discussed in detail in the following sections.

III. Digital LDI and LDD Ladder Filters

The signal flow graph of a third-order LDI ladder filter is shown in Fig. 2. By carrying out $z^{1/2}$ impedance scaling, we end up with the signal flow graph of Fig. 3. Note that no approximation has been made in the signal flow graph manipulation and therefore the synthesis procedure yields exactly the desired frequency response.

It has been shown [1], [2] that the lossless discrete integrator $1/(z^{1/2} - z^{-1/2})$ forms the basic building block

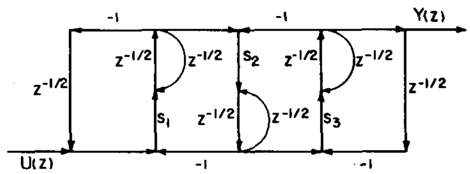


Fig. 2. Signal flow graph of a third-order LDI ladder filter.

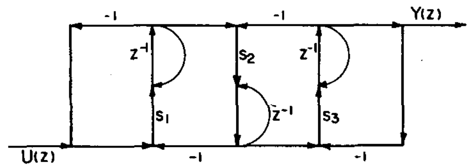


Fig. 3. Signal flow graph of a third-order LDI ladder filter with $z^{-1/2}$ eliminated.

of low sensitivity digital low-pass ladder filters. Similarly, another operator called the lossless discrete differentiator (LDD) $1/(z^{1/2}+z^{-1/2})$ is expected to be able to serve as the basic building block of low sensitivity digital high-pass ladder filters. This lossless discrete differentiator is equivalent to the one proposed in [1]. Indeed, a simple transformation can be used to transform an LDI low-pass ladder filter to an LDD high-pass ladder filter.

Consider the well-known low-pass to high-pass transformation:

$$z \to -z$$

thus

$$z^{1/2} \to jz^{1/2}, \qquad z^{-1/2} \to -jz^{-1/2}$$

and

$$(z^{1/2}-z^{-1/2}) \to j(z^{1/2}+z^{-1/2}). \tag{14}$$

Letting $z=e^{j\omega}$ on the LHS and $z=e^{j\omega'}$ on the RHS of (14) yields

$$\omega' = \pi - \omega$$

where ω is the original frequency variable and ω' is the transformed frequency variable. The new transformed transfer function is thus related to the original transfer function by

$$H'(e^{j\omega'}) = H(e^{j(\pi-\omega)}). \tag{15}$$

Obviously, $H'(e^{j\omega'})$ is a mirror image of $H(e^{j\omega})$ with the mirror located at $\omega=\pi/2$.

Substituting $z^{-1/2}$ by $-jz^{-1/2}$ in Fig. 2 and manipulating it appropriately yields the signal flow graph in Fig. 4. Thus the transformation only changes the signs of certain branches in the ladder filter signal flow graph. The overall topology remains unchanged and thus the low sensitivity property of the LDI ladder filter is preserved in the LDD ladder filter.

It is obvious from (14) that if $p_i = r_i e^{j\vartheta_i}$ is a pole of the LDI low-pass ladder filter, then $p_i' = -r_i e^{j\vartheta_i}$ is a pole of the LDD high-pass ladder filter. Therefore, (14) transforms a stable LDI low-pass ladder filter to a stable LDD high-pass ladder filter.

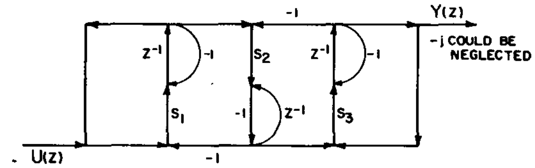


Fig. 4. Signal flow graph of a third-order LDD ladder filter with $z^{-1/2}$ eliminated.

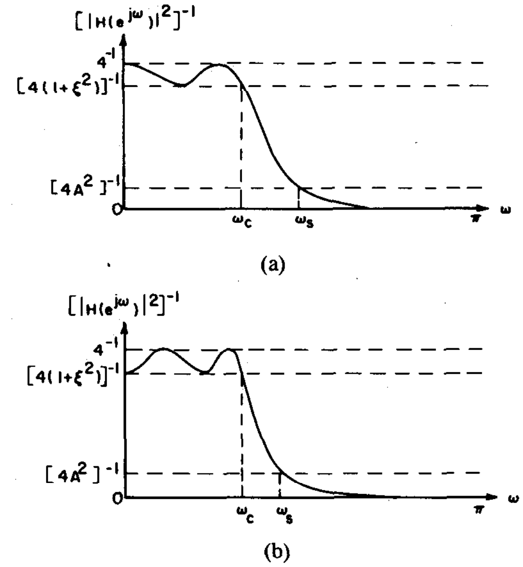


Fig. 5. Magnitude-squared transfer characteristic of, (a) a third- (odd) order, and (b) a fourth (even)-order Chebyshev low-pass filter.

IV. Digital Chebyshev Transfer Functions

The derivation of a digital all-pole Chebyshev transfer function closely parallels to the derivation of an analog Chebyshev transfer function [11], [12].

The magnitude-squared transfer characteristics of odd- and even-order Chebyshev low-pass filters that can be used for LDI ladder filter implementation are given in Fig. 5(a) and (b), respectively. In these figures,

$$\omega_c = \text{passband edge}$$
$$\omega_s = \text{stopband edge}$$
$$\epsilon = \text{passband ripple}$$

and

$$A = \text{stopband attenuation}.$$

Suppose $H(z)$ is the reciprocal of an all-pole Z-transform transfer function. The magnitude-squared response $|H(e^{j\omega})|^2$ with a Chebyshev transfer characteristic can be written as [11], [12]

$$|H(e^{j\omega})|^2 = 4[1+\epsilon^2F^2(\omega)] \tag{16}$$

where $F(\omega)$ varies between zero and one within the passband and is larger than one within the stopband, $\epsilon^2 = 10^{0.1A_p}-1$ and A_p = passband ripple in decibels. Furthermore, it can be shown that [9], [10]

$$|H(e^{j\omega})|^2 = H(z)H(z^{-1})|_{z=e^{j\omega}} = \sum_{i=0}^{n} m_i \cos^{n-i}\omega. \tag{17}$$

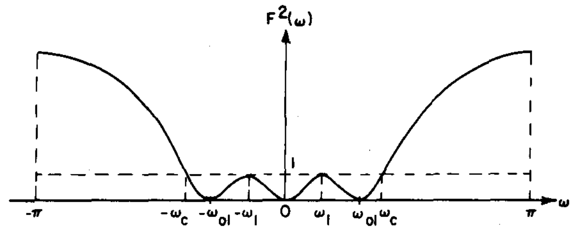


Fig. 6. $F^2(\omega)$ of a third (odd)-order Chebyshev low-pass filter.

From (17), if $\boldsymbol{H}(z)$ is a polynomial in z of order n, then $F^2(\omega)$ is a polynomial in $\cos\omega$ of order n, and $F(\omega)$ is a polynomial in $\cos\omega$ of order $n/2$. The situation that n is an odd number will be clarified in the subsequent discussion.

Similar to the procedure outlined in [11], [12], the derivation of $\boldsymbol{H}(z)$ involves the following steps:

1) Deduce the exact form of $F(\omega)$ such that the transfer characteristics in Fig. 5 is achieved.

2) Obtain the exact form of $|\boldsymbol{H}(e^{j\omega})|^2$.

3) Calculate the zeros of $|\boldsymbol{H}(z)|^2$, where $z=e^{j\omega}$, and then assign the zeros inside the unit circle to $\boldsymbol{H}(z)$.

The derivation of the digital all-pole Chebyshev low-pass filter design formulas will start with a third-order (odd-order) low-pass filter example. The result is then generalized to the case of nth-order low-pass filters.

Comparing Fig. 5(a) and (16), $F^2(\omega)$ of a third-order (odd-order) Chebyshev low-pass filter is given in Fig. 6.

$F(\omega)$ in this case is a polynomial in $\cos\omega$ of order $3/2$. Furthermore, it has the following properties [11], [12]

1) $F(\omega)=0$ if $\omega=0, \pm\omega_{01}$

2) $F^2(\omega)=1$ if $\omega=\pm\omega_1, \pm\omega_c$

3) $dF^2(\omega)/d\omega=0$ if $\omega=0, \pm\omega_1, \pm\omega_{01}$.

Observe that a factor of the form $(\cos\omega-\cos\omega_{01})$ has zeros at $\omega=\omega_{01}$ and $\omega=-\omega_{01}$. Since every factor of this form possess two zeros, $(\cos 0-\cos\omega)$, or equivalently $(1-\cos\omega)$ has double zeros at $\omega=0$. The first property of $F(w)$ indicates a single zero at $\omega=0$. If this corresponds to a factor $(1-\cos\omega)^{1/2}$, then

$$F(\omega)=M_1(1-\cos\omega)^{1/2}(\cos\omega-\cos\omega_{01}), \tag{18}$$

which is a polynomial in $\cos\omega$ of order $3/2$.

Since $F^2(\omega)$ is a polynomial in $\cos\omega$ of order 3, the derivative of $F^2(\omega)$ with respect to ω is a polynomial in $\cos\omega$ of order 2, multiplied by $-\sin\omega$, which is the derivative of $\cos\omega$. The above observation together with property 3 imply that

$$dF^2(\omega)/d\omega=M_2(\cos\omega-\cos\omega_1)(\cos\omega-\cos\omega_{01})(\sin\omega) \tag{19}$$

Next, the factor $(\cos\omega-\cos\omega_1)$ in $dF^2(\omega)/d\omega$ indicates that $[1-F^2(\omega)]$ must have the factor $(\cos\omega-\cos\omega_1)^2$. Consequently, from property 2

$$1-F^2(\omega)=M_3(\cos\omega-\cos\omega_1)^2(\cos\omega-\cos\omega_c) \tag{20}$$

Manipulating equations (18), (19), and (20) as in [11], we obtain

$$\int\frac{dF(\omega)}{\sqrt{1-F^2(\omega)}}=M_4\int\frac{d(\cos\omega)}{\sqrt{(1-\cos\omega)(\cos\omega-\cos\omega_c)}}.$$

The solution of this equation is

$$\begin{matrix}\sin^{-1}\\ \text{or}\\ \cos^{-1}\end{matrix}[F(\omega)]=M_4\begin{matrix}\sin^{-1}\\ \text{or}\\ \cos^{-1}\end{matrix}\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right] \tag{21}$$

where $M_0=1$. (The use of this notation will facilitate the subsequent discussion.) The choice of $\sin^{-1}$ or $\cos^{-1}$ depends on the properties of $F(\omega)$ and it will be shown that the following choice of $F(\omega)$ is suitable for an odd-order Chebyshev filter:

$$F(\omega)=\sin\left[\frac{n}{2}\cos^{-1}\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]$$

or

$$\begin{aligned}F^2(\omega)&=\frac{1}{2}-\frac{1}{2}\cos\left[n\cos^{-1}\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]\\&=\frac{1}{2}-\frac{1}{2}T_n\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\end{aligned} \tag{22}$$

where T_n is the nth-order Chebyshev polynomial.

To show that (22) is a suitable choice of $F(\omega)$, we find by direct substitution,

$$F(0)=0 \quad\text{and}\quad F(\pm\omega_c)=1,$$

which satisfy properties 1 and 2.

Furthermore if the derivation procedure is to be valid, $F(\omega)$ must have the form of (18) or in general

$$F(\omega)=(1-\cos\omega)^{1/2}R(\cos\omega) \tag{23}$$

where $R(\cos\omega)$ is a polynomial in $\cos\omega$. This can be proven by using mathematical induction technique. However, the complete proof is lengthy and thus will not be included here. We will only demonstrate that the factor $(1-\cos\omega)^{1/2}$ actually exist.

The presence of the Chebyshev polynomial in (22) shows that $F^2(\omega)$ is a polynomial in $\cos\omega$ of order n. In addition to this, at $\omega=0$, or equivalently, at $\cos\omega=1$,

$$F^2(0)=1/2-1/2\cos[n\cos^{-1}(1)]=0.$$

Therefore, $(1-\cos\omega)$ is a factor of $F^2(\omega)$. In other words, $(1-\cos\omega)^{1/2}$ is a factor of $F(\omega)$.

By the same argument, it can be shown that the following form of $F(\omega)$ is suitable for even order Chebyshev low-pass transfer functions

$$F(\omega)=\cos\left[\frac{n}{2}\cos^{-1}\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]$$

or

$$\begin{aligned}F^2(\omega)&=\frac{1}{2}+\frac{1}{2}\cos\left[n\cos^{-1}\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]\\&=\frac{1}{2}+\frac{1}{2}T_n\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\end{aligned} \tag{24}$$

with $M_0=1$.

Referring to Fig. 5, an odd-order Chebyshev low-pass filter has the characteristic $|\boldsymbol{H}(e^{j0})|=2$. Careful examination of the LDI ladder structure indicates that this results in unity rightmost ladder branch multiplier, i.e., $s_0=1$ in (12). Unfortunately, an even-order Chebyshev low-pass filter does not possess this property, so that two extra multipliers are needed to realize an even-order filter. Therefore, it would be desirable to modify an even-order Chebyshev low-pass filter so that $|\boldsymbol{H}(e^{j0})|=2$.

The modification can be achieved by eliminating one of the ripples in the passband of an even-order Chebyshev low-pass filter so that the passband of a, say, fourth-order filter is similar to the passband of a third-order filter [12]. The modified Chebyshev low-pass filter is suboptimal in the sense that the stopband performance of the modified filter is not as good as the original filter, but, in general, better than the next lower odd-order Chebyshev low-pass filter.

A fourth-order filter example will be used to derive the modified Chebyshev low-pass filter design formulas. The result is then generalized to the case of nth-(even) order low-pass filters.

$F^2(\omega)$ of a fourth-order modified Chebyshev low-pass filter is very similar to $F^2(\omega)$ of a third-order Chebyshev low-pass filter (Fig. 5). However, ω_0 and ω_{01} of a modified fourth-order filter would in general be different from that of a third-order filter. The three properties of $F^2(\omega)$ of the third-order Chebyshev low-pass filter also apply to the modified fourth-order filter except that $F(\omega)$ in this case is a polynomial in $\cos\omega$ of order 2.

From property 1, we can write

$$F(\omega)=M_5(1-\cos\omega)(\cos\omega-\cos\omega_{01}) \tag{25}$$

which means we forced double zeros at $\omega=0$. Since

$$F^2(\omega)=M_5^2(1-\cos\omega)^2(\cos\omega-\cos\omega_{01})^2$$

the derivative of $F^2(\omega)$ must possess the factors $(1-\cos\omega)$ and $(\cos\omega-\cos\omega_{01})$. This together with property 3 imply

$$dF^2(\omega)/d\omega=M_6(1-\cos\omega)(\cos\omega-\cos\omega_1)\cdot(\cos\omega-\cos\omega_{01})(\sin\omega). \tag{26}$$

Finally, $1-F^2(\omega)$ has the factor $(\cos\omega-\cos\omega_1)^2$ as argued before and the factor $(\cos\omega-\cos\omega_c)$. Note that the factor $(\cos\omega-\cos\omega_c)$ can only have multiplicity 1, otherwise, this factor will show up in $dF^2(\omega)/d\omega$. Therefore,

$$1-F^2(\omega)=M_7(\cos\omega-\cos\omega_1)^2\cdot(\cos\omega-\cos\omega_c)(M_0-\cos\omega) \tag{27}$$

where $|M_0|>1$, otherwise $F^2(\omega)=1$ at a certain frequency other than ω_1 and ω_c, which is not allowed according to Fig. 5, or property 2.

Manipulating (25), (26), and (27) yields

$$\int\frac{dF(\omega)}{\sqrt{1-F^2(\omega)}}=M_8\int\frac{d\cos\omega}{\sqrt{(M_0-\cos\omega)(\cos\omega-\cos\omega_c)}}.$$

The solution to this equation is the same as (22) except that $M_0>1$. Substituting $\omega=0$ in (22) and using the property $F(0)=0$ yields

$$M_0=1+\frac{(1-\cos\omega_c)(1-\cos(\pi/n))}{1+\cos(\pi/n)}. \tag{28}$$

From this, $M_0>1$, which fulfills the requirement of (27).

Equations (22) and (24) give $F(\omega)$ for $|\omega|<|\omega_c|$. For $|\omega|\geqslant|\omega_c|$, we have

$$\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\leqslant -1 \tag{29}$$

and thus

$$\begin{aligned}|\boldsymbol{H}(e^{j\omega})|^2&=4\left[1+\epsilon^2F^2(\omega)\right]\\&=4+2\epsilon^2\left[1\pm\cos\left[n\cos^{-1}\cdot\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]\right]\\&=4+2\epsilon^2\left[1\pm\cos\left[n\pi+nj\cosh^{-1}\cdot\left[\frac{2\cos\omega-(M_0+\cos\omega_c)}{M_0-\cos\omega_c}\right]\right]\right],\\&\quad +:\text{even } n;\quad -:\text{odd } n\end{aligned}$$

or

$$|\boldsymbol{H}(e^{j\omega})|^2=4+2\epsilon^2\left[1+\cosh\left[n\cosh^{-1}\cdot\left[\frac{(M_0+\cos\omega_c)-2\cos\omega}{M_0-\cos\omega_c}\right]\right]\right] \tag{30}$$

where $M_0=1$ for both odd- and even-order Chebyshev low-pass filters, and M_0 is given by (28) for modified even-order Chebyshev filters.

Any four of the five parameters ω_c, ω_s, ϵ, A, and n determine the remaining one in a Chebyshev transfer function. Substituting $|\boldsymbol{H}(e^{j\omega})|$ by $2A$ at $\omega=\omega_s$ in equation (30) and rearranging, we get

$$n\geqslant\frac{\cosh^{-1}\left[2\dfrac{A^2-1}{\epsilon^2}-1\right]}{\cosh^{-1}\left[\dfrac{(M_0+\cos\omega_c)-2\cos\omega_s}{M_0-\cos\omega_c}\right]},\qquad n\text{ integer}. \tag{31}$$

This formula can be used to determine the order of a Chebyshev filter in order to meet a particular filter specification.

If we make the substitution $z=e^{j\omega}$ in (30) and set $|\boldsymbol{H}(z)|=0$, then

$$1+\frac{\epsilon^2}{2}\left[1+\cosh\left[n\cosh^{-1}\left[\frac{(M_0+\cos\omega_c)-(z+z^{-1})}{M_0-\cos\omega_c}\right]\right]\right]=0. \tag{32}$$

This equation enables us to find the zeros of $\boldsymbol{H}(z)$. Let

$$\beta+j\gamma=\frac{(M_0+\cos\omega_c)-(z+z^{-1})}{M_0-\cos\omega_c}. \tag{33}$$

From (32) and (33), it can be shown that β and γ are given by [11]

$$\beta=\cos\left[\frac{2m-1}{n}\pi\right]\cosh\left[\frac{1}{n}\cosh^{-1}\left[1+\frac{2}{\epsilon^2}\right]\right]$$
$$\gamma=\sin\left[\frac{2m-1}{n}\pi\right]\sinh\left[\frac{1}{n}\cosh^{-1}\left[1+\frac{2}{\epsilon^2}\right]\right] \quad (34)$$

where $m=0,1,2,\cdots,n-1$. Having determined β and γ, $\boldsymbol{H}(z)$ can readily be found by solving equation (33) and assigning the roots inside the unit circle to $\boldsymbol{H}(z)$.

Now, the reciprocal of the transfer function is given by

$$\boldsymbol{H}(z)=h_0(z-p_1)(z-p_2)\cdots(z-p_n). \quad (35)$$

Thus

$$h_0=|\boldsymbol{H}(e^{j0})|/[(1-p_1)(1-p_2)\cdots(1-p_n)] \quad (36)$$

where $|\boldsymbol{H}(e^{j0})|=2$ for odd-order Chebyshev filters and even-order modified Chebyshev filters, and $|\boldsymbol{H}(e^{j0})|=2/\sqrt{1+\epsilon^2}$ for even-order Chebyshev filters.

Note that the above discussion on all-pole digital transfer functions is completely general. It can be used to design LDI ladder filters and any other suitable filter structures as well. The design formulas can also be extended to include high-pass, bandpass, or even multiband filters.

V. Factorization and Root-Squaring

The second step in the design procedure is to calculate $\boldsymbol{H}(z)\boldsymbol{H}(z^{-1})$ from $\boldsymbol{H}(z)$, find $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ from equation (11) and then factorize $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ to $\boldsymbol{K}(z)$ and $\boldsymbol{K}(z^{-1})$. The last part of this step poses a major difficulty in the LDI ladder filter synthesis.

The most straightforward method of performing the factorization is first find all the zeros of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ and then arbitrarily assign the zeros in the interior of the unit circle to $\boldsymbol{K}(z)$ [2]. However, numerical ill-conditioning usually occurs in the required root-finding procedure [2]. This is because the zeros of $\boldsymbol{K}(z)$ are usually very close to the unit circle.

Suppose z_i and $\overline{z_i}$ is a complex-conjugate zero pair of $\boldsymbol{K}(z)$, then z_i^{-1} and $\overline{z_i}^{-1}$ is a complex-conjugate zero pair of $\boldsymbol{K}(z^{-1})$. Let

$$z_i=r_ie^{j\vartheta_i}, \qquad \text{where } r\approx 1$$

then

$$\overline{z_i}^{-1}=r_i^{-1}e^{j\vartheta_i}.$$

Since

$$\arg(z_i)=\arg\left(\overline{z_i}^{-1}\right)$$

we have

$$|z_i|\approx\left|\overline{z_i}^{-1}\right|.$$

The two zeros z_i and $\overline{z_i}^{-1}$, any i, of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ are very close to each other. The situation is even worse for high-order narrow-band low-pass filters, in which case all the zeros cluster around $1+j0$. This renders the usual root-finding algorithms inaccurate. On the other hand, the coefficients of $\boldsymbol{H}(z)\boldsymbol{H}(z^{-1})$ and $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ could be different by only a very small fraction. Therefore, in excess of twelve digits numerical accuracy could be required in the design of high-order narrow-band filters.

In this paper, a new factorization technique is devised to tackle the numerical problem. The mathematical basis of this technique is given in the following paragraphs.

Consider an nth-order polynomial,

$$\boldsymbol{K}(z)=k_0z^n+k_1z^{n-1}+\cdots+k_{n-1}z+k_n \quad (37)$$

then

$$\begin{aligned}\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})&=\left(k_0z^n+k_1z^{n-1}+\cdots+k_n\right)\\&\quad\cdot\left(k_0z^{-n}+k_1z^{-n+1}+\cdots+k_n\right)\\&=l_0(z^n+z^{-n})+l_1(z^{n-1}+z^{-(n-1)})\\&\quad+\cdots+l_{n-1}(z+z^{-1})+l_n.\end{aligned} \quad (38)$$

From (37) and (38),

$$\begin{aligned}l_0&=k_0k_n\\l_1&=k_0k_{n-1}+k_1k_n\\&\vdots\\l_i&=\sum_{p=0}^{i}k_pk_{p+n-i}\\&\vdots\\l_n&=k_0k_0+k_1k_1+\cdots+k_{n-1}k_{n-1}+k_nk_n.\end{aligned} \quad (39)$$

The factorization problem can be stated as follows:

Given the coefficients l_i, $0\leqslant i\leqslant n$, of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$, find the coefficients k_i, $0\leqslant k\leqslant n$, of $\boldsymbol{K}(z)$. The relationship between l_i's and k_i's are given by (39).

This problem can be solved by using Newton's method in several variables.

If $k(m)=[k_0(m)\quad k_1(m)\quad\cdots\quad k_n(m)]^T$ is the approximation of $k_f=[k_0\quad k_1\quad\cdots\quad k_n]^T$ after m iterations, Newton's method states that [13], [14]

$$k(m+1)=k(m)-J_f^{-1}(m)\left[F_f(m)-L\right] \quad (40)$$

where

$$L=[l_0\quad l_1\quad\cdots\quad l_n]^T$$

$$F_f(m)=\begin{bmatrix}f_{f_0}(m)\\f_{f_1}(m)\\\vdots\\f_{f_i}(m)\\\vdots\\f_{f_n}(m)\end{bmatrix}=\begin{bmatrix}k_0k_n\\k_0k_{n-1}+k_1k_n\\\vdots\\\sum_{p=0}^{i}k_pk_{p+n-i}\\\vdots\\k_0k_0+k_1k_1+\cdots+k_nk_n\end{bmatrix}_{k=k(m)} \quad (41)$$

and

$$J_f(m)=\left[\frac{\partial f_{f_i}}{\partial k_j}\right]_{k=k(m)}, \qquad 0\leqslant i,j\leqslant n$$

where

$$\frac{\partial f_{f_i}}{\partial k_j} = \frac{\partial}{\partial k_j}\left[\sum_{p=0}^{i} k_p k_{p+n-i}\right] = \begin{cases} \frac{\partial}{\partial k_j}\left[\cdots + k_j k_{j+n-i} + \cdots + k_{j+i-n} k_j + \cdots\right], & i \neq n \\ \frac{\partial}{\partial k_j}\left[\cdots + k_j k_j + \cdots\right], & i = n. \end{cases}$$

In both cases,

$$J_f(m) = \left[\frac{\partial f_{f_i}}{\partial k_j}\right]_{k=k(m)} = k_{j-i+n}(m) + k_{j+i-n}(m) \tag{42}$$

provided that the indexes $(j-i+n)$ and $(j+i-n)$ are between 0 and n. If any of these indices are not inside the range, then the corresponding term is zero. (This will be referred as the "index in range" condition in the subsequent discussion.) Furthermore, it is shown in the Appendix that in general

$$J_f^{-1}(m) F_f(m) = k(m)/2. \tag{43}$$

Therefore, (40) becomes

$$J_f(m)[k(m+1) - k(m)/2] = L. \tag{44}$$

The iteration procedure is carried out in the following steps:

1) Calculate $J_f(m)$ from $k(m)$;

2) Solve the matrix equation $[J_f(m)][c(m)] = L$ by one of the standard method, as for example, Gauss elimination. Here $[c(m)]$ is the unknown vector;

3) Calculate the new approximation by $k(m+1) = c(m) + k(m)/2$.

It can be shown that in the case of odd-order Chebyshev low-pass filters and even order modified Chebyshev low-pass filters, two of the zeros of $K(z)K(z^{-1})$ are $1 + j0$. Due to the limitation of numerical accuracy, they become $(1+\delta_1) + j0$ and $(1+\delta_2) + j0$ where δ_1 and δ_2 are very small numbers. In other words, they become extremely closely paired zeros. These zeros must be removed before applying Newton's method for factorization in order to avoid potential instability. This step is unnecessary in the case of even order Chebyshev low-pass filters. Experimental observations shows that the factorization is usually stabilized in within 50 iterations.

Newton's method for factorizing $K(z)K(z^{-1})$ to $K(z)$ and $K(z^{-1})$ produces accurate results in most situations. However, if higher numerical accuracy is desired, a root-squaring technique can be used. It has been shown that if $z_i = r_i e^{j\vartheta_i}$ is a zero of $K(z)K(z^{-1})$ then $\overline{z_i}^{-1} = r_i^{-1} e^{j\vartheta_i}$ is also a zero of $K(z)K(z^{-1})$. Since r_i is close to unity, $|z_i|$ and $|\overline{z_i}^{-1}|$ are very close to each other. However, consider the square of these zeros

$$(z_i)^2 = r_i^2 e^{j2\vartheta_i}$$

and

$$\left(\overline{z_i}^{-1}\right)^2 = r_i^{-2} e^{j2\vartheta_i}.$$

If $r_i < 1$, then $r_i^2 < r_i$ and $r_i^{-2} > r_i^{-1}$. In other words, the zero that is inside the unit circle becomes closer to the origin and the zero that is outside the unit circle becomes further apart from the origin. Therefore, the two zeros becomes more separated from each other. Furthermore, the arguments of the squared zeros are twice the arguments of the original zeros, so the squaring process would spread any clustered zeros that occur in the design of high-order narrow-band LDI ladder filters.

In order to apply this principle, we need to calculate the coefficients of a new polynomial whose zeros are the square of the zeros of the original polynomial. This process is called root-squaring which is well known in numerical mathematical analysis [14].

Now, the factorization of $K(z)K(z^{-1})$ could be carried out as follows:

1) Perform root-squaring on $K(z)K(z^{-1})$ so that the zeros of a new polynomial $\overline{K(z)K(z^{-1})}$ are the square of the zeros of $K(z)K(z^{-1})$.

2) Use Newton's method to factorize $\overline{K(z)K(z^{-1})}$ to $\overline{K(z)}$ and $\overline{K(z^{-1})}$.

3) Perform square-rooting on $\overline{K(z)}$ so that the zeros of the resulting polynomial $K(z)$ are the square roots of the zeros of $\overline{K(z)}$.

Note that the root-squaring and thus the square-rooting process could be applied several times in order to separate highly clustered zeros. Now, the root-squaring process, which is well documented in the literature [14], will be introduced in detail and then the mathematics of the square-rooting process will be developed.

Consider the following polynomials,

$$\begin{aligned} z^n K(z) K(z^{-1}) &= z^n\left[l_0(z^n + z^{-n}) + \cdots + l_{n-1}(z + z^{-1}) + l_n\right] \\ &= l_0(z - z_1)\cdots(z - z_n)(z - z_1^{-1})\cdots(z - z_n^{-1}) \end{aligned} \tag{45}$$

$$\begin{aligned} z^n K(-z) K(-z^{-1}) &= z^n\left[l_0(z^n + z^{-n}) + \cdots \right. \\ &\quad \left. + (-1)^{n-1} l_{n-1}(z + z^{-1}) + (-1)^n l_n\right] \\ &= l_0(z + z_1)\cdots(z + z_n)(z + z_1^{-1})\cdots(z + z_n^{-1}). \end{aligned} \tag{46}$$

Let

$$\begin{aligned} z^{2n}\overline{K(z)}\,\overline{K(z^{-1})} &= \left[z^n K(z) K(z^{-1})\right]\left[z^n K(-z) K(-z^{-1})\right] \\ &= z^{2n}\left[\overline{l_0}(z^{2n} + z^{-2n}) + \cdots + \overline{l_{n-1}}(z^2 + z^{-2}) + \overline{l_n}\right] \\ &= l_0^2(z^2 - z_1^2)\cdots(z^2 - z_n^2)(z^2 - z_1^{-2})\cdots(z^2 - z_n^{-2}). \end{aligned} \tag{47}$$

Therefore, the zeros of $z^{2n}\overline{K(z)K(z^{-1})}$ are the square of the zeros of $z^n K(z)K(z^{-1})$. From (45), (46), and (47), it can be shown that

$$\overline{l_i} = (-1)^i \left[l_i^2 + 2\sum_{p=1}^{i} (-1)^p l_{i-p} l_{i+p} \right] \tag{48}$$

provided the "index in range" condition $0 \leqslant i-p, i+p \leqslant 2n$ is satisfied, and if $i-p, i+p > n$, they are to be replaced by $2n-(i-p)$ and $2n-(i+p)$, respectively.

After $\overline{K(z)K(z^{-1})}$ is found, we can use Newton's method developed in the previous discussion to factorize $\overline{K(z)K(z^{-1})}$ to $\overline{K(z)}$ and $\overline{K(z^{-1})}$, where

$$\overline{K(z)} = \overline{k_0} z^{2n} + \overline{k_1} z^{2(n-1)} + \cdots + \overline{k_{n-1}} z^2 + \overline{k_n}. \tag{49}$$

Equations (40)–(44) remain unchanged except that k_i's and l_i's become $\overline{k_i}$'s and $\overline{l_i}$'s, respectively.

Next, $K(z)$ has to be determined from $\overline{K(z)}$ such that the zeros of $K(z)$ are the square-roots of the zeros of $\overline{K(z)}$. Since

$$\begin{aligned} K(z) &= k_0 z^n + k_1 z^{n-1} + \cdots + k_{n-1} z + k_n \\ &= k_0 (z - z_1) \cdots (z - z_n) \end{aligned} \tag{50}$$

and

$$\begin{aligned} (-1)^n K(-z) &= k_0 z^n - k_1 z^{n-1} + \cdots \\ &\quad + (-1)^{n-1} k_{n-1} z + (-1)^n k_n \\ &= k_0 (z + z_1) \cdots (z + z_n) \end{aligned} \tag{51}$$

therefore,

$$\begin{aligned} \overline{K(z)} &= \overline{k_0} z^{2n} + \overline{k_1} z^{2(n-1)} + \cdots + \overline{k_{n-1}} z^2 + \overline{k_n} \\ &= k_0^2 (z^2 - z_1^2) \cdots (z^2 - z_n^2) \\ &= (-1)^n K(z) K(-z). \end{aligned} \tag{52}$$

From (50), (51), and (52),

$$\begin{aligned} \overline{k_0} &= k_0^2 \\ \overline{k_1} &= -k_1^2 + 2k_0 k_2 \\ &\;\vdots \\ \overline{k_i} &= (-1)^i [k_i^2 - 2k_{i-1}k_{i+1} + 2k_{i-2}k_{i+2} \\ &\quad - 2k_{i-3}k_{i+3} + \cdots] \\ &= (-1)^i \left[k_i^2 + 2\sum_{p=1}^{i} (-1)^p k_{i-p} k_{i+p} \right] \\ &\;\vdots \\ \overline{k_n} &= (-1)^n k_n^2 \end{aligned} \tag{53}$$

provided the "index in range" condition $0 \leqslant i-p, i+p \leqslant n$ is satisfied. The square-rooting process can be stated as follows:

> Given the coefficients $\overline{k_i}$, $0 \leqslant i \leqslant n$, of $\overline{K(z)}$, find k_i, $0 \leqslant i \leqslant n$, of $K(z)$, such that $\overline{k_i}$ and k_i are related by (53).

Again, this problem can be solved by using Newton's method in several variables.

If $k(m) = [k_0(m) \;\; k_1(m) \;\; \cdots \;\; k_n(m)]^T$ is the approximation of $k_s = [k_0 \;\; k_1 \;\; \cdots \;\; k_n]^T$ after m iterations, Newton's method states that

$$k(m+1) = k(m) - J_s^{-1}(m)[F_s(m) - S] \tag{54}$$

where

$$S = [\overline{k_0} \;\; \overline{k_1} \;\; \cdots \;\; \overline{k_n}]^T,$$

$$F_s(m) = \begin{bmatrix} f_{s_0}(m) \\ f_{s_1}(m) \\ \vdots \\ f_{s_i}(m) \\ \vdots \\ f_{s_n}(m) \end{bmatrix} = \begin{bmatrix} k_0^2 \\ -k_1^2 + 2k_0 k_2 \\ \vdots \\ (-1)^i \left[k_i^2 + 2\sum_{p=1}^{i} (-1)^p k_{i-p} k_{i+p} \right] \\ \vdots \\ (-1)^n k_n^2 \end{bmatrix}_{k=k(m)} \tag{55}$$

and

$$J_s(m) = \left[\frac{\partial f_{s_i}}{\partial k_j} \right]_{k=k(m)}, \qquad 0 \leqslant i, j \leqslant n$$

where

$$\frac{\partial f_{s_i}}{\partial k_j} = \frac{\partial}{\partial k_j} \left[(-1)^i \left[k_i^2 + 2\sum_{p=1}^{i} (-1)^p k_{i-p} k_{i+p} \right] \right].$$

If $i \neq j$, the relevant terms correspond to

$$\begin{aligned} i - p = j \;&\rightarrow\; p = i - j \;\rightarrow\; i + p = 2i - j \\ i + p = j \;&\rightarrow\; p = j - i \;\rightarrow\; i - p = 2i - j. \end{aligned}$$

Since p must be positive, only either one of them is true, that is, either

$$\begin{aligned} \frac{\partial f_{s_i}}{\partial k_j} &= 2(-1)^i \frac{\partial}{\partial k_j} [\cdots + (-1)^{i-j} k_j k_{2i-j} + \cdots] \\ &= 2(-1)^{2i-j} k_{2i-j} \end{aligned}$$

or

$$\begin{aligned} \frac{\partial f_{s_i}}{\partial k_j} &= 2(-1)^i \frac{\partial}{\partial k_j} [\cdots + (-1)^{j-i} k_{2i-j} k_j + \cdots] \\ &= 2(-1)^j k_{2i-j} \end{aligned}$$

is true. However, they are the same because $(-1)^{2i-j} =$

$(-1)^j$. If $i=j$

$$\frac{\partial f_{s_i}}{\partial k_j}=\frac{\partial}{\partial k_j}\left[(-1)^i k_i^2+\cdots\right]$$
$$=2(-1)^i k_i=2(-1)^{2i-j}k_{2i-j}.$$

Therefore, in all cases

$$J_s(m)=\left[\frac{\partial f_{s_i}}{\partial k_j}\right]_{k=k(m)}\left[=2(-1)^{2i-j}k_{2i-j}(m)\right] \quad (56)$$

with the "index in range" condition.

Similar to the case of factorization, it can be shown that $J_s(m)F_s(m)=k(m)/2$. Therefore, (54) becomes $J_s(m)[k(m+1)-k(m)/2]=S$. The iteration procedure is similar to that of Newton's method for factorization.

The root-squaring technique has been successfully applied to design both even- and odd-order Chebyshev lowpass filters. However, due to unknown reasons, any attempt to apply the root-squaring technique to design even-order modified Chebyshev filters results in unstable iterations. Fortunately, unless extreme accuracy is required, the root-squaring procedure is not strictly necessary in the design of LDI ladder filters.

VI. The Chain Matrix Elements

The first part of the third step in the design procedure is to find $E(z)$ and $F(z)$ from $\boldsymbol{H}(z)$ and $\boldsymbol{K}(z)$ according to (8) and (9). Rewrite these equations as

$$(z+1)E(z)=\boldsymbol{K}(z)+z\boldsymbol{H}(z) \quad (57)$$

and

$$z^{-1/2}(z+1)F(z)=\boldsymbol{K}(z)-\boldsymbol{H}(z) \quad (58)$$

where $\boldsymbol{H}(z)$ and $\boldsymbol{K}(z)$ are to be scaled by $z^{-n/2}$ [2]. Let

$$\boldsymbol{K}(z)=\left[k_0z^n+k_1z^{n-1}+\cdots+k_{n-1}+k_n\right]z^{-n/2}$$
$$\boldsymbol{H}(z)=\left[h_0z^n+h_1z^{n-1}+\cdots+h_{n-1}+h_n\right]z^{-n/2}$$
$$(z+1)E(z)=\left[e_0'z^{n+1}+e_1'z^n+\cdots+e_n'z+e_{n+1}'\right]z^{-n/2}$$
$$z^{-1/2}(z+1)F(z)=\left[f_0'z^n+f_1'z^{n-1}+\cdots+f_{n-1}'z+f_n'\right]z^{-n/2}$$
$$E(z)=\left[e_0z^n+e_1z^{n-1}+\cdots+e_{n-1}z+e_n\right]z^{-n/2}$$

and

$$F(z)=\left[f_0z^{n-1}+f_1z^{n-2}+\cdots+f_{n-2}z+f_{n-1}\right]z^{-(n-1)/2}. \quad (59)$$

From (57), (58) and (59)

$$e_0'=h_0,$$
$$e_i'=k_{i-1}+h_i, \qquad i=1,2,\cdots,n$$
$$e_{n+1}'=k_n$$

and

$$f_i'=k_i-h_i, \qquad i=0,1,\cdots,n. \quad (60)$$

Now, e_i and f_i can be found from e_i' and f_i', respectively, by removing the factor $(z+1)$ [13]. Therefore,

$$e_0=e_0'$$
$$e_i=e_i'-e_{i-1}, \qquad i=1,2,\cdots,n$$

and

$$f_0=f_0'$$
$$f_i=f_i'-f_{i-1}, \qquad i=1,2,\cdots,n-1. \quad (61)$$

Combining (60) and (61), we finally have

$$e_0=h_0,$$
$$e_i=k_i+h_i-e_{i-1}, \qquad i=1,2,\cdots,n$$

and

$$f_0=k_0-h_0$$
$$f_i=k_i-h_i-f_{i-1}, \qquad i=1,2,\cdots,n \quad (62)$$

which can easily be programmed.

Now we have to separate $E(z)$ and $F(z)$ to the chain matrix elements $A(z)$, $B(z)$, $C(z)$, and $D(z)$ which are image polynomials. Let

$$A(z)=d_0(z^{1/2}-z^{-1/2})^n+d_2(z^{1/2}-z^{-1/2})^{n-2}+d_4(z^{1/2}-z^{-1/2})^{n-4}+\cdots \quad (63)$$

and

$$B(z)=d_1(z^{1/2}-z^{-1/2})^{n-1}+d_2(z^{1/2}-z^{-1/2})^{n-3}+d_5(z^{1/2}-z^{-1/2})^{n-5}+\cdots. \quad (64)$$

From (4), (63), (64) and (59)

$$\begin{aligned}
&A(z)-z^{-1/2}B(z)\\
&=\Big[d_0(z^{1/2}-z^{-1/2})^n-d_1z^{-1/2}(z^{1/2}-z^{-1/2})^{n-1}\\
&\quad+d_2(z^{1/2}-z^{-1/2})^{n-2}-d_3z^{-1/2}(z^{1/2}-z^{-1/2})^{n-3}\\
&\quad+\cdots+d_{2m}(z^{1/2}-z^{-1/2})^{n-2m}\\
&\quad-d_{2m+1}z^{-1/2}(z^{1/2}-z^{-1/2})^{n-2m-1}+\cdots\Big]\\
&=\Big[d_0z^0(z-1)^n-d_1z^0(z-1)^{n-1}\\
&\quad+d_2z^1(z-1)^{n-2}-d_3z^1(z-1)^{n-3}+\cdots\\
&\quad+d_{2m}z^m(z-1)^{n-2m}\\
&\quad-d_{2m+1}z^m(z-1)^{n-2m-1}+\cdots\Big]z^{-n/2}\\
&=\left[e_0z^n+e_1z^{n-1}+e_2z^{n-2}+\cdots+e_{n-1}z^1+e_nz^0\right]z^{-n/2}.
\end{aligned} \quad (65)$$

It is possible to write down a matrix equation that compares the coefficients associated to equal powers of z on both sides of this equation:

$$\begin{bmatrix} q_{00}z^n & q_{01}z^n & \cdots & q_{0n}z^n\\ q_{10}z^{n-1} & q_{11}z^{n-1} & \cdots & q_{1n}z^{n-1}\\ \vdots & \vdots & & \vdots\\ q_{n0}z^0 & q_{n1}z^0 & \cdots & q_{nn}z^0\end{bmatrix}\begin{bmatrix} d_0\\ d_1\\ \vdots\\ d_n\end{bmatrix}=\begin{bmatrix} e_0z^n\\ e_1z^{n-1}\\ \vdots\\ e_nz^0\end{bmatrix}. \quad (66)$$

Consider a particular coefficient d_j, we have

$$\sum_{i=0}^{n} d_j q_{ij} z^{n-i} = d_{2m} z^m (z-1)^{n-2m}, \quad \text{for } j = 2m = \text{even} \tag{67}$$

$$\sum_{i=0}^{n} d_j q_{ij} z^{n-i} = d_{2m+1} z^m (z-1)^{n-2m-1}, \quad \text{for } j = 2m+1 = \text{odd}. \tag{68}$$

However, the RHS of (67) is

$$d_{2m} z^m (z-1)^{n-2m} = d_{2m} z^m \sum_{p=0}^{n-2m} \binom{n-2m}{p} (-1)^p z^{n-2m-p} = d_{2m} \sum_{p=0}^{n-2m} \binom{n-2m}{p} (-1)^p z^{n-m-p}.$$

Let $j = 2m$ and then $i = p + j/2$, so that at $p = 0$, $i = j/2$ and at $p = n - j$, $i = n - j/2$. Thus

$$d_{2m} z^n (z-1)^{n-2m} = d_j \sum_{p=0}^{n-j} \binom{n-j}{p} (-1)^p z^{n-p-j/2} = d_j \sum_{i=j/2}^{n-j/2} \binom{n-j}{i-j/2} (-1)^{i-j/2} z^{n-i} = \sum_{i=0}^{n} d_j q_{ij} z^{n-i}$$

from LHS of (67). Hence,

$$q_{ij} = \begin{cases} (-1)^{i-j/2} \binom{n-j}{i-j/2}, & \text{for } j/2 \leqslant (\text{even } j) \leqslant n - j/2 \\ 0, & \text{for other even } j \end{cases} \tag{69}$$

Similarly, equation (68) yields

$$q_{ij} = \begin{cases} (-1)^{i+1-(j+1)/2} \binom{n-j}{i-(j+1)/2}, \\ \quad \text{for } (j+1)/2 \leqslant (\text{odd } j) \leqslant n-(j-1)/2 \\ 0, \quad \text{for other odd } j. \end{cases} \tag{70}$$

Disregard z^{n-i} in (66), it becomes

$$\begin{bmatrix} q_{00} & q_{01} & \cdots & q_{0n} \\ q_{10} & q_{11} & \cdots & q_{1n} \\ \vdots & \vdots & & \vdots \\ q_{n0} & q_{n1} & \cdots & q_{nn} \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix}. \tag{71}$$

Since the matrix $[q_{ij}]$ in (71) can be calculated from (69) and (70), the coefficients of $A(z)$ and $B(z)$ can be solved by using Gauss elimination. The coefficients of $C(z)$ and $D(z)$ can be found by exactly the same method.

Having obtained the chain matrix elements, it is straightforward to find the ladder multiplier values by carrying out the ladder decomposition according to (13).

VII. Computer Program and Design Examples

The above design formulas have been integrated in a comprehensive LDI ladder filter design program (Fig. 7). This program implements the design of odd-order Chebyshev low-pass filters and even-order modified Chebyshev low-pass filters. The root-squaring procedure is not included for simplicity. The program is written in Fortran 77 and runs on a DEC VAX-11/780 computer system under the UNIX operating system. The investigator specifies the order, passband edge and ripple of the LDI ladder filter, then the program returns the ladder section and the rightmost ladder branch multiplier values.

This program can design a wide range of LDI ladder filters with different specifications. Experiments shows that up to tenth-order filters can be designed. In addition to this, it is observed that the accuracy of the result is basically limited by the accuracy of the factorization step. The polynomial product $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ obtained by multiplying the factorized $\boldsymbol{K}(z)$ must be close to the desired $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ obtained from equation (11) as much as possible, usually two digits after the decimal point.

Because of this limiting factor, it is desirable to derive a relationship between n, ω_c, and ϵ such that under certain condition the comprehensive LDI ladder filter design program can be used confidently.

For 64 bits double precision arithmetic, numerical values are accurate up to about 16 digits. Allowing for 2 digits numerical error in the factorization step, the coefficients in $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ must be of order 10^{12} or less. Now the problem becomes estimating the order of magnitude of the coefficients of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ for any combinations of n, ω_c, and ϵ. From (11) and (38)

$$\boldsymbol{H}(z)\boldsymbol{H}(z^{-1}) = l_0(z^n + z^{-n}) + \cdots + (l_{n-1}+1)(z+z^{-1}) + (l_n+2)$$

and thus

$$\boldsymbol{H}(-1)\boldsymbol{H}(-1) = \pm(2l_0 - 2l_1 + \cdots - 2l_{n-1} + l_n). \tag{72}$$

In all the observed cases, the coefficients l_i's of $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ have alternating signs. Therefore, (72) can be written as

$$[\boldsymbol{H}(-1)]^2 \approx 2 \sum_{i=0}^{n} |l_i| \tag{73}$$

and this gives the approximate order of magnitude of l_i's. Setting $z = e^{j\omega} = -1$ in equation (30) and using the formulas

$$\cosh m_1 = (e^{m_1} + e^{-m_1})/2 \approx e^{m_1}/2, \quad \text{for large } m_1$$

$$\cosh^{-1} m_2 = \ln\left(m_2 + \sqrt{m_2^2 - 1}\right) \approx \ln(2m_2), \quad \text{for large } m_2$$

and

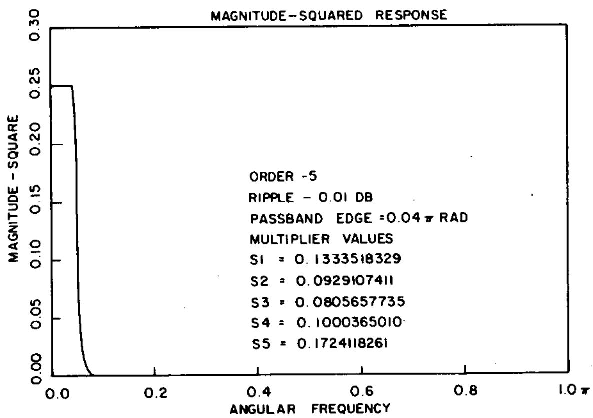
$$\cos\omega_c = \sqrt{1-\omega_c^2} \approx 1 - \omega_c^2/2, \quad \text{for small } \omega_c$$

```
c           DIGITAL LDI LADDER FILTER DESIGN PROGRAM
c                           by
c                E.S.K. LIU - December 1982
c          Department of Electrical Engineering
c                The University of Calgary
c                 Calgary  Alberta  Canada
c
c     A comprehensive digital LDI ladder filter design program.
c     The investigator specifies the order, ripple ( in db ) and
c     passband edge ( in rad ), then the program returns the
c     ladder section multiplier values of the LDI ladder filter.
c     The right-most ladder branch multipliers are 1.
c
      double complex    alpha,zero1,zero2,zero(10),p(0:10)
      double precision  pi,rn,ripple,edge,
     &             rk,gain,phi,esilon,angle,beta,gamma,
     &             h(0:10),k(0:10),e(0:10),f(0:10),s(10),r(0:10),
     &             l(0:10),a(0:10),c(0:10),q(0:10,0:10)
c
      read(5,*)n,ripple,edge
      pi = 3.141592653589793d0
      rn = float(n)
c
c        Calculate phi from the ripple specification
c
      esilon = 1.d0 + 2.d0/(10.d0**(0.1d0*ripple)-1.d0)
      phi = esilon + dsqrt(esilon**2-1.d0)
      phi = dlog(phi)/rn
c
c        Calculate the gain and zeros
c
      gain = 2.d0
      rk = 1.d0
      if(mod(n,2).eq.0) rk = 1.d0 + (1.d0 - dcos(edge))
     &            * (1.d0 - dcos(pi/rn)) / (1.d0 + dcos(pi/rn))
      do 100 m = 1,n
        angle = (2.d0*float(m)-1.d0) * pi/rn
        beta = dcos(angle) * dcosh(phi)
        gamma = dsin(angle) * dsinh(phi)
        alpha = dcmplx(rk+dcos(edge),0.d0)
     &       - dcmplx(rk-dcos(edge),0.d0) * dcmplx(beta,gamma)
        zero1 = (alpha - zsqrt(alpha**2-(4.d0,0.d0))) / (2.d0,0.d0)
        zero2 = (alpha + zsqrt(alpha**2-(4.d0,0.d0))) / (2.d0,0.d0)
        if(zabs(zero1).lt.1.d0)zero(m) = zero1
        if(zabs(zero2).lt.1.d0)zero(m) = zero2
        gain = gain / zabs((1.d0,0.d0)-zero(m))
100   continue
c
c        Calculate H(z) from the zeros
c
      p(0) = dcmplx(gain,0.d0)
      do 200 m = 1,n
        p(m) = dcmplx(0.d0,0.d0)
        do 200 i = m,1,-1
200       p(i) = p(i)-p(i-1)*zero(m)
      do 300 i = 0,n
        h(i) = real(p(i))
300     k(i) = h(i)
c
c        Calculate H(z)H(1/z) and then K(z)K(1/z) from H(z)
c
      do 320 i = 0,n
        l(i) = 0.d0
        do 320 j = 0,i
320       l(i) = l(i) + h(j)*h(j+n-i)
      l(n-1) = l(n-1) - 1.d0
      l(n) = l(n) - 2.d0
c
c        Remove the factor (z-1) out of the approximate
c        K(z) and the factor -(z-1)**2 out of K(z)K(1/z).
c
      do 350 i = 1,n
        l(i) = l(i-1)+l(i)
350     k(i) = k(i-1)+k(i)
      do 360 i = 1,n
360     l(i) = l(i-1)+l(i)
      do 370 i = 0,n
370     l(i) = -l(i)
c
c        Do factorization
c
      do 420 iterat = 1,50
c
c        Construct the Jacobian matrix
c
        do 400 i = 0,n-1
          do 400 j = 0,n-1
            q(i,j) = 0.d0
            m = j+i-n+1
            if(m.ge.0 .and. m.le.n-1)q(i,j) = q(i,j)+k(m)
            m = j-i+n-1
            if(m.ge.0 .and. m.le.n-1)q(i,j) = q(i,j)+k(m)
400     continue
c
c        Solve the matrix equation and update k(m)
c
        do 410 i = 0,n-1
410       a(i) = l(i)
        call gauss (n,q,a,c)
        do 420 i = 0,n-1
420       k(i) = c(i) + k(i)/2.d0
c
c        Multiply K(z) by K(z-1).
c
      k(n) = 0.
      do 450 i = n,1,-1
450     k(i) = k(i)-k(i-1)
c
c        Calculate E(z) and F(z) from H(z) and K(z)
c
      e(0) = h(0)
      do 500 i = 1,n
500     e(i) = k(i-1) + h(i) - e(i-1)
      f(0) = k(0) - h(0)
      do 550 i = 1,n
550     f(i) = k(i) - h(i) - f(i-1)
c
c        Calculate the image polynomials A,B,C and D
c
      call qmat  (n,q)
      call gauss (n+1,q,e,a)
      call qmat  (n-1,q)
      call gauss (n,q,f,c)
c
c        Do ladder decomposition
c
      do 800 m = n,1,-1
        s(n-m+1) = - c(0) / a(0)
        c(m) = 0.d0
        do 700 i = 2,m,2
700       r(i-2) = a(i) + c(i) / s(n-m+1)
        do 800 i = 0,m-1,2
          a(i) = - c(i)
800       c(i) = - r(i)
c
      write(6,920)(i,s(i),i = 1,n)
920   format(1x,"ladder section s(",i2,") = ",g18.10)
      stop
      end
c
c
c     Subroutine "qmat" constructs the Q matrix for
c     image polynomial calculations.
c     Subroutine parameters:
c     n = order of polynomial
c     q = the returned Q matrix
c
      subroutine qmat (n,q)
      double precision  q(0:10,0:10)
      do 500 i = 0,n
        do 500 j = 0,n
          q(i,j) = 0.d0
          if(mod(j,2).eq.0)then
            if(i.ge.j/2.and.i.le.n-j/2)then
              q(i,j) = (-1.)**(i-j/2)
              do 100 m = 1,i-j/2
100             q(i,j) = q(i,j) * float( (n-j)-(i-j/2)+m ) / float(m)
            endif
          else
            if(i.ge.(j+1)/2.and.i.le.n-(j-1)/2)then
              q(i,j) = (-1.)**(i+1-(j+1)/2)
              do 300 m = 1,i-(j+1)/2
300             q(i,j) = q(i,j) * float( (n-j)-(i-(j+1)/2)+m ) / float(m)
            endif
          endif
500   continue
      return
      end
c
c
c     Subroutine "gauss" uses Gauss elimination with partial pivoting
c     to solve a set of simultaneous linear equations [13].
c     Subroutine parameters:
c     n = number of simultaneous equations
c     a = n x n matrix A in AX=B
c     b = n vector B in AX=B
c     x = returned solution n vector X
c
      subroutine gauss (n,a,b,x)
      double precision  r,a(11,11),b(11),x(11)
c
      do 500 i = 1,n-1
c
c        Partial pivoting
```

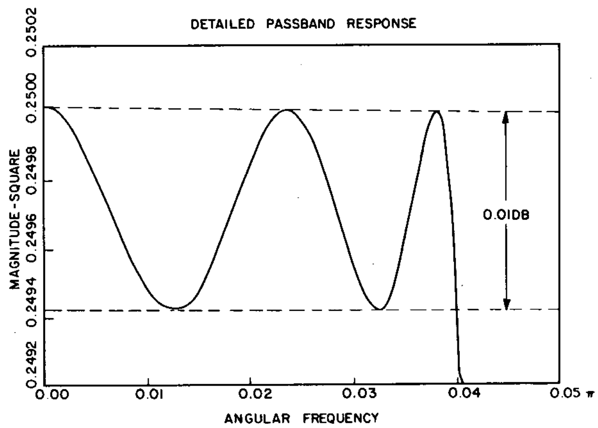
Fig. 7. A comprehensive LDI ladder filter design program.

```
c
        k = i
        do 100 j = i+1,n
100       if( dabs(a(j,i)) .gt. dabs(a(k,i)) )k = j
        r = b(k)
        b(k) = b(i)
        b(i) = r
        do 300 j = i,n
          r = a(k,j)
          a(k,j) = a(i,j)
300       a(i,j) = r
c
c         Transform A to an upper trianglar matrix
c
        do 500 k = i+1,n
          r = -a(k,i)/a(i,i)
          b(k) = r*b(i) + b(k)
          do 500 j = i,n
500         a(k,j) = r*a(i,j) + a(k,j)
c
c         Backsolve the trianglar equations
c
      x(n) = b(n)/a(n,n)
      do 700 k = 1,n-1
        r = 0.d0
        do 600 j = 1,k
600       r = r + a(n-k,n+1-j) * x(n+1-j)
700     x(n-k) = (b(n-k)-r) / a(n-k,n-k)
c
      return
      end
```

Fig. 7. (contd.)



(a)

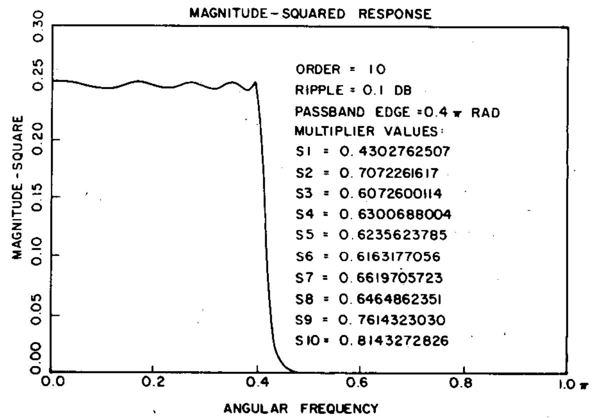


(b)

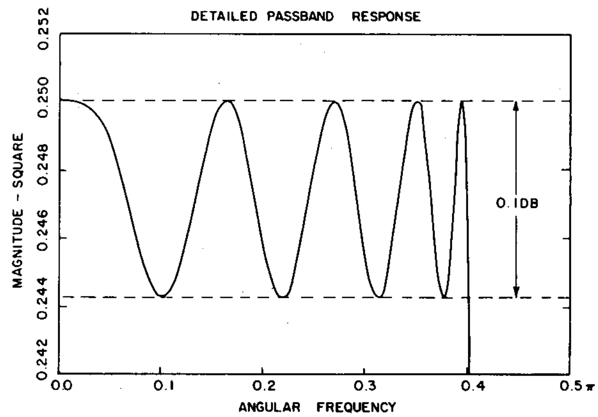
Fig. 8. The (a) overall and (b) detailed passband magnitude-squared response of a fifth-order LDI ladder filter designed by the program in Fig. 7.

yields

$$[H(-1)]^2 \approx 4+2\epsilon^2\left[1+\cosh\left[n\cosh^{-1}\left[\frac{3+\cos\omega_c}{1-\cos\omega_c}\right]\right]\right]$$
$$\approx \epsilon^2\left[\frac{4}{\omega_c}\right]^{2n}.$$



(a)



(b)

Fig. 9. The (a) overall and (b) detailed passband magnitude-squared response of a tenth-order LDI ladder filter designed by the program in Fig. 7.

Therefore, if $\epsilon^2(4/\omega_c)^{2n} \leqslant 10^{12}$ and $n \leqslant 10$, the LDI ladder filter design program can be used confidently.

The program had been used to design a narrow-band LDI low-pass ladder filter with specifications $n=5$, $\omega_c = 0.04\pi$ and $A_p = 0.01$ dB. The magnitude response of the designed LDI ladder filter (without coefficient quantization) is plotted in Fig. 8. A wideband LDI low-pass ladder filter example with specifications $n=10$, $\omega_c = 0.4\pi$, and $A_p = 0.1$ dB is given in Fig. 9. The accuracy of the design program is fully demonstrated.

VIII. Conclusion

It has been shown that LDI low-pass ladder filters can be transformed to LDD high-pass ladder filters. The topology and thus the low sensitivity property of the LDI low-pass ladder filters are preserved in the LDD high-pass ladder filters. Exact design of an LDI ladder filter requires the determination of a digital all-pole Chebyshev transfer function which is traditionally done by computer iterative methods. In this paper, a set of closed-form design formulas is derived for such transfer functions. A new factorization technique is developed to improve the numerical accuracy of the design, several synthesis equations are reformulated and they are incorporated into a comprehensive LDI ladder filter design program.

Appendix I
Proof of $J_f^{-1}(m)F_f(m) = k(m)/2$ in the $\boldsymbol{K}(z)\boldsymbol{K}(z^{-1})$ Polynomial Factorization Procedure

Since both $J_f(m)$ and $F_f(m)$ are calculated at $k = k(m)$, the iteration index m could be omitted in the following discussion. The equation

$$J_f^{-1}F_f = k/2$$

is equivalent to

$$J_f k = 2F_f$$

where

$$k = \begin{bmatrix} k_0 & k_1 & k_2 & \cdots & k_n \end{bmatrix}^T$$

$$F_f = \begin{bmatrix} f_{f_0} & f_{f_1} & f_{f_2} & \cdots & f_{f_n} \end{bmatrix}^T$$

$$f_{f_i} = \sum_{p=0}^{i} k_p k_{p+n-i}$$

and the Jacobian matrix

$$J_f = \left[\frac{\partial f_{f_i}}{\partial k_j}\right].$$

Now,

$$J_f k = \left[\frac{\partial f_{f_i}}{\partial k_j}\right][k_j]$$

$$= \left[\sum_{j=0}^{n} k_j \frac{\partial f_{f_i}}{\partial k_i}\right].$$

But

$$2F_f = [2f_{f_i}].$$

In other words, it is required to prove

$$\sum_{j=0}^{n}\left[k_j \frac{\partial f_{f_i}}{\partial k_j}\right] = 2f_{f_i}, \qquad \text{any } i.$$

Recall (42) that

$$\frac{\partial f_{f_i}}{\partial k_j} = k_{j-i+n} + k_{j+i-n}$$

with the "index in range" condition. Now,

$$\sum_{j=0}^{n}\left[k_j \frac{\partial f_{f_i}}{\partial k_j}\right] = \sum_{j=0}^{n}[k_j k_{j-i+n}] + \sum_{j=0}^{n}[k_j k_{j+i-n}].$$

Let $j = p$ in the first summation and $j = p - i + n$ in the second summation on the RHS of this equation, then

$$\sum_{j=0}^{n}\left[k_j \frac{\partial f_{f_i}}{\partial k_j}\right] = \sum_{p=0}^{n}[k_p k_{p-i+n}] + \sum_{p=i-n}^{i}[k_{p-i+n} k_p]. \tag{A1}$$

However, the upper bound of p in the first summation can be truncated to i, and the lower bound of p in the second summation can be truncated to 0, because all the terms truncated have at least one index less than zero or greater than n and thus are zero by the "index in range" definition. Therefore, equation (A1) becomes

$$\sum_{j=0}^{n}\left[k_j \frac{\partial f_{f_i}}{\partial k_j}\right] = 2\sum_{p=0}^{i}[k_p k_{p-i+n}] = 2f_i$$

and thus

$$J_f k = 2F_f$$

or

$$J_f^{-1}(m)F_f(m) = k(m)/2$$

as required.

References

[1] L. T. Bruton, "Low sensitivity digital ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 168–176, Mar. 1975.

[2] D. A. Vaughan-Pope and L. T. Bruton, "Transfer function synthesis using generalized doubly terminated two-pair networks," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 79–88, Feb. 1977.

[3] L. T. Bruton, "Transformations and terminating networks in discrete leapfrog ladder structures," in *Int. Symp. Circuits Systems*, (San Francisco, CA) pp. 692–696, Apr. 1974.

[4] D. A. Vaughan-Pope and L. T. Bruton, "Synthesis of singly-terminated lossless digital integrator ladder filters," in *Proc. 1975 Midwest Symp. Circuits and Systems*, pp. 314–318, 1975.

[5] G. M. Jacobs, D. J. Allstot, R. W. Brodersen and P. R. Gray, "Design techniques for MOS switched capacitor ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-25, pp. 1014–1021, Dec. 1978.

[6] K. Martin, "Improved circuits for the realization of switched-capacitor filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, No. 4, pp. 237–244, Apr. 1980.

[7] M. S. Lee and C. Chang, "Switched-capacitor filters using the LDI and bilinear transformations," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 265–270, Apr. 1981.

[8] R. D. Davis and T. N. Trick, "Optimum design of lowpass switched-capacitor ladder filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, No. 6, pp. 522–527, June 1980.

[9] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1975.

[10] L. R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing," Englewood Cliffs, NJ: Prentice Hall, 1975.

[11] A. Antoniou, *Digital Filters: Analysis and Design.* New York: McGraw Hill, 1979.

[12] R. W. Daniels, *Approximation Methods for Electronic Filter Design.* New York: McGraw Hill, 1974.

[13] L. W. Johnson and R. D. Riess, "Numerical Analysis," Addison-Wesley, 1977.

[14] J. B. Scarborough, "Numerical Mathematical Analysis," The John Hopkins Press, 1966.

[15] S. O. Scanlan, "Analysis and synthesis of switched-capacitor state-variable filters,' *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 85–93, Feb. 1981.

E. S. K. Liu was born in Hong Kong on December 30, 1955. He received the B.A.Sc. degree in electrical engineering from the University of Ottawa, Ottawa, Ont., Canada, in 1980, and the M.Sc. degree in electrical engineering from The University of Calgary, Calgary, Alta., Canada, in 1983. He is currently a graduate student in the Department of Computer Science at The University of Calgary and has been awarded the City of Calgary/McMahon Stadium Society Graduate Scholarship.

His areas of interest include digital filter design and analysis, compiler design and integrated circuit design.

L. E. Turner received the B.Sc. and Ph.D. degrees in electrical engineering from The University of Calgary, Calgary, Alta., Canada, in 1974 and 1979, respectively.

He is at present Assistant Professor in the Department of Electrical Engineering, The University of Calgary, Calgary, Alta., Canada. His research interests include digital signal processing, practical recursive digital filter implementation and nonlinear effects in recursive digital filters.

L. T. Bruton is a Professor of Electrical Engineering and Dean of Engineering at the University of Victoria, Victoria, B.C., Canada. His research interests are in the areas of analog and digital signal processing with particular emphasis on the design and implementation of microelectronic active and digital filters.