Where $\overline{A}_i(1/z)$ is defined as

$$\overline{A}_i(1/z) = 1 - \sum_{j=1}^{i-1} a_{i-j}^* z^{-j}.$$  (8)

From (1), the prediction error can now be written as

$$E_i(z) = A_i(z) S(z) - k_i z^{-l(i)} L_i(z), \quad (9)$$

Note that the first term is essentially the error at state $i = 1$. Let the backward prediction error be defined from the second term of (9), that is,

$$A_i(z) = z^{-l(i)} L_i(z) S(z). \quad (10)$$

The inverse z-transform of (11) yields, after an index change,

$$b(m) = s(m - i) - \sum_{j=1}^{i} a_{i-j}^* s(m + j - i). \quad (12)$$

The backward prediction error is seen as the error in predicting the sample at time $m - i$ from all future samples.

The z-transform of the backward prediction error at state $i = 1$, from (11), can be written as

$$B_1(z) = z^{-l} \left[ 1 - \sum_{j=1}^{i-1} a_{i-j}^* z^{-j} \right] S(z). \quad (13)$$

The z-transform of the forward error, using (9), can be written as

$$E_i(z) = E_i - k_i z^{-l} S(z). \quad (14)$$

Using (13), the prediction error is found from (14) via the inverse z-transform to be

$$e_i(m) = d_i(m) - k_i b_{i-1}(m - 1). \quad (15)$$

Thus, the lattice can be viewed as the combination of a forward and backward predictor.

An expression for the backward prediction error similar to (15) can be found [4] by combining (6) and (10), yielding

$$b_i(z) = z^{-l} B_{i-1}(z) S(z). \quad (16)$$

After substituting (1) into (16), the inverse z-transform will yield

$$b_i(m) = b_{i-1}(m - 1) - k_i e_{i-1}(m). \quad (17)$$

Equations (15) and (17), of course, allow the reflection coefficients to be computed in the lattice filter formulation of Fig. 1.

The stepdown procedure can now be obtained [4] using the similarity between the forward and backward predictors, and one additional relationship for $A_i(1/z)$ that follows from (8), namely,

$$\overline{A}_i(1/z) = \overline{A}_i(z)/l - k_i z^{-l} A_i(1/z). \quad (18)$$

Substituting (18) into (8), we obtain

$$A_i(z) = A_{i-1}(z) [1 - k_i^2] - k_i z^{-l} \overline{A}_i(1/z). \quad (19)$$

Solving for $A_{i-1}(1/z)$, we find that

$$A_{i-1}(z) = [A_i(z) + k_i z^{-l} \overline{A}_i(1/z)]/[1 - k_i^2]. \quad (20)$$

where $|k_i| < 1$. Recall that the stepdown recursion assumes a stable filter.

Substituting (2) and (20) into (22) gives the desired result,

$$1 - S_{i-1}^{-1} f_i = \sum_{j=1}^{i-1} a_{i-j}^* f_j + \eta_i z^{-l}.$$  (21)

Equation (21) is solved recursively by equating polynomial coefficients for $i = p, p - 1, \ldots, 1$ to obtain a set of reflection coefficients from a set of predictor coefficients. The computational procedure can be summed up as follows:

$$a_{i-1} = \frac{a_0^* + k_j a_{i-j}^*}{1 - k_i^2}.$$  (22)

for $i = p, p - 1, \ldots, 1$, and $j = 0, 1, \ldots, 1 - 1$, where $a_{(p)} = k_1 = 1$ for $l \neq p$ and $k_i, l < 1$. Equation (22) represents a generalization of the stepdown procedure described in [11. For the case of real coefficients, it is identical to that presented in [13. The lattice realization is equivalent to the predictor parameter realization, from a digital filtering standpoint.

111. CONCLUSIONS

In this correspondence, a technique to transform complex predictor parameters to complex reflection coefficients has been presented. This has been used in processing speech as an analytic (complex) signal [4]. Other applications include complex filter design, complex coefficient coding, and complex predictor parameter to real predictor parameter transformations [4].

REFERENCES


Realization of First-Order Two-Dimensional All-Pass Digital Filters

M. SUDHAKARA REDDY, S. C. DUTTA ROY, AND S. N. HAZRA

Abstract—A structure to realize a first-order two-dimensional allpass transfer function with five multipliers and two delays has been proposed. This has been achieved by modifying the signal Bowgraph of an existing structure which uses six multipliers and two delays. The multipliers of the proposed structure are shown to be real for stable filters.

I. INTRODUCTION

Two-dimensional all-pass first-order functions are used as a mapping function for transformation of 1-D IIR filters to 2-D IIR filters and in cascade with recursive filters to improve the overall phase response of the system. The realization of a general first-order 2-D all-pass function using six multipliers and three delays is described in [1]. More recently, Ganapathy et al. [2] have realized the same with six multipliers and two delays. In this correspondence we propose a structure which uses five multipliers and...
The stability constraints on the coefficients of a first-order 2-D filter. Using stability constraints on the coefficients of the filter function, it is shown that the branch transmittances are always real for stable filters.

For the realization of the above function we consider the signal flowgraph of Fig. 1, is shown in Fig. 2. By transposing Fig. 1, we have another configuration of the signal flowgraph proposed in [2]. The coefficients of the filter function, it is shown that the branch transmittances are always real for stable filters.

For $K_2$ to be real, we must have
\[ a_0 - a_1 + a_2 + 1 > 0, \quad (15a) \]
\[ a_0 - a_1 - a_2 - 1 > 0, \quad (15b) \]
\[ a_0 + a_1 - a_2 + 1 > 0, \quad (15c) \]
\[ a_0 + a_1 + a_2 + 1 > 0. \quad (15d) \]

The condition (14) follows if we multiply (16) and (17), respectively. The condition (14) follows if we multiply (16) and (17). Thus, $K_2$ is always real for stable filters.

We note, however, from (3)-(5) that if $a_l = -1$, then there are no solutions for $K_2$ and $K_3$. In such case the signal flowgraph of Fig. 1 is slightly modified by interchanging the gain of the branches CD and EF. That is, the gain becomes 1 for the branch CD and -1 for the branch EF. With this modification the values of $K_2$ and $K_3$ are given by
\[ K_2 = (a_0 - a_2)/(1 - a_1). \quad (18) \]
and
\[ K_3 = (a_0 a_2 - a_1)/(1 - a_1). \quad (19) \]

The expression for all other branch gains remain unchanged, and their values also remain real for stable filters.

The realization of $H(Z_1, Z_2')$, using the signal flowgraph of Fig. 1, is shown in Fig. 2. By transposing Fig. 1, we have another implementation of (1) using the same number of multiplications and delays.

If we make $a_l = a_0$, then $H(Z_1', Z_2')$ becomes a separable filter. Each of the resulting first-order 1-D all-pass filters can be implemented using one multiplier [5]. Hence, if the first-order 2-
IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING

[Page dimensions: 615.0x796.0]

D all-pass filter is a separable one, then only two multiplications are required to implement it. However, if Fig. 1 is used to implement the separable filter, then $k_3 = 0$, and it requires four multiplications.

III. CONCLUSIONS

A new signal flowgraph is used to implement a 2-D all-pass first-order filter section with five multipliers and two delays. This is an improvement over an existing implementation using six multipliers and two delays. If the first-order all-pass function is a separable one, then the number of multipliers reduces to two.

REFERENCES


Error Feedback in a Class of Orthogonal Polynomial Digital Filter Structures

D. WILLIAMSON AND S. SRIDHARAN

Abstract—A method of applying error feedback in three digital ladder and lattice filter structures derived from the theory of orthogonal polynomials is proposed. It is shown that error feedback leads to significant improvement in both the roundoff noise characteristics and limit cycle behavior. In particular, the error-compensated one-multiplier structure is superior to the (uncompensated) normalized ladder. This results in significant savings in both hardware and speed of implementation.

I. INTRODUCTION

When coefficient quantization is ignored, error feedback and feedforward (which will jointly be abbreviated as EFB) have been shown to significantly reduce both the mean-square amplitude round-off noise and zero input limit cycles in the finite word length (FWL) implementation of direct, normal, and state space digital filters [11-4]. In this correspondence, a method of applying EFB in three ladder and lattice digital filter structures derived from the theory of orthogonal polynomials [5], [6] is presented. In the absence of EFB, Markel and Gray [7] have shown that in an FWL implementation of a narrow-band filter having clustered poles, the normalized (i.e., four-multiplier) ladder is superior to the one-multiplier structure. We show that EFB can produce a significant improvement in the signal-to-quantization-noise ratio (SNR) with only a minimal cost in hardware and speed (only extra adders are required). Furthermore, the one-multiplier structure is shown to provide better performance than the normalized ladder. Since the number of multipliers is reduced by four per section, the savings are significant.

II. ORTHOGONAL POLYNOMIAL STRUCTURES

Consider a filter transfer function defined by

$$H(z) = \frac{P_d(z)}{A_d(z)}$$

where

$$P_d(z) = \sum_{i=0}^{N} p_i z^{-i}; A_d(z) = \sum_{i=0}^{N} a_i z^{-i}; (a_0 = 1).$$

This filter can be implemented via three types of ladder or lattice structures. All realizations involve the "reflection coefficients" $(k)$ and the "tap coefficients" $(v)$. The actual characterization of $(k)$ and $(v)$ may be found in Gray and Markel [5]. A computationally simple procedure, particularly in realizing filter zeros, may be found in Williamson [8]. Any FWL implementation of the structures requires scaling of the filter as well as quantization of filter coefficients and node signals. We will assume that the L2 norm scaling procedure described in [5]-[7] is employed to scale the filter. We do not consider coefficient quantization. Accordingly, we assume that all coefficients are exactly represented by $B$ bit (signed) fractions, with the most significant $B$ bits indicated, producing errors in the filter are carried out to $B$ bit (signed) fractions. For each structure considered, the EFB coefficients will be assumed to be integers, chosen optimally to maximize the output SNR of the filter. We shall see in Section III, that these optimum integer EFB coefficients shall be either $\pm 1$ or 0 so that the implementation of the EFB requires at most an addition or subtraction.

A. Two-Multiplier Lattice

Consider an FWL implementation

$$x_{n+1} = x_n + y_{n} - k_n Q[x_{n}]* - k_n e_{n}$$

$$y_{n+1} = k_n Q[x_{n}]* + x_{n} + \epsilon_{n}$$

(2a)

where $(k_n)$ are the EFB coefficients. The terms $Q[x_{n}]*$ denote the errors resulting from the quantization. That is, $Q[x_{n}]* = x_{n} - (n)$ and similarly for $\epsilon_{n}$. Note that these errors, which result from quantization of the $B + B$ bit nodal signals, are $B + B$ bit fractions, with the most significant $B$ bits, equal to the sign bit. The corresponding input and output are given be

$$\frac{1}{\|F\|} u(n) = x_{n}$$

$$y(n) = \sum_{m=0}^{N} \{Q[x_{n}]* \} 2^{n-m} + 2^{n-m+1}$$

(2c)

where $\{f_{x}\}$ are the scaled tap coefficients given by

$$\{f_{x}\} = \frac{2^{n-m} \{f_{x}\}}{w}$$

(2d)

where $\|F\|_1$ is the input scale factor. Procedures for determining $\{f_{x}\}$, the factor $w$, and the power of two shifts $b$, are explained by Gray and Markel in [5]-[7]. In (2c), $(\{u\})$ are integer EFB coefficients. Fig. 1 illustrates an $M$th order general filter with correction. Note that for the purpose of clarity of Fig. 1, two quantizers are indicated, producing errors $e_{n}$ and $e_{n}$. For the mth section. However, there is only one quantizer. The error $e_{n}$ will