is, all components are calculated in place in the same way as the ordinary DIT algorithm.

In Fig. 1, the first weight in each bunch of butterflies of each stage is unity: $W^0 = 1$, while it is not always so in Fig. 2. Thus, the modified algorithm apparently seems to have more complex multiplications. The brief observation, however, reveals one of the weights in the bunch to be either $W^0 = 1$ or $W^{N/2} = -1$. Since a subtraction is equivalent in complexity to an addition, the modified algorithm has same load of calculations as the ordinary one.

Pruning

Pruning the modified algorithm is shown in Fig. 2. Here again, only calculations which correspond to bold solid lines are necessary. They have a repetitive pattern between adjacent stages, in contrast to the apparent random pattern in Fig. 1.

The repetitive pattern simplifies the modified algorithm. In order to obtain the components of the Kth to $(K + L - 1)$th frequencies, one has only to compute simply from the first to Lth or all butterflies in each bunch of the ith stage, depending on whether $2^{i-1} > L$ or not, respectively. Owing to the repetitive pattern, the bold solid lines exist always only in the first to Lth, or all butterflies in each bunch. This results in a very simple Fortran program as shown in Fig. 3.

This program requires shuffled data and calculates components within desired frequency band effectively. There are shortened butterflies to be calculated in pruned stages, as can be seen in Fig. 2; but this program calculates complete butterflies. For the example of Fig. 2, therefore, it calculates not only the outputs K and $K + 1$, but also $K + 8$ and $K + 9$. Addition (subtraction) time is negligible, and complete butterfly evaluation, which is also employed in [2], fairly simplifies the program.

So far, only the case to compute narrow-band components has been considered. If time sequence has trailing zeros for high resolution of frequency, the algorithm given by [4] can be taken in as similarly as in [2].

References

[1] K. Nagai, "Fourier domain reconstruction of synthetic focus acoustic imaging system," *Proc. IEEE*, vol. 72, p. 748, June 1984.

[2] T. V. Sreenivas and P. V. S. Rao, "High resolution narrow-band spectra by FFT pruning," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 254-257, Apr. 1980.

[3] J. Markel, "FFT pruning," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 305-311, Dec. 1971.

[4] D. P. Skinner, "Pruning the decimation-in-time FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 193-194, Apr. 1976.

The Stepdown Procedure for Complex Predictor Coefficients

JOSEPH PICONE, DIMITRIOS P. PREZAS, WALTER T. HARTWELL, AND JOSEPH L. LoCICERO

Abstract—**The stepdown procedure is an algorithm in which the predictor parameters of a direct form digital filter are converted to the corresponding real reflection coefficients of a digital lattice filter. In this correspondence, a derivation of the stepdown procedure for *complex* predictor parameters is presented. The complex stepdown procedure is a useful tool for applications ranging from complex filter design to complex predictor coefficient transformations.**

Manusript received April 30, 1985; revised November 19, 1985. This work was supported by the Advanced Switching Services Laboratory, AT&T Bell Laboratories, Naperville, IL. This work was submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in electrical engineering in the Graduate School of the Illinois Institute of Technology, Chicago, IL.

J. Picone, D. P. Prezas, and W. T. Hartwell are with AT&T Bell Laboratories—Indian Hill, Naperville, IL 60566.

J. L. LoCicero is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616.

IEEE Log Number 8608699.

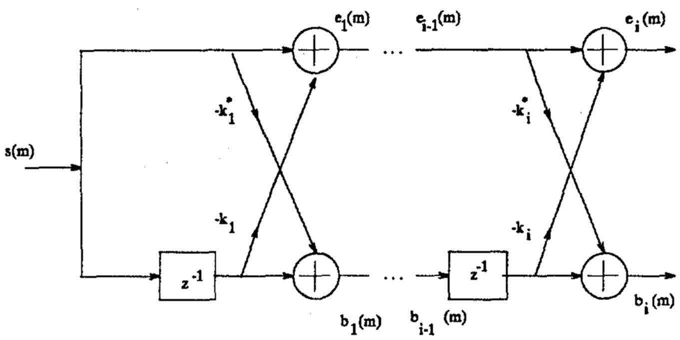

Fig. 1. Complex lattice filter analyzer.

I. Introduction

When modeling a complex time series as an autoregressive (AR) process, or equivalently, as a linearly predictable signal, it is frequently convenient to be able to convert from a lattice filter to a finite impulse response (FIR) filter realization. This is commonly referred to as the process of converting reflection coefficients to predictor parameters. This can be accomplished via the Levinson-Robinson algorithm [1], or the Durbin recursion (or stepup procedure) [2]. It is also useful to be able to convert in the reverse direction, a process that is know as the stepdown procedure [1]. In this correspondence, the stepdown procedure is extended to the complex case. It is assumed that only the highest order predictor coefficients are available.

The complex digital lattice filter is depicted in Fig. 1. The sequence at stage i of the lattice filter, defined as the forward prediction error $e_i(m)$, can be written in z-transform notation as

$$E_i(z) = A_i(z)\, S(z), \tag{1}$$

where

$$A_i(z) = 1 - \sum_{j=1}^{i} a_j z^{-j}, \tag{2}$$

$$S(z) = \mathcal{Z}\{s(mT)\}. \tag{3}$$

The sequence $s(mT)$ represents the input signal $s(t)$ sampled at its Nyquist rate ($1/T$ Hz).

The filter parameters of (2), defined as predictor parameters a_j, can be obtained from reflection coefficients k_j using the Durbin recursion [1]-[3]:

$$a_{i,i} = k_i \tag{4}$$

$$a_{i,j} = a_{i-1,j} - k_i a^*_{i-1,i-j}, \quad 1 \le j \le i - 1. \tag{5}$$

The procedure in (4) and (5) is performed recursively for $i = 1, 2, \cdots, p$, and when $i = p$, the final predictor coefficients are obtained.

II. Derivation of the Complex Stepdown Procedure

The recursion relations of (4) and (5) can be substituted into (2), where $a_j = a_{i,j}$, to give

$$A_i(z) = A_{i-1}(z) - k_i z^{-i}\left[1 - \sum_{j=1}^{i-1} a^*_{i-1,i-j} z^{i-j}\right] \tag{6}$$

$$= A_{i-1}(z) - k_i z^{-i}\overline{A}_{i-1}(1/z), \tag{7}$$

Where $\overline{A}_{i-1}(1/z)$ is defined as

$$\overline{A}_{i-1}(1/z) = 1 - \sum_{j=1}^{i-1} a^*_{i-1,i-j} z^{i-j}. \tag{8}$$

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

$$E_i(z) = A_{i-1}(z)\, S(z) - k_i z^{-i} \overline{A}_{i-1}(1/z)\, S(z). \tag{9}$$

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,

$$B_i(z) = z^{-i} \overline{A}_i(1/z)\, S(z) \tag{10}$$

$$= z^{-i}\left[1 - \sum_{j=1}^{i} a^*_{i,i+1-j} z^{i+1-j}\right] S(z). \tag{11}$$

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

$$b_i(m) = s(m - i) - \sum_{j=1}^{i} a^*_{i,j} s(m + j - i). \tag{12}$$

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

$$B_{i-1}(z) = z^{-i+1}\left[1 - \sum_{j=1}^{i-1} a^*_{i-1,j} z^{j}\right] S(z). \tag{13}$$

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

$$E_i(z) = E_{i-1}(z) - k_i z^{-1}[z^{-i+1} \overline{A}_{i-1}(1/z)\, S(z)]. \tag{14}$$

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

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

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

$$B_i(z) = z^{-1} B_{i-1}(z) - k_i^* A_{i-1}(z)\, S(z). \tag{16}$$

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

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

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 relationships between the forward and backward predictors, and one additional relationship for $A_i(1/z)$ that follows from (8), namely,

$$\overline{A}_i(1/z) = \overline{A}_{i-1}(1/z) - k_i^* z^i A_{i-1}(z). \tag{18}$$

Substituting (18) into (8), we obtain

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

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

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

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

Substituting (2) and (8) into (20) gives the desired result,

$$1 - \sum_{j=1}^{i-1} a_{i-1,j} z^{-j} = \left\{1 - \sum_{j=1}^{i} a_{i,j} z^{-j} + k_i z^{-i} \times \left[1 - \sum_{j=1}^{i} a^*_{i,i+1-j} z^{i+1-j}\right]\right\} \Big/ [1 - |k_i|^2]. \tag{21}$$

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

$$a_{i-1,j} = \frac{a_{i,j} + k_j a^*_{i,i-j}}{1 - |k_i|^2}, \tag{22}$$

with

$$k_i = a_{i,i},$$

for $i = p, p - 1, \cdots, 1$, and $j = 0, 1, \cdots, i - 1$, where $a_{i,0} = k_0 = 1$ for $1 \le i \le p$ and $|k_i| < 1$. Equation (22) represents a generalization of the stepdown procedure described in [1]. For the case of real coefficients, it is identical to that presented in [1]. The lattice filter realization is equivalent to the predictor parameter realization, from a digital filtering standpoint.

III. 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

[1] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.

[2] J. Durbin, "The fitting of time series models," *Rev. Inst. Int. de Stat.*, vol. 28, no. 2, pp. 233–244, 1960.

[3] S. M. Kay and S. L. Marple, Jr., "Spectrum analysis—A modern perspective," *Proc. IEEE*, vol. 69, pp. 1380–1419, Nov. 1981.

[4] J. Picone, "Analytic signal processing," Ph.D. dissertation, Illinois Inst. Technol., Chicago, IL, 1983.

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 all-pass transfer function with five multipliers and two delays has been proposed. This has been achieved by modifying the signal flowgraph 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

Manuscript received May 23, 1985; revised December 10, 1985.

The authors are with the Department of Electrical Engineering, Indian Institute of Technology, New Delhi 110 016, India.

IEEE Log Number 8608144.