Lattice Implementation of Two-Dimensional Recursive Digital Filters

HON KEUNG KWAN, SENIOR MEMBER, IEEE, AND YING CHUN LUI, STUDENT MEMBER, IEEE

Abstract—The implementation of a quarter-plane autoregressive 2-D filter using a newly developed 2-D Levinson algorithm is presented. The resulting structure inherits most of the nicer properties of the well-established 1-D lattice filter such as high modularity, low coefficient sensitivities, low roundoff noise, and elimination of internal overflow. The low sensitivity property of the structure is shown by numerical examples.

I. INTRODUCTION

For 1-D digital filters, there exist many efficient implementation structures such as the parallel form, cascade form, state-space implementation and lattice implementation [1]–[3]. Of special significance is the normalized lattice filter structure [1] developed by Gray and Markel which has many desirable properties such as low coefficient sensitivities and low roundoff noise [2]. The structure has unity energy at all nodes so that no scaling is required and internal overflow cannot occur. The implementation is also highly modular in structure, making it a competitive candidate for VLSI implementation.

Unfortunately, most of the above-mentioned implementation methods cannot be extended in a straightforward manner to 2-D case due to the added dimension and the lack of fundamental theorems of algebra for multidimensional polynomials. There still does not exist an efficient implementation method for general 2-D digital filters. For this reason, it is usually necessary to tailor a design algorithm to suit a particular implementation.

Motivated by the success of 1-D lattice structure, there has been quite a lot of research effort directed to the development of 2-D equivalent lattice structures [4]–[6]. However, all these formulations are capable of implementing only a restricted class of transfer functions. Recently, Lenk and Parker have extended the well-established Levinson algorithm to the 2-D case for modeling 2-D stationary random fields [7]. This algorithm bears a lot of resemblance to its 1-D counterpart which is the backbone of the 1-D lattice structure. The algorithm implies a 2-D lattice structure which consists of arrays of basic 1-D lattice sections. The structure is free from internal overflow so no scaling is required at the input. In addition, the realization is found to have good coefficient quantization properties. In this paper, we include a brief review of the 2-D Levinson algorithm, a description of the 2-D lattice structure and the method of computing the reflection coefficients from a given all-pole transfer function. Finally, numerical examples are presented to demonstrate the low sensitivity property of the implementation. A preliminary version of this paper has already appeared in [8].

II. 2-D LEVINSON ALGORITHM

For a random process \( Y^2 = \{(y(i, j), i = 0, \ldots, M, j = 0, \ldots, M) \) , a quarter-plane support for an arbitrary sample \( y(m, n) \) can be defined as \( Q_N = \{(y(i, j), i = m+1, \ldots, m+N, j = n+1, \ldots, n+N) \) . To make indexing simpler, we can order the elements within the support into a 1-D array:

\[
L_N = \{ y(m, n), y(m-1, n), \ldots, y(m-N, n), y(m, n-1), y(m-1, n-1), \ldots, y(m-N, n-1), \ldots, y(m, n-N), y(m-1, n-N), \ldots, y(m-N, n-N) \}.
\]

The elements of this array can be numbered from 0 to \((N+1)^2 - 1\), this indexing scheme corresponds to a row-wise scanning of \( y(i, j) \) within the support \( Q_N \). The notation \( y((i, j) - p) \) denotes the \( p \)th element behind \( y(i, j) \) in \( L_N \). The forward error associated with the prediction of the sample \( y(m, n) - p \) from the previous \( q \) samples can be defined as

\[
e_p^{(m, n) - p} = \sum_{(i, j) \in ((m, n) - p)} a_{(m, n) - p}^q (i, j) y(i, j)
\]

for \( p = 0, \ldots, N+1, q = 0, \ldots, (N+1)^2 - 1 \).

Similarly, we can define the backward error associated with the prediction of the sample \( y(m, n) - p - q \) from the \( q \) samples prior to it in \( L_N^b \):

\[
r_p^{(m, n) - p} = \sum_{(i, j) \in ((m, n) - p)} b_{(m, n) - p}^q (i, j) y(i, j)
\]

for \( p = 0, \ldots, N+1, q = 0, \ldots, (N+1)^2 - 1 \).

The notation \( a_{(m, n) - p}^q (i, j) \) and \( b_{(m, n) - p}^q (i, j) \) in (2) and (3) are, respectively, the 2-D forward and backward prediction error coefficients. Fig. 1 illustrates the notations defined...
above for the case of $N=2$, $p=1$ and $q=3$. The shaded region is the prediction mask for both the forward and backward errors. The forward error $e_{0,(0,n)}^{(q-1)}$ is associated with the prediction of the point $y(-1,0)$ from all other points in the shaded area while the backward error $r_{0,(0,n)}^{(q-1)}$ is the prediction error of the point $y(-1,-1)$ from all other points within the mask. The forward and backward errors can be updated in order through the use of the following recursions:

$$
\begin{bmatrix}
e_{0,(m,n)-p}^{q} \\
r_{0,(m,n)-p}^{q}
\end{bmatrix} = \Theta_p^q 
\begin{bmatrix}
e_{0,(m,n)-p}^{(q-1)} \\
r_{0,(m,n)-p}^{(q-1)}
\end{bmatrix}
$$

(4)

where $\Theta_p^q$ is the chain scattering matrix:

$$
\Theta_p^q = \frac{1}{\sqrt{1 - (k_p^q)^2}} 
\begin{bmatrix}
1 & -k_p^q \\
-k_p^q & 1
\end{bmatrix}
$$

(5)

$$
k_p^q = E \left\{ e_{0,(m,n)-p}^{(q-1)} r_{0,(m,n)-p}^{(q-1)} \right\}
$$

(6)

is known as the reflection coefficients and $E\{\cdot\}$ denotes statistical expectation. It can be shown that if the original random process has unit energy, i.e., $E\{(y(m,n))^2\} = 1$, then, all the forward and backward error fields will also have unit energy. It can be shown using (2) and (3) that

$$
e_{0,(m,n)-N-1}^{q} = e_{0,(m,n-1)}^{q}
$$

(7)

$$
r_{0,(m,n)-N-1}^{q} = r_{0,(m,n-1)}^{q}
$$

(8)

in other words, $e_{0,(m,n)-N-1}^{q}$ and $r_{0,(m,n)-N-1}^{q}$ are, respectively, equal to $e_{0,(m,n)}^{q}$ and $r_{0,(m,n)}^{q}$ delayed by one row. Starting with $e_{0,(m,n)-p}^{q} = r_{0,(m,n)-p}^{q} = y(m,n) \cdot p$ for $p = 0$ to $N$, (4)-(6) and (7)-(8) can be used recursively to compute the error fields of subsequent lattice stages. The forward error field of the final lattice stages $e_{0,(m,n)}^{q-N+1}$ will then be the optimal error of prediction of the point $y(m,n)$ from all the other points within the support $Q_N$.

III. FILTER IMPLEMENTATION

The transfer function relating the forward error field of the final stage and the input is of the form:

$$
\frac{Y(z_1, z_2)}{E(z_1, z_2)} = \frac{1}{\sum_{i=0}^{N} \sum_{j=0}^{N} a(i,j)z_1^{-i}z_2^{-j}}
$$

(9)

where $E(z_1, z_2)$ and $Y(z_1, z_2)$ represent, respectively, the $z$-transforms of $e_{0,(m,n)}^{(q-N+1)}$ and $y(m,n)$. Now if the input to the 2-D lattice model is the impulse response of an all-pole transfer function of the form in (9), then, the final stage error at the output will be white noise with $z$-transforms equal to unity. By redirecting the signal flow of the lattice model so that the input of the model becomes an output and the output at the final lattice stage becomes an input, we obtain a lattice structure which will generate the impulse responses of the given transfer function for a unit impulse input.

The 2-D lattice structure consists of two types of modules which are depicted in Fig. 2. The operation performed by the type $B$ module is described by the order update equation (4) and the type $A$ module performs the operation:

$$
\begin{bmatrix}
e_{0,(m,n)-p}^{(q-1)} \\
r_{0,(m,n)-p}^{(q-1)}
\end{bmatrix} = \frac{1}{\sqrt{1 - (k_p^q)^2}} 
\begin{bmatrix}
k_p^q & -k_p^q \\
-k_p^q & \sqrt{1 - (k_p^q)^2}
\end{bmatrix} 
\begin{bmatrix}
e_{0,(m,n)-p}^{q} \\
r_{0,(m,n)-p}^{q}
\end{bmatrix}
$$

(10)

It is of interest to note that both types of modules are basic 1-D lattice sections. The overall 2-D lattice structure for the case $N = 2$ is shown in Fig. 3. The structure repeats itself for $q=1$ to $5$. For $q=6$ to $8$, the number of modules for each $q$ decreases progressively and some of the module outputs become redundant.
The following algorithm can be used to calculate the reflection coefficients from a given all-pole quarter-plane transfer function $H(z_1, z_2) = 1/D(z_1, z_2)$ of unit norm:

**Step 0:**

$$A_0^0(z_1, z_2) = B_0^0(z_1, z_2) = z_1^{-p}$$

$$A_{N+1}^0(z_1, z_2) = B_{N+1}^0(z_1, z_2) = z_2^{-1}$$

for $p = 0, \ldots, N$, where $N$ is the order of the transfer function.

**Step 1:**

$$k_p^q = \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{2\pi} A_p^{(q-1)}(z_1, z_2) B_p^{(q-1)}(z_1^{-1}, z_2^{-1}) D(z_1, z_2) D(z_1^{-1}, z_2^{-1}) \, dw_1 \, dw_2$$

(11)

for $z_1 = e^{i\omega}$ and $z_2 = e^{i\omega}$, $q = 1, \ldots, (N+1)^2 - 1$, $p = 0, \ldots, N$. The integral in (11) is usually done by numerical integration. In our case, Simpson’s one-third rule [9] is used to evaluate the integral.

**Step 2: Update Polynomial**

$$\begin{bmatrix} A_p^q(z_1, z_2) \\ B_p^q(z_1, z_2) \end{bmatrix} = \Theta_p^q \begin{bmatrix} A_p^{(q-1)}(z_1, z_2) \\ B_p^{(q-1)}(z_1, z_2) \end{bmatrix}$$

(12)

for $q = 1, \ldots, (N+1)^2 - 1$, $p = 0, \ldots, N$, where $\Theta_p^q$ is defined in (4).
IV. NUMERICAL EXAMPLES

To illustrate the robust numerical characteristics of the 2-D lattice structure, we consider the implementation of a fourth-order all-pole separable transfer function:

\[ H(z_1, z_2) = H_i(z_1) H_j(z_2) \]  

(15)

where \( H_i(z_i) \) for \( i = 1, 2 \) is a fourth-order all-pole 1-D transfer function with poles located at \( 0.8 (\pm n\theta) \) for \( n = 1, 2 \). Different values of \( \theta \) are used so that the degree of pole clustering can be controlled. The computer used for simulation was the VAX 11/750 minicomputer and double precision arithmetic was used throughout. For each transfer function corresponding to a certain value of \( \theta \), the frequency responses of the transfer functions with quantized and unquantized coefficients were compared. The performance criterion used is the normalized mean square error:

\[
\text{ERROR} = \frac{\int_{0}^{2\pi} \int_{0}^{2\pi} |H(z_1, z_2) - H'(z_1, z_2)|^2 \, dw_1 \, dw_2}{\int_{0}^{2\pi} \int_{0}^{2\pi} |H(z_1, z_2)|^2 \, dw_1 \, dw_2}
\]  

(16)

for \( z_1 = e^{i\theta} \) and \( z_2 = e^{i\phi} \) where \( H'(z_1, z_2) \) is the transfer function with quantized coefficient. Table I compares the values of ERROR for different values of \( \theta \) and coefficient wordlength for the direct form and the lattice form. All coefficients are in floating point representation and the wordlength shown in Table I is the number of bits for the mantissa only, the exponent is not quantized. From the tabulated results, it can be seen that the sensitivity of the direct form increases drastically as the pole separation decreases and in some cases, the filter becomes unstable. The sensitivities of both forms of realization are minimum at \( \theta = 60^\circ \) where the poles are evenly distributed around the origin. In all cases, the performance of the lattice form is superior to the direct form and the filter is always stable.

V. CONCLUSIONS

The implementation of an all-pole quarter-plane 2-D digital filter based upon a newly developed 2-D form of Levinson algorithm has been presented. The implementation is free from internal overflow and has good sensitivity properties which have been illustrated by numerical examples. In general, the transfer function of 2-D recursive digital filter consists of a numerator part and a denominator part. In practice, the denominator part can be implemented by the lattice implementation presented in this paper whereas the numerator part can be implemented by a nonrecursive method such as that advanced in [10]. The overall implementation can be obtained by a cascade connection of both parts.

REFERENCES


