

# A HIGHLY MODULAR NORMALIZED ADAPTIVE LATTICE ALGORITHM FOR MULTICHANNEL LEAST SQUARES FILTERING

George-Othon Glentis \* and Cornelis H. Slump

UNIVERSITY OF TWENTE

Department of Electrical Engineering

Laboratory for Network Theory and VLSI design

P.O. Box 217, 7500 AE Enschede, The Netherlands

## ABSTRACT

In this paper a highly modular normalized adaptive lattice algorithm for multichannel Least Squares FIR filtering and multivariable system identification, is presented. Multichannel filters with different number of delay elements per input channel are allowed. The main features of the proposed multichannel adaptive lattice least squares algorithm is the use of scalar only operations, multiplications/divisions and square roots, and the local communication which enables the development of a fully pipelinable architecture.

## 1. INTRODUCTION

Adaptive lattice algorithms update the so called error parameters, i.e., the difference between system's output and desired response signal, for all intermediate filter orders, [1]-[6]. The number of error variables used as well as the operations needed for their time update, depends linearly on the dimension of system's parameters. The error variables are utilized for the computation of the *reflection coefficients* which in turn are used for the update of the error parameters themselves. Moreover, they can serve for the computation of the corresponding transversal filters, using order update recursions.

The proposed normalized adaptive lattice LS algorithm, although it deals with the multichannel problem, it manages to get free of matrix operations altogether, in contrast to known adaptive schemes that require matrix manipulations, [2]-[3]. The multichannel formulation is naturally decomposed into  $k$  single channel subsystems and a single block step is replaced by a sequence of  $k$  successive phases allowing for full pipelining. The proposed scheme is capable for handling multichannel filters with different number of delay elements

assigned to each input channel. The power/angle normalization method is adopted. As a result, all internal variables utilized by the algorithm are within the range of  $[-1, 1]$ .

## 2. PROBLEM FORMULATION

A multi-input single-output FIR filter is described by the following difference equation

$$y(n) = - \sum_{i=1}^k \sum_{l=1}^{m_i} x_i(n-l+1)c_l^i \quad (1)$$

where  $k$  is the number of input channels, and  $x_i(n)$ ,  $i = 1, \dots, k$ , are the input signals. Integer  $m_i$ , named *the order* of the filter with respect to input  $i$ , is, in general, different for each input signal channel,  $m_i \neq m_j$ ,  $i, j = 1, 2, \dots, k$ , [5]-[6].

The vector that carries the filter coefficients is defined as

$$\mathbf{C}_{\mathbf{m}_k} = [\mathbf{c}_{m_i}]_{i=1 \dots k}, \mathbf{c}_{m_i} = [c_l^i]_{l=1 \dots m_i}$$

where,  $\mathbf{m}_k = [m_1, m_2, \dots, m_k]$  is a multi index that consists of the individual filter orders. A set of  $k$  regressor vectors is defined as

$$\mathcal{X}_{\mathbf{m}_k}^j(n) = [\mathbf{x}_{m_i}^i(n-\delta)]_{i=1 \dots k}, \delta = \begin{cases} 1, & i \leq j \\ 0, & i > j \end{cases} \quad (2)$$

where  $\mathbf{x}_{m_i}^i(n) = [x_i(n+1-l)]_{l=1 \dots m_i}$  and  $j = 1 \dots k$ . Then, eq. (1) takes the form

$$y(n) = -\mathcal{X}_{\mathbf{m}_k}^{0t}(n)\mathbf{C}_{\mathbf{m}_k}$$

Given a desired response signal  $z(n)$  and input signals  $x_1(n)$ ,  $x_2(n)$  ..  $x_k(n)$ , the optimum filter, in the Least Squares sense, minimizes the total squared error over a finite data horizon, weighted by an exponentially decay factor,  $0 < \lambda \leq 1$ ,

$$E_{\mathbf{m}_k}(N) = \langle e(n), e(n) \rangle, \quad e(n) = z(n) - y(n)$$

Supported by the European Community, Human Capital Mobility, EUROCHIP program

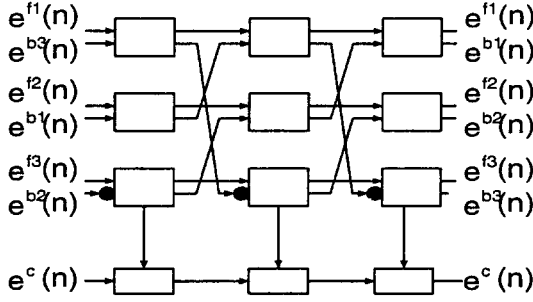


Figure 1: Error variables update ( $k = 3$ ). • represents a unit time delay.

where  $\langle x, y \rangle = \sum_{n=0}^N \lambda^{N-n} xy^t$ . The pertinent LS filter satisfies the set of linear equations

$$\mathbf{R}_{\mathbf{m}_k}^0(N) \mathbf{C}_{\mathbf{m}_k}(N) = -\mathbf{d}_{\mathbf{m}_k}(N) \quad (3)$$

where  $\mathbf{R}_{\mathbf{m}_k}^0(N)$  is the sampled autocorrelation matrix corresponding to the regressor  $\mathcal{X}_{\mathbf{m}_k}^0(n)$ . Following eq. (2), a set of sampled autocorrelation matrices is defined as

$$\mathbf{R}_{\mathbf{m}_k}^j(N) = \langle \mathcal{X}_{\mathbf{m}_k}^j(n) \mathcal{X}_{\mathbf{m}_k}^{jt}(n) \rangle, j = 0, \dots, k$$

$\mathbf{d}_{\mathbf{m}_k}(N)$  is the cross correlation between the inputs and the desired response signal, i.e.,

$$\mathbf{d}_{\mathbf{m}_k}(N) = \langle \mathcal{X}_{\mathbf{m}_k}^0(n) z(n) \rangle$$

While a standard linear system solver can be utilized for the solution of the normal equations, the structure of matrix  $\mathbf{R}_{\mathbf{m}_k}^0(N)$  enables the development of efficient algorithms for the computation of the optimal filter  $\mathbf{C}_{\mathbf{m}_k}(N)$ . The derivation of such fast algorithms is based on the nesting properties of the autocorrelation matrix that permits the order recursive estimation of the optimal filter, starting from  $\mathbf{C}_1(N)$  up to the final filter  $\mathbf{C}_{\mathbf{m}_k}(N)$ , [5]-[6]. Two sets of auxiliary variables are utilized,  $\mathcal{A}^\nu(N)$ ,  $\nu = 1, 2, \dots, k$ , and  $\mathcal{B}^\mu(N)$ ,  $\mu = 1, 2, \dots, k$ , corresponding to multi-input single output forward and backward linear predictors, respectively, [5], [6]. They are defined by the linear system of equations

$$\begin{aligned} \mathbf{R}_{\mathbf{m}_k}^{(\nu)}(N) \mathcal{A}_{\mathbf{m}_k}^{(\nu)}(N) &= -\mathbf{r}_{\mathbf{m}_k}^{f(\nu)}(N), \nu = 1, 2, \dots, k \\ \mathbf{r}_{\mathbf{m}_k}^{f(\nu)}(N) &= \langle \mathcal{X}_{\mathbf{m}_k}^{(\nu)}(n), x_\nu(n) \rangle \\ \mathbf{R}_{\mathbf{m}_k}^{(\mu)}(N) \mathcal{B}_{\mathbf{m}_k}^{(\mu)}(N) &= -\mathbf{r}_{\mathbf{m}_k}^{b(\mu+1)}(N), \mu = 0, 1, \dots, k-1 \\ \mathbf{r}_{\mathbf{m}_k}^{b(\mu+1)}(N) &= \langle \mathcal{X}_{\mathbf{m}_k}^{(\mu)}(n), x_{\mu+1}(n - m_{\mu+1}) \rangle \end{aligned}$$

### 3. THE ADAPTIVE LATTICE ALGORITHM

Filter and forward and backward predictors are adaptively estimated as, [6]

$$\begin{aligned} \mathbf{C}_{\mathbf{m}_k}(N) &= \mathbf{C}_{\mathbf{m}_k}(N-1) + \mathcal{W}_{\mathbf{m}_k}^0(N) \epsilon_{\mathbf{m}_k}^c(N) \\ \epsilon_{\mathbf{m}_k}^c(N) &= z(N) + \mathcal{X}_{\mathbf{m}_k}^{0t}(N) \mathbf{C}_{\mathbf{m}_k}(N) \\ \mathcal{B}_{\mathbf{m}_k}^{(\mu+1)}(N) &= \mathcal{B}_{\mathbf{m}_k}^{(\mu+1)}(N-1) + \mathcal{W}_{\mathbf{m}_k}^\mu(N) \epsilon_{\mathbf{m}_k}^{b(\mu+1)}(N) \\ \epsilon_{\mathbf{m}_k}^{b(\mu+1)}(N) &= x_{\mu+1}(N - m_{\mu+1}) + \mathcal{X}_{\mathbf{m}_k}^{(\mu)t}(N) \mathcal{B}_{\mathbf{m}_k}^{\mu+1}(N) \\ \mathcal{A}_{\mathbf{m}_k}^{(\nu)}(N) &= \mathcal{A}_{\mathbf{m}_k}^{(\nu)}(N-1) + \mathcal{W}_{\mathbf{m}_k}^\nu(N) \epsilon_{\mathbf{m}_k}^{f(\nu)}(N) \\ \epsilon_{\mathbf{m}_k}^{f(\nu)}(N) &= x_\nu(N) + \mathcal{X}_{\mathbf{m}_k}^{(\nu)t}(N) \mathcal{A}_{\mathbf{m}_k}^{(\nu)}(N) \end{aligned}$$

The set of  $k$  Kalman gain vectors are utilized above is defined as

$$\mathbf{R}_{\mathbf{m}_k}^j(N-1) \mathcal{W}_{\mathbf{m}_k}^j(N) = -\mathcal{X}_{\mathbf{m}_k}^j(N), j = 0, 1, 2, \dots, k-1$$

Notice that  $\mathcal{W}_{\mathbf{m}_k}^k(N) = \mathcal{W}_{\mathbf{m}_k}^0(N-1)$ . The a posteriori errors defined above, are related to the corresponding a priori error as  $\epsilon_{\mathbf{m}_k}^c(N) = \epsilon_{\mathbf{m}_k}^c(N) \alpha_{\mathbf{m}_k}^{w(0)}(N)$ ,  $\epsilon_{\mathbf{m}_k}^{f(\nu)}(N) = \epsilon_{\mathbf{m}_k}^{f(\nu)}(N) \alpha_{\mathbf{m}_k}^{w(\nu)}(N)$ ,  $\epsilon_{\mathbf{m}_k}^{b(\mu+1)}(N) = \epsilon_{\mathbf{m}_k}^{b(\mu+1)}(N) \alpha_{\mathbf{m}_k}^{w(\mu)}(N)$ , where  $\alpha_{\mathbf{m}_k}^{w(j)}(N) = 1 - \mathcal{X}_{\mathbf{m}_k}^{jt}(N) \mathcal{W}_{\mathbf{m}_k}^j(N-1)$ . Based on the above time updates and the order recursive scheme of [5], an a posteriori adaptive lattice algorithm has been derived in [6]. During each iteration, a set of reflection coefficients is updated which are turn is utilized for the order update of the error parameters themselves. The algorithm is summarized on Table 1.

A posteriori error variables for the filter as well as for the forward and backward predictors, are estimated using a recursive lattice-ladder scheme. The passage from error variables of order  $\mathbf{m}_k$  to the successive  $\mathbf{m}_k + k$  is accomplished via a set of  $k$  single step recursions. During the first iteration ( $i = 0$ ) of a new step, the forward prediction errors are coupled together with the backward prediction errors in a particular way which is:  $[\epsilon^{f1}, \epsilon^{b1}]$ ,  $[\epsilon^{f2}, \epsilon^{b1}]$ ,  $[\epsilon^{f3}, \epsilon^{b2}]$ ,  $[\epsilon^{fk}, \epsilon^{b(k-1)}]$ . When ( $i = 1$ ) prediction errors are coupled together as:  $[\epsilon^{f1}, \epsilon^{b(k-1)}]$ ,  $[\epsilon^{f2}, \epsilon^{bk}]$ ,  $[\epsilon^{f3}, \epsilon^{b1}]$ ,  $[\epsilon^{fk}, \epsilon^{b(k-2)}]$ . Finally, ( $i = k-1$ ) the pairs are:  $[\epsilon^{f1}, \epsilon^{b1}]$ ,  $[\epsilon^{f2}, \epsilon^{b2}]$ ,  $[\epsilon^{f3}, \epsilon^{b3}]$ ,  $[\epsilon^{fk}, \epsilon^{bk}]$  (see Figure 1).

### 4. THE NORMALIZED ADAPTIVE ALGORITHM

Let us define the power/angle normalized errors, for the filter as well as for the forward and backward predictors,

$$\begin{aligned} c_{\mathbf{m}_k}(N) &= \epsilon_{\mathbf{m}_k}(N) \alpha_{\mathbf{m}_k}^{-\frac{b(1)}{2}}(N) \alpha_{\mathbf{m}_k}^{\frac{w(0)}{2}}(N) \\ b_{\mathbf{m}_k}^{\mu+1}(N) &= \epsilon_{\mathbf{m}_k}^{b(\mu+1)}(N) \alpha_{\mathbf{m}_k}^{-\frac{b(\mu+1)}{2}}(N) \alpha_{\mathbf{m}_k}^{\frac{w(\mu)}{2}}(N) \\ f_{\mathbf{m}_k}^\nu(N) &= \epsilon_{\mathbf{m}_k}^{f(\nu)}(N) \alpha_{\mathbf{m}_k}^{-\frac{f(\nu)}{2}}(N) \alpha_{\mathbf{m}_k}^{\frac{w(\nu)}{2}}(N) \end{aligned}$$

$$\begin{aligned}\epsilon_0^c(N) &= \epsilon_{\mathbf{m}_k}^c(N), \epsilon_0^{b(\mu)}(N) = \epsilon_{\mathbf{m}_k}^{b(\mu)}(N), \epsilon_0^{f(\nu)}(N) = \epsilon_{\mathbf{m}_k}^{f(\nu)}(N) \\ \alpha_0^c(N) &= \alpha_{\mathbf{m}_k}^c(N), \alpha_0^w(N) = \alpha_{\mathbf{m}_k}^w(N) \\ \alpha_0^{b(\mu)}(N) &= \alpha_{\mathbf{m}_k}^{b(\mu)}(N), \alpha_0^{f(\nu)}(N) = \alpha_{\mathbf{m}_k}^{f(\nu)}(N)\end{aligned}$$

FOR  $i = 0$  TO  $(k - 1)$   
 LET  $\mathbf{I} = \text{right\_rotate}[12 \dots k - 1]_{i+1}$   
 FOR  $\mu = 0$  TO  $(k - 1)$ , LET  $\nu = \mathbf{I}(\mu + 1)$

$$\begin{aligned}e_i^{b(\mu+1)}(N) &= \epsilon_i^{b(\mu+1)}(N) \alpha_i^{w(\nu)}(N) \\ e_i^{f(\nu)}(N) &= \epsilon_i^{f(\nu)}(N) Z_N^h \alpha_i^{w(\nu)}(N) \\ \beta_{i+1}^{b(\mu+1)}(N) &= \lambda \beta_{i+1}^{b(\mu+1)}(N - 1) + \epsilon_i^{f(\nu)}(N) Z_N^h e_i^{b(\mu+1)}(N) \\ k_{i+1}^{b(\mu+1)}(N) &= -\beta_{i+1}^{b(\mu+1)}(N) / \alpha_i^{f(\nu)}(N) \\ k_{i+1}^{f(\nu)}(N) &= -\beta_{i+1}^{b(\mu+1)}(N) / Z_N^h \alpha_i^{b(\mu+1)}(N) \\ \epsilon_{i+1}^{b(\mu+1)}(N) &= Z_N^h \epsilon_i^{b(\mu+1)}(N) + \epsilon_i^{f(\nu)}(N) k_{i+1}^{b(\mu+1)}(N) \\ \epsilon_{i+1}^{f(\nu)}(N) &= \epsilon_i^{f(\nu)}(N) + Z_N^h \epsilon_i^{b(\mu+1)}(N) k_{i+1}^{f(\nu)}(N) \\ \alpha_{i+1}^{f(\nu)}(N) &= \alpha_i^{f(\nu)}(N) + \beta_{i+1}^{b(\mu+1)}(N) k_{i+1}^{f(\nu)}(N) \\ &= \lambda \alpha_i^{f(\nu)}(N - 1) + \epsilon_{i+1}^{f(\nu)}(N) e_{i+1}^{f(\nu)}(N) \\ \alpha_{i+1}^{b(\mu+1)}(N) &= Z_N^h \alpha_i^{b(\mu+1)}(N) + \beta_{i+1}^{b(\mu+1)}(N) k_{i+1}^{b(\mu+1)}(N) \\ &= \lambda \alpha_i^{b(\mu+1)}(N - 1) + \epsilon_{i+1}^{b(\mu+1)}(N) e_{i+1}^{b(\mu+1)}(N) \\ a_{i+1}^{w(\nu)}(N) &= a_i^{w(\nu)}(N) + (e_i^{b(\mu+1)}(N))^2 / Z_N^h \alpha_i^{b(\mu+1)}(N)\end{aligned}$$

ENDFOR  $\mu$

$$\begin{aligned}e_i^c(N) &= \epsilon_i^c(N) \alpha_i^{w(i)}(N) \\ \beta_{i+1}^c(N) &= \lambda \beta_{i+1}^c(N) + \epsilon_i^c(N) \epsilon_i^{b(i+1)}(N) \alpha_i(N) \\ k_{i+1}^c(N) &= -\beta_{i+1}^c(N) / \alpha_i^{b(i+1)}(N) \\ \epsilon_{i+1}^c(N) &= \epsilon_i^c(N) + \epsilon_i^{b(i+1)}(N) k_{i+1}^c(N) \\ \alpha_{i+1}^c(N) &= \alpha_i^c(N) + \beta_{i+1}^c(N) k_{i+1}^c(N) \\ &= \lambda \alpha_i^c(N - 1) + \epsilon_{i+1}^c(N) e_{i+1}^c(N)\end{aligned}$$

ENDFOR  $i$

$$\begin{aligned}\epsilon_{\mathbf{m}_k+k}^c(N) &= \epsilon_k^c(N) \\ \epsilon_{\mathbf{m}_k+k}^{b(\mu)}(N) &= \epsilon_k^{b(\mu)}(N), \epsilon_{\mathbf{m}_k+k}^{f(\nu)}(N) = \epsilon_k^{f(\nu)}(N) \\ \alpha_{\mathbf{m}_k+k}^c(N) &= \alpha_k^c(N), \alpha_{\mathbf{m}_k+k}^w(N) = \alpha_k^w(N) \\ \alpha_{\mathbf{m}_k+k}^{b(\mu)}(N) &= \alpha_k^{b(\mu)}(N), \alpha_{\mathbf{m}_k+k}^{f(\nu)}(N) = \alpha_k^{f(\nu)}(N)\end{aligned}$$

Table 1. The a posteriori adaptive lattice algorithm.  $Z_N$  represents a delay with respect to  $N$ . It is activated by  $h$ , as:  $h = 1$ , if  $\nu = k$ ,  $h = 0$ , otherwise.

Moreover, define the normalized filter reflection coefficients, and a single set of normalized combined forward/backward reflection coefficients, for all intermediate orders  $i = 0, 1 \dots k$ ,

$$\begin{aligned}l_{i+1}(N) &= \beta_{i+1}^c(N) \alpha_i^{-\frac{c}{2}}(N) \alpha_i^{-\frac{b(i+1)}{2}}(N) \\ k_{i+1}^{b(\mu+1)}(N) &= \beta_{i+1}^{\mu+1}(N) \alpha_i^{-\frac{f(\nu)}{2}}(N) Z_N^h \alpha_i^{-\frac{b(\mu+1)}{2}}(N)\end{aligned}$$

Then, it is easy to prove that

$$\begin{aligned}\alpha_{i+1}^{b(\mu+1)}(N) (Z_N^h \alpha_i^{-b(\mu+1)}(N)) &= \\ \alpha_{i+1}^{f(\nu)}(N) \alpha_i^{-f(\nu)}(N) &= 1 - (k_{i+1}^{\mu+1}(N))^2 \\ \alpha_{i+1}^c(N) \alpha_i^{-c}(N) &= 1 - (l_{i+1}(N))^2\end{aligned}$$

Thus  $|k_{i+1}^{\mu+1}(N)| \leq 1$  and  $|l_{i+1}(N)| \leq 1$ . Moreover,

$$\begin{aligned}\alpha_i^c(N) \alpha_i^{-c}(N - 1) &= \lambda - (c_i(N))^2 \\ \alpha_i^{b(\mu+1)}(N) \alpha_i^{-b(\mu+1)}(N - 1) &= \lambda - (b_i^{\mu+1}(N))^2 \\ \alpha_i^{f(\nu)}(N) \alpha_i^{-f(\nu)}(N - 1) &= \lambda - (f_i^\nu(N))^2\end{aligned}$$

which result to  $|c_i(N)| \leq 1$ ,  $|b_i^{\mu+1}(N)| \leq 1$ , and  $|f_i^\nu(N)| \leq 1$ .

The normalized algorithm of Table 2 is then derived, by applying the above definitions to the corresponding variables of Table 1. The lattice algorithm of Table 2 is depicted in Figure 2, for a special case of a three channel problem,  $k = 3$ , and final multi index  $\mathbf{m}_3 = [m_1, m_2, m_3] = [7, 3, 1]$ .

## 5. CONCLUSIONS

The highly modular normalized adaptive lattice algorithm for multichannel Least Squares FIR filtering presented. Multichannel filters with different number of delay elements per input channel are allowed. Two basic units, the lattice cell and ladder cell, are required for the implementation of the algorithm. Scalar only operations, multiplications/divisions and square roots, are utilized. In addition, the inherent modularity and the local communication enables the development of a fully pipelinable architecture.

## 6. REFERENCES

- [1] N. Kalouptsidis and S. Theodoridis, (eds), Adaptive System Identification and Signal Processing, Prentice Hall, 1993.
- [2] D. Lee, M. Morf, and B. Friedlander, 'Recursive Least squares ladder estimation algorithm,' IEEE ASSP-32, pp. 381-389, April 1984.
- [3] F. Ling, and J.G. Proakis, 'A generalized multichannel least squares lattice algorithm based on sequential processing stages,' IEEE ASSP-29, pp. 627-641, 1981.
- [4] H. Levi-Ari, 'Modular architecture for adaptive multichannel algorithms, IEEE ASSP-87, pp. 543-52, 1987.
- [5] G. Glentis and N. Kalouptsidis, Efficient Order Recursive Algorithms for Multichannel Least Squares Filtering, IEEE SP-40, pp.1354-1374, June 1992.
- [6] G. Glentis and N. Kalouptsidis, Fast Adaptive Algorithms for Multichannel Filtering and System Identification, IEEE SP-40, pp. 2433-2458, October 1992.

$$\alpha_{[0]}^c(N) = \lambda \alpha_{[0]}^c(N-1) + z^2(N), \alpha_{[0]}^{fj}(N) = \lambda \alpha_{[0]}^{fj}(N-1) + x_j^2(N)$$

$$c_{[0]}(N) = z(N) \alpha_{[0]}^{-\frac{s}{2}}(N), f_{[0]}^j(N) = b_{[0]}^j(N) = x_j(N) \alpha_{[0]}^{-\frac{f_j}{2}}(N)$$

FOR  $s = 1$  TO  $k$   
 FOR  $j = 1$  TO  $m_s - m_{s+1}$ ,  $c_0(N) = c_{\mathbf{m}_s}(N), b_0^\mu(N) = b_{\mathbf{m}_s}^\mu(N), f_0^\nu(N) = f_{\mathbf{m}_s}^\nu(N)$   
 FOR  $i = 0$  TO  $(s-1)$  LET  $\mathbf{I} = \text{right\_rotate}[12 \dots s-1]_{i+1}$

$$\ell_{i+1}(N) = \frac{\sqrt{1 - (b_i^{i+1}(N))^2} \sqrt{1 - (c_i(N))^2} \ell_{i+1}(N-1) + b_i^{i+1}(N) c_i(N)}{c_i(N) - b_i^{i+1}(N) \ell_i(N)}$$

$$c_{i+1}(N) = \frac{c_i(N) - b_i^{i+1}(N) \ell_i(N)}{\sqrt{1 - (b_i^{i+1}(N))^2} \sqrt{1 - (\ell_{i+1}(N))^2}}$$

FOR  $\mu = 0$  TO  $(s-1)$  LET  $\nu = \mathbf{I}(\mu+1)$

$$k_{i+1}^{\mu+1}(N) = \frac{\sqrt{1 - (f_i^\nu(N))^2} \sqrt{1 - (Z_N^h b_i^{\mu+1}(N))^2} k_{i+1}^{\mu+1}(N-1) + Z_N^h b_i^{\mu+1}(N) f_i^\nu(N)}{f_i^\nu(N) - Z_N^h b_i^{\mu+1}(N) k_{i+1}^{\mu+1}(N)}$$

$$f_{i+1}^\nu(N) = \frac{\sqrt{1 - (Z_N^h b_i^{\mu+1}(N))^2} \sqrt{1 - (k_{i+1}^{\mu+1}(N))^2}}{Z_N^h b_i^{\mu+1}(N) - f_i^\nu(N) k_{i+1}^{\mu+1}(N)}$$

$$b_{i+1}^{\mu+1}(N) = \frac{Z_N^h b_i^{\mu+1}(N) - f_i^\nu(N) k_{i+1}^{\mu+1}(N)}{\sqrt{1 - (f_i^\nu(N))^2} \sqrt{1 - (k_{i+1}^{\mu+1}(N))^2}}$$

ENDFOR  $\mu$

FOR  $\mu = s+1$  TO  $k$

$$k_{i+1}^\mu(N) = \frac{\sqrt{1 - (f_i^\mu(N))^2} \sqrt{1 - (b_i^{i+1}(N-1))^2} k_{i+1}^\mu(N-1) + b_i^{i+1}(N-1) f_i^\mu(N)}{f_i^\mu(N) - b_i^{i+1}(N-1) k_{i+1}^\mu(N)}$$

$$b_{i+1}^\mu(N) = f_{i+1}^\mu(N) = \frac{f_i^\mu(N) - b_i^{i+1}(N-1) k_{i+1}^\mu(N)}{\sqrt{1 - (b_i^{i+1}(N-1))^2} \sqrt{1 - (k_{i+1}^\mu(N))^2}}$$

ENDFOR  $\mu$

ENDFOR  $i$   $c_{\mathbf{m}_s+s}(N) = c_s(N), b_{\mathbf{m}_s+s}^\mu(N) = b_s^\mu(N), f_{\mathbf{m}_s+s}^\nu(N) = f_s^\nu(N)$

ENDFOR  $j$

ENDFOR  $s$

Table 2. The normalized adaptive lattice algorithm

lattice cell

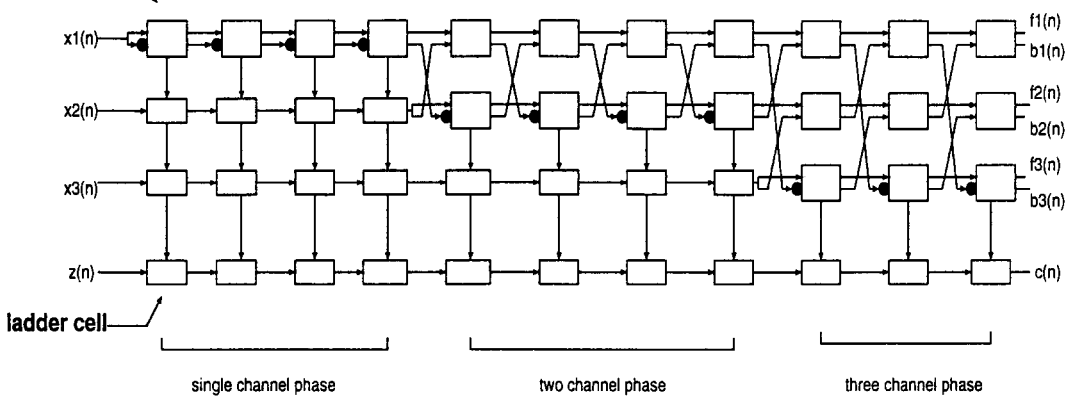


Figure 2. The highly modular normalized adaptive lattice structure for  $\mathbf{m}_3 = [7, 3, 1]$ .