

A highly modular adaptive lattice algorithm for multichannel least squares filtering

George-Othon Glentis^{a,*}, Nicholas Kalouptsidis^b

^a *University of Twente, Department of Electrical Engineering, EL-BSC, Laboratory for Network Theory and VLSI-Design, P.O. Box 217, 7500 AE Enschede, The Netherlands*

^b *University of Athens, Department of Informatics, Panepistimiopolis, TYPA Buildings, Athens 157 71, Greece*

Received 18 June 1993; revised 8 May 1995

Abstract

In this paper a highly modular 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 additions, and the local communication which enables the development of a fully pipelining architecture. The tracking capability and the numerical stability and accuracy of the proposed technique are illustrated by simulations.

Zusammenfassung

Ein weitgehend modularer adaptiver Lattice-Algorithmus zur mehrkanaligen Kleinste-Quadrate-FIR-Filterung und zur mehrdimensionalen Systemidentifikation wird vorgestellt. Mehrkanalige Filter mit unterschiedlichen Anzahlen von Verzögerungs-elementen je Kanal sind erlaubt. Hauptmerkmale des vorgeschlagenen adaptiven Mehrkanal-Lattice-Algorithmus' auf Kleinste-Quadrate-Basis sind die Verwendung rein skalarer Operationen, Multiplikationen/Divisionen und Additionen und die lokale Kommunikation, welche die Entwicklung einer vollständigen Pipelinestruktur ermöglicht. Die Trackingfähigkeit sowie die numerische Stabilität und Genauigkeit der vorgeschlagenen Technik werden durch Simulationen illustriert.

Résumé

Cet article présente un algorithme en treillis adaptatif hautement modulaire pour le filtrage RIF aux moindres carrés multicanal. Les filtres multicanaux avec un nombre différent d'éléments de délai par entrée sont autorisés. Les caractéristiques principales de l'algorithme des moindres carrés en treillis, adaptatif et multicanal est l'utilisation d'opérations scalaires uniquement, multiplications/divisions et additions, et la communication locale qui permet le développement d'une architecture entièrement en pipeline. La capacité de suivi ainsi que la stabilité numérique et la précision de la technique proposée sont illustrées par des simulations.

Keywords: Multichannel filtering and system identification; Least squares estimation; Adaptive lattice algorithms

* Corresponding author.

1. Introduction

In this paper a fast adaptive algorithm for an equivalent description of the filter parameters is developed, using lattice structures. Adaptive lattice realizations are popular due to a number of nice properties they share. They offer modular structure suitable for VLSI implementation, decoupling between successive stages which allow for pipelining, and insensitivity to round-off noise [1–3, 7–9]. The proposed adaptive lattice least squares algorithm for the multichannel filtering has an additional feature. 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 [10]. 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. Moreover, the proposed scheme is capable for handling multichannel filters with different number of delay elements assigned to each input channel. When the estimation of the filter coefficients is the desired task, supplementary two terms Levinson type recursions can be employed, for the efficient computation of the filter coefficients in a purely order recursive way.

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 sequences $x_1(n), x_2(n), \dots, x_k(n)$, $n \geq 0$, 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, $m_i \neq m_j$, $i, j = 1, 2, \dots, k$ [4–6]. For each input i let us define the coefficients vector $\mathbf{c}_{m_i}^i = [c_1^i \ c_2^i \ \dots \ c_{m_i}^i]^T$ and the corresponding data vector $\mathbf{x}_{m_i}^i(n) = [x_i(n)x_i(n-1) \ \dots \ x_i(n-m_i+1)]^T$, of dimensions $m_i \times 1$. We insert these vectors into the block vectors

$\mathbf{c}_{m_k} = [\mathbf{c}_{m_i}^i]_{i=1,2,\dots,k}$ and $\mathbf{x}_{m_k}(n) = [\mathbf{x}_{m_1}^{1T}(n) \ \mathbf{x}_{m_2}^{2T}(n) \ \dots \ \mathbf{x}_{m_k}^{kT}(n)]^T$, where the *multi-index* m_k consists of the individual filter orders, i.e., $m_k = [m_1, m_2, \dots, m_k]$. $\mathbf{x}_{m_k}(n)$ and \mathbf{c}_{m_k} are block vectors of block order k , with entries vectors of dimensions $m_i \times 1$. They have dimension $M \times 1$, where M is the sum of entries dimensions, $M = \sum_{i=1}^k m_i$. Then, Eq. (1) is readily expressed in a compact form as

$$y(n) = - \mathbf{x}_{m_k}^T(n) \mathbf{c}_{m_k}.$$

The least squares design of a multichannel FIR filter if formulated as follows. Let $e_{m_k}^c(n) = z(n) - y(n) = z(n) + \mathbf{x}_{m_k}^T(n) \mathbf{c}_{m_k}$ be the error between filter's output $y(n)$ and a desired response signal $z(n)$ at time instant n . The optimum filter, in the LS sense, minimizes the total squared error over a finite data horizon, i.e., $n \in [0, N]$, weighted by an exponentially decay factor, $0 < \lambda \leq 1$,

$$E_{m_k}^c(N) = \sum_{n=0}^N \lambda^{N-n} (e_{m_k}^c(n))^2.$$

The pertinent LS filter, $\mathbf{c}_{m_k}^{LS} \equiv \mathbf{c}_{m_k}(N)$, satisfies the set of linear equations

$$\mathbf{R}_{m_k}(N) \mathbf{c}_{m_k}(N) = - \mathbf{d}_{m_k}(N), \quad (2)$$

where

$$\mathbf{R}_{m_k}(N) = \sum_{n=0}^N \lambda^{N-n} \mathbf{x}_{m_k}(n) \mathbf{x}_{m_k}^T(n),$$

$$\mathbf{d}_{m_k}(N) = \sum_{n=0}^N \lambda^{N-n} \mathbf{x}_{m_k}(n) z(n).$$

A primary objective in adaptive filtering is the time update of the parameters vector and the estimation of the output error on the basis of the newcoming information, i.e.,

$$\begin{aligned} \mathbf{c}_{m_k}(N+1) &= \mathcal{G}(\mathbf{c}_{m_k}(N), z(N+1), [x_i(N+1)]_{i=1,\dots,k}), \\ e_{m_k}^c(N+1) &= z(N+1) - y(N+1). \end{aligned}$$

Both tasks can be accomplished by means of fast transversal and fast lattice adaptive algorithms, as is described in the following sections.

2.1. The transversal multichannel LS algorithm

Eq. (2) determines the optimal FIR filter in the least squares sense. The pertinent filter is obtained as the solution of the linear system of equations (2). While a standard linear system solver can be utilized, the structure of matrix $\mathbf{R}_{m_k}(N)$ enables the development of efficient algorithms for the computation of the optimal filter $\mathbf{c}_{m_k}(N)$. The derivation of such fast algorithms is based on the nesting properties of matrix $\mathbf{R}_{m_k}(N)$ that permits the order recursive estimation of the optimal filter, starting from $\mathbf{c}_1(N)$ up to the final filter $\mathbf{c}_{m_k}(N)$ [4–6]. Two sets of auxiliary vectors are utilized, $\mathbf{a}^\nu(N)$, $\nu = 1, 2, \dots, k$, and $\mathbf{b}^\mu(N)$, $\mu = 1, 2, \dots, k$, corresponding to multi-input single output forward and backward linear predictors, respectively [4, 6].

Linear prediction can be viewed as a special case of filtering, where the desired response signal is a future value of the underline process. Consider the data vector

$$\begin{aligned} \mathbf{x}_{m_k}^{(\mu)}(n) &= [\mathbf{x}_{m_1}^{1T}(n-1) \ \mathbf{x}_{m_2}^{2T}(n-1) \ \dots \\ &\quad \mathbf{x}_{m_\mu}^{\mu T}(n-1) \ \mathbf{x}_{m_{\mu+1}}^{(\mu+1)T}(n) \ \dots \ \mathbf{x}_{m_k}^{kT}(n)]^T, \\ \mu &= 0, 1, 2, \dots, k, \end{aligned} \quad (3)$$

where the μ first entry vectors are delayed by a unit shift. Clearly, $\mathbf{x}_{m_k}^0(n) = \mathbf{x}_{m_k}(n)$ and $\mathbf{x}_{m_k}^k(n) = \mathbf{x}_{m_k}(n-1)$. Then the following multi-input single-output type forward and backward predictors are defined:

(a) The ν -channel one step ahead forward predictor $\mathbf{a}_{m_k}^{(\nu)}(N)$, $\nu = 1, 2, \dots, k$, is obtained by minimizing the total squared error

$$E_{m_k}^{f(\nu)}(N) = \sum_{n=0}^N \lambda^{N-n} (\mathbf{x}_\nu(n) + \mathbf{x}_{m_k}^{(\nu)T}(n) \mathbf{a}_{m_k}^{(\nu)}(N))^2,$$

and hence satisfies the normal equations

$$\mathbf{R}_{m_k}^{(\nu)}(N) \mathbf{a}_{m_k}^{(\nu)}(N) = -\mathbf{r}_{m_k}^{f(\nu)}(N), \quad (4)$$

where

$$\mathbf{R}_{m_k}^{(\nu)}(N) = \sum_{n=0}^N \lambda^{N-n} \mathbf{x}_{m_k}^{(\nu)}(n) \mathbf{x}_{m_k}^{(\nu)T}(n),$$

$$\mathbf{r}_{m_k}^{f(\nu)}(N) = \sum_{n=0}^N \lambda^{N-1} \mathbf{x}_{m_k}^{(\nu)}(n) x_\nu(n).$$

(b) The $\mu + 1$ -channel one step backward predictor $\mathbf{b}_{m_k}^{(\mu+1)}(N)$, $\mu = 0, 1, \dots, k-1$, minimizes the total squared error

$$\begin{aligned} E_{m_k}^{b(\mu+1)}(N) &= \sum_{n=0}^N \lambda^{N-n} (\mathbf{x}_{\mu+1}(n - m_{\mu+1}) \\ &\quad + \mathbf{x}_{m_k}^{(\mu)T}(n) \mathbf{b}_{m_k}^{(\mu+1)}(N))^2 \end{aligned}$$

and is obtained by the linear system

$$\mathbf{R}_{m_k}^{(\mu)}(N) \mathbf{b}_{m_k}^{(\mu+1)} = -\mathbf{r}_{m_k}^{b(\mu+1)}(N), \quad (5)$$

where

$$\mathbf{R}_{m_k}^{(\mu)}(N) = \sum_{n=0}^N \lambda^{N-n} \mathbf{x}_{m_k}^{(\mu)}(n) \mathbf{x}_{m_k}^{(\mu)T}(n),$$

$$\mathbf{r}_{m_k}^{b(\mu+1)}(N) = \sum_{n=0}^N \lambda^{N-n} \mathbf{x}_{m_k}^{(\mu)}(n) x_{\mu+1}(n - m_{\mu+1}).$$

The basic recursions of the efficient order updating algorithm for the determination of the increased order optimal filter $\mathbf{c}_{m_k+k}(N)$ in terms of the lower dimensions counterpart $\mathbf{c}_{m_k}(N)$ are summarized in Table 1. The derivation of the algorithm

Table 1

The multichannel LS filter order recursions. The operator z_N^{h1} denotes a shift with respect to N . It is activated by the superscript $h1$, $h1 = 1$ if $\nu = k$, otherwise $h1 = 0$

FOR $i = 0$ TO $(k-1)$ DO

$$\mathbf{S}_{m_k+i+1}^{(i+1)} \mathbf{c}_{m_k+i+1}(N) = \begin{pmatrix} \mathbf{c}_{m_k+i}(N) \\ 0 \end{pmatrix} + \begin{pmatrix} \mathbf{b}_{m_k+i}^{(i+1)}(N) \\ 1 \end{pmatrix} k_{m_k+i+1}^{c_{m_k+i+1}}(N)$$

LET $\mathbf{I} = \text{right_rotate}[1 \ 2 \ \dots \ k]_{i+1}$

FOR $\mu = 0$ TO $(k-1)$ DO

LET $\nu = \mathbf{I}(\mu + 1)$

$$\begin{aligned} \mathbf{T}_{m_k+i+1}^{(\nu)} \mathbf{b}_{m_k+i+1}^{(\mu+1)}(N) \\ = \begin{pmatrix} 0 \\ z_N^{h1} \mathbf{b}_{m_k+i}^{(\mu+1)}(N) \end{pmatrix} + \begin{pmatrix} 1 \\ \mathbf{a}_{m_k+i}^{(\nu)}(N) \end{pmatrix} k_{m_k+i+1}^{b_{m_k+i+1}}(N) \end{aligned}$$

$$\begin{aligned} \mathbf{S}_{m_k+i+1}^{(\mu+1)} \mathbf{a}_{m_k+i+1}^{(\nu)}(N) \\ = \begin{pmatrix} \mathbf{a}_{m_k+i}^{(\nu)}(N) \\ 0 \end{pmatrix} + \begin{pmatrix} z_N^{h1} \mathbf{b}_{m_k+i}^{(\mu+1)}(N) \\ 1 \end{pmatrix} k_{m_k+i+1}^{a_{m_k+i+1}}(N) \end{aligned}$$

ENDFOR μ

ENDFOR i

as well as the detailed organization are presented in [4]. Each block step $m_k \rightarrow m_k + k$ is decomposed into k successive single step iterations, $i = 1, 2, \dots, k$, where the solution vector $c_{m_k+i+1}(N)$ is augmented by a new element. Thus, after the completion of a single block step, $c_{m_k}(N)$ has been increased by k elements. A notable feature of the algorithm is the cyclic way in which the forward predictors, a^v , and the backward predictors, b^μ , are updated. The cyclic nature of the parameters coupling is nicely described by introducing the index sequence $[1 \ 2 \ \dots \ k]$ and the rotation operator

$$I = \text{right_rotate}[1, 2, \dots, k]_{i+1},$$

which rotates $[1, 2, \dots, k]$ $(i + 1)$ times to the right. In this way, the backward predictors b^μ associated with the forward predictors a^v at phase i are found if we set $v = I(\mu + 1)$, and $I(\mu + 1)$ means the $(\mu + 1)$ th element of the index sequence I , $\mu = 0, 1, \dots, k$.

The shifted backward predictor $b_{m_k+i}^{i+1}(N-1)$, $i = 0, 1, \dots, k-1$, required by the order recursive algorithm of Table 1 when parameter $h1 = 1$ can directly be estimated from $b_{m_k+i}^{i+1}(N)$ as

$$b_{m_k+i}^{i+1}(N-1) = b_{m_k+i}^{i+1}(N) - w_{m_k+i}(N) \varepsilon_{m_k+i}^{b(i+1)}(N), \quad (6)$$

$w_{m_k+i}(N)$ is the so-called *dual Kalman gain* and it is defined by the linear system $\lambda R_{m_k+i}(N-1) \times w_{m_k+i}(N) = -x_{m_k+i}(N)$. It is recursively updated as [4]

$$\begin{aligned} S_{m_k+i+1}^{(i+1)} w_{m_k+i+1}(N) \\ = \begin{pmatrix} w_{m_k+i}(N) \\ 0 \end{pmatrix} + \begin{pmatrix} b_{m_k+i}^{(i+1)}(N-1) \\ 1 \end{pmatrix} k_{m_k+i+1}^w(N), \end{aligned} \quad (7)$$

$$k_{m_k+i+1}^w(N) = -e_{m_k+i}^{b(i+1)}(N) / (\lambda \alpha_{m_k+i}^{b(i+1)}(N-1)). \quad (8)$$

3. The lattice-ladder multichannel LS algorithm

Adaptive lattice algorithms update the so-called error parameters, that is, the difference between system's output and a desired response signal, for all intermediate filter orders [1–3, 7–9]. 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. An efficient adaptive lattice algorithm for LS multichannel filtering was presented in [5], based on the so-called a posteriori error parameterization for the time update of β variables and error power variables, α . The reflection coefficients sought were estimated as the ratio β/α . Following [8], an alternative adaptive lattice scheme is developed that explicitly update reflection coefficients themselves, utilizing both a priori and a posteriori error variables. As is discussed in [8] the direct updating of the reflection coefficients offers better numerical accuracy and numerical stability.

Let us consider the instantaneous a priori and a posteriori filter error associated with filter $c_{m_k}(N)$ at time instant N , designated by latin e and greek ε typesetting, respectively,

$$\begin{aligned} e_{m_k}^c(N) &= z(N) + x_{m_k}^T(N) c_{m_k}(N-1), \\ \varepsilon_{m_k}^c(N) &= z(N) + x_{m_k}^T(N) c_{m_k}(N). \end{aligned} \quad (9)$$

We also define the a priori and the a posteriori prediction errors associated with the multi-input single output forward and backward predictors, Eqs. (4) and (5),

$$\begin{aligned} e_{m_k}^{f^v}(N) &= x_v(N) + x_{m_k}^{vT}(N) a_{m_k}^v(N-1), \\ \varepsilon_{m_k}^{f^v}(N) &= x_v(N) + x_{m_k}^{vT}(N) a_{m_k}^v(N), \\ v &= 1, \dots, k; \end{aligned} \quad (10)$$

$$\begin{aligned} e_{m_k}^{b(\mu+1)}(N) &= x_{\mu+1}(N - m_{\mu+1}) + x_{m_k}^{\mu T}(N) b_{m_k}^{\mu+1}(N-1), \\ \varepsilon_{m_k}^{b(\mu+1)}(N) &= x_{\mu+1}(N - m_{\mu+1}) + x_{m_k}^{\mu T}(N) b_{m_k}^{\mu+1}(N), \\ \mu &= 0, 1, \dots, k-1; \end{aligned} \quad (11)$$

where the data vectors $x_{m_k}^v(n)$ and $x_{m_k}^\mu(n)$ are given by Eq. (3).

A priori and 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 m_k to the successive $m_k + k$ is accomplished via a set of k single step recursions. The lattice-ladder equations are obtained using the increased order recursions for the filter and forward and

Table 2
Order update recursions for the a priori and a posteriori errors

| A posteriori error updates | A priori error updates |
|---|---|
| FOR $i = 0$ TO $(k - 1)$ DO | For $i = 0$ TO $(k - 1)$ DO |
| $e_{m_k+i+1}^c(N) = e_{m_k+i}^c(N) + e_{m_k+i}^{b(i+1)}(N)k_{m_k+i+1}^c(N)$ | $e_{m_k+i+1}^c(N) = e_{m_k+i}^c(N) + e_{m_k+i}^{b(i+1)}(N)k_{m_k+i+1}^c(N - 1)$ |
| LET $I = \text{right_rotate}[1\ 2 \dots k]_{i+1}$ | LET $I = \text{right_rotate}[1\ 2 \dots k]_{i+1}$ |
| FOR $\mu = 0$ TO $(k - 1)$ DO | FOR $\mu = 0$ TO $(k - 1)$ DO |
| LET $v = I(\mu + 1)$ | LET $v = I(\mu + 1)$ |
| $e_{m_k+i+1}^{b(\mu+1)}(N) = z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N) + \varepsilon_{m_k+i}^{f(v)}(N)k_{m_k+i+1}^{b(\mu+1)}(N)$ | $e_{m_k+i+1}^{b(\mu+1)}(N) = z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N) + \varepsilon_{m_k+i}^{f(v)}(N)k_{m_k+i+1}^{b(\mu+1)}(N - 1)$ |
| $e_{m_k+i+1}^{f(v)}(N) = e_{m_k+i}^{f(v)}(N) + z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N)k_{m_k+i+1}^{f(v)}(N)$ | $e_{m_k+i+1}^{f(v)}(N) = e_{m_k+i}^{f(v)}(N) + z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N)k_{m_k+i+1}^{f(v)}(N - 1)$ |
| ENDFOR μ | ENDFOR μ |
| ENDFOR i | ENDFOR i |

backward predictors of Table 1 together with the lower and upper partitions of the increased order data vector (3) [5]. The lattice-ladder recursions for the a priori and the a posteriori errors are tabulated in Table 2.

Forward and backward error variables, $e^{f(v)}$ and $e^{b(\mu)}$ are coupled together in a similar way as the transversal filter counterparts of Table 1. 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 $[e^{f1}, e^{bk}]$, $[e^{f2}, e^{b1}]$, $[e^{f3}, e^{b2}]$, $[e^{fk}, e^{b(k-1)}]$. When ($i = 1$) prediction errors are coupled together as $[e^{f1}, e^{b(k-1)}]$, $[e^{f2}, e^{bk}]$, $[e^{f3}, e^{b1}]$, $[e^{fk}, e^{b(k-2)}]$. Finally ($i = k - 1$) the pairs are $[e^{f1}, e^{b1}]$, $[e^{f2}, e^{b2}]$, $[e^{f3}, e^{b3}]$, $[e^{fk}, e^{bk}]$. A similar scheme holds for the a posteriori error variables.

The lattice-ladder recursions of Table 2 can be transformed to an adaptive algorithm once reflection coefficients designated by the letter k are sequentially updated. It is shown in [5] that the reflection coefficients of order $m_k + i + 1$, $i = 0, 1, \dots, k - 1$, are computed as

$$k_{m_k+i+1}^c(N) = -\beta_{m_k+i+1}^c(N)/\alpha_{m_k+i}^{b(i+1)}(N), \quad (12)$$

$$k_{m_k+i+1}^{b(\mu+1)}(N) = -\beta_{m_k+i+1}^{b(\mu+1)}(N)/\alpha_{m_k+i}^{f(v)}(N), \quad (13)$$

$$k_{m_k+i+1}^{f(v)}(N) = -\beta_{m_k+i+1}^{b(\mu+1)}(N)/z_N^{h1} \alpha_{m_k+i}^{b(\mu+1)}(N), \quad (14)$$

where

$$\beta_{m_k+i+1}^c(N) = \lambda \beta_{m_k+i+1}^c(N) + e_{m_k+i}^c(N) \varepsilon_{m_k+i}^{b(i+1)}(N), \quad (15)$$

$$\begin{aligned} \beta_{m_k+i+1}^{b(\mu+1)}(N) \\ = \lambda \beta_{m_k+i+1}^{b(\mu+1)}(N - 1) + e_{m_k+i}^{f(v)}(N) (z_N^{h1} \varepsilon_{m_k+i}^{b(\mu+1)}(N)), \end{aligned} \quad (16)$$

and

$$\begin{aligned} \alpha_{m_k+i+1}^{f(v)}(N) \\ = \lambda \alpha_{m_k+i+1}^{f(v)}(N - 1) + \varepsilon_{m_k+i+1}^{f(v)}(N) e_{m_k+i+1}^{f(v)}(N), \end{aligned} \quad (17)$$

$$\begin{aligned} \alpha_{m_k+i+1}^{b(\mu+1)}(N) \\ = \lambda \alpha_{m_k+i+1}^{b(\mu+1)}(N - 1) + \varepsilon_{m_k+i+1}^{b(\mu+1)}(N) e_{m_k+i+1}^{b(\mu+1)}(N). \end{aligned} \quad (18)$$

Following [8] we seek to determine direct time updating recursions that sequentially estimate the reflection coefficients, Eqs. (12)–(14). A simple glance at Eqs. (12)–(18) reveals that the reflection coefficients can be obtained by the solution of a modified equation using the matrix inversion lemma as

$$(\lambda \alpha + \varepsilon e)k = -(\lambda \beta + \varepsilon \hat{e}), \quad (19)$$

$$k = -\beta/\alpha + (\hat{e} + e(-\beta/\alpha))k^w,$$

$$k^w = -\varepsilon/(\lambda \alpha + \varepsilon e). \quad (20)$$

Applying identity (20) to Eqs. (12)–(14) we get the lattice-ladder algorithm of Table 3. The basic processor units implementing the lattice and the ladder part of the propose adaptive algorithm are depicted in Fig. 1.

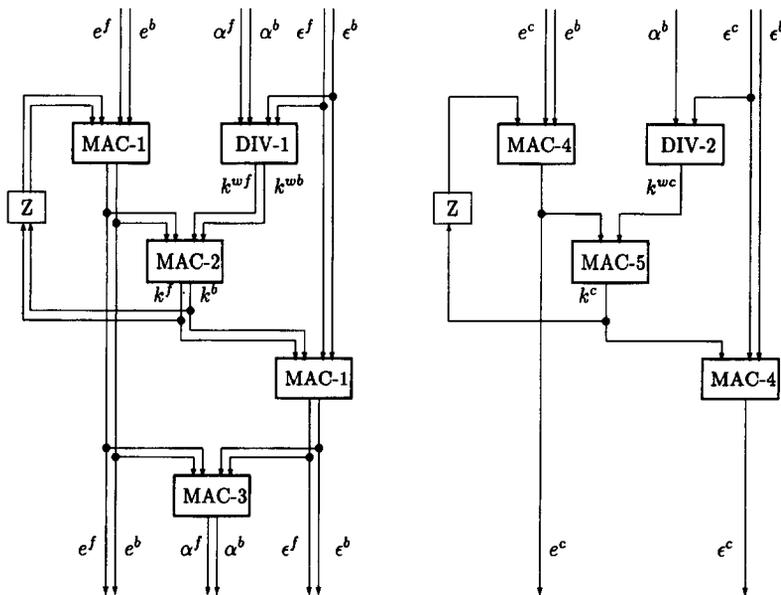


Fig. 1. The basic lattice and ladder cells utilized by the multichannel adaptive lattice algorithm. Two types of processor units are needed. The first, denoted by MAC, performs multiply and accumulate operations. The second, denoted by DIV, performs divisions. Operator Z represents a unit delay shift. A detailed description of each processor is given below.

| | |
|-------|--|
| MAC-1 | $e^f = e^f + e^b * k^f, e^b = e^b + e^f * k^b$ |
| MAC-2 | $k^f = k^f + e^f * k^{wf}, k^b = k^b + e^b * k^{wb}$ |
| MAC-3 | $\alpha^f = \lambda * \alpha^f + e^f * \epsilon^f, \alpha^b = \lambda * \alpha^b + e^b * \epsilon^b$ |
| MAC-4 | $e^c = e^c + e^b * k^c$ |
| MAC-5 | $k^c = k^c + e^b * k^{wc}$ |
| DIV-1 | $k^{wf} = -e^b / \alpha^b, k^{wb} = -e^f / \alpha^f$ |
| DIV-2 | $k^{wc} = -e^c / \alpha^b$ |

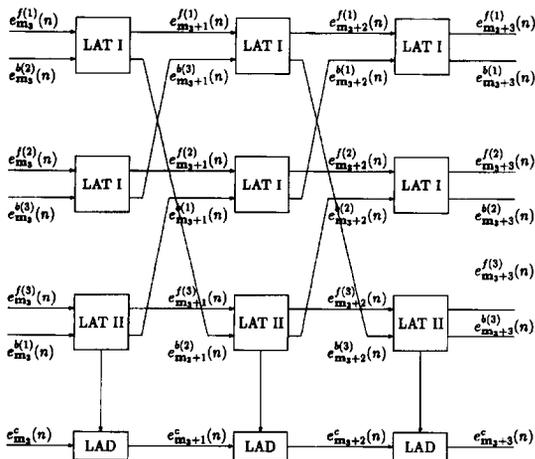


Fig. 2. The basic lattice/ladder cells of the multichannel lattice algorithm, for the special case $k = 3$.

3.1. Algorithm organization

The basic lattice stage of the proposed multichannel adaptive lattice algorithm updates the error variables from order m_k to the following $m_k + k$, through a sequence of k intermediate steps. The basic lattice stage and the corresponding signal flow graph illustrating the circulation of the a priori error variables are depicted in Fig. 2 (the remaining parameters are circulated in a similar way). It is built on $k^2 + k$ locally interconnected processing units that process single channel signals. Two types of lattice units are used, named LAT I and LAT II. The internal organization of LAT I is described in Fig. 1. LAT II is identical to LAT I, apart from a unit delay assigned to the backward errors and powers. In addition, a ladder unit, named LAD, is

Table 3
A posteriori/a priori adaptive lattice multichannel algorithm

FOR $i = 0$ TO $(k - 1)$ DO
 $e_{m_k+i+1}^c(N) = e_{m_k+i}^c(N) + e_{m_k+i}^{b(i+1)}(N)k_{m_k+i+1}^c(N - 1)$
 $k_{m_k+i}^{wc}(N) = -e_{m_k+i}^{b(i)}(N)/\alpha_{m_k+i}^{b(i)}(N)$
 $k_{m_k+i+1}^c(N) = k_{m_k+i+1}^c(N - 1) - e_{m_k+i+1}^c(N)k_{m_k+i}^{wc}(N)$
 $e_{m_k+i+1}^c(N) = e_{m_k+i}^c(N) + e_{m_k+i}^{b(i+1)}(N)k_{m_k+i+1}^c(N)$
LET $I = \text{right_rotate}[1\ 2 \dots k]_{i+1}$
FOR $\mu = 0$ TO $(k - 1)$ DO
LET $v = I(\mu + 1)$
 $e_{m_k+i+1}^{b(\mu+1)}(N) = z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N) + e_{m_k+i}^{f(v)}(N)k_{m_k+i+1}^{b(\mu+1)}(N - 1)$
 $e_{m_k+i+1}^{f(v)}(N) = e_{m_k+i}^{f(v)}(N) + z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N)k_{m_k+i+1}^{f(v)}(N - 1)$
 $k_{m_k+i+1}^{wf(v)}(N) = -e_{m_k+i}^{f(v)}(N)/\alpha_{m_k+i}^{f(v)}(N)$
 $k_{m_k+i}^{wb(\mu+1)}(N) = -(z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N))/\alpha_{m_k+i}^{b(\mu)}(N)$
 $k_{m_k+i+1}^{b(\mu+1)}(N) = k_{m_k+i+1}^{b(\mu+1)}(N - 1) + e_{m_k+i+1}^{b(\mu)}(N)k_{m_k+i+1}^{wf(v)}(N)$
 $k_{m_k+i+1}^{f(v)}(N) = k_{m_k+i+1}^{f(v)}(N - 1) + e_{m_k+i+1}^{f(v)}(N)k_{m_k+i+1}^{wb(\mu+1)}(N)$
 $e_{m_k+i+1}^{b(\mu+1)}(N) = z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N) + e_{m_k+i}^{f(v)}(N)k_{m_k+i+1}^{b(\mu+1)}(N)$
 $e_{m_k+i+1}^{f(v)}(N) = e_{m_k+i}^{f(v)}(N) + z_N^{h1} e_{m_k+i}^{b(\mu+1)}(N)k_{m_k+i+1}^{f(v)}(N)$
 $\alpha_{m_k+i}^{b(\mu+1)}(N) = \lambda \alpha_{m_k+i}^{b(\mu+1)}(N - 1) + e_{m_k+i}^{b(\mu+1)}(N)e_{m_k+i}^{b(\mu+1)}(N)$
 $\alpha_{m_k+i}^{f(v)}(N) = \lambda \alpha_{m_k+i}^{f(v)}(N - 1) + e_{m_k+i}^{f(v)}(N)e_{m_k+i}^{f(v)}(N)$
ENDFOR μ
ENDFOR i

utilized. It is depicted in Fig. 1. In the case of a multichannel filter having equal filter orders for all channels, i.e., $\mathbf{p}_k = [p\ p \dots p]$, p such devices must be interconnected. Then, the overall hardware required is $p(k^2 - k)$ LAT I units, pk LAT II units and pk LAD units.

When the multi index of the final filter $\mathbf{c}_{p_k} = [p_1\ p_2 \dots p_k]$, where $p_i \neq p_j$, in general, a different updating scheme has to be adopted. The order update procedure starting from $e_1^c(N)$ to the final $e_{p_k}^c(N)$ consists of k phases. During each phase s , error variables $e_{m_s}^c(N)$, $e_{m_s}^{b(i)}(N)$ and $e_{m_s}^{f(i)}(N)$, $i = 1, \dots, s$, are updated from stage m_s to m_{s+s} via s scalar iterations using the basic lattice cell. The remaining error variables $e_{m_s}^{f(i)}(N)$ and $e_{m_s}^{b(i)}(N)$, $j = s + 1, \dots, k$, are updated using the ladder type recursions

$$e_{m_s+i+1}^{f(j)}(N) = e_{m_s+i}^{f(j)}(N) + e_{m_s+i}^{b(i+1)}(N - 1)k_{m_s+i+1}^{fj}(N), \quad (21)$$

$$e_{m_s+i+1}^{b(i)}(N) = e_{m_s+i+1}^{f(i)}(N),$$

$$e_{m_s+i+1}^{f(j)}(N) = e_{m_s+i}^{f(j)}(N) + e_{m_s+i}^{b(i+1)}(N - 1)k_{m_s+i+1}^{fj}(N), \quad (22)$$

$$e_{m_s+i+1}^{b(i)}(N) = e_{m_s+i+1}^{f(i)}(N),$$

$$k_{m_s+i}^{wf(j)}(N) = -e_{m_s+i}^{b(i)}(N)/\alpha_{m_s+i}^{b(i)}(N - 1),$$

$$k_{m_s+i+1}^{f(j)}(N) = k_{m_s+i+1}^{f(j)}(N - 1) + e_{m_s+i+1}^{f(j)}(N)k_{m_s+i}^{wf(j)}(N). \quad (23)$$

Let $\mathcal{L}[s]$ represent a lattice cell of s -channels. Let $I^c[s]$ denote the ladder part that corresponds to the joint process and $I^{fj}[s]$, $j = s + 1, \dots, k$, denote the ladder parts that correspond to the forward predictors of order bigger than s . During phase s , the error variables s^c are updated using ladder $I^c[s]$. The forward and backward prediction errors e^{bj} and e^{fj} , $j = 1, \dots, s$, are updated using the lattice $\mathcal{L}[s]$ while prediction errors e^{fj} , $j = s + 1, \dots, k$, are updated using the ladders $I^{fj}[s]$, $j = s + 1, \dots, k$, respectively. The whole procedure consists of $\sum_{s=1}^k p_{s,s+1} = p_1$ steps. The lattice algorithm of Table 3 is depicted in Fig. 3, for a special case of a three channel problem, $k = 3$, and final mutli index $\mathbf{p} = [p_1\ p_2\ p_3] = [7\ 4\ 2]$.

In the special case when the channel lengths are equal, i.e., $p_i = p_j$, $\mathbf{p}_k = [p\ p \dots p]$, the last phase, $\mathcal{L}[k]$, $I^c[k]$, is only utilized.

The computational complexity of the proposed highly modular adaptive lattice LS algorithm of Table 3 is $O(kP)$ MADS for a k -channel filter of order $\mathbf{p}_k = [p_1\ p_2 \dots p_k]$, $P = \sum_{i=1}^k p_i$. When the filter coefficients are requested, additional $O(kP^2)$ MADS are needed as dictated by Table 1. The time and order recursions of Table 1 can be transformed to a static memoryless system once Eqs. (6) and (8) that compensate for the time shift in the backward prediction are taken into account. In the latter case, at each time, pure order recursions are performed for the computation of filter parameters.

4. Simulation

The performance of the proposed adaptive least squares lattice algorithm is illustrated by

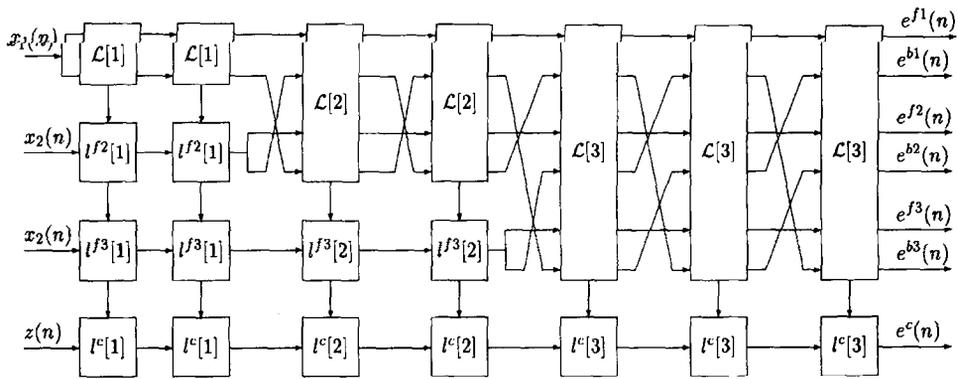


Fig. 3. The complete lattice structure for a three channel filter of order $p_3 = [7\ 4\ 2]$.

simulations. A three channel FIR filter of order $p_3 = [7\ 4\ 2]$ was used as a test system. To investigate the tracking capabilities of the algorithm, the coefficients of the FIR filter were changed in the middle of the observation interval. The input signals $x_1(n)$, $x_2(n)$ and $x_3(n)$ were generated by a three-channel AR model driven by independent white noise signals. A white noise signal of zero mean and unit variance is added at the output of the system, thus resulting to a signal to noise ratio (SNR) of 20 dB. Four different values of the forgetting factor λ have been investigated, i.e., $\lambda_1 = 0.99$, $\lambda_2 = 0.975$, $\lambda_3 = 0.95$ and $\lambda_4 = 0.9$. The condition

number of the autocorrelation matrix associated with Eq. (2) was within the range of $[200, 400]$, depending on the value of the forgetting factor λ .

The lattice algorithm of Table 3 was implemented in FORTRAN using single precision IEEE floating arithmetic (24-bits of mandissa). In addition, to investigate the effects of finite arithmetic, the experiments were repeated using reduced precision arithmetic. The performance of the algorithm was monitored by the ensemble averaged, squared, a priori prediction error, $e_{m_k}^c(N)$. The ensemble average was taken over 1000 independent realizations of the experiments.

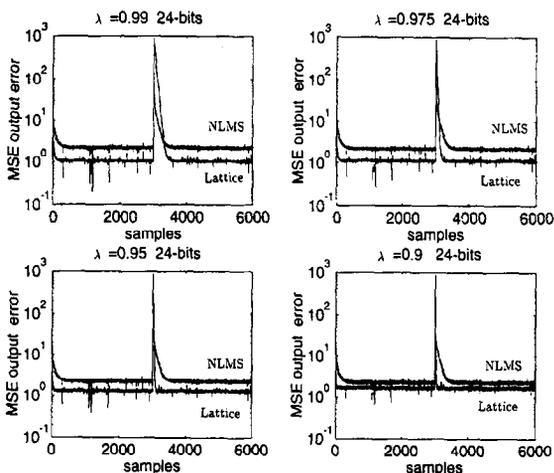


Fig. 4(a). A comparison between the fast lattice LS adaptive algorithm and the NLMS, when full precision (24 bits of mandissa) is used.

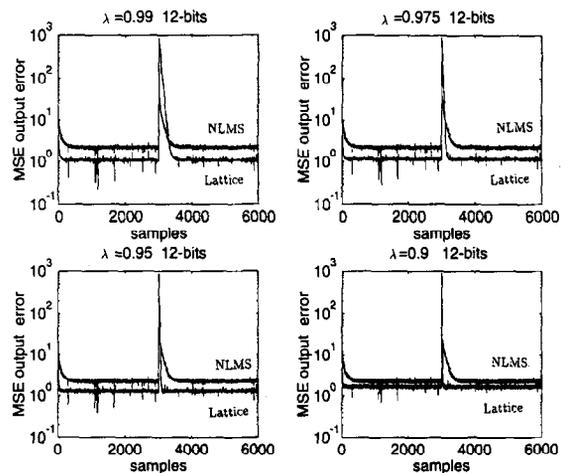


Fig. 4(b). A comparison between the fast lattice LS adaptive algorithm and the NLMS, when reduced precision (12 bits of mandissa) is used.

The proposed algorithm was tested against the multichannel fast RLS stabilized schemes [5], and the normalized LMS algorithm (read as the projection algorithm in [7]). The F-RLS and the proposed fast lattice algorithm were both performed better than the NLMS algorithm, with respect to the convergence speed, the tracking capabilities and the minimum error attained. In fact, when no divergence effect occurred to the fast RLS, both LS algorithm followed the same trajectory. However, the fast lattice scheme were noticed to be more robust compared the fast RLS counterpart, when reduced precision is used and when the forgetting factor takes values not very close to one. The proposed fast lattice LS algorithm performed satisfactory in all experiments, for all $\lambda \in \{0.9, 0.95, 0.975, 0.99\}$ and for mandissa length up to 8 bits. On the contrary, the ‘stabilized’ fast RLS was noticed to be sensitive in cases when the forgetting factor was not kept very close to one, i.e., $\lambda \in \{0.9, 0.95\}$ and when less than 16 bits of mandissa were utilized. In these cases, re-initialization of the F-RLS was very often necessary. The performance of the proposed fast lattice LS algorithm compared to the NLMS algorithm is illustrated in Figs. 4(a) and 4(b), for 24 bits and 12 bits of mandissa precision, respectively. The simulation results indicate that the proposed fast lattice LS adaptive algorithm has a good tracking capability and, in addition, has a nice numerical behavior in reduced arithmetic precision.

5. Conclusion

A highly efficient adaptive lattice algorithm for least squares multichannel FIR filtering has been developed. The proposed algorithm is explicitly update reflection coefficients, based on a priori and a posteriori error parametrization. It handles filters

with different number of delay elements per input channel. Scalar only operations (addition, multiplication and division) are utilized and communications is kept at a local level, allowing for full pipelining. The tracking capability, as well as the numerical stability and accuracy of the method, were experimentally demonstrated. FORTRAN code implementation of the algorithm can be provided by authors.

References

- [1] M.G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker, New York, 1997.
- [2] G. Carayannis, D. Manolakis and N. Kalouptsidis, “A unified view of parametric processing algorithms for prewindowed signals”, *Signal Processing*, Vol. 10, No. 4, June 1986, pp. 335–368.
- [3] B. Friedlander, “Lattice filters for adaptive processing”, *Proc. IEEE*, Vol. 70, August 1982, pp. 829–867.
- [4] G. Glentis and N. Kalouptsidis, “Efficient order recursive algorithms for multichannel least squares filtering”, *IEEE Trans. Signal Process.*, June 1992, pp. 1354–1374.
- [5] G. Glentis and N. Kalouptsidis, “Fast adaptive algorithms for multichannel filtering and system identification”, *IEEE Trans. Signal Process.*, October 1992, pp. 2433–2458.
- [6] G. Glentis and N. Kalouptsidis, “Efficient adaptive algorithms for multichannel least squares filtering using a channel decomposition technique”, *Computers and Electrical Engineering, Special Issue on Adaptive Signal Processing*, Vol. 2, May 1992.
- [7] G. Goodwin and K. Sin, *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
- [8] F. Ling, D. Manolakis and J. Proakis, “Numerically robust least squares lattice-ladder algorithm for direct updating of the reflection coefficients”, *IEEE Trans. Acoust. Speech Signal Process.*, August 1986, pp. 837–845.
- [9] F. Ling and J.G. Proakis, “A generalized multichannel least squares lattice algorithm based on sequential processing stages”, *IEEE Trans. Acoust. Speech Signal Process.*, Vol. ASSP-32, April 1984, pp. 381–389.
- [10] A. Pertropulu, C. Nikias and J. Proakis, “Efficient realization structure of the a priori lattice-ladder recursive least squares algorithm”, *IEEE Trans. Signal Process.*, April 1991, pp. 992–996.