DIXIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

LEAST SQUARES LATTICE ALGORITHMS WITH DIRECT UPDATING OF THE REFLECTION COEFFICIENTS +

Fuyun Ling*
Dimitris Manolakis**
John G. Proakis**

* Codex Corporation, Mansfield, MA 02048
**Department of Electrical & Computer Engineering, Northeastern University, 360 Huntington Ave., Boston, MA 02115
+ This work was supported by the National Science Foundation, under Grant ECS-8205596.

RESUME

Ce papier decrit une nouvelle approche au refraichissement temporel des parametres des filtres de moindre carres en treillis. Par rapport aux algorithmes d'utilisation courante, des formules explicites, recursives dans le domaine temporel, pour le rafraichissement des coefficients de reflexion et des gains du treillis sont developpees. Ces formules originales ameliorent d'une facon significative les proprietes numeriques (par example, la precision numerique et la stabilite numerique) des algorithmes de moindre carres pour les filtres en treillis. Des resultats analytiques et des simulations des proprietes de ces algorithmes sont presentes.

THE CONVENTIONAL A-POSTERIORI LS LATTICE-LADDER ALGORITHM		
	Time initialization	
1	$\alpha^f_m(M-1)=\alpha^b_m(M-1)=\sigma I, \sigma>0$	m=0,1, ...,p-1
2	$\beta_m(M-1)=0, \beta^z_m(M-1)=0, \beta^z_p(M-1)=0$	
3	$\varepsilon^b_m(M-1)=0$	
	Order initialization	MAD
4	$\varepsilon^f_o(n)=\varepsilon^b_o(n)=x(n)$, $\varepsilon_o(n)=z(n)$, $\alpha^*_o(n)=1$	0
5	$\alpha^f_o(n)=\alpha^b_o(n)=\lambda\alpha^f_o(n-1)+x(n)x^t(n)$	$\ell^2+\ell$
	LATTICE PART m=0,1,...p-2	MAD
6	$\beta_{m+1}(n)=\lambda\beta_{m+1}(n-1)+\varepsilon^b_m(n-1)\varepsilon^{ft}_m(n)/\alpha^*_m(n-1)$	$2\ell^2+\ell$
7	$k^f_{m+1}(n)=-\alpha^{-f}_m(n)\beta^t_{m+1}(n)$	$4\ell^2$
8	$k^b_{m+1}(n)=-\alpha^{-b}_m(n-1)\beta_{m+1}(n)$	
9	$\varepsilon^f_{m+1}(n)=\varepsilon^f_m(n)+k^{bt}_{m+1}(n)\varepsilon^b_m(n-1)$	
10	$\varepsilon^b_{m+1}(n)=\varepsilon^b_m(n-1)+k^{ft}_{m+1}(n)\varepsilon^f_m(n)$	
11	$\alpha^f_{m+1}(n)=\lambda\alpha^f_{m+1}(n-1)+\varepsilon^f_{m+1}(n)\varepsilon^{ft}_{m+1}(n)/\alpha^*_{m+1}(n-1)$	$O(\ell^2)$
12	$\alpha^*_{m+1}(n)=\alpha^*_m(n)-\varepsilon^{bt}_m(n)\alpha^{-b}_m(n)\varepsilon^b_m(n)$	$\ell^2+\ell$
13	$\alpha^b_{m+1}(n)=\lambda\alpha^b_{m+1}(n-1)+\varepsilon^b_{m+1}(n)\varepsilon^{bt}_{m+1}(n)/\alpha^*_{m+1}(n)$	$O(\ell^2)$
	LADDER PART m=0,1,...p-1	MAD
14	$\beta^z_{m+1}(n)=\lambda\beta^z_{m+1}(n-1)+\varepsilon^b_m(n)\varepsilon^t_m(n)/\alpha^*_m(n)$	$2q\ell+\ell$
15	$k_{m+1}(n)=-\alpha^{-b}_m(n)\beta^z_{m+1}(n)$	$\ell^2+q\ell$
16	$\varepsilon_{m+1}(n)=\varepsilon_m(n)+k^t_{m+1}(n)\varepsilon^b_m(n)$	

Table I. Computational organization and complexity of the LS a-posteriori lattice-ladder structure.

SUMMARY

This paper describes a new approach to the time-updating of the parameters of a least squares lattice-ladder filter. In contrast to conventional algorithms, time-recursive formulas for the explicit updating of the reflection coefficients and the ladder gains are developed. These new formulas dramatically improve the numerical properties, e.g., numerical accuracy and numerical stability, of the least squares lattice algorithms. Analytical and simulations results on the numerical properties of the new algorithms are presented.

THE CONVENTIONAL A-PRIORI LS LATTICE-LADDER ALGORITHM		
	Time initialization	
1	$\alpha^f_m(M-1)=\alpha^b_m(M-1)=\alpha^b_m(M-2)=\sigma I$, $\sigma>0$	m=0,1,..., p-1
2	$\beta_m(M-2)=0, \beta^z_m(M-2)=0, \beta^z_p(M-2)=0$	
3	$e^b_m(M-1)=0, e_m(M-1)=0, e_p(M-1)=0, \alpha^*_m(M-2)=1$	
	Order Initialization	MAD
4	$e^f_o(n)=e^b_o(n)=x(n), e_o(n)=z(n), \alpha^*_o(n-1)=1$	0
5	$\alpha^f_o(n)=\alpha^b_o(n)=\lambda\alpha^f_o(n-1)+x(n)x^t(n)$	$\ell^2+\ell$
	LATTICE PART m=0,1,...,p-2	MAD
6	$\beta_{m+1}(n-1)=\lambda\beta_{m+1}(n-2)+e^b_m(n-2)e^{ft}_m(n-1)\alpha^*_m(n-2)$	$2\ell^2+\ell$
7	$k^f_{m+1}(n-1)=-\alpha^{-f}_m(n-1)\beta^t_{m+1}(n-1)$	$4\ell^2$
8	$k^b_{m+1}(n-1)=-\alpha^{-b}_m(n-2)\beta_{m+1}(n-1)$	
9	$e^f_{m+1}(n)=e^f_m(n)+k^{bt}_{m+1}(n-1)e^b_m(n-1)$	
10	$e^b_{m+1}(n)=e^b_m(n-1)+k^{ft}_{m+1}(n-1)e^f_m(n)$	
11	$\alpha^f_{m+1}(n-1)=\lambda\alpha^f_{m+1}(n-2)+$ $e^f_{m+1}(n-1)e^{ft}_{m+1}(n-1)/\alpha^*_{m+1}(n-2)$	$O(\ell^2)$
12	$\alpha^*_{m+1}(n-1)=\alpha^*_m(n-1)-$ $e^{bt}_m(n-1)\alpha^{-b}_m(n-1)e^b_m(n-1)\alpha^{*2}_m(n-1)$	$\ell^2+\ell+2$
13	$\alpha^b_{m+1}(n-1)=\lambda\alpha^b_{m+1}(n-2)+$ $e^b_{m+1}(n-1)e^{bt}_{m+1}(n-1)\alpha^*_{m+1}(n-1)$	$O(\ell^2)$
	LADDER PART m=0,1,...,p-1	MAD
14	$\beta^z_{m+1}(n-1)=\lambda\beta^z_{m+1}(n-2)+e^b_m(n-1)e^t_m(n-1)\alpha^*_m(n-1)$	$2q\ell+\ell$
15	$k_{m+1}(n-1)=-\alpha^{-b}_m(n-1)\beta^z_{m+1}(n-1)$	$\ell^2+q\ell$
16	$e_{m+1}(n)=e_m(n)+k^t_{m+1}(n-1)e^b_m(n)$	

Table II. Computational organization and complexity of the a-priori lattice-ladder structure.

LEAST SQUARES LATTICE ALGORITHMS WITH DIRECT UPDATING OF THE REFLECTION COEFFICIENTS

1. INTRODUCTION

Lattice-ladder structures are extensively used in various signal processing applications. After the introduction of the classical two-multiplier lattice by Itakura and Saito, a number of adaptive lattice filters have been developed and applied in various fields [1].

Exact least-squares (LS) algorithms for all-zero modeling have been extensively studied in the literature [1-3]. In addition, much attention has been given to the case in which suboptimum ("non-exact" LS) gradient-type algorithms are used for the adaptation [4-5]. Recent research has been devoted to the investigation of the numerical properites of both types of algorithms in the case where finite precision arithmetic is used [6-8].

As is well known, lattice structures are described in terms of the so-called reflection coefficients and the ladder gains. Both parameters are recursively updated in time, in the case of adaptive filtering, in an effort to minimize the accumulated squared error.

In this paper we describe a new approach to the time updating of the parameters of a LS lattice-ladder filter. In contrast to the conventional algorithms, we have developed time-recursive formulas for the explicit updating of the lattice parameters, i.e., of the reflection coefficients and the ladder gains (direct updating). To distinguish the new structures from the conventional ones we use the term "modified" LS lattice algorithms.

The modified structures are developed for both a priori and a posteriori error-based LS algorithms. Finally, simulation results are given that support the superiority of the modified algorithms, as compared with the conventional ones.

2. THE CONVENTIONAL LS LATTICE-LADDER ALGORITHMS

We consider a multichannel FIR filter of order m with the following input-output relationship

$$y_m(n) = -\sum_{j=1}^{m} c_j^t(n)\, x(n+1-j) \triangleq -\underline{c}_m^t(n)\underline{x}_m(n) \tag{1}$$

where

$$\underline{c}_m(n) \triangleq [c_1^t(n)c_2^t(n) \;\ldots\; c_m^t(n)]^t \tag{2}$$

$$\underline{x}_m(n) \triangleq [x^t(n)x^t(n-1) \;\ldots\; x^t(n-m+1)]^t \tag{3}$$

The input signal $x(n)$ consists of ℓ-channels whereas the output $y_m(n)$ is a q-channel signal. As a consequence, the filter coefficients $\{c_j(n)\ ,\ j=1,2,\ldots,m\}$ are ℓxq matrices with scalar entries.

Suppose now that $z(n)$ is a q-channel desired response signal. The LS filter $\underline{c}_m(n)$ is obtained by minimizing the following total squared error

$$E_m(n) = \sum_{j=M}^{n} \lambda^{n-j}\, \varepsilon_m^t(n,j)\varepsilon_m(n,j) \tag{4}$$

where

$$\varepsilon_m(n,j) \triangleq z(j) + \underline{c}_m^t(n)\, \underline{x}_m(j) \tag{5}$$

is the estimation error at time j based on data up to the time n. In case j=n we will simplify the notation to $\varepsilon_m(n)$. The parameter λ , $0<\lambda<1$, is the well-known exponential forgetting factor. A value of λ close to but less than one makes it possible to realize time recursive LS filters that track slowly varying parameters [1], [9].

In this paper we will limit our discussion to the so-called prewindowed signal case. In this case, we assume that the input signal $x(n)=0$ for $n<M$. Obviously this is an arbitrary assumption, but it has the advantage of leading to simpler fast algorithms.

Minimization of $E_m(n)$ in (4) leads to the following set of normal equations

$$R_m(n)\underline{c}_m(n) = -\underline{r}_m(n) \tag{6}$$

where

$$R_m(n) = \sum_{j=M}^{n} \lambda^{n-j}\underline{x}_m(j)\underline{x}_m^t(j) \tag{7}$$

$$\underline{r}_m(n) = \sum_{j=M}^{n} \lambda^{n-j}\underline{x}_m(j)z^t(j) \tag{8}$$

The LS filter $\underline{c}_m(n)$ for a fixed time n can be obtained by solving (6) using fast order-recursive algorithms [1], [10], [11].

However, in many adaptive signal processing applications it is desirable to solve for $\underline{c}_m(n)$ recursively in time. This can be attained by using any of the conventional recursive LS (RLS) or square-root, or orthogonalizing algorithms [9], [12], [13] with a computational complexity of $O(m^2)$ per time update.

Besides these methods more efficient "fast RLS algorithms" are available which offer a computational complexity of O(m) per time updating [14], [13], [12]. We note that all these algorithms are recursive in time but the order m is fixed, say, to p.

In many cases of practical interest, as in adaptive equalization and echo cancellation, we do not need the LS filter $\underline{c}_m(n)$ but, instead, the error $\varepsilon_m(n)$ or the estimate $y_m(n)$ is desired. For such problems we can use the so-called LS lattice-ladder algorithms [1], [15]. These methods are recursive in order and time in the sense that all errors $\varepsilon_m(n)$, $m=1,2,\ldots,p$ are produced by the algorithm at each time instant.

As it is known, we can distinguish between two families of fast LS algorithms [13]. Algorithms in the first family primarily use the a-posteriori errors, of the form

$$\varepsilon_m(n) = z(n) + \underline{c}_m^t(n)\underline{x}_m(n) \tag{9}$$

In contrast, algorithms in the second family use a-priori errors of the form

$$e_m(n) = z(n) + \underline{c}_m^t(n-1)\underline{x}_m(n) \tag{10}$$

The difference between these two errors is that the a-priori error is computed using the optimum filter parameters at the previous time instant. The a-priori error based algorithms are best suited for applications such as adaptive equalization and adaptive noise canceling. We will use the Greek letter ε for a-posteriori errors and the English letter e for a-priori errors [13].

The LS lattice-ladder algorithms can be derived by using either a geometric approach [1], [2], [8] or a matrix approach [3], [12], [15]. Table I summarizes the a-posteriori LS lattice-ladder algorithm whereas Table II gives the a-priori one.

The algorithm in Table I uses the a-posteriori forward and backward prediction errors, defined by

$$\varepsilon_m^f(n) = x(n) + \underline{a}_m^t(n)\underline{x}_m(n-1) \tag{11}$$

$$\varepsilon_m^b(n) = x(n-m) + \underline{b}_m^t(n)\underline{x}_m(n) \tag{12}$$

where $\underline{a}_m(n)$ and $\underline{b}_m(n)$ are the LS forward and backward predictors at time n.

The algorithm in Table II is based on the a-priori forward and backward prediction errors, given by

LEAST SQUARES LATTICE ALGORITHMS WITH DIRECT UPDATING OF THE REFLECTION COEFFICIENTS

$$e_m^f(n) = x(n) + \underline{a}_m^t(n-1)\underline{x}_m(n-1) \tag{13}$$

$$e_m^b(n) = x(n-m) + \underline{b}_m^t(n-1)\underline{x}_m(n) \tag{14}$$

Furthermore, both algorithms, use the following variables.

Forward and backward variances

$$\alpha_m^f(n) = \sum_{j=M}^{n} \lambda^{n-j}\varepsilon_m^f(n,j)\varepsilon_m^{ft}(n,j) \tag{15}$$

and

$$\alpha_m^b(n) = \sum_{j=M}^{n} \lambda^{n-j}\varepsilon_m^b(n,j)\varepsilon_m^{bt}(n,j) \tag{16}$$

Partial correlation coefficients

$$\beta_m(n) = \sum_{j=M}^{n} \lambda^{n-j}\varepsilon_{m-1}^b(n-1,j-1)\varepsilon_{m-1}^{ft}(n,j) \tag{17}$$

and

$$\beta_m^z(n) = \sum_{j=M}^{n} \lambda^{n-j}\varepsilon_{m-1}^b(n,j)\varepsilon_{m-1}^t(n,j) \tag{18}$$

Forward and backward reflection coefficients

$$k_m^f(n) = -\alpha_{m-1}^{-f}(n)\beta_m^t(n) \tag{19}$$

$$k_m^b(n) = -\alpha_{m-1}^{-b}(n-1)\beta_m(n) \tag{20}$$

Ladder gain

$$k_m(n) = -\alpha_{m-1}^{-b}(n)\, b_m^z(n) \tag{21}$$

Note that we use the notation $\alpha_m^{-f}(n) \triangleq [\alpha_m^f(n)]^{-1}$

Angle Variable

$$\alpha_m^*(n) = 1 - \underline{x}_m^t(n)R_m^{-1}(n)\underline{x}_m(n) \tag{22}$$

It can also be shown that

$$0 \le \alpha_m^*(n) = \lambda^n \frac{\det R_m(n-1)}{\det R_m(n)} \le 1 \tag{23}$$

This last relation allows for the interpretation of $\alpha_m^*(n)$ as an angle variable. It is worth mentioning that this angle variable is directly related to the likelihood variable $\gamma_m(m) = 1-\alpha_m^*(n)$ and the optimum gain $\alpha_m(n) = 1/\alpha_m^*(n)$ [1], [15].

We conclude this section with some remarks regarding the computational complexity of the algorithms in Tables I and II. From the tables we see that both schemes require the inversion of the $\ell x \ell$ matrices $\alpha_m^f(n)$ and $\alpha_m^b(n)$ which requires a computational complexity of $O(\ell^3)$ per time update. However, these inversions can be avoided by updating directly the inverse matrices $\alpha_m^{-f}(n)$ and $\alpha_m^{-b}(n)$ or their square-root factors using rank one decompositions. Another approach based on the modified Gram-Schmidt algorithm is described in [19]. All these approaches reduce the total complexity of the LS lattice-ladder algorithms to $O(\ell^2 m)$ per time update.

3. THE MODIFIED LS LATTICE-LADDER ALGORITHMS

In this section we derive formulas for the direct updating of the reflection coefficients and the ladder gains. The derivations are mainly based on the following matrix identity which is a general form of the well-known matrix inversion lemma [9], [12].

$$(\lambda A+\underline{x}\underline{y}^t)^{-1} = \frac{1}{\lambda}A^{-1} - \frac{\frac{1}{\lambda}A^{-1}\underline{x}\;\frac{1}{\lambda}\underline{y}^tA^{-1}}{1+\frac{1}{\lambda}\underline{y}^tA^{-1}\underline{x}} \tag{24}$$

We will first derive the recursions for the modified a-posteriori LS lattice-ladder algorithm. For convenience we will refer to relations in Tables I or II by using the notation (I-) or (II-).

To obtain a direct updating formula for the forward reflection coefficient $k_m^f(n)$ we should combine time recursions for the quantities $\alpha_m^{-f}(n)$ and $\beta_m(n)$ (see eq. (19)). Since a time updating for $\beta_m(n)$ is already available (see eq. (I-6)) we should obtain an update for $\alpha_m^{-f}(n)$. To this end we first recall from [13] that

$$\alpha_m^{-f}(n)\varepsilon_m^f(n) = \frac{1}{\lambda}\alpha_m^{-f}(n-1)e_m^f(n)\alpha_{m+1}^*(n) \tag{25}$$

$$1+\frac{1}{\lambda}e_m^{ft}(n)\alpha_m^{-f}(n-1)\varepsilon_m^f(n) = \alpha_m^*(n-1)/\alpha_{m+1}^*(n) \tag{26}$$

$$1+\frac{1}{\lambda}e_m^{bt}(n)\alpha_m^{-b}(n-1)e_m^b(n)\alpha_m^*(n) = \alpha_m^*(n)/\alpha_{m+1}^*(n) \tag{27}$$

Substitution of (I-6) into (I-7) and applying (24) for (11), after some algebraic manipulations gives the desired formula

$$k_{m+1}^f(n) = k_{m+1}^f(n-1) - \frac{\alpha_m^{-f}(n)\varepsilon_m^f(n)}{\alpha_m^*(n-1)}\left[\varepsilon_m^{bt}(n-1)+\varepsilon_m^{ft}(n)k_{m+1}^f(n-1)\right] \tag{28}$$

For the single-channel case (28) is simplified into

$$k_{m+1}^f(n) = \frac{\alpha_{m+1}^*(n)}{\alpha_m^*(n-1)}\left\{k_{m+1}^f(n-1) - \frac{1}{\lambda}\alpha_m^{-f}(n-1)\varepsilon_m^f(n)\varepsilon_m^{bt}(n-1)/\alpha_m^*(n-1)\right\} \tag{29}$$

This recursion offers a direct time updating of the forward reflection coefficient.

To obtain the corresponding formula for the backward reflection coefficient we proceed in a similar manner. Indeed, by combining (I-13) with (24) and (I-12), we obtain

$$\lambda\alpha_m^{-b}(n)\varepsilon_m^b(n) = \alpha_m^{-b}(n-1)e_m^b(n)\alpha_{m+1}^*(n) \tag{30}$$

Substitution of (30) and (I-6) into (I-8) leads to the desired recursion

$$k_{m+1}^b(n) = k_{m+1}^b(n-1) - \frac{\alpha_m^{-b}(n-1)\varepsilon_m^b(n-1)}{\alpha_m^*(n-1)}\left[\varepsilon_m^{ft}(n)+\varepsilon_m^{bt}(n-1)k_{m+1}^b(n-1)\right] \tag{31}$$

To complete the algorithm we need a corresponding formula for the ladder gain $k_m(n)$. This can be easily obtained by combining (30), (I-14) and (I-15). Indeed, it is readily shown that

$$k_{m+1}(n) = k_{m+1}(n-1) - \frac{\alpha_m^{-b}(n)\varepsilon_m^b(n)}{\alpha_m^*(n)} \left[\varepsilon_m^t(n)+\varepsilon_m^{bt}(n)k_{m+1}(n-1) \right] \quad (32)$$

The modified a-posteriori LS lattice-ladder algorithm is obtained from Table I if we make the following changes.

- . Move (I-12) in place of (I-6) which is not needed any more.
- . Replace (I-7) by (28)
- . Replace (I-8) by (31)
- . Delete (I-14)
- . Replace (I-15) by (32).

This algorithm was first introduced in [16].

Next, we turn our attention to the derivation of the modified a-priori LS lattice-ladder algorithm.

We start by substituting eq. (II-6) into (II-7), which gives

$$k^f_{m+1}(n) = -\lambda\alpha_m^{-f}(n)\beta^t_{m+1}(n-1) - \alpha_m^{-f}(n)e_m^f(n)e_m^{bt}(n-1)\alpha_m^*(n-1) \quad (33)$$

To continue, we note that

$$-\lambda\alpha_m^{-f}(n)\beta^t_{m+1}(n-1) = \lambda\alpha_m^{-f}(n)\alpha_m(n-1)\alpha_m^{-f}(n-1)\beta^t_{m+1}(n-1)$$
$$= \lambda\alpha_m^{-f}(n)\alpha_m^f(n-1)k^f_{m+1}(n-1) \quad (34)$$

However (see (II-11), we

$$\alpha_m^f(n-1) = \frac{1}{\lambda}\{\alpha_m^f(n) - e_m^f(n)e_m^{ft}(n)\alpha_m^*(n-1)\}$$

Using this relation, (34) becomes

$$-\lambda\alpha_m^{-f}(n)\beta^t_{m+1}(n-1) = k^f_{m+1}(n-1) - \alpha_m^{-f}(n)\ e_m^f(n)e_m^{ft}(n)\alpha_m^*(n-1)k^f_{m+1}(n-1) \quad (35)$$

Substituting (35) into (33) and then using (II-10) gives the following direct updating recursion for the forward reflection coefficient

$$k^f_{m+1}(n) = k^f_{m+1}(n-1) - \alpha_m^{-f}(n)e_m^f(n)e_{m+1}^{bt}(n)\alpha_m^*(n-1) \quad (36)$$

To obtain the recursion for the backward reflection coefficient we proceed in a similar way. Indeed, we first use (I-8) and (I-6) to derive

$$k^b_{m+1}(n) = -\lambda\alpha_m^{-b}(n-1)\beta_{m+1}(n-1) - \alpha_m^{-b}(n-1)e_m^b(n-1)e_m^{ft}(n)\alpha_m^*(n-1) \quad (37)$$

Then using (II-13) we successively have

$$-\lambda\alpha_m^{-b}(n-1)\beta_{m+1}(n-1) = \lambda\alpha_m^{-b}(n-1)\alpha_m^b(n-2)k^b_{m+1}(n-1)$$
$$= k^b_{m+1}(n-1) - \alpha_m^{-b}(n-1)e_m^b(n-1)e_m^{bt}(n-1)\alpha_m^*(n-1)k^b_{m+1}(n-1)$$

Substituting the last relation into (37) and using (II-9) gives

$$k^b_{m+1}(n) = k^b_{m+1}(n-1)-\alpha_m^{-b}(n-1)e_m^b(n-1)e_m^{ft}(n)\alpha_m^*(n-1) \quad (38)$$

By proceeding in exactly the same way we can derive the following formulas for the direct updating of the ladder gain

$$k_m(n) = k_{m+1}(n-1)-\alpha_m^{-b}(n)e_m^b(n)e_{m+1}^t(n)\alpha_m^*(n) \quad (39)$$

The modified a-priori LS lattice-ladder algorithm is obtained from Table II if we make the following changes.

- . Delete (II-6) and (II-14)
- . Replace (II-7) by (36)
- . Replace (II-8) by (38)
- . Replace (II-15) by (39)

Obviously, before the replacement, the time shift $n \to n-1$ is required.

This algorithm was first introduced in [18].

4. NUMERICAL PROPERTIES AND PERFORMANCE COMPARISON

LS lattice algorithms using the new formulas given above to directly update their reflection coefficients have similar computational complexity as the conventional LS lattice algorithm. However, the modified LS lattice algorithm exhibits better numerical properties [18]. In this section we discuss the numerical behavior of the various LS lattice algorithms mentioned in this paper.

4.1. Comparison of the Numerical Accuracy of LS Lattice Algorithms

The numerical accuracy of the direct updating formula for the a priori errors modified LS lattice algorithm has been tested through computer simulations. Table III provides the output mean square error of the LS lattice algorithms using both the conventional form and the error-feedback form. Fixed point arithmetic with a word-length from 8 to 15 bits was used in the simulation. The improvement of the new formula is obvious from these results. This result agrees with the experiments for the gradient lattice algorithm [6]. In both cases, the modified LS or the gradient lattice algorithms, which provide better numerical accuracy, adopt an approach to directly update the reflection coefficients without first computing the crosscorrelation and autocorrelation of the estimation errors, as does the conventional LS lattice algorithm. Since the LS lattice algorithm using the error feedback formula performs exact LS estimation, it yields a faster initial convergence than the gradient lattice algorithm.

Because the parameters estimated in the LS lattice algorithm are the reflection coefficients and the joint estimation coefficients, the extra output error is mainly caused by the inaccurate estimation of these coefficients. The estimation error in the coefficients of previous stages will also propagate from stage to stage. In this paper we only consider the estimation error introduced in each stage. Below, we provide a simple analysis of the estimation error of the reflection coefficients in the LS lattice algorithms due to round-off error.

4.2. Estimation Error in Coefficients of the Conventional Lattice Algorithms

In the conventional form of the a posteriori LS lattice algorithm, given in Table I, the forward and backward reflection coefficients are computed according to (7) and (8), where the quantitites $\beta_{m+1}(t)$, $\alpha_m^f(n)$ and $\alpha_m^b(n-1)$ are computed by using (6), (11), and (13) respectively. For the a priori algorithm the situation is similar (see Table II).

We observe from (6), (11) and (13) that all three of these equations have the same form. In the steady state $\alpha_m^f(n)$ is a constant, which is its mean value, with a fluctuation around it. For λ close to one, $\alpha_m^*(n)$ is also close to one, and hence, we may ignore its effects. If infinite precision is used in the algorithm, the mean value of $\alpha_m^f(n)$ is equal to the mean value of $[\varepsilon_m^f(n)]^2$. The variance of the fluctuation can be computed by dividing the variance of $[\varepsilon_m^f(n)]^2$ by $(1-\lambda)$. If λ is close to 1, this variance is much smaller than the squared mean of $\alpha_m^f(n)$. The situation for $\beta_m(n)$ and $\alpha_m^b(n-1)$ is very similar. The fluctuation is the cause of the self-noise.

The round-off error affects both the mean values and the variances of $\alpha_m^f(n)$, $\alpha_m^b(n-1)$ and $\beta_m(n)$. By considering the round-off error, (11) can be rewritten as

$$\alpha_m^f(n) = \lambda\alpha_m^f(n-1)+[\varepsilon_m^f(n)]^2/\alpha_m^*(n-1)+w(n) \qquad (40)$$

where w(n) is the total round-off error. Since $\alpha_m^f(n)$ is $1/(1-\lambda)$ larger than the mean value of $[\varepsilon_m^f(n)]^2$, when λ is close to one, the magnitude of the first term on the left side of (40) is much larger than the magnitude of the second term. From numerical analysis we know that the total round-off error, w(n), in this case is determined mainly by addition and its peak value is approximately equal to $2^{-b}\alpha_m^f(n)$ for floating point arithmetic, and 2^{-b} for fixed point arithmetic where b is the word-length used. Assuming that the optimum scaling is used for fixed-point arithmetic, namely $\alpha_m^f(n)$ is close to 1, both cases yield similar relative errors. In practical cases, $[\varepsilon_m^f(n)]^2/\alpha_m^*(n-1)$ is always greater than w(n). Hence, we can ignore the contribution of the round-off error to the variance of $\alpha_m^f(n)$. A similar conclusion is obtained in [17] for the normalized LS lattice algorithm. The main effect of the round-off error to the estimation of $k_m^f(n)$ and $k_m^b(n)$ is to introduce a bias in the estimates. We investigate this effect below.

The effects of the round-off error on the bias of the computed values of $k_m^f(n)$ and $k_m^b(n)$ depend on the distribution of $\varepsilon_m^f(n)$ and $\varepsilon_m^b(n-1)$ as well as on the method of rounding. From the discussion given above, the dominant term is the error that occurs in addition. If we use b bits to represent $\alpha_m^f(n)$ or $\alpha_m^b(n)$, only about $b+\log_2(1-\lambda)$ bits are used to represent $[\varepsilon_m^f(n)]^2$ or $[\varepsilon_m^b(n)]^2$. In particular, when $[\varepsilon_m^f(n)]^2$ or $[\varepsilon_m^b(n)]^2$ have a magnitude less than the least significant bit (LSB) when using truncation (or half of the LSB when rounding is used), they will not contribute to the accumulation. In other words, there is a "dead zone" around zero which has a width equal to 2^{-b+1} (or 2^{-b}). This dead zone causes a loss in the computed results when a short word-length with a magnitude smaller than the magnitude computed using infinite precision. This loss could be quite significant depending on the distribution of $\varepsilon_m^f(n)$ and $\varepsilon_m^b(n)$. For example, assuming $\varepsilon_m^f(n)$ is Gaussian, $[\varepsilon_m^f(n)]^2$ will be chi-square with one degree of freedom. This density function becomes infinite at zero. In other words, the loss of the terms of $[\varepsilon_m^f(n)]^2$ and $[\varepsilon_m^b(n)]^2$ with a small magnitude which fall into the "dead zone", will cause a significant accumulated error. Even when rounding is used instead of truncation, the computed value of $\alpha_m^f(n)$, or $\alpha_m^b(n)$, will still be less than its true value in magnitude.

For $\beta_m(n)$, the effect is even more complicated. We provide a rough discussion below. First, the distribution of $\varepsilon_m^f(n)\varepsilon_m^b(n)$ must have some symmetry around its mean value. The effect of the dead zone is to remove the samples around zero and, hence, to move the estimated mean toward zero. As a result the magnitude of the estimated $b_m(n)$ is smaller than its true value computed using infinite precision. However, the amount of the shift depends on the distribution of $\varepsilon_m^f(n)$ and $\varepsilon_m^b(n)$, and it is difficult to analyze. Secondly, we know that $\beta_m(n)$ is an estimate of the correlation between $\varepsilon_m^f(n)$ and $\varepsilon_m^b(n)$, and usually $\varepsilon_m^f(n)$ and $\varepsilon_m^b(n)$ have the same variance. If the correlation coefficient is close to unity, the distribution of $\varepsilon_m^f(n)\varepsilon_m^b(n)$ will be close to the distribution of $[\varepsilon_m^f(m)]^2$ and $[\varepsilon_m^b(n)]^2$. As a consequence, the effects of the dead zone will be similar to the computed values of both $\beta_m(n)$ and $\alpha_m^b(n)$. Since the forward and the backward reflection coefficients are obtained by dividing $\beta_m(a)$ by $\alpha_m^f(n)$ and $\alpha_m^b(a)$, respectively, their computed values are close to the true values computed by using infinite precision. On the other hand, if the correlation coefficient is close to zero, we expect the distribution of $\varepsilon_m^f(n)\varepsilon_m^b(n)$ and the distribution of $[\varepsilon_m^f(n)]^2$ or $[\varepsilon_m^b(n)]^2$ to be different. The computed reflection coefficients in this case will be less accurate than in the former case.

We have designed a computer simulation of a simple scalar LS estimation to demonstrate the effects of the round-off errors discussed above. The data sequence, x(n), and the desired signal sequence, z(n), used in the simulation are jointly distributed Gaussian random scalar sequences with zero means and unit variance. The correlation between x(n) and z(n) are chosen to be 0.2, 0.5 and 0.9. The LS error of z(j) based on x(j), for j=0 to n, is computed according to (compared to (I.16) for m=0),

$$e(n,j) = z(j)-k(n)x(j) \quad (j=0\ldots n) \qquad (41)$$

where

$$b(n) = r_{zx}(n)/r_{xx}(n) \qquad (42)$$

In (42) $r_{zx}(n)$ and $r_{xx}(n)$ are defined as

$$r_{zx}(n)=\sum_{j=0}^{n} \lambda^{n-j}z(j)x(j) \ ; \ r_{xx}(n)=\sum_{j=0}^{n}\lambda^{n-j}x^2(j) \qquad (43)$$

$r_{zx}(n)$ and $r_{xx}(n)$ are computed recursively as

$$r_{zx}(n)=\lambda r_{zx}(n-1)+z(n)x(n) \ ; \ r_{xx}(n)=\lambda r_{xx}(n-1)+x^2(n) \qquad (44)$$

The coefficients, computed using fixed point arithmetic with a short word length, are given in Table IV. Truncation is used in the computations. As a comparison, the coefficients computed using floating point arithmetic with 22 bits precision are also given. This can be viewed as having infinite precision. From this table we notice that, in general, when the word-length is less than 9 bits, the accuracy of the estimated coefficients is unsatisfactory.

4.3 Error Analysis for the Error Feedback Formula

From the equations given in sections 2 and 3, we note that the new formulas for estimating the reflection coefficients are similar to the stochastic gradient algorithm if we view $\alpha_m^*(n)\alpha_m^{-b}(n)$ in (39) as a variable step size. Its robustness to round-off error is mainly due to this similarity. To understand this, we discuss how the stochastic gradient algorithm works.

First, we consider, as an example, the scalar LS estimation given by Eqs. (41) through (44). If we use the gradient algorithm in this case, the algorithm will be

$$\varepsilon(n)=z(n)-k(n-1)x(n) \ ; \ k(n)=k(n-1)+\Delta\varepsilon(n)x(n) \qquad (45)$$

From the orthogonality principle we know that the optimum coefficient is achieved when $\varepsilon(n)$ is orthogonal to x(n), or $E[\varepsilon(n)x(n)]=0$. From (45), we note that if at time n, k(n) is not equal to its optimum value, $\tilde{k}$, we can write

$$k(n) = \tilde{k}+\delta k \qquad (46)$$

and assume $\delta k>0$. After N iterations, if we think k(n) is a constant in this period, we have

$$k(n+N)=k(n)+\Delta\sum_{j=n+1}^{n+N} \varepsilon(j)x(j)$$

$$= k(n)+\Delta\sum_{j=n+1}^{n+N} [z(j)-\tilde{r}_x(j)-\Delta\sum_{j=n+1}^{n+N} x^2(j)\tilde{k} \qquad (47)$$

Since $\tilde{k}$ is the optimum coefficient, the first summation of (47) is approximately equal to zero and the second summation is positive. Hence the algorithm will reduce the value of k(n) and, hence, bring it towards its optimum value. If a dead zone exists, as long as the tails of the probability density of $\varepsilon(j)x(j)$ outside the dead zone are not symmetric, the algorithm will still bring the coefficient towards the right direction. We can conclude from the above discussion that the estimation accuracy of the gradient algorithm is based on the detection of the assymmetry of the density function of $\varepsilon(j)x(j)$ with respect to the origin, rather than the accuracy of computation.

The accuracy of the the coefficients estimated using the error feedback formula for the simple model given by (41) through (44) has been evaluated through computer simulations to compare with the results obtained previously. The results given in Table IV show that the error-feedback formula provides unbiased estimates even when a word-length of 7 bits is used.

In conclusion, we have presented two new formulas to directly update the coefficients of the LS lattice algorithm without the computation of the crosscorrelation and autocorrelation of the estimation errors. The numerical characteristics of the different versions of the LS lattice algorithms have been analyzed and supported by simulation results.

TABLE 3. Numerical Accuracy of LS Lattice Algorithms Using the Error-Feedback Formula

(Output MSE x 10^{-3})

Algorithm	Number of Bits 15	12	10	8	Floating Point
LSLA	2.18	3.09	25.2	365	2.10
LSLAEF	2.16	2.22	3.09	31.6	2.10

LSLA - Least Squares (LS) Lattice Algorithm
LSLAEF - LS Lattice Algor. (Error-Feedback Formula)

TABLE 4. The Effects of Finite Word-Length to the Accuracy of Estimation of Correlation Coefficients

(1) k=0.89878 (Floating Point 22 bits)

Number of bits used	r_{xy}	r_{yy}	$k=r_{xy}/r_{yy}$ (direct form)	$k=r_{xy}/r_{yy}$ (error-feedback)
15	0.9577	1.066	0.8989	0.8988
12	0.8844	0.9853	0.8984	0.8998
10	0.7912	0.8856	0.8945	0.9003
8	0.6371	0.7145	0.8932	0.8996
7	0.4705	0.5295	0.9017	0.9055

(2) k=0.52400 (Floating Point 22 bits)

Number of bits used	r_{xy}	r_{yy}	$k=r_{xy}/r_{yy}$ (direct form)	$k=r_{xy}/r_{yy}$ (error-feedback)
15	0.5425	1.0373	0.5234	0.5242
12	0.4906	0.9559	0.5134	0.5238
10	0.4297	0.8591	0.5000	0.5258
8	0.3150	0.6898	0.4548	0.5269
7	0.1368	0.5015	0.2480	0.9055

(3) k=0.22815 (Floating Point 22 bits)

Number of bits used	r_{xy}	r_{yy}	$k=r_{xy}/r_{yy}$ (direct form)	$k=r_{xy}/r_{yy}$ (error-feedback)
15	0.2261	0.9986	0.2268	0.2282
12	0.1882	0.9195	0.2048	0.2300
10	0.1476	0.8243	0.1786	0.2302
8	0.0829	0.6656	0.1232	0.2309
7	0.0134	0.4717	0.0270	0.2306

REFERENCES

1. B. Friedlander, "Lattice filters for adaptive processing", Proc. IEEE, Vol. 70, No. 8, Aug. 1982, pp. 829-867.
2. D. Lee, B. Friedlander, and M. Morf, "Recursive square-root ladder estimation algorithms", IEEE Trans. on ASSP, Vol. 29, No.3, June 1981, pp. 627-641.
3. F. Satorius and J. D. Pack, "Applications of LS adaptive lattice algorithms to adaptive equalization", IEEE Trans. on Comm. Vol. 29, Feb. 1981, pp. 136-142.
4. J. Makhoul and R. Viswanathan, "Adaptive lattice method for linear prediction", Proc. ICASSP-78, April 1978, pp. 83-86.
5. E. Satorius and S. Alexander, "Channel equalization using adaptive lattice algorithms", IEEE Trans. on Comm., Vol. 27, no. 6, June 1979, pp. 899-905.
6. E. Satorius, S. Lansch, S. Lee and D.O. Griffiths "Fixed point implementation of adaptive digital filters", Proc. ICASSP-83, Boston, MA, April 1983.
7. C. Samson and V. Reddy, "Fixed point error analysis of the normalized ladder algorithm", Proc. ICASSP-82 , Paris, France, May 1982.
8. F. Ling and J. Proakis, "Numerical accuracy and stability: two problems of adaptive estimation algorithms caused by round-off error", Proc. of ICASSP-84, San Diego, CA, March 1984.
9. L. Ljung and T. Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, 83.
10. L. Marple, "Efficient LS FIR system identification", IEEE Trans. on ASSP, Feb. 1981.
11. N. Kalouptsidis, G. Carayannis and D. Manolakis, "Efficient recursive-in-order LS FIR filtering and prediction", IEEE Trans. on ASSP, to appear.
12. J. G. Proakis, "Digital Communications", McGraw-Hill, 1983.
13. G. Carayannis, D. Manolakis and N. Kalouptsidis, "A Fast sequential algorithm for LS filtering and prediction", IEEE Trans. ASSP, Dec. 1983.
14. L. Ljung, M. Morf and D. Falconer, "Fast calculation of gain matrices for recursive estimation schemes", Int. J. of Control, Jan. 1978.
15. G. Carayannis, D. Manolakis and N. Kalouptsidis, "Efficient processing of prewindowed signals using direct and lattice-ladder structures", submitted for publication.
16. D. Manolakis, F. Ling, and J. G. Proakis, "Multichannel Lattice-Ladder Structures with applications to Pole-Zero Modeling", Proc. 1984 Int. Symp. on Circuits and Systems, Montreal, Canada, May 1984.
17. C. Samson and V.U. Reddy, "Fixed-Point Error Analysis of the Normalized Ladder Algorithm", IEEE Trans. on ASSP Vol. ASSP-31, No. 5, pp. 1177-1191, Oct. 1983.
18. F. Ling, "Rapidly Convergent Adaptive Filtering Algorithms with Applications to Adaptive equalization and Channel Estimation", Ph.D. Thesis, Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA, Sept. 1984.