Echo Cancellation of Voiceband Data Signals Using Recursive Least Squares and Stochastic Gradient Algorithms

MICHAEL L. HONIG, MEMBER, IEEE

Abstract—**The convergence properties of adaptive least squares (LS) and stochastic gradient (SG) algorithms are studied in the context of echo cancellation of voiceband data signals. The algorithms considered are the SG transversal, SG lattice, LS transversal (fast Kalman), and LS lattice. It is shown that for the channel estimation problem considered here, LS algorithms converge in approximately $2N$ iterations where N is the order of the filter. In contrast, both SG algorithms display inferior convergence properties due to their reliance upon statistical averages. Simulations are presented to verify this result, and indicate that the fast Kalman algorithm frequently displays numerical instability which can be circumvented by using the lattice structure. Finally, the equivalence between an LS algorithm and a fast converging modified SG algorithm which uses a maximum length input data sequence is shown.**

I. INTRODUCTION

THE application of recursive least squares (LS) algorithms to speech modeling, channel equalization, and echo cancellation of speech signals has been reported respectively in [1]-[3]. In all cases, improved performance (i.e., speed of convergence) has been observed when compared to more conventional recursive least mean square or stochastic gradient (SG) techniques. Here we examine the performance of recursive LS algorithms in the context of echo cancellation of voiceband data signals. This case deserves special consideration, since the input to the canceller is significantly different from the corresponding inputs used in the previous applications mentioned.

The echo cancellation problem considered here is illustrated in Fig. 1. A detailed exposition of this problem in a practical communication system is given in [4]. Echoes in the received signal are caused by impedance mismatches in the hybrid couplers which interface the two-wire and four-wire circuits. Focusing on the far left receiver, transmitted energy from the far left transmitter can leak through both the far left hybrid ("near-end" echo) and through the hybrid at the far end of the four-wire circuit ("far-end" echo). The purpose of the echo canceller shown in Fig. 1(b) is to compensate for this distortion by synthesizing a replica of the system impulse response as seen between the far left transmitter and receiver. If this synthesis is exact, then the inputs to the summer will be equal, and the residual echo at point A will be completely nulled. In practice, however, the echo canceller is typically a linear finite impulse response (FIR) filter, so that some residual echo at point A will remain uncancelled. Because the impulse response to be synthesized is initially unknown and may vary with time, the canceller must be an adaptive filter.

For purposes of this study, we need only concentrate on Fig. 2 which shows the data samples $\{a_i\}$ as inputs to a noisy channel and the canceller. If the impulse response of the canceller is the same as that of the channel, then the output error e_i will equal the noise sample η_i. Echo cancellers have typically used the SG transversal algorithm [4] to adjust the coefficients so as to minimize the output mean squared error (MSE) $E[e_i^2]$. Notice that, assuming the input and noise are uncorrelated, the minimum MSE achievable is the noise variance $E[\eta_i^2]$.

The distinguishing features of this application from the applications mentioned at the beginning of this section are 1) the number of coefficients in the adaptive filter must be quite large (i.e., greater than 100) in order to cancel the far-end echo, and 2) the input to the adaptive filter is in this case binary uncorrelated data. The performance degradation of the SG transversal algorithm due to a highly correlated input (i.e., large "eigenvalue spread" [5]) is therefore eliminated. One might suspect that under these ideal conditions, recursive LS algorithms offer no significant improvement over the SG transversal algorithm. It is shown that this is not the case. Assuming a noiseless channel, an LS algorithm can perfectly estimate N impulse response values given only N data samples, whereas the SG algorithm requires approximately $5N$ to $7N$ data samples to converge, depending on the step size. When added noise is considered, an LS algorithm requires approximately $2N$ data samples in order to achieve an output MSE which is 3 dB above the channel noise variance, whereas the

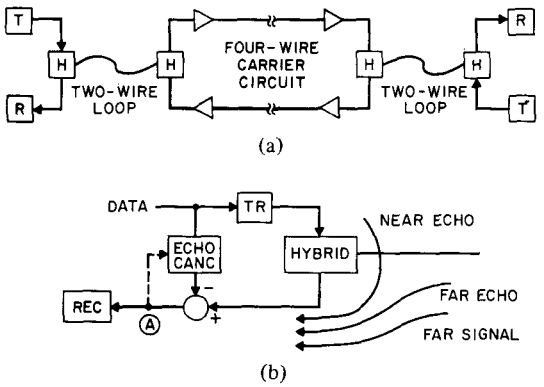


Fig. 1. (a) Typical dialed connection. (b) Use of data-driven echo canceller at station location. (Copied from [4].)

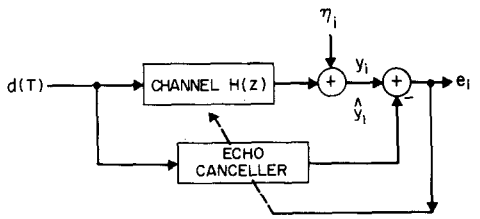


Fig. 2. Illustration of the channel estimation problem.

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication without oral presentation. Manuscript received September 30, 1983; revised April 10, 1984.

The author is with Bell Communications Research, Inc., Morristown, NJ 07960.

0090-6778/85/0100-0065$01.00

SG algorithm requires approximately $7N$ to $9N$ data samples to achieve the same level of cancellation.

In the next section, the performances of LS and SG algorithms are compared analytically. In Section III, it is shown that by choosing the input sequence in an appropriate fashion, the computational complexity of an LS algorithm can be greatly reduced. Using a maximum length input sequence in fact leads to a slightly modified SG transversal algorithm. This algorithm was proposed by Farrow and Salz and is also discussed in [6]. Section IV describes the algorithms considered in this paper and Section V presents simulation results which verify the discussion in preceding sections.

II. LS Versus SG Comparison

In this section the performances of LS and SG algorithms are compared with respect to speed of convergence. The input to the canceller is assumed to be a sequence of independent data samples $\{a_j\}$, each of which assumes a value of $+1$ or -1 with probability $1/2$. The output of the channel in Fig. 2 at iteration i is denoted as

$$y_i = \sum_{j=0}^{i} a_j r_{i-j} + \eta_i \tag{2.1}$$

where a_i is the most recent input, r_j, $0 \leqslant j \leqslant i$, are the first $i+1$ samples of the channel impulse response, and η_i is additive channel noise. For convenience, we assume that

$$r_j = 0 \qquad \text{for } j > N-1 \tag{2.2}$$

where N is some prespecified integer. Defining the N-dimensional vectors

$$r^T = [r_0 r_1 \cdots r_{N-1}] \tag{2.3}$$

and

$$a_{i|N}{}^T = [a_i a_{i-1} \cdots a_{i-N+1}], \tag{2.4}$$

(2.1) can be rewritten as

$$y_i = r^T a_{i|N} + \eta_i. \tag{2.5}$$

The objective of the canceller is to estimate the vector r. Denoting the estimate of r at time i as c_i, the output error is

$$e_i = y_i - c_i{}^T a_{i|N}. \tag{2.6}$$

Assuming the noise samples η_i are uncorrelated with the data samples, the value of c_i which minimizes the output MSE $E[e_i^2]$ is

$$c_{\text{opt}} = [E(a_{i|N} a_{i|N}{}^T)]^{-1} E(y_i a_{i|N}) = r \tag{2.7}$$

and

$$\min E[e_i^2] = E[\eta_i^2] \equiv \sigma^2. \tag{2.8}$$

The LS estimate of r at time i selects c_i such that the sum

$$S \equiv \sum_{j=0}^{i} w_j e_j^2 \tag{2.9}$$

is minimized, where the coefficients w_j are introduced to weight recent errors more than past errors. This technique allows nonstationary channel statistics to be tracked. Typically, the errors are weighted exponentially, i.e.,

$$w_j = w^{i-j} \tag{2.10}$$

where w is a constant slightly less than unity. Setting the derivative of (2.9) with respect to c_i equal to zero gives

$$c_i = \left(\sum_{j=0}^{i} w_j a_{j|N} a_{j|N}{}^T \right)^{-1} \left(\sum_{j=0}^{i} w_j y_j a_{j|N} \right). \tag{2.11}$$

Notice that if the first data sample is received at time $i = 0$, then some type of assumption must be made concerning the data samples a_j, $j < 0$, which enter the vectors $a_{0|N}$, $a_{1|N}$, $\cdots$, $a_{N-2|N}$. "Prewindowed" LS estimation [1] assumes that

$$a_j = 0, \qquad j < 0. \tag{2.12}$$

The estimate of c_i, (2.11), assuming that the (nonzero) data values $a_{-1}, a_{-2}, \cdots, a_{-N+1}$ are available at the receiver at iteration $i = 0$ will be called the "covariance LS estimate" of c_i, in analogy with the covariance LS estimation used in speech processing [1]. Both "prewindowed" and "covariance" estimates are nearly equal, given a moderate number of data samples (i.e., $2N$ to $3N$). However, the startup procedures to obtain each type of estimate recursively will be different. It is apparent that as $i \to \infty$, the estimate (2.11) approaches c_{opt} in (2.7).

The LS estimate (2.11) can be written recursively as follows:

$$c_i = c_{i-1} + e_i' \Phi_{i|N}{}^{-1} a_{i|N} \tag{2.13}$$

where e_i' is the "causal" prediction error

$$e_i' \equiv y_i - c_{i-1}{}^T a_{i|N} \tag{2.14}$$

and the sample covariance matrix is

$$\Phi_{i|N} \equiv \sum_{j=0}^{i} w_j a_{j|N} a_{j|N}{}^T. \tag{2.15}$$

Notice that the error e_i' is to be distinguished from the error e_i defined by (2.6). Computationally efficient recursive LS algorithms have received much attention in the literature and will be described in Section IV.

Notice that the covariance LS estimate of c_i at iteration i requires N more data samples than the prewindowed estimate. The time it takes an LS algorithm to converge will always be referred to in terms of the *number of input data samples received*. If it is stated, for example, that the output MSE has "converged" (i.e., is within 3 dB of its asymptotic value) at time $i = N$, this means that if prewindowed LS estimation is assumed, the algorithm has converged after N iterations, whereas if covariance LS estimation is assumed, the algorithm has converged after $2N$ iterations (N iterations are required to receive the first N data samples).

It will be instructive to examine the prewindowed LS estimate c given m data samples where $m \leqslant N$. Consider first the case where the additive noise $\eta_i = 0$. Notice that for $m < N$ the matrix $\Phi_{m|N}$, which appears in (2.11), is singular; however, an LS solution for the first m values of c is obtained by solving the following triangular set of linear equations:

$$\begin{vmatrix} a_0 & 0 & 0 & \cdots & 0 \\ a_1 & a_0 & 0 & & \vdots \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & & 0 \\ a_{m-1} & a_{m-2} & a_{m-3} & \cdots & a_0 \end{vmatrix} \begin{vmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_m \end{vmatrix} = \begin{vmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{m-1} \end{vmatrix}. \tag{2.16}$$

In this case the error sequence given by (2.6) is zero for $i = 0, 1, \cdots, m-1$, and hence, the sum (2.9) assumes the minimum value of zero. When a new data sample a_m is received, the same system of equations (2.16) is solved where m is replaced by $m+1$. Notice that the first m coefficients remain the same, so that only c_{m+1} must be determined. This procedure is continued until N data samples are received, in which case the estimated vector $\boldsymbol{c}_{N-1} = \boldsymbol{r}$.

If the additive noise is nonzero, the system of equations (2.16) still yields the LS estimate for $c_1, \cdots, c_m$; however, after N iterations the estimate $\boldsymbol{c}_{N-1}$ may deviate substantially from $\boldsymbol{r}$. Examination of (2.16) reveals that the deleterious effect of the added noise is cumulative, and that for moderately large N, the last few estimated impulse response values become dominated by channel noise. The effect of additive noise upon the estimate $\boldsymbol{c}$ was found empirically to be so devastating that computer roundoff error alone can cause successive estimated impulse response coefficients to become extremely large for moderate values of N (i.e., $N = 30$), depending upon the input data sequence.

Notice, however, that if $N+1$, rather than N, data samples are used to obtain an LS estimate for N impulse response values, the sequence of errors e_i, $i = 0, 1, \cdots, N$ (with additive channel noise) cannot in general be set to zero, and the triangular system of equations (2.16) no longer applies. The expected value of the sum (2.9), where $w_j = 1$, cannot be less than the sum of the noise variances,

$$E\left[\sum_{i=0}^{N} e_i^2\right] \geq (N+1)\sigma^2, \tag{2.17}$$

and the effect of the channel noise upon the estimate of the latter impulse response values is not nearly as severe as when given N data samples.

This startup problem has typically been avoided by initializing the diagonal elements of the matrix $\Phi_{i|n}$, given by (2.15), to some small value δ such that the algorithm (2.13) is numerically stable. Although this technique destroys the exactness of the LS solution, results in Section V and in [2] and [3] indicate that when this type of initialization procedure is used with the LS lattice algorithm, the degradation in performance is not noticeable. Use of this initialization procedure with the LS transversal (fast Kalman) algorithm [7], however, did not prevent the algorithm from diverging.

To compute the output MSE resulting from the LS estimate (2.11), (2.5) and (2.11) can be combined to give

$$\begin{aligned}\boldsymbol{c}_i &= \Phi_{i|N}^{-1}\left[\sum_{j=0}^{i} w_j \boldsymbol{a}_{j|N}(\boldsymbol{a}_{j|N}^T \boldsymbol{r} + \eta_j)\right] \\ &= \Phi_{i|N}^{-1}\left[\left(\sum_{j=0}^{i} w_j \boldsymbol{a}_{j|N}\boldsymbol{a}_{j|N}^T\right)\boldsymbol{r} + \sum_{j=0}^{i} w_j \eta_j \boldsymbol{a}_{j|N}\right] \\ &= \boldsymbol{r} + \Phi_{i|N}^{-1}\left(\sum_{j=0}^{i} w_j \eta_j \boldsymbol{a}_{j|N}\right).\end{aligned} \tag{2.18}$$

The covariance of the estimate conditioned on the data sequence is

$$\begin{aligned}&E[(\boldsymbol{c}_i - \boldsymbol{r})(\boldsymbol{c}_i - \boldsymbol{r})^T \mid a_0, a_1, \cdots, a_i] \\ &= \Phi_{i|N}^{-1}\left[\sum_{j=0}^{i}\sum_{l=0}^{i} w_j w_l E(\eta_j \eta_l)\boldsymbol{a}_{j|N}\boldsymbol{a}_{l|N}^T\right]\Phi_{i|N}^{-1} \\ &= \sigma^2 \Phi_{i|N}^{-1}\left[\sum_{j=0}^{i} w_j^2 \boldsymbol{a}_{j|N}\boldsymbol{a}_{j|N}^T\right]\Phi_{i|N}^{-1}.\end{aligned} \tag{2.19}$$

Setting $w_j = 1$ for $0 \leq j \leq i$ gives

$$E[(\boldsymbol{c}_i - \boldsymbol{r})(\boldsymbol{c}_i - \boldsymbol{r})^T \mid a_0, \cdots, a_i] = \sigma^2 \Phi_{i|N}^{-1}. \tag{2.20}$$

The output causal MSE conditioned on the input sequence in this case is therefore

$$\begin{aligned}&E[e_i'^2 \mid a_0, \cdots, a_{N-1}] \\ &= E\{[\boldsymbol{a}_{i|N}^T(\boldsymbol{r} - \boldsymbol{c}_i) + \eta_i]^2 \mid a_0, \cdots, a_{i-1}\} \\ &= \boldsymbol{a}_{i|N}^T\{E[(\boldsymbol{c}_i - \boldsymbol{r})(\boldsymbol{c}_i - \boldsymbol{r})^T \mid a_0, \cdots, a_{N-1}]\}\boldsymbol{a}_{i|N} + \sigma^2 \\ &= \sigma^2[\boldsymbol{a}_{i|N}^T \Phi_{i|N}^{-1} \boldsymbol{a}_{i|N} + 1].\end{aligned} \tag{2.21}$$

The unconditional output MSE is the average of (2.21) over all possible (2^{i+1}) input sequences. Unfortunately, this average is extremely difficult to compute; however, it is shown in the Appendix that, assuming exponential weighting and covariance estimation, as a first-order approximation,

$$E[e_i'^2] \approx \sigma^2\left[1 + N\frac{1-w}{1+w}\,\frac{1+w^{i+1}}{1-w^{i+1}}\right] \tag{2.22}$$

for large i, where w is the exponential weighting constant. As $i \to \infty$,

$$E[e_i'^2] \approx \sigma^2\left(1 + N\frac{1-w}{1+w}\right). \tag{2.23}$$

The exponential weighting therefore decreases the asymptotic level of cancellation. A similar result was presented in [8] and [9]. As $w \to 1$, (2.22) becomes

$$E[e_i'^2] \approx \sigma^2\left(1 + \frac{N}{i+1}\right). \tag{2.24}$$

This behavior was first pointed out in [10]. Notice that for $i = N-1$,

$$E[e_{N-1}'^2] \approx 2\sigma^2 \tag{2.25}$$

indicating that at time $i = N-1$, or after $2N$ iterations, (i.e., $2N$ data samples have been received), the output MSE is 3 dB above the channel noise variance. Although these results have been derived assuming covariance estimation, the corresponding prewindowed behavior should not be significantly different. This is verified via simulation in Section V.

It is instructive to compare the behavior of the LS algorithm (2.13) with that of the SG transversal algorithm [5]. A comparison with the SG lattice algorithm is postponed to Section V. The SG transversal algorithm is given by

$$\boldsymbol{c}_i = \boldsymbol{c}_{i-1} + \alpha e_i' \boldsymbol{a}_{i|N} \tag{2.26}$$

where α is the step size, which controls the size of the adjustment at each iteration. The convergence properties of this algorithm were studied in [4]. It was found that for the baseband canceller considered here,

$$\begin{aligned}E[e_i'^2] &\approx (1 - 2\alpha + \alpha^2 N)^i E[e_0^2] \\ &\quad + \frac{1 - (1 - 2\alpha + \alpha^2 N)^i}{\alpha(2 - \alpha N)}\,2\alpha\sigma^2\end{aligned} \tag{2.27}$$

where $E[e_0^2] = E[y_0^2]$, assuming the elements of $\boldsymbol{c}_0$ are

zero. Letting $\alpha = 1/N$, the value computed in [4] which gives the fastest convergence to the steady state MSE, (2.27) becomes

$$E[e_i'^2] = \left(1 - \frac{1}{N}\right)^i (E[e_0^2] - 2\sigma^2) + 2\sigma^2. \tag{2.28}$$

For this value of α, the output MSE only asymptotically approaches $2\sigma^2$, the value approximately obtained using an LS algorithm after $2N$ iterations. If at iteration i we have that $E[e_i'^2] \gg 2\sigma^2$, then (2.28) can be accurately approximated as

$$E[e_i'^2] \approx \left(1 - \frac{1}{N}\right)^i E[e_0^2] \tag{2.29}$$

which says that $E[e_i'^2]$ is $10i \log [1 - (1/N)]$ dB below $E[e_0^2]$. As an example, if $N = 100$ and if we assume the signal-to-noise ratio is 30 dB, after 200 iterations the output MSE is 21.3 dB above the noise level. After 500 iterations, the output MSE is 8.1 dB above the noise level. The discussion in this section indicates that an LS algorithm converges approximately 4–5 times faster than the SG transversal algorithm.

III. Selection of Startup Sequence

An important design parameter offered by the application considered here is the binary startup sequence, which is the initial input to the canceller. This is in contrast to other applications such as channel equalization, where the input to the equalizer is the output of an unknown channel. By selecting the startup sequence in an appropriate fashion, the complexity of the matrix inversion needed to obtain the LS estimate (2.11) can be dramatically reduced.

Assume, for example, that the data vectors $\boldsymbol{a}_{0|N}$, $\boldsymbol{a}_{i|N}$, $\cdots$, $\boldsymbol{a}_{N-1|N}$ are rotated versions of an N-term maximum length sequence [11], and therefore have the following properties:

$$\boldsymbol{a}_{j|N}{}^T \boldsymbol{a}_{i|N} = -1, \qquad j \neq i \tag{3.1a}$$

$$\mathbf{1}^T \boldsymbol{a}_{j|N} = 1 \tag{3.1b}$$

where $\mathbf{1}$ is the N-dimensional vector whose elements are unity. (Since $\boldsymbol{a}_{0|N}$ is assumed to contain no zero elements, covariance estimation is assumed.) The covariance matrix at the Nth iteration is

$$\Phi_{N-1|N} = (N+1)\boldsymbol{I} - \mathbf{1}\mathbf{1}^T. \tag{3.2}$$

It is easily verified that

$$\Phi_{N-1|N}^{-1} = \frac{1}{N+1} = (\boldsymbol{I} + \mathbf{1}\mathbf{1}^T). \tag{3.3}$$

Substituting (3.3) into (2.11), assuming $w_j = 1$, and using (3.1) gives

$$\boldsymbol{c}_{N-1} = \frac{1}{N+1}(\boldsymbol{I} + \mathbf{1}\mathbf{1}^T)\left(\sum_{j=0}^{N-1} y_j \boldsymbol{a}_{j|N}\right)$$

$$= \frac{1}{N+1}\sum_{j=0}^{N-1} y_j(\boldsymbol{a}_{j|N} + \mathbf{1}) \tag{3.4}$$

which can be written recursively as

$$\boldsymbol{c}_i = \boldsymbol{c}_{i-1} + \frac{1}{N+1} y_i(\boldsymbol{a}_{i|N} + \mathbf{1}) \qquad \text{for } i \leq N. \tag{3.5}$$

It is easily verified that replacing y_i by e_i' also yields (3.4) at iteration $i = N - 1$.

If a maximum length sequence with period N is the input to the canceller, the following algorithm therefore gives an LS solution for the coefficient vector after N iterations:

$$\boldsymbol{c}_i = \boldsymbol{c}_{i-1} + \frac{1}{N+1} e_i'(\boldsymbol{a}_{i|N} + \mathbf{1}). \tag{3.6}$$

This algorithm was originally presented by Farrow as a desirable modification of the SG algorithm (2.26) [12]. It is pointed out in [13], however, that *any* startup sequence can be assumed as long as the covariance matrix $\Phi_{i|N}$ in (2.13) is nonsingular for $i = 0, 1, \cdots, N-1$. (The singularity problem can always be avoided by substituting $\Phi_{i|N} + \delta \boldsymbol{I}$ for $\Phi_{i|N}$ where δ is a small constant.) Since the training vectors $\boldsymbol{a}_{0|N}$, $\boldsymbol{a}_{1|N}$, $\cdots$, $\boldsymbol{a}_{N-1|N}$ are known *a priori*, the gain vectors $\Phi_{i|N}{}^{-1}\boldsymbol{a}_{i|N}$ for $i = 0, 1, \cdots, N$ can be computed off line and stored prior to transmission. Furthermore, either prewindowed or covariance LS estimation can be used. It is in fact suggested [13] that a maximum length training sequence may not result in the fastest possible convergence, and that other training sequences (in particular, a training sequence which produces a diagonal covariance matrix at iteration $N-1$) may be more desirable. Ignoring the increase in required memory, the complexity (number of multiplies and additions per iteration) of the resulting prewindowed LS algorithm is identical to the complexity of the SG transversal algorithm.

Assuming a periodic maximum length training sequence of length N is transmitted for $i < N$, (2.18), (2.20), and (2.21) (assuming $w_j = 1$) can be used directly with (3.1)–(3.4) to give

$$\boldsymbol{c}_{N-1} = \boldsymbol{r} + \frac{1}{N+1}\sum_{j=0}^{N-1} \eta_j(\boldsymbol{a}_{j|N} + \mathbf{1}) \tag{3.7}$$

$$E[(\boldsymbol{c}_{N-1} - \boldsymbol{r})(\boldsymbol{c}_{N-1} - \boldsymbol{r})^T] = \frac{\sigma^2}{N+1}(\boldsymbol{I} + \mathbf{1}\mathbf{1}^T) \tag{3.8}$$

and

$$E[e_{N-1}'^2] = 2\sigma^2. \tag{3.9}$$

The properties of maximum length sequences therefore enable a convenient characterization of the impulse response estimate after N iterations. This example serves to illustrate the essential difference between the SG algorithm (2.26), which relies upon second-order *statistical* properties of the input, and LS algorithms which attempt to exploit the *algebraic* properties of the given input sequence [14].

IV. Algorithm Descriptions

In this section the specific algorithms used to generate the simulation results in Section V are described. Recursive SG and LS algorithms can assume either a tapped-delay line (transversal) or order-recursive (lattice) type of filter structure. The algorithms (2.13) and (2.26) recursively compute the coefficient vector $\boldsymbol{c}$ and, hence, assume the transversal filter structure shown in Fig. 3. In order to define the lattice, or order-recursive filter structure [15], we first define the following prediction errors. Given an arbitrary time series $a_0, a_1, \cdots, a_i$, where a_i is the current data sample, the nth-order forward prediction residual at time i is

$$e_f(i|n) = a_i - \boldsymbol{f}^T(i|n)\boldsymbol{a}_{i-1|n} \tag{4.1}$$

where

$$\boldsymbol{f}^T(i|n) = [f_{1|n} \cdots f_{n|n}]$$

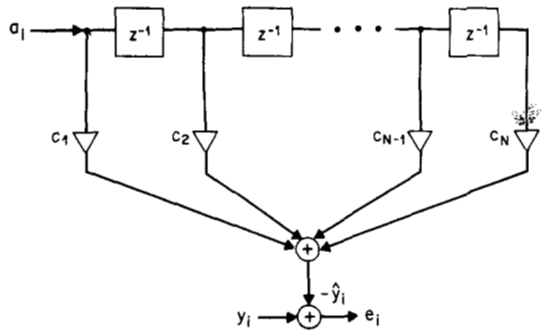


Fig. 3. Echo canceller using the transversal filter structure.

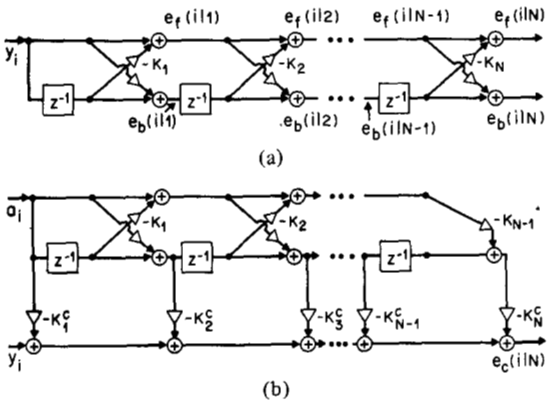


Fig. 4. (a) Lattice structure. (b) Echo canceller using the lattice structure.

is the vector of *forward* prediction coefficients. Similarly, the nth-order backward prediction residual at time i is

$$e_b(i\,|\,n) = a_{i-n} - \boldsymbol{b}^T(i\,|\,n)\boldsymbol{a}_{i|n} \tag{4.2}$$

where

$$\boldsymbol{b}^T(i\,|\,n) = [b_{1|n} \cdots b_{n|n}]$$

is the vector of *backward* prediction coefficients. If the second-order statistics of the data sequence $\{a_i\}$ are known and stationary, then the optimal values of the vectors $\boldsymbol{f}(i|N)$ and $\boldsymbol{b}(i|N)$ which respectively minimize $E[e_f{}^2(i|N)]$ and $E[e_b{}^2(i|N)]$ can be computed. The lattice structure is shown in Fig. 4(a). It follows directly from the following order recursions for the error sequences, given that the prediction vectors $\boldsymbol{f}$ and $\boldsymbol{b}$ are chosen optimally [15]:

$$e_f(i\,|\,n) = e_f(i\,|\,n-1) - K_n e_b(i-1|\,n-1) \tag{4.3a}$$

$$e_b(i\,|\,n) = e_b(i-1\,|\,n-1) - K_n e_f(i\,|\,n-1) \tag{4.3b}$$

for $1 \leqslant n \leqslant N$, where N is the filter order. The lattice, or PARCOR, coefficients K_n are given by

$$K_n = \frac{E[e_f(i\,|\,n-1)e_b(i-1\,|\,n-1)]}{E[e_f{}^2(i\,|\,n-1)]}. \tag{4.4}$$

An important property of the backward residuals is that they are uncorrelated, that is,

$$E[e_b(i\,|\,m)e_b(i\,|\,n)] = \delta_{mn}E[e_b{}^2(i\,|\,n)] \tag{4.5}$$

where $\delta_{mn} = 1$, for $m = n$, and $\delta_{mn} = 0$, for $m \neq n$. In analogy with the transversal structure, the filter coefficients K_n, $1 \leqslant n \leqslant N$, can be updated in time if the second-order statistics of the input sequence are unknown or time varying.

The filter structure shown in Fig. 4(a) can be used to generate prediction residuals given a *single* input time series. The application considered here, however, requires that a linear combination of the data samples a_{i-N+1}, a_{i-N+2}, $\cdots$, a_i be formed to predict the channel output y_i. The appropriate lattice structure for the echo cancellation application is shown in Fig. 4(b). The input data sequence is used as the input to the lattice structure to form the sequence of backward prediction errors $e_b(i|0)$, $e_b(i|1)$, $\cdots$, $e_b(i|N)$. A linear combination of the backward residuals, rather than the input data, is formed to predict y_i. The filter residual is therefore

$$e_i = y_i - \sum_{n=0}^{N-1} K_{n+1}^c e_b(i\,|\,n). \tag{4.6}$$

The coefficient K_n^c which minimizes $E[e_i{}^2]$ is

$$K_n^c = \frac{E[y_i e_b(i\,|\,n-1)]}{E[e_b{}^2(i\,|\,n-1)]} = \frac{E[e_c(i\,|\,n-1)e_b(i\,|\,n-1)]}{E[e_b{}^2(i\,|\,n-1)]} \tag{4.7}$$

where $e_c(i|\,n-1)$ is the $(n-1)$st-order filter residual e_i. From (4.6) it follows that

$$e_c(i\,|\,n) = e_c(i\,|\,n-1) - K_n^c e_b(i\,|\,n-1) \tag{4.8}$$

which, when combined with the recursions (4.3), results in the filter structure shown in Fig. 4(b).

Recursive SG and LS estimation techniques exist for both the transversal and lattice structures. The LS algorithms will be described first. Recursive covariance LS estimation [9], [15]-[18] requires substantially more computation than prewindowed estimation, and should not give substantial improvements in performance, so that we consider only the prewindowed algorithms.

Computationally efficient prewindowed recursive LS algorithms which require on the order of N operations per iteration have been presented in [1], [7], [15], and [19]-[21]. The following algorithm updates the lattice coefficients K_n and K_n^c, $1 \leqslant n \leqslant N$, so as to minimize the sum

$$R_c(i\,|\,n) = \sum_{j=0}^{i} w^{i-j}e_c{}^2(j\,|\,n), \quad 1 \leqslant n \leqslant N \tag{4.9}$$

where the error sequence $e_c(j|N)$, $j = 0, 1, \cdots, i$, results from passing the input data through a filter with *fixed* coefficients computed at time i.

Initialization:

$$\gamma_{-1|n} = e_b(-1\,|\,n) = k_{n+1}(-1) = 0 \qquad \text{for } 0 \leqslant n \leqslant N. \tag{4.10}$$

For $i = 0, 1, 2, \cdots$,

$$\gamma_{i|0} = 0 \tag{4.11}$$

$$e_f(i\,|\,0) = e_b(i\,|\,0) = a_i \quad e_c(i\,|\,0) = y_i \tag{4.12}$$

$$R_b(i\,|\,0) = R_f(i\,|\,0) = wR_f(i-1|\,0) + a_i{}^2. \tag{4.13}$$

For $n = 0$ to $n = n'$,

$$k_{n+1}(i) = wk_{n+1}(i-1) + e_f(i\,|\,n)e_b(i-1\,|\,n)\frac{1}{1-\gamma_{i-1|n}} \tag{4.14}$$

$$K_{n+1}^{b}(i)=\frac{k_{n+1}(i)}{R_b=(i-1\,|\,n)}\,,\qquad K_{n+1}^{f}(i)=\frac{k_{n+1}(i)}{R_f(i\,|\,n)} \tag{4.15a}$$

$$e_f(i\,|\,n+1)=e_f(i\,|\,n)-K_{n+1}^{b}(i)e_b(i-1\,|\,n) \tag{4.15b}$$

$$e_b(i\,|\,n+1)=e_b(i-1\,|\,n)-K_{n+1}^{f}(i)e_f(i\,|\,n) \tag{4.15c}$$

$$R_f(i\,|\,n+1)=R_f(i\,|\,n)-k_{n+1}(i)K_{n+1}^{b}(i) \tag{4.16a}$$

$$R_b(i\,|\,n+1)=R_b(i-1\,|\,n)-k_{n+1}(i)K_{n+1}^{f}(i) \tag{4.16b}$$

$$\gamma_{i|n+1}=\gamma_{i|n}+\frac{e_b^{\,2}(i\,|\,n)}{R_b(i\,|\,n)} \tag{4.17}$$

$$k_{n+1}^{c}(i)=wk_{n+1}^{c}(i-1)+e_c(i\,|\,n)e_b(i\,|\,n)\frac{1}{1-\gamma_{i|n}} \tag{4.18}$$

$$K_{n+1}^{c}(i)=\frac{k_{n+1}^{c}(i)}{R_b(i\,|\,n)} \tag{4.19a}$$

$$e_c(i\,|\,n+1)=e_c(i\,|\,n)-K_{n+1}^{c}(i)e_b(i\,|\,n) \tag{4.19b}$$

where

$$k_n(i)=\sum_{j=0}^{i}e_f^{*}(j\,|\,n)e_b^{*}(j-1\,|\,n) \tag{4.20}$$

and the LS cost functions are

$$R_f(i\,|\,n)=\sum_{j=0}^{i}e_f^{*2}(j\,|\,n) \tag{4.21}$$

$$R_b(i\,|\,n)=\sum_{j=0}^{i}e_b^{*2}(j\,|\,n). \tag{4.22}$$

Also, the gain variable

$$\gamma_{i\,|n}=\boldsymbol{a}_{i|n}{}^{T}\Phi_{i|n}{}^{-1}\boldsymbol{a}_{i|n} \tag{4.23}$$

where $\Phi_{i|\,n}$ is the covariance matrix (2.15). The variable γ does not appear in SG lattice algorithms and is analogous to the Kalman gain, which appears in recursive LS transversal algorithms [10], [19]-[21]. The residuals $e_f^{*}(j|n)$ and $e_b^{*}(j|n)$, $0\leqslant j\leqslant i$, result from passing the input through a predictor with *fixed* coefficients computed at time i.

The value of n', which specifies the upper limit for the order recursions (4.14)-(4.19), depends upon the initialization procedure. Notice that the recursions (4.15) cannot be computed unless $R_f(i|n)\neq 0$ and $R_b(i-1|n)\neq 0$. This will not be true for all $n<N$ when $i<N$. One procedure to alleviate this problem is to perform the computations (4.14)-(4.19) for $n=1,\cdots,$ min $(i-1, N-1)$. In this way the filter is "built" in an order-recursive fashion, i.e., given $i<N$ data samples, an ith-order LS filter is computed. This startup procedure is equivalent to solving the triangular set of equations (2.16). Alternatively, we can set

$$R_f(-1\,|\,n)=R_b(-1\,|\,n)=\delta \qquad \text{for } 0\leqslant n\leqslant N-1 \tag{4.24}$$

where δ is some small constant chosen to ensure that the algorithm remains stable. This initialization is analogous to initializing the diagonal elements of the matrix $\Phi_{i|N}$ to δ.

A description of the computationally efficient version of (2.13), i.e., the fast Kalman algorithm, is omitted since it was not used to generate the results in the next section. Ignoring finite word length effects, the performance of the fast Kalman algorithm should be identical to that of the LS lattice given identical startup procedures, since both minimize the sum (4.9). We point out that both startup procedures described for the LS lattice can also be used to start the fast Kalman algorithm. In particular, the diagonal elements of $\Phi_{i|\,n}$ can be set to a small constant, or the coefficient vector $\boldsymbol{c}$ can be "built up" in an order-recursive fashion by using order recursions for the prediction vectors $\boldsymbol{f}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ in analogy with (4.15) and (4.19) [16], [17], [20]. The fast Kalman algorithm often diverges given a random input sequence, however, and hence, the lattice algorithm is to be preferred for most applications. Recently, new fixed-order recursive LS algorithms have been presented which require somewhat less computation than the fast Kalman algorithm, and have superior numerical properties [20], [21]. These algorithms were not simulated.

The simulated SG algorithms are now described. The simulated SG transversal algorithm is given by (2.26) with the additional recursion

$$\beta=\frac{1}{wR(i-1)+a_i^{\,2}}\,,\qquad R(0)=\delta \tag{4.25}$$

where w is an exponential weighting constant close to unity, and δ is a small constant. The step size β is therefore inversely proportional to a time-varying estimate of the input variance. The reason for this modification is that the optimal step size is inversely proportional to the expected value of the signal energy present in the transversal filter [4], which is increasing for $i<N$.

SG lattice algorithms can be obtained by replacing the expectations in (4.4) and (4.7) with time averages. There are numerous ways of doing this [15], [22]-[25], all of which are heuristic. The SG lattice algorithm simulated in the next section is given by the LS recursions (4.10)-(4.19) where the LS gain $\gamma_{i|\,n}$, $0\leqslant n\leqslant N$, is set to zero, and the order recursions (4.16) are replaced by the time averages

$$R_f(i\,|\,n)=wR_f(i-1\,|\,n)+e_f^{\,2}(i\,|\,n) \tag{4.26a}$$

and

$$R_b(i\,|\,n)=wR_b(i-1\,|\,n)+e_b^{\,2}(i\,|\,n). \tag{4.26b}$$

This algorithm is almost identical to an SG algorithm which adapts K_{n-1}^{f} and K_{n-1}^{b} independently to minimize $E[e_b^{\,2}(i|n)]$ and $E[e_f^{\,2}(i|n)]$, respectively [22]. The essential difference is that here the lattice coefficients are updated *before* the order updates (4.15b), (4.15c), and (4.19b), whereas in [22]-[25] they are updated *after* these order updates. The SG algorithm used here more closely parallels the LS lattice algorithm, but may in fact give inferior performance relative to alternative SG lattice algorithms.

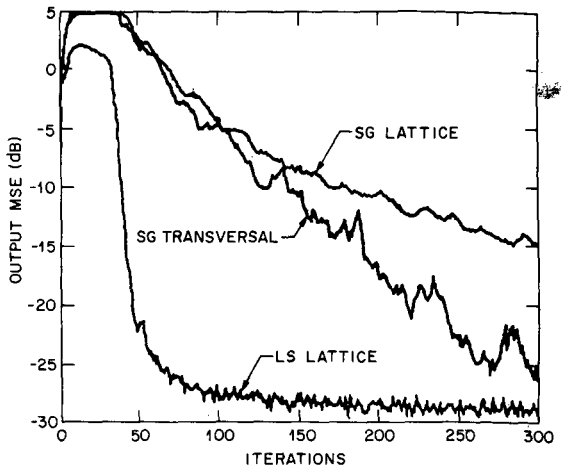
V. Simulation Results

This section presents simulation results comparing the performance of the (prewindowed) LS lattice, SG transversal, and SG lattice algorithms described in the last section. In all cases the input data are assumed to be independent binary samples (± 1), the channel impulse response is given by

$$r_i=(0.96)^{i},\qquad i=0,1,\cdots,N, \tag{5.1}$$

and the signal-to-noise ratio is

$$\text{SNR}=\frac{E[y_i^{\,2}]}{E[\eta_i^{\,2}]}=40\text{ dB}. \tag{5.2}$$



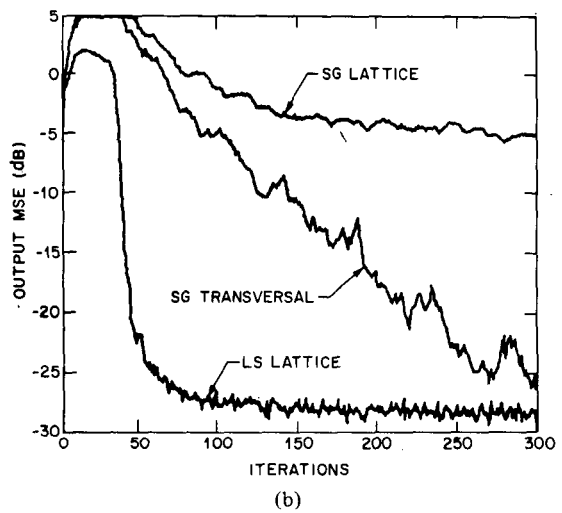
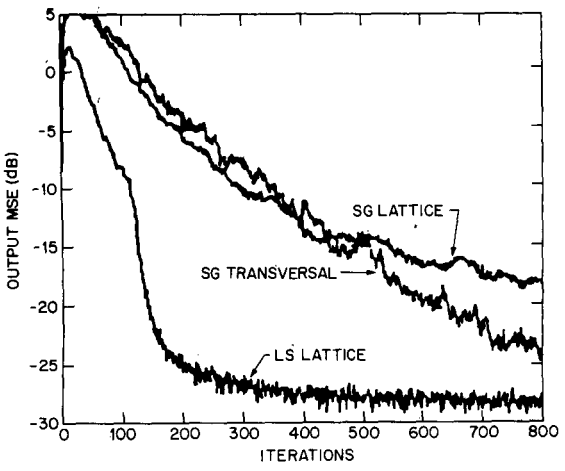


Fig. 5. (a) Averaged output squared error versus time obtained from the SG transversal algorithm, the SG lattice algorithm, and the LS lattice algorithm in the context of Fig. 2 for $N = 30$ and $w = 1$. (b) Averaged output squared error versus time obtained from the SG transversal and lattice algorithms and the LS lattice algorithm for $N = 30$ and $w = 0.967$.



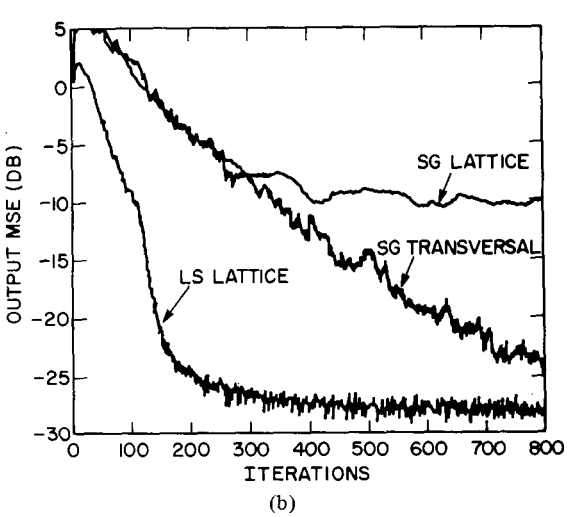


Fig. 6. Averaged output squared error versus time obtained from the SG transversal and lattice algorithms, and the LS lattice algorithm. (a) $N = 100$ and $w = 1$. (b) $N = 100$ and $w = 0.99$.

Fig. 5 shows plots of output squared error versus time averaged over 200 individual runs of each of the algorithms mentioned above. The filter order N in each case was 30. The exponential weighting constant for the lattice algorithms was 1.0 in Fig. 5(a), whereas in Fig. 5(b), $1 - w = 1/30$, the optimal SG transversal step size. Fig. 6 shows analogous plots for $N = 100$. The value of w used in the lattice algorithms was again 1.0 in Fig. 6(a), and $1 - w = 0.01$ in Fig. 6(b). In all cases the value of w in (4.25) was selected to ensure that the step size β converged to the optimal step size $1/N$. The SG transversal convergence curves in Figs. 5 and 6 were also compared to curves obtained using a constant optimal step size $\beta = 1/N$. Both the constant and time varying step size algorithms yield approximately the same performance.

The order-recursive startup procedure described for the lattice algorithms in Section IV often caused the LS algorithm to diverge. The reason for this stems from the poor estimate of the coefficient vector $\boldsymbol{c}$, which is obtained by inverting the triangular set of linear equations (2.16) with additive noise. To alleviate this problem, the initialization (4.24) was used.

The error plotted for each of the three algorithms is the causal error $e_c'(i|N)$ defined by (2.14). Notice that $E[e_c'^2(i)]$ decreases monotonically from its maximum value to its minimum value as i increases. In contrast, $E[e_c^2(i)]$, which results from an LS algorithm with an exact order-recursive startup, is zero for $i < N$, jumps to some positive value at $i = N$, and then decreases monotonically. Also, when exponential weighting is used, recent errors are weighted more than past errors, so that the current error is small relative to past errors. The asymptotic value of $E[e_c^2(i)]$ with exponential weighting can therefore be less than the channel noise variance and, hence, is physically less meaningful than $E[e_c'^2(i)]$.

It is straightforward to show that for the LS lattice algorithm [17], [20], [21]

$$e_c'(i|N) = \frac{e_c(i|N)}{1 - \gamma_{i-1|N}}. \tag{5.3}$$

Unfortunately, no such simple relation exists for determining the error e_c', which results from the SG lattice algorithm simulated here. Consequently, this error must be obtained at iteration i by freezing the filter coefficients K_n^f, K_n^b, and K_n^c, $1 \leq n \leq N$, at iteration $i - 1$, and computing the error $e_c(i|N)$ that results from passing the data samples y_{i-N+1}, $\cdots$, y_{i+1} through the fixed coefficient lattice. For large values of N this procedure becomes computationally infeasible; however, for $N = 30$, averaged plots of the following squared error,

$$e_i''^2 = \left[y_i - \sum_{j=0}^{N-1} K_{j+1}^c (i-1) e_b(i|j) \right]^2, \tag{5.4}$$

were found to be nearly identical to averaged plots of $e_i'^2$. (Notice that e_i'' is not the same as e_i', since $e_b(i|j)$ depends upon the current lattice coefficients $K_m^f(i)$ and $K_m^b(i)$, for $0 \leqslant m \leqslant j$.) Consequently, for computational convenience, the SG lattice curves correspond to averages of $e_i''^2$ rather than $e_i'^2$.

The LS lattice output MSE plots in Figs. 5 and 6 indicate that the output MSE is less than 3 dB from its asymptotic value after $2N$ iterations, which is consistent with the discussion in Section II. The value of δ in (4.24) was 0.1. The fast Kalman algorithm was also simulated; however, it was found that the value of δ required to ensure that the algorithm remains stable is quite large (i.e., $\delta = 10$). The resulting convergence of the output MSE is considerably slower than the simulated LS lattice curves in Figs. 5 and 6.

To see why the LS lattice algorithm should have superior numerical properties in comparison to the fast Kalman algorithm during startup, consider the case where $i < N$ data samples are used to compute an LS filter. If the transversal structure is used, then the triangular set of equations (2.16) applies. As discussed in Section II, as i increases, the estimate $\boldsymbol{c}_i$ degrades rapidly in the presence of additive noise. This estimate may in fact become numerically very large, in which case a very high degree of numerical precision is required to accurately compute successive coefficient vectors. Also, once $N + 1$ data values are available, (2.16) no longer yields an LS solution and the LS estimate $\boldsymbol{c}_N$ may be drastically different from $\boldsymbol{c}_{N-1}$. Because the fast Kalman algorithm consists of a large number of recursions, some of which involve N adds and multiplies, it contains numerous sources of roundoff error. This roundoff error can easily lead to instability, due to the extreme sensitivity to noise exhibited by the estimate $\boldsymbol{c}_m$ for $m < N$ and the large range in values which it may assume.

If $i < N$ data samples are used to compute an LS lattice filter, the first m stages of the lattice, where $m < i$, constitute an mth-order LS filter based upon i data samples. Computation of the mth-order coefficients is therefore well determined (i.e., the sequence of mth-order output errors cannot be zero) and does not exhibit the numerical problems associated with computing $\boldsymbol{c}_m$ via (2.16). Given i data samples, it follows that only the ith-order lattice coefficients may be ill-determined (i.e., the ith-order residuals are zero), and that when the $(i + 1)$st data sample is received, the estimated ith-order coefficients are well-behaved. Because the estimate $\boldsymbol{c}_m$ obtained via (2.16) may be extremely poor given i data samples, however, it must be expected that the estimated ith-order lattice coefficients will also be poor, and that numerical instability may result in analogy with the fast Kalman algorithm. This was in fact found to be the case, and hence, the order-recursive lattice startup procedure described in Section IV cannot be used in this application. Because the lattice structure is order-recursive, however, latter stages can diverge without affecting preceding stages, and hence, the value of δ in (4.24) needed to ensure stability is much less than that needed for the fast Kalman algorithm. The fast LS convergence described in Section II can therefore be obtained by using the lattice structure. An analysis of roundoff error in the fast Kalman and LS lattice algorithms is given in [26].

The slower convergence of the SG algorithms in Figs. 5 and 6 is due to their reliance upon time averages, such as (4.26), to compute second-order statistics. The poor performance of the SG lattice algorithm relative to the SG transversal algorithm is due to statistical fluctuations in the coefficients K_j^f and K_j^b, $1 \leqslant j \leqslant N$. Since the data sequence $\{a_i\}$ is uncorrelated, the optimal value of these coefficients, given by (4.4), is zero. Adapting these coefficients to estimate the second-order statistical properties of the input a_i, therefore, cannot improve upon the performance of the SG transversal algorithm, which is obtained by arbitrarily setting these coefficients to zero. The effect of these coefficient fluctuations is to increase the variance of the backward errors which are used in the tapped delay line (bottom half of the filter structure) shown in Fig. 4(b). This in turn increases the output MSE. (This increase in output MSE due to lattice coefficient fluctuations can be approximated analytically [23], [24].) In other studies, which have reported improved performance of the SG lattice relative to the SG transversal algorithm, the input to the predictor part of the lattice is *correlated* [25], [27]. The coefficients K_j^f and K_j^b in this case are adapted to decorrelate the input sequence and, therefore, have nonzero optimal values.

It is emphasized that the previous argument does *not* hold when comparing the LS lattice to the SG transversal algorithm. In particular, the LS coefficients K_n^f and K_n^b, $1 \leqslant n \leqslant N$, are *not* adapted to estimate second-order statistical properties. Instead, they are adapted to orthogonalize the input in an algebraic sense. The SG, or statistical property (4.5), is replaced by the following LS property:

$$\sum_{j=0}^{i} e_b^*(j|m) e_b^*(j|n) = \delta_{mn} R_b(i|N) \tag{5.5}$$

where e_b^* and R_b are defined as in Section IV. The optimal LS values of K_j^f and K_j^b in this case are therefore *not* equal to zero at any given time instant, although the asymptotic mean values of K_j^f and K_j^b are zero.

VI. Conclusions

The performance of the SG transversal, SG lattice, fast Kalman, and LS lattice algorithms has been studied in the context of echo cancellation of voiceband data signals. Ignoring finite precision problems, LS algorithms converge in approximately $2N$ iterations in the presence of channel noise, where N is the order of the filter. This is 4–5 times faster than the convergence speed of the SG algorithms considered. Due to the poor numerical properties of the estimated channel impulse response given $m < N$ data samples, the fast Kalman algorithm often diverges given a random training sequence. This problem can be circumvented by using the lattice structure. Finally, the SG lattice exhibits the poorest performance of the four algorithms compared, since the lattice PARCOR coefficients are adapted to optimal values of zero via statistical averaging.

Some of the results presented in this paper may not carry over to other applications (such as time series prediction) because of the special properties of the application considered. Nevertheless, this particular application has provided insight into the behavior of LS and SG algorithms which has thus far been absent in the literature.

Appendix

We wish to approximate output MSE using the LS estimate (2.13) with exponential weighting. Combining (2.19) and (2.21) gives

$$E[e_i'^2 \mid a_0, \cdots, a_i] = \sigma^2 \left\{ 1 + \boldsymbol{a}_{i|N}{}^T \Phi_{i|N}{}^{-1} \left(\sum_{j=0}^{i} w_j{}^2 \boldsymbol{a}_{j|N} \boldsymbol{a}_{j|N}{}^T \right) \Phi_{i|N}{}^{-1} \boldsymbol{a}_{i|N} \right\}. \tag{A.1}$$

To compute the unconditional output MSE, the average

$$\epsilon = E \left\{ \boldsymbol{a}_{i|N}{}^T \Phi_{i|N}{}^{-1} \left(\sum_{j=0}^{i} w_j{}^2 \boldsymbol{a}_{j|N} \boldsymbol{a}_{j|N}{}^T \right) \Phi_{i|N}{}^{-1} \boldsymbol{a}_{i|N} \right\} \tag{A.2}$$

must be computed over all possible 2^{i+1} sequences $(a_0, \cdots, a_i)$. For large i we approximate

$$\sum_{j=0}^{i} w_j \boldsymbol{a}_{j|N} \boldsymbol{a}_{j|N}{}^T \approx \sum_{j=0}^{i} w_j E[\boldsymbol{a}_{j|N} \boldsymbol{a}_{j|N}{}^T] = \boldsymbol{I} \sum_{j=0}^{i} w_j. \quad \text{(A.3)}$$

Using this approximation in (A.2) gives

$$\epsilon \approx N \frac{\sum_{j=0}^{i} w_j{}^2}{\left(\sum_{j=0}^{i} w_j\right)^2}. \quad \text{(A.4)}$$

Substituting $w_j = w^{i-j}$, where w is constant, gives

$$\epsilon \approx N \frac{\left(\frac{1 - w^{2(i+1)}}{1 - w^2}\right)}{\left(\frac{1 - w^{i+1}}{1 - w}\right)^2} = N \frac{1 - w}{1 + w} \frac{1 + w^{i+1}}{1 - w^{i+1}} \quad \text{(A.5)}$$

and substituting (A.5) into (A.1) gives (2.22).

Acknowledgment

The author thanks R. D. Gitlin, J. J. Werner, and C. W. Farrow for many valuable discussions concerning this work.

References

[1] M. Morf and D. T. Lee, "Fast algorithms for speech modeling," Inform. Syst. Lab., Stanford Univ., Stanford, CA, Tech. Rep. M308-1, Dec. 1978.

[2] E. Satorius and J. Pack, "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-29, pp. 136–142, Feb. 1981.

[3] F. K. Soong and A. M. Peterson, "Fast least-squares (LS) in the voice echo cancellation application," in *Proc. IEEE ICASSP,* Paris, France, May 1982.

[4] J. J. Werner, "An inband data-driven echo canceler for 4800 bps full-duplex data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 722–730, Sept. 1984.

[5] R. D. Gitlin, J. E. Mazo, and M. G. Taylor, "On the design of gradient algorithms for digitally implemented adjustment filters," *IEEE Trans. Circuit Theory,* vol. CT-20, pp. 125–136, Mar. 1973.

[6] J. Salz, "On the start-up problem in digital echo cancelers," *Bell Syst. Tech. J.*, vol. 62, pp. 1353–1364, July–Aug. 1983, part 1.

[7] D. D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446, Oct. 1978.

[8] R. S. Medaugh and L. J. Griffiths, "A comparison of two fast linear predictors," in *Proc. IEEE ICASSP,* Atlanta, GA, Apr. 1981.

[9] D. Manolakis, F. Ling, and J. G. Proakis, "Efficient least-squares algorithms for finite memory adaptive filtering," presented at Conf. Inform. Sci. Syst., Princeton, NJ, Mar. 1984.

[10] D. Godard, "Channel equalization using Kalman filter for fast data transmission," *IBM J. Res. Develop.*, pp. 267–273, May 1974.

[11] W. W. Peterson, *Error-Correcting Codes.* Cambridge, MA: M. I. T. Press, 1961.

[12] C. W. Farrow, unpublished work.

[13] J. M. Cioffi and T. Kailath, "An efficient, high-performance-least-squares, data-driven echo canceller for full-duplex data transmission," *IEEE Trans. Commun.*, to be published.

[14] M. Mueller, "On the rapid initial convergence of least squares equalizer adjustment algorithms," *Bell Syst. Tech. J.*, vol. 60, pp. 2345–2358, Dec. 1981.

[15] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE,* vol. 70, pp. 829–867, Aug. 1982.

[16] B. Porat, B. Friedlander, and M. Morf, "Square-root covariance ladder algorithms," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 813–829, Aug. 1982.

[17] M. L. Honig, "Recursive fixed-order covariance least-squares algorithms," *Bell Syst. Tech. J.*, vol. 62, pp. 2961–2992, Dec. 1983, part 1.

[18] J. M. Cioffi and T. Kailath, "Windowing methods and their efficient transversal-filter implementation for the RLS adaptive-filtering algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* submitted for publication.

[19] D. T. Lee, B. Friedlander, and M. Morf, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 627–641, June 1981.

[20] J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares, transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-32, pp. 304–337, Apr. 1984.

[21] D. Manolakis, G. Carayannis, and N. Kalouptsidis, "On the computational organization of fast sequential algorithms," in *Proc. IEEE ICASSP,* San Diego, CA, Mar. 1984, pp. 43.7.1–43.7.4.

[22] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. IEEE ICASSP,* Hartford, CT, May 1977, pp. 683–686.

[23] M. L. Honig and D. G. Messerschmitt, "Convergence properties of an adaptive digital lattice filter," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 642–653, June 1981.

[24] M. L. Honig, "Convergence models for lattice joint process estimator and least squares algorithms," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-31, pp. 415–425, Apr. 1983.

[25] M. L. Honig and D. G. Messerschmitt, "Comparison of LS and SG lattice predictor algorithms using two performance criteria," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-32, pp. 441–445, Apr. 1984.

[26] S. Ljung and L. Ljung, "Error propagation properties of recursive least squares adaptation algorithms," Dep. Elec. Eng., Lindkoping Univ., Lindkoping, Sweden, Internal Rep. LITH-ISY-I-0620, Sept. 1983.

[27] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899–905, June 1979.

★

Michael L. Honig (S'80-M'81) was born in Phoenix, AZ, on December 23, 1955. He received the B.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1978 and 1981, respectively.

From July 1981 to October 1983 he was a member of the Technical Staff at AT&T Information Systems, formerly part of Bell Laboratories, Holmdel, NJ. He subsequently transferred to Bell Communications Research, Inc., Morristown, NJ, where he is currently a member of the Communications Sciences Research Division. His research interests include adaptive signal processing and data communications.

Dr. Honig is a member of Tau Beta Pi and Phi Beta Kappa.