where A and B are square matrices of order M, s and t are vectors of dimension M, and ρ is a scalar. The matrix J is the M-dimensional contraidentity matrix. The matrices A and B are *general* matrices and need not possess any special structure. The above partitioning leads to slightly more general factorization results than can be found in the literature [1], [2]. This is particularly true in the case of an odd order matrix, where most previous results impose the conditions $s = t$, submatrix A be symmetric (i.e., $A = A^T$), and submatrix B be persymmetric (i.e., $B^T = JBJ$). This is done, of course, to restrict the class under consideration to doubly-symmetric matrices. This condition is necessary, for example, in order to represent covariance matrices. Our more general class of matrices may contain covariance matrices, but it also contains matrices which are negative definite and indefinite. The matrices in Examples 1, 2, and 3 given previously are negative definite, positive definite, and indefinite, respectively. This broader class of matrices exhibits some interesting properties and has a variety of applications. These issues are discussed in [3].

ACKNOWLEDGMENT

The authors acknowledge H. Krishna, Concordia University, for helpful discussions regarding this work.

REFERENCES

[1] S. D. Morgera, "On the reducibility of finite toeplitz matrices–Applications in speech analysis and pattern recognition," *Signal Processing*, vol. 4, pp. 425–443, Oct. 1982.

[2] A. Cantoni and P. Butler, "Eigenvalues and eigenvectors of symmetric centrosymmetric matrices," *Linear Algebra App.*, vol. 13, pp. 275–288, 1976.

[3] L. Datta and S. D. Morgera, "On the reducibility of generalized centrosymmetric matrices–Applications in engineering problems," *Proc. IEEE*, submitted for publication.

Comparison of Least-Squares and Stochastic Gradient Lattice Predictor Algorithms Using Two Performance Criteria

MICHAEL L. HONIG AND DAVID G. MESSERSCHMITT

Abstract–**The least-squares (LS) and stochastic gradient (SG) lattice prediction algorithms are compared using two different performance criteria. These are a) output mean squared error and b) the accuracy of the autoregressive spectral estimate obtained from the mean values of the lattice coefficients, assuming a stationary input. It is found that the second performance criterion is more sensitive than the first. This "spectral" performance criterion is a measure of the accuracy of the estimated autoregressive model coefficients. Bias in the LS and SG coefficient estimates can cause significant deviation of the asymptotic spectral estimates from the actual input spectrum. The similarly between the LS and SG lattice algorithms enables comparative simulations with analogous initial conditions and exponential weighting constants. For both types of comparisons, the LS algorithm offers a modest performance improvement over the SG algorithms simulated. This improvement is more noticeable when the input is highly correlated. It is also found that slight changes in the SG lattice algorithm may significantly affect its performance.**

Manuscript received June 13, 1981; revised March 15, 1982, July 15, 1983. This work was supported by the National Science Foundation under Grant ENG78-16966.

M. L. Honig was with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA. He is now with Bell Laboratories, Holmdel, NJ 07733.

D. G. Messerschmitt is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94705.

I. INTRODUCTION

The least-squares (LS) lattice algorithm has recently received considerable attention in applications of adaptive filtering [1]. The advantages of the lattice structure, combined with a computationally efficient method for recursively computing the filter coefficients which give an exact least squares solution to the prediction problem, has made the LS lattice an attractive alternative to other linear predictive techniques. In recent years, alternative stochastic gradient (SG) techniques have also been suggested as methods for adapting lattice coefficients [2], [3]. These SG algorithms require somewhat less computation than the LS algorithms; however, claims in the literature tend to suggest that a significant improvement in performance (i.e., convergence speed) can be achieved by using LS rather than SG adaptation techniques [1], [4]. This paper presents a comparison of LS and SG lattice prediction algorithms using two different performance criteria, output mean square error (MSE) and the accuracy of the autoregressive spectral estimate computed from mean coefficient values. The latter criterion is more sensiive than the first, since it depends upon the accuracy of the estimated filter coefficients.

Because of the similarity between SG and LS lattice algorithms, the comparative simulations presented in Section III use equivalent initial conditions and coefficient adaptation constants for each algorithm simulated. Section III-A presents results from simulations of LS and SG lattice predictors with correlated Gaussian noise inputs. Averaged output squared error versus time are shown for three different input spectra. In Section III-B the LS and SG algorithms are compared in the context of spectral estimation. Spectral estimates with time as a parameter are computed from the mean values of the lattice coefficients estimated via the LS and SG algorithms. Deviation of the (asymptotic) estimated spectrum from the actual input spectrum (for both algorithms) is caused by biased coefficient estimates.

II. LATTICE ALGORITHMS

In this section, the LS and SG lattice prediction algorithms which were used to generate the simulation results in the next sections are specified. The input sequence is denoted as y_i, the nth order forward prediction residual is

$$e_f(i|n) = y_i - \sum_{j=1}^{n} f_{j|n} y_{i-j} \tag{2.1}$$

where the $f_{j|n}$, $1 \leqslant j \leqslant n$ are the forward prediction coefficients, and the backward prediction residual is

$$e_b(i|n) = y_{i-n} - \sum_{j=0}^{n-1} b_{j|n} y_{i-j} \tag{2.2}$$

where the $b_{j|n}$, $1 \leqslant j \leqslant n$ are the backward prediction coefficients. The least squares lattice algorithm recursively adapts the PARCOR coefficients in a lattice structure so as to simultaneously minimize

$$R_f(i|n) \equiv \sum_{j=0}^{i} w^{i-j} e_f^2(j|n) \tag{2.3}$$

0096-3518/84/0400-0441$01.00

and

$$R_b(i|n) \equiv \sum_{j=0}^{i} w^{i-j} e_b^2(j|n) \tag{2.4}$$

for $1 \leqslant n \leqslant N$, where N is the order of the filter, and w is some exponential weighting constant close (or equal) to unity. The (prewindowed) algorithm is given as follows:

For $i = 0$ (initialization)

$$R_f(0|n) = R_b(0|n) = \delta \geqslant 0, \qquad 0 \leqslant n \leqslant N-1 \tag{2.5a}$$

$$e_b(0|n) = 0, \qquad 0 \leqslant n \leqslant N-1. \tag{2.5b}$$

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

$$e_f(i|0) = e_b(i|0) = y_i, \qquad \gamma(i|0) = 0 \tag{2.6a}$$

$$R_f(i|0) = wR_f(i-1|0) + y_i^2 \tag{2.6b}$$

$$k_{n+1}(i) = wk_{n+1}(i-1) + \frac{e_f(i|n)\, e_b(i-1|n)}{1-\gamma(i-1|n)} \tag{2.6c}$$

$$K_{n+1}^{(f)}(i) = \frac{k_{n+1}(i)}{R_f(i|n)} \qquad K_{n+1}^{(b)}(i) = \frac{k_{n+1}(i)}{R_b(i-1|n)} \tag{2.6d}$$

$$e_f(i|n+1) = e_f(i|n) - K_{n+1}^{(b)}(i)\, e_b(i-1|n) \tag{2.6e}$$

$$e_b(i|n+1) = e_b(i-1|n) - K_{n+1}^{(f)}(i)\, e_f(i|n) \tag{2.6f}$$

$$R_b(i|n+1) = R_b(i-1|n) - k_{n+1}(i)\, K_{n+1}^{(f)}(i) \tag{2.6g}$$

$$R_f(i|n+1) = R_f(i|n) - k_{n+1}(i)\, K_{n+1}^{(b)}(i) \tag{2.6h}$$

$$\gamma(i|n+1) = \gamma(i-1|n) + \frac{e_b^2(i|n)}{R_b(i|n)}. \tag{2.6i}$$

The order recursions (2.6e) and (2.6f) define the lattice structure shown in Fig. 1. The variable $\gamma(i|n)$ has been interpreted as an optimal weighting factor [1] and is given by

$$\gamma(i|n) = \boldsymbol{y}_{i|n}^T \boldsymbol{\Phi}_{i|n}^{-1} \boldsymbol{y}_{i|n} \tag{2.7}$$

where $\boldsymbol{y}_{i|n}$ is the data vector

$$\boldsymbol{y}_{i|n}^T = [\, y_i\, y_{i-1}, \cdots, y_{i-n+1}\,] \tag{2.8}$$

and $\boldsymbol{\Phi}_{i|n}$ is the sample covariance matrix

$$\boldsymbol{\Phi}_{i|n} \equiv \sum_{j=0}^{i} w^{i-j} \boldsymbol{y}_{j|n} \boldsymbol{y}_{j|n}^T + w^i \delta \boldsymbol{I} \tag{2.9}$$

where $\boldsymbol{I}$ is the identity matrix.

Stochastic gradient lattice algorithms start with the least mean square recursions (2.6e) and (2.6f) [1], [3] and attempt to estimate the values of $K_{n+1}^{(f)}$ and $K_{n+1}^{(b)}$ which minimize $E[e_f^2(i|n+1)]$ and $E[e_b^2(i|n+1)]$, respectively. These values are

$$K_{n+1,\text{opt}}^{(f)} = \frac{E[e_f(i|n)\, e_b(i-1|n)]}{E[e_f^2(i|n)]} \tag{2.10a}$$

and

$$K_{n+1,\text{opt}}^{(b)} = \frac{E[e_f(i|n)\, e_b(i-1|n)]}{E[e_b^2(i-1|n)]}. \tag{2.10b}$$

Assuming $K_j^{(f)} = K_{j,\text{opt}}^{(f)}$ and $K_j^{(b)} = K_{j,\text{opt}}^{(b)}$, $1 \leqslant j \leqslant N$ implies that

$$E[e_f^2(i|n)] = E[e_b^2(i-1|n)], \qquad 1 \leqslant n \leqslant N \tag{2.11}$$

and

$$K_{n+1,\text{opt}}^{(f)} = K_{n+1,\text{opt}}^{(b)} = \frac{2E[e_f(i|n)\, e_b(i-1|n)}{E[e_f^2(i|n) + e_b^2(i-1|n)]}. \tag{2.12}$$

The optimal coefficient values specified by (2.10) and (2.12) can be estimated via time averages. The SG algorithm considered here estimates the coefficients specified by (2.12) as follows:

$$k_{n+1}(i) = wk_{n+1}(i-1) + 2e_f(i|n)\, e_b(i-1|n) \tag{2.13a}$$

$$R_f(i|n) = wR_f(i-1|n) + e_f^2(i|n) + e_b^2(i-1|n) \tag{2.13b}$$

$$K_{n+1}^{(f)}(i) = K_{n+1}^{(b)}(i) = \frac{k_{n+1}(i)}{R_f(i|n)}. \tag{2.13c}$$

It was suggested to the authors by Griffiths [5] that if the constraint (2.13c) is assumed, then (2.13) should be used to estimate the lattice coefficients, as opposed to alternative "one-coefficient" techniques [3]. An alternative "two-coefficient" SG algorithm, which is essentially obtained by setting the LS gains in the LS lattice algorithm $\gamma(i|n) = 0$, for all i and n, was also simulated; however, results indicate that indeed (2.13) performs better than both alternative one- and two-coefficient estimation techniques.

III. Comparative Simulations

A. Output MSE

In this section, an empirical comparison of SG and LS lattice performance is made using output MSE as the performance criterion. Figs. 2–4 show results from three simulations of LS lattice and SG transversal and lattice predictors. In each case, the input was generated by passing a white Gaussian noise with variance 10^4 through a ten pole filter. The spectra of the inputs are shown in Figs. 2(a), 3(a), and 4(a), and the corresponding averaged trajectories of output squared prediction error versus time are shown respectively in Figs. 2(b), 3(b), and 4(b). Notice that the spectral peaks shown in Figs. 2–4 are progressively sharper, implying that the associated input processes are progressively more correlated (i.e., have a larger eigenvalue spread [6]). For all three cases the order of the filter was 10, the value of w used in the SG and LS lattice simulations was 0.97, and the initial value of R_f for both the LS and SG lattice simulations was 50. The SG transversal simulations used parameters $w = 0.95$ and $R_f(0) = 10^6$ in Figs. 2 and 3, and the parameters $w = 0.97$ and $R_f(0) = 10^8$ in Fig. 4. The initial coefficient values in each case were zero. Smaller initial values of R_f with the SG transversal algorithm generally caused instability.[1] Also shown in Figs. 2–4 are curves of MSE versus time computed via the analytical techniques in [3] and [8].

Figs. 2–4 indicate that the more "jagged" the spectrum of the input process, the greater the difference in convergence behavior for all three algorithms simulated. Both the LS and SG lattice algorithms appear to converge with approximately 150 samples in all three cases, indicating that the convergence speed of these algorithms is insensitive to second-order input statistics. The LS and SG algorithms behave similarly in Figs. 2 and 3; however, the SG lattice MSE levels out at a somewhat higher value than the LS MSE in Fig. 4 and at a slightly lower value than the LS MSE in Fig. 2. Changing the exponential weight in the LS lattice algorithm to $w = 1.0$ made little difference with the inputs in Figs. 2 and 3; however, this caused noticeably slower convergence when used with the input corresponding to Fig. 4. The convergence time with Gaussian noise inputs is significantly longer than the convergence time which occurs in different contexts such as channel equalization [4]. In particular, it has been pointed out by Mueller [9] that an LS equalizer requires only N linearly independent N element input data vectors to converge, assuming the additive channel noise is relatively small.

[1] See also [1] and [7] for additional comparisons of LS and SG lattice predictor performance using Gaussian noise inputs.

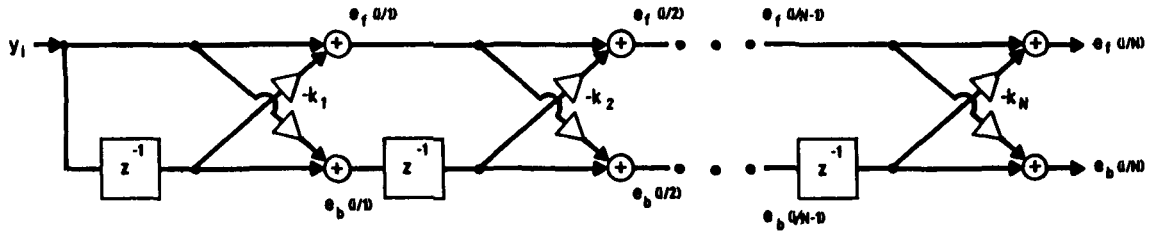
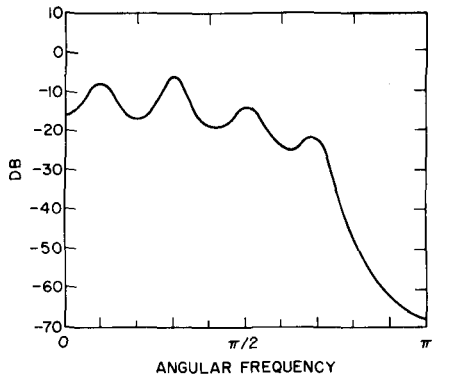
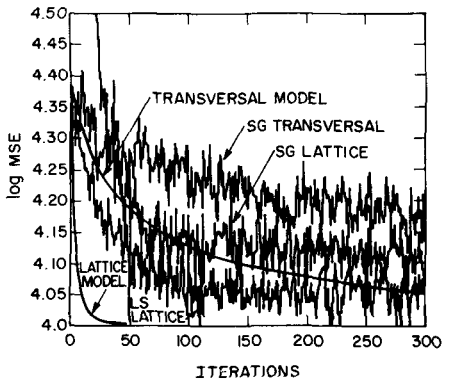


Fig. 1. Lattice structure.

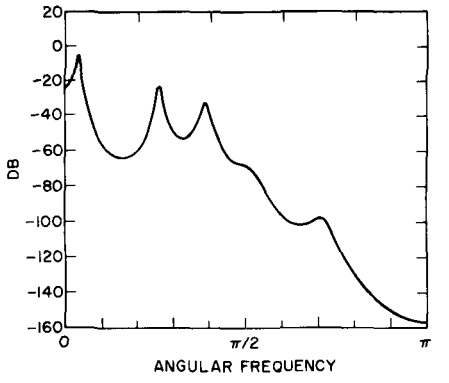


(a)

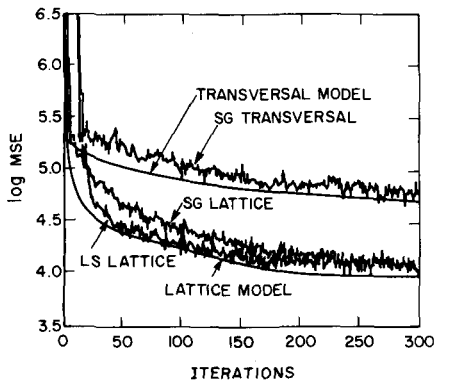


(b)

Fig. 2. (a) Spectrum of input noise used to generate Fig. 2(b). (b) Simulated MSE of the SG transversal predictor, the SG lattice predictor, and the LS lattice predictor. Also shown is MSE computed by the SG transversal and lattice convergence models.



(a)



(b)

Fig. 3. (a) Spectrum of input noise used to generate Fig. 3(b). (b) Simulated MSE of the SG transversal predictor, the SG lattice predictor, and the LS lattice predictor. Also shown is MSE computed by the SG transversal and lattice convergence models.

B. Spectral Estimates

In many applications of adaptive linear prediction, output MSE is an inappropriate performance criterion. In linear predictive coding of speech, for example, the accuracy of the estimated short-term speech spectrum is of paramount importance. For this case, the estimated filter coefficients rather than output MSE are of interest. In order to compare the performance of LS and SG lattice algorithms using a spectral criterion, we show in Figs. 5–7 spectra obtained respectively from the LS and SG lattice algorithms using the same correlated Gaussian noise input which was used to generate Fig. 3. The spectral estimates were obtained by averaging 200 simulations of each algorithm and sampling the mean coefficient values at $i = 20$, $i = 50$, $i = 100$, $i = 200$, and $i = 1000$. In the case of the SG lattice algorithm, the PARCOR coefficients K_n, $1 \leqslant n < 10$, can be computed from (2.13c), averaged, and converted to prediction coefficients [1] in order to compute the estimated spectrum. The LS coefficients used to generate Figs. 6 and 7 are similarly given by

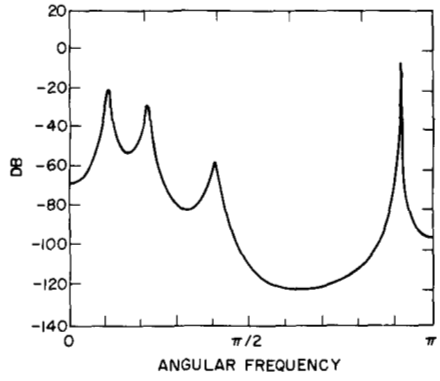
$$K_{n+1}(i) = \frac{2k_{n+1}(i)}{R_f(i|n) + R_b(i|n)}. \tag{3.1}$$

Spectra was also generated using averages of the coefficients

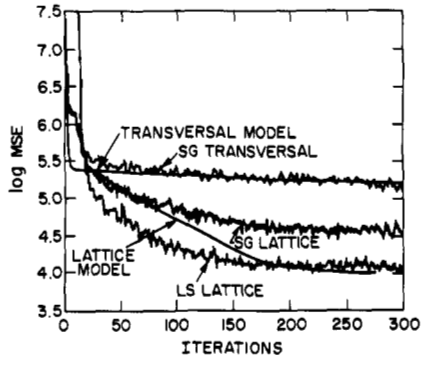
$$K_{n+1}(i) = \frac{k_{n+1}(i)}{\sqrt{R_f(i|n)\, R_b(i|n)}} \tag{3.2}$$

which enter the normalized LS lattice algorithm [1]; however, the estimates obtained from using (3.1) were found to be somewhat more accurate. Also shown in Figs. 5–7 is the actual input spectrum. The SG and LS parameters used in Figs. 5 and 6, respectively, were $w = 0.97$ and $R_f(0|n) = 50$. Fig. 7 shows spectral estimates obtained from the LS algorithm with $w = 1.0$.

Deviation of the estimated spectrum after convergence ($i = 1000$) in Figs. 5 and 6 is due to biased coefficient estimates. This effect becomes more pronounced as the exponential weight w decreases. Biased coefficient estimates are caused



(a)



(b)

Fig. 4. (a) Spectrum of input noise used to generate Fig. 4(b). (b) Simulated MSE of the SG transversal predictor, SG lattice predictor, and LS lattice predictor. Also shown is MSE computed by the SG transversal and lattice convergence models.

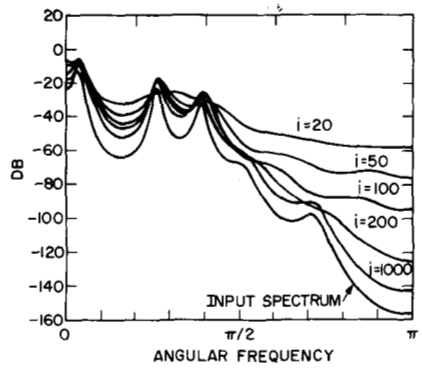


Fig. 5. Autoregressive spectral estimates versus time obtained from the SG lattice algorithm ($w = 0.97$).

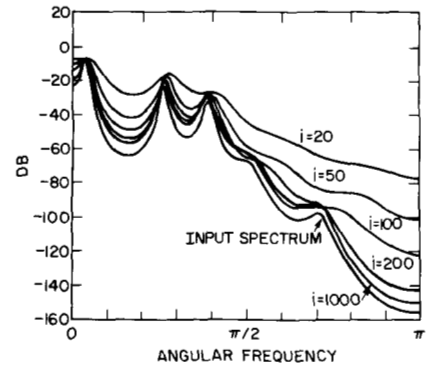


Fig. 6. Autoregressive spectral estimates versus time obtained from LS lattice algorithm ($w = 0.97$).

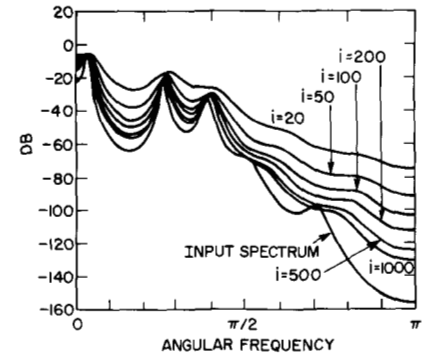


Fig. 7. Autoregressive spectral estimates versus time obtained from the LS lattice algorithm ($w = 1.0$).

by correlations between the filter coefficients and the inputs to each corresponding stage and by statistical fluctuations in filter coefficients which alter the statistics of input signals to successive stages [3], [8]. The difference in performance between the SG and LS lattice algorithms is more noticeable in Figs. 5 and 6 than in Fig. 3. In particular, the asymptotic spectral estimate obtained from the LS algorithm is slightly closer to the actual spectrum than the asymptotic spectral estimate obtained from the SG lattice algorithm. In both cases, the spectral estimate obtained at $i = 300$ was the same as that shown for $i = 1000$. Because exponential weighting was not used in the LS simulation shown in Fig. 7, the estimated coefficients converge in probability to the actual reflection coefficients associated with the input [10], and the asymptotic estimated spectrum is therefore the same as the input spectrum. Notice, however, that even at $i = 1000$, which is long after the output MSE is close to its asymptotic value, the estimated spectrum deviates considerably from the actual input spectrum. Figs. 5 and 7 therefore indicate that in applications of adaptive filtering where accurate estimates of the input spectrum are desired given a moderate number of samples, an exponential forgetting factor $w < 1$ is desirable.

IV. Conclusions

Comparative simulations of LS and SG lattice algorithms have been presented using stationary Gaussian input sequences. Two performance criteria, output MSE and the accuracy of the spectral estimate obtained from averaged coefficients, have been used as a basis for comparison. It has been emphasized that the essential difference between the LS lattice and SG lattice algorithms is the presence of LS gains $\gamma(i|n)$. In order to measure the affect of these weighting factors most of the comparative simulations in this paper have used equivalent parameters and initial conditions for both algorithms. Results indicate that the LS lattice algorithm offers modest improvement over the SG algorithms simulated. This difference in performance is more pronounced when the spectral criterion is used. In contrast to applications such as channel equalization, where the input to the adaptive filter is derived from a binary sequence, both LS and SG predictors require considerably more than $2N$ iterations to converge with correlated Gaussian noise inputs.

For many applications of adaptive linear prediction, es-

pecially those which involve highly nonstationary environments, the difference in performance exhibited by the LS and SG lattice algorithms may be more pronounced. A comparative study of algorithm performance in these remaining applications should yield interesting and useful results.

ACKNOWLEDGMENT

The authors are grateful to L. Griffiths for suggesting that the "one-coefficient" lattice simulations be performed.

REFERENCES

[1] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, Aug. 1982.
[2] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. 1977 IEEE ICASSP*, Hartford, CT, May 1977, pp. 683-686.
[3] 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.
[4] 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.
[5] L. Griffiths, personal communication.
[6] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res. Develop.*, pp. 546-555, Nov. 1972.
[7] R. S. Medaugh and L. J. Griffiths, "A comparison of two fast linear predictors," in *Proc. 1981 IEEE ICASSP*, Atlanta, GA, Mar. 1981, pp. 293-297.
[8] M. L. Honig, "Convergence models for lattice joint process estimators and least squares algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 415-425, Apr. 1983.
[9] M. S. Mueller, "On the rapid initial convergence of least squares equalizer adjustment algorithms," *Bell Syst. Tech. J.*, vol. 60, pp. 2345-2358, Dec. 1981.
[10] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*. New York: Academic, 1977.

Use of the Most Significant Autocorrelation Lags in Iterative ME Spectral Estimation

MIGUEL A. LAGUNAS-HERNANDEZ

Abstract—Many methods of spectral analysis are based either directly or indirectly on a set of autocorrelation values estimated from the available data. A good selection of autocorrelation Lags can improve the quality of the spectral estimate at a given computational cost.

To demonstrate the above possibility, this paper shows how to select Lags corresponding to the most significant values of the autocorrelation. In this way, one obtains better estimates than those found using the standard method, namely, the technique proposed by Lim and Malik [1] for (iterative) ME spectral analysis.

Several examples are considered to illustrate this possibility.

Manuscript received May 14, 1982; revised June 23, 1983 and October 10, 1983.

The author was on leave with the Department of Electrical Engineering, University of Colorado, Boulder, CO 80309. He is now with E.T.S.I. de Telecomunicación, c/ Jordi Girona Salgado, Barcelona-34, Spain.

I. Introduction

In autocorrelation based methods for spectral estimation, we form an estimate $\{\hat{r}_x(0), \cdots, \hat{r}_x(M)\}$ of the autocorrelation function of the signal x from the available data, and then we compute the power spectrum of x by direct or parametric methods using such an estimate.

It seems clear that the quality of the final power spectrum estimate will depend essentially on the amount of signal information which is retained in $\{\hat{r}_x(0), \cdots, \hat{r}_x(M)\}$. This paper attempts to demonstrate how an appropriate selection of lags (different from the standard one, $\{0, \cdots, M\}$), can considerably improve the quality of power spectrum (in a sense which depends on the criterion employed for lag selection).

This possibility is of paramount importance in cases where a high computational effort is necessary; for example, when the criterion is to minimize (or maximize) an objective function under autocorrelation restrictions. (Iterative algorithms or linear programming are then usually required.) Also, the convergence time and properties are better because a good startup of the algorithms is provided.

Although the idea is of general interest, we will focus our attention on ME iterative techniques; first, we show the potential advantage of the proposed procedure in 1-D cases in a clear context; second, we apply this procedure to the more critical 2-D problems, in which memory and computational restrictions force us to use only a very reduced number of autocorrelation values.

We will choose lags which have the largest absolute values of the autocorrelation. This approach shows that, at a fixed computational cost, there are improvements in the resolution and more noise immunity when compared with currently reported works.

II. Lag Selection in Spectral Estimation

The classical selection of the first $M + 1$ autocorrelation lags, $\{0, \cdots, M\}$, is supported by the fact of that this set offers the largest statistical stability among all the possible choices. We claim that using other selections will yield more desirable properties (using entropy as the objective function, the zero lag has to be included to ensure that a maximum exists).

It is clear that for stationary processes and even for nonstationary ones (like sinusoids in noise), the number of samples between highly correlated samples [i.e., number of lags between maxima of $r_x(m)$] provides a good deal of information about periodicities. Then, selecting the M lags (and zero) with largest absolute values will result in an estimate which is better than the classical one in the following sense: the peaks will be reinforced, the noise effects reduced. This approach will be referred to as the modified approach in the rest of this work.

Note that the modified method has a drawback; namely, it reduces the statistical stability since it uses autocorrelation values with lags greater than M; but like in prediction [2] and interpolation problems, the location of the most significant autocorrelation samples can be more important than their concrete values as concerns with some properties, such as resolution in the final spectral estimate, and convergence rate in adaptive or iterative methods for spectral estimation.

Of course, the use of all the lags available (or significant) will result in a better final estimate; but in most 2-D real problems, the use of a large correlation support is not allowed because it will increase the computational load beyond reasonable limits.

0096-3518/84/0400-0445$01.00