

*Ident. Syst. Param. Estim.*, June 1982, pp. 481-486. Also to appear *Int. J. Contr.*

- [31] J. A. Cadzow and R. L. Moses, "An adaptive ARMA spectral estimator: Parts 1 and 2," in *Proc. ASSP Workshop Spectral Estim.*, McMaster University, Hamilton, Ont., Canada, Aug. 1981.
- [32] B. Porat, B. Friedlander, and M. Morf, "Square-root covariance ladder algorithms," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 813-829, Aug. 1982.
- [33] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, pp. 829-867, Aug. 1982.
- [34] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. Berlin: Springer-Verlag, 1976.
- [35] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1982.
- [36] V. Solo, "The convergence of AML," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 958-962, Dec. 1979.
- [37] B. Friedlander, "Lattice methods for spectral estimation," *Proc. IEEE*, vol. 70, pp. 990-1017, Sept. 1982.
- [38] T. Söderström, "Convergence of identification methods based on the instrumental variable approach," *Automatica*, vol. 10, pp. 685-688, 1974.
- [39] Y. T. Chan, J. M. M. Lavoie, and J. B. Plant, "A parameter estimation approach to estimation of frequencies of sinusoids," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 214-219, Apr. 1981.
- [40] B. Friedlander, "The overdetermined recursive instrumental variable algorithm," *IEEE Trans. Automat. Contr.*, submitted for publication.



**Benjamin Friedlander** (S'74-M'76-SM'82) was born on February 24, 1947. He received the B.Sc. and the M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1968 and 1972, respectively, and the Ph.D. degree in electrical engineering and the M.Sc. degree in statistics from Stanford University, Stanford, CA, in 1976.

From 1968 to 1972 he served in the Israel Defense Force as an electronic engineer. In 1976 he joined Systems Control Technology, Inc., Palo Alto, CA, where he is now Manager of the Adaptive Systems Department. He was responsible for the analysis and development of signal processing and control algorithms for various applications, including: detection and tracking of multiple targets by acoustic sensors and radar systems, speech compression and transmission, synthetic aperture radar, tracking radars for fire control and homing missiles, electronic countermeasures for radars, optimal resource allocation, and decentralized computation and control in large-scale systems. He developed several new computationally efficient techniques for adaptive filtering, spectral estimation, and adaptive control. During this period he was also a lecturer at Stanford University, teaching graduate courses on linear systems, nonlinear estimation, and system identification. His current research interests include adaptive signal processing, system identification, adaptive control, and the development of estimation and control algorithms for large scale systems.

Dr. Friedlander is a member of Sigma Xi.

## Convergence Models for Lattice Joint Process Estimators and Least Squares Algorithms

MICHAEL L. HONIG, MEMBER, IEEE

**Abstract**—A simple model characterizing the convergence properties of an adaptive digital lattice filter using gradient algorithms has been reported [1]. This model is extended to the least mean square (LMS) lattice joint process estimator [5], and to the least squares (LS) lattice and "fast" Kalman algorithms [9]–[16]. The models in each case are compared with computer simulation. The single-stage LMS lattice analysis presented in [1] is also applied to the LS lattice. Results indicate that for stationary inputs, the LMS lattice and LS algorithms exhibit similar behavior.

### I. INTRODUCTION

**I**N [1] the convergence properties of a continuously adaptive lattice filter using least mean square (LMS) gradient algorithms were discussed. In particular, by making approxima-

tions, a simple model for convergence was described along with a first order estimate of coefficient variance and its affect upon the filter's mean squared output signal. In this paper these results are extended to the lattice joint process estimator [5], [7] and to the least squares (LS) lattice algorithm [12]–[16]. A simple convergence model similar to that presented in [1] is also derived for the "fast" Kalman algorithm [9]. In each case the accuracy of the convergence model is tested by computer simulation.

In Section II we extend the model for convergence presented in [1] to two (similar) versions of the LMS lattice joint process estimator. In Section III the single-stage analysis presented in [1] is extended to the least squares lattice filter. A comparison of these results with those given for the LMS lattice emphasizes the similarities between the two algorithms. In Section IV-A a convergence model is derived for the LS lattice predictor, and in Section IV-B for the LS lattice joint process estimator. In Section V a convergence model for the least squares transversal filter based upon the "fast" Kalman algorithm is derived.

Manuscript received September 17, 1981; revised October 15, 1982. This work was supported by the National Science Foundation under Grant ENG78-16966.

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

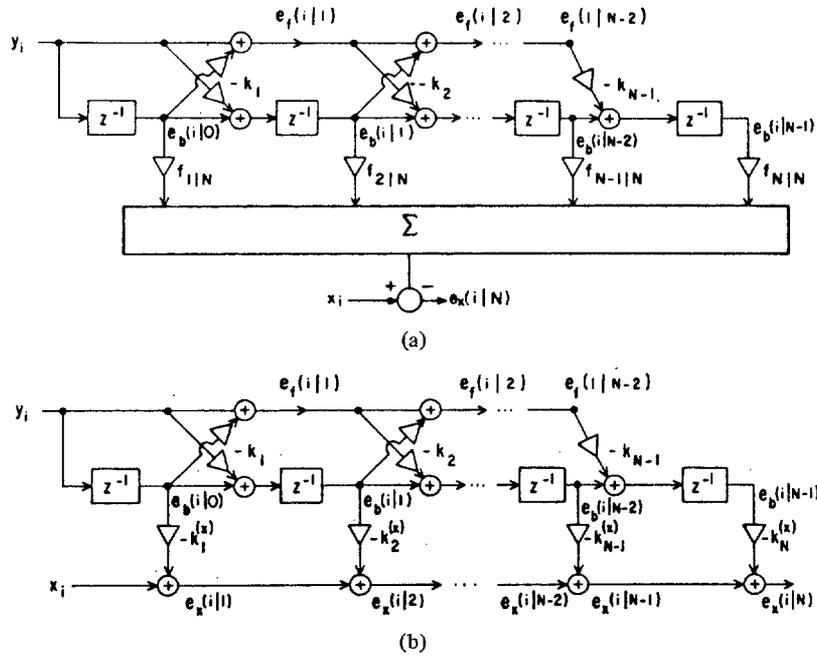


Fig. 1. (a) Lattice joint process estimator (version 1). (b) Lattice joint process estimator (version 2).

## II. LMS LATTICE JOINT PROCESS ESTIMATOR CONVERGENCE MODEL

We begin by extending the convergence model for the LMS lattice predictor to the two versions of the LMS lattice joint process estimator shown in Fig. 1. In particular, we are given a data sequence  $\{y_i\}$  which is used to estimate some other sequence  $\{x_i\}$ . Although numerous LMS methods exist for adapting the PARCOR coefficients in the lattice structure [4]–[6], the following lattice algorithm is used here,

$$e_f(i|0) = y_i, \quad e_b(i|0) = y_{i-1}, \quad e_x(i|0) = x_i$$

$$e_f(i|n+1) = e_f(i|n) - k_{n+1}(i)e_b(i|n) \quad (2.1a)$$

$$e_b(i|n+1) = e_b(i-1|n) - k_{n+1}(i-1)e_f(i-1|n) \quad (2.1b)$$

$$B_n(i) = (1 - \beta)B_n(i-1) + e_b^2(i|n) \quad (2.2a)$$

$$C_n(i) = (1 - \beta)C_n(i-1) + e_f(i|n)e_b(i|n) \quad (2.2b)$$

$$k_{n+1}(i) = \frac{C_n(i)}{B_n(i)} \quad (2.2c)$$

where  $0 \leq n \leq N$  and  $N$  is the filter order,  $\beta$  is the adaptation step-size,  $e_f(i|n)$  is the  $n$ th-stage forward residual,  $e_b(i|n)$  is the  $n$ th-stage backward residual, and  $k_n(i)$  is the  $n$ th-stage lattice PARCOR coefficient, all at time  $i$ . Notice that  $e_b(i|n)$  is defined as a function of  $y_{i-1}, y_{i-2}, \dots, y_{i-n-1}$ , i.e.,

$$e_b(i|n) = y_{i-n-1} - \sum_{j=1}^n b_{j|n} y_{i-j} \quad (2.3)$$

where the  $b_{j|n}$  are the backward prediction coefficients.

The two structures shown in Fig. 1 are equivalent; however, the “tap” coefficients which multiply the backward residuals of the lattice may be adapted in two ways. In particular, the

algorithm corresponding to Fig. 1(a) is

$$e_x(i|N) = x_i - f^T(i|N)e_b(i|N), \quad (2.4)$$

$$f_{j|N}(i+1) = f_{j|N}(i) + \frac{1}{B_j(i)} e_b(i|j-1)e_x(i|N) \quad (2.5)$$

where

$$f^T(i|N) = [f_{1|N}(i), f_{2|N}(i), \dots, f_{N|N}(i)], \quad (2.6)$$

$$e_b^T(i|N) \equiv [e_b(i|0), e_b(i|1), \dots, e_b(i|N-1)], \quad (2.7)$$

$f_{j|N}(i)$  is the  $j$ th “tap” coefficient, and  $e_x(i|N)$  is the  $N$ th-order filter prediction error, all at time  $i$ . The algorithm corresponding to Fig. 1(b) is

$$e_x(i|n+1) = e_x(i|n) - k_{n+1}^{(x)}(i)e_b(i|n) \quad (2.8)$$

$$C_n^{(x)}(i) = (1 - \beta)C_n^{(x)}(i-1) + e_x(i|n)e_b(i|n) \quad (2.9a)$$

$$k_{n+1}^{(x)}(i) = \frac{C_n^{(x)}(i)}{B_n(i)} = \left\{ 1 - \frac{e_b^2(i|n)}{B_n(i)} \right\} k_{n+1}^{(x)}(i-1) + \frac{e_x(i|n)e_b(i|n)}{B_n(i)} \quad (2.9b)$$

where in this case  $k_n^{(x)}(i)$  denotes the  $n$ th “tap” weight at time  $i$ . The tap weights are denoted differently in each case in order to emphasize the similarity between the first algorithm (2.5) and the LMS transversal algorithm, and the similarity between the second algorithm (2.9) and a lattice gradient algorithm. Although the second method is more popular [7] and is analogous to the least squares lattice joint process estimation algorithm, convergence models for both methods are described.

The optimal value of  $f(i|N)$ , obtained by minimizing  $E[e_x^2(i|N)]$ , is

$$f_{opt} = \{E[\epsilon_b(i|N)\epsilon_b^T(i|N)]\}^{-1}E[x_i\epsilon_b(i|N)]. \quad (2.10)$$

If the lattice coefficients are optimal,  $E[\epsilon_b(i|N)\epsilon_b^T(i|N)]$  is diagonal [5] and hence

$$[f(i)]_{i, opt} = \frac{E[x_i e_b(i|j-1)]}{E[e_b^2(i|j-1)]}. \quad (2.11)$$

Similarly, minimizing  $E[e_x^2(i|n)]$  with respect to  $k_n^{(x)}$  gives

$$k_{n, opt}^{(x)}(i) = \frac{E[e_x(i|n-1)e_b(i|n-1)]}{E[e_b^2(i|n-1)]}. \quad (2.12)$$

If the PARCOR coefficients are fixed at their optimal values, from the principle of orthogonality,  $f_{j, opt} = k_{j, opt}^{(x)}$ . If coefficients  $k_j$  or  $k_j^{(x)}$ ,  $1 \leq j < n$ , are adapting,  $e_x$  and  $e_b$  will be nonstationary, causing  $k_{n, opt}^{(x)}$  and  $f_{n, opt}$  to vary with time.

To model the lattice joint process estimator using (2.1), (2.4), and (2.5), we rewrite (2.5) as

$$\begin{aligned} f(i+1|N) &= f(i|N) + \bar{\beta}(i+1)\epsilon_b(i|N)[x_i - \epsilon_b^T(i|N)f(i|N)] \\ &= [I - \bar{\beta}(i+1)\epsilon_b(i|N)\epsilon_b^T(i|N)]f(i|N) \\ &\quad + \bar{\beta}(i+1)x_i\epsilon_b(i|N) \end{aligned} \quad (2.13)$$

where  $\bar{\beta}$  is a diagonal (normalized) step size matrix, i.e.,

$$\bar{\beta}(i+1) \equiv \text{diag} \left[ \frac{1}{B_1(i)}, \frac{1}{B_2(i)}, \dots, \frac{1}{B_N(i)} \right]. \quad (2.14)$$

Taking expectations of both sides of (2.13) assuming  $\epsilon_b(i|N)$  and  $f(i|N)$  are approximately independent gives

$$\begin{aligned} E[f(i+1|N)] &\approx \{I - E[\bar{\beta}(i+1)]E[\epsilon_b(i|N)\epsilon_b^T(i|N)]\} \\ &\quad \cdot E[f(i|N)] + E[\bar{\beta}(i+1)]E[x_i\epsilon_b(i|N)]. \end{aligned} \quad (2.15)$$

Assuming that the lattice PARCOR coefficients are following trajectories close to their mean value trajectories, we can use the convergence model for the lattice predictor [1] to compute the second-order statistics of  $\epsilon_b(i|N)$ . Specifically,

$$\begin{aligned} E[\epsilon_b(i|N)\epsilon_b^T(i|N)]_{lm} &= E[e_b(i|l-1)e_b(i|m-1)] \\ &= E \left[ \left( y_{i-l} - \sum_{j=1}^{l-1} b_{j|l-1}(i)y_{i-j} \right) \right. \\ &\quad \cdot \left. \left( y_{i-m} - \sum_{j=1}^{m-1} b_{j|m-1}(i)y_{i-j} \right) \right] \\ &\approx R_y(l-m) - \sum_{j=1}^{m-1} b_{j|m-1}(i)R_y(l-j) \\ &\quad - \sum_{j=1}^{l-1} b_{j|l-1}(i)R_y(m-j) \\ &\quad + \sum_{q=1}^{l-1} \sum_{j=1}^{m-1} b_{q|l-1}(i)b_{j|m-1}(i)R_y(q-j) \end{aligned} \quad (2.16)$$

and

$$\begin{aligned} \{E[x_i\epsilon_b(i|N)]\}_m &= E[x_i e_b(i|m-1)] \\ &= E \left[ x_i \left( y_{i-m} - \sum_{j=1}^{m-1} b_{j|m-1}(i)y_{i-j} \right) \right] \\ &= R_{xy}(m) - \sum_{j=1}^{m-1} b_{j|m-1}(i)R_{xy}(j) \end{aligned} \quad (2.17)$$

where  $R_y(j) = E[y_i y_{i+j}]$ ,  $R_{xy}(j) = E[x_i y_{i-j}]$ , and the  $b_{j|m-1}(i)$ ,  $1 \leq j \leq m-1$ ,  $1 \leq m \leq N$ , are computed from the convergence model for the lattice predictor. The value of

$$\{E[\bar{\beta}(i+1)]\}_{nm} \equiv \frac{1}{E[B_n(i)]}, \quad 1 \leq n \leq N$$

is computed by taking expectations of both sides of (2.2a), using the value of  $E[e_b^2(i|n)]$  calculated from the lattice predictor model. The output mean-squared error is computed as follows:

$$\begin{aligned} E[e_x^2(i|N)] &= E[x_i - f^T(i|N)\epsilon_b(i|N)]^2 \\ &\approx R_x(0) - 2E[f^T(i|N)]E[x_i\epsilon_b(i|N)] \\ &\quad + E[f^T(i|N)]E[\epsilon_b(i|N)\epsilon_b^T(i|N)]E[f(i|N)] \end{aligned} \quad (2.18)$$

where  $R_x(j) = E[x_i x_{i+j}]$  and  $E[f(i|N)]$ ,  $E[\epsilon_b(i|N)\epsilon_b^T(i|N)]$ , and  $E[x_i\epsilon_b(i|N)]$  are given, respectively, by (2.15), (2.16) and (2.17). Combining (2.15) through (2.18) with the model in [1] therefore completes the convergence model for the LMS lattice joint process estimator shown in Fig. 1(a). Since the same assumptions were made to derive this model as were made to derive the LMS lattice predictor model, the accuracy of the two models should be similar. Fig. 2 compares the mean value trajectories for the tap weights  $f_{j|10}(i)$ ,  $j = 4$  and  $10$ , computed from the model with the mean value trajectories obtained by averaging 200 separate simulations of the algorithm with a step size  $\beta = 0.01$ . (For a more detailed description of the simulation, see [2].) Fig. 3 shows the output mean squared error obtained from the model and by averaging 200 simulations. As pointed out in [1], as the step-size  $\beta$  decreases, the accuracy of the model should improve. Since the value of  $\beta$  used for this example is less than half that used in [1], the model curves shown here appear to be more accurate than those shown in [1].

To model the LMS lattice joint process estimator shown in Fig. 1(b), note from (2.4) that  $e_x(i|n)$  is a linear combination of  $x_i, y_{i-1}, y_{i-2}, \dots, y_{i-n}$ , i.e.,

$$e_x(i|n) = \bar{f}^T(i|n)y_{i|n}^{(x)}$$

where the  $(N+2)$ -dimensional vectors

$$\bar{f}^T(i|n) \equiv [1, -f_{1|n}(i), \dots, -f_{n|n}(i), 0, 0, \dots, 0] \quad (2.19)$$

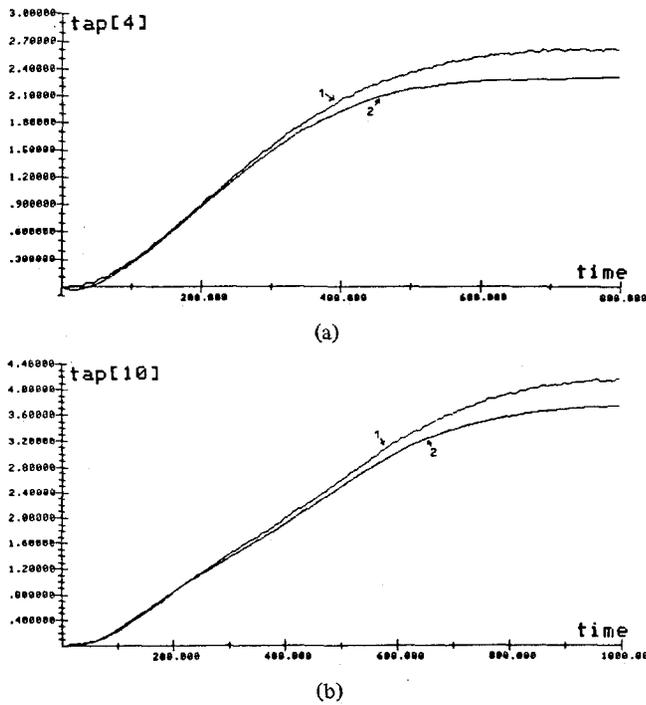


Fig. 2. Mean value trajectories of  $f_{j|n}^{(x)}(i)$ ,  $j$  equal to 4 and 10, in a lattice joint process estimator using (2.5) by (curve 1) simulation and (curve 2) from the convergence model.

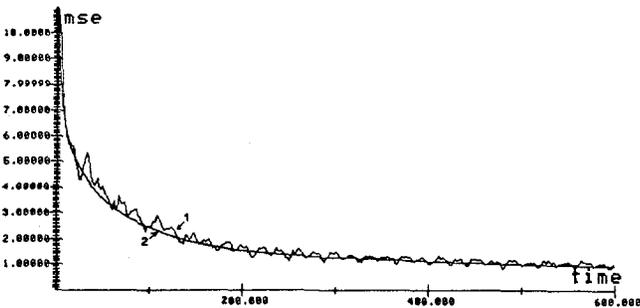


Fig. 3. Output MSE of a tenth-order lattice joint process estimator using (2.5) by (curve 1) simulation and (curve 2) from the convergence model.

and

$$y_{i|n}^{(x)T} \equiv [x_i, y_{i-1}, \dots, y_{i-n}, \dots, y_{i-N-1}]. \quad (2.20)$$

We therefore rewrite (2.8) with  $k_{n+1}^{(x)}(i)$  replaced by its mean value as

$$\bar{f}(i|n+1) = \bar{f}(i|n) - E[k_{n+1}^{(x)}(i)] \bar{b}(i|n) \quad (2.21)$$

where

$$\bar{b}(i|n) \equiv [0, -b_{1|n}(i), -b_{2|n}(i), \dots, -b_{n|n}(i), 1, 0, 0, \dots, 0], \quad (2.22)$$

and the  $b_{j|n}$  are the backward prediction coefficients in (2.3). Taking expectations of (2.9b) assuming  $k_n^{(x)}(i)$  is approximately independent of  $e_b(j|n)$  gives

$$E[k_{n+1}^{(x)}(i)] \approx \left\{ 1 - \frac{E[e_b^2(i|n)]}{E[B_n(i)]} \right\} E[k_{n+1}^{(x)}(i-1)] + \frac{E[e_x(i|n)e_b(i|n)]}{E[B_n(i)]}. \quad (2.23)$$

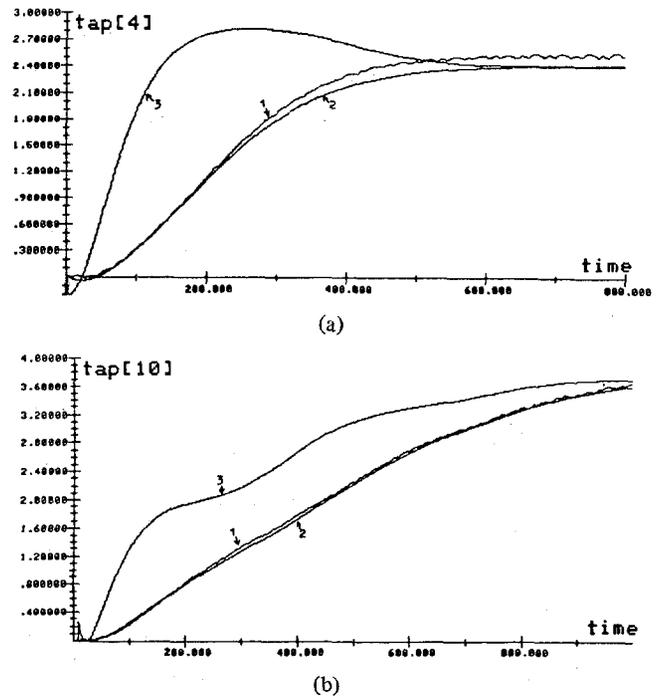


Fig. 4. Mean value trajectories of  $k_j^{(x)}(i)$ ,  $j$  equal to 4 and 10, in a lattice joint process estimator using (2.9) by (curve 1) simulation and (curve 2) from the convergence model. Curve 3 shows the trajectory of  $k_{j,opt}^{(x)}(i)$  computed by the model.

As previously mentioned,  $E[e_b^2(i|n)]$ , and hence  $E[B_n(i)]$ ,  $1 \leq n \leq N$ , are available from the lattice predictor model in [1]. In addition, we approximate

$$E[e_x^2(i|n)] = E\{\bar{f}^T(i|n)y_{i|n}^{(x)}y_{i|n}^{(x)T}\bar{f}(i|n)\} \approx E[\bar{f}^T(i|n)]E[y_{i|n}^{(x)}y_{i|n}^{(x)T}]E[\bar{f}(i|n)] \quad (2.24)$$

and

$$E[e_x(i|n)e_b(i|n)] = E[\bar{f}^T(i|n)y_{i|n}^{(x)}y_{i|n}^{(x)T}\bar{b}(i|n)] \approx E[\bar{f}^T(i|n)]E[y_{i|n}^{(x)}y_{i|n}^{(x)T}]E[\bar{b}(i|n)]. \quad (2.25)$$

Combining (2.21), (2.23), (2.24), and (2.25) with the model in [1] therefore completes our convergence model for the LMS lattice joint process estimator shown in Fig. 1(b). Fig. 4 compares the mean value trajectories of the coefficients  $k_j^{(x)}(i)$ ,  $j = 4$  and 10, computed from the model with the mean value trajectories obtained by averaging 200 separate simulations of the algorithm using the same input statistics which were used to generate Fig. 2. Also shown is the time-varying optimal value of  $k_n^{(x)}(i)$ , given by (2.12). Output MSE versus time obtained both by simulation and by the convergence model is virtually identical to that shown in Fig. 3 and is therefore omitted.

A comparison of Figs. 2 and 4 suggests that the speeds of convergence for both algorithms (2.5) and (2.9) are similar. Intuitively, this seems reasonable since in both cases the lattice predictor is attempting to orthogonalize the set of inputs driving the second set of coefficients. If we therefore assume that the backward error covariance matrix  $E[\epsilon_b(i|N)\epsilon_b^T(i|N)]$  is diagonal, and that  $E[x_i\epsilon_b(i|n)] \approx E[e_x(i|N)\epsilon_b(i|n)]$  (which

is exactly true if the coefficients are fixed at their optimal values), then (2.9) is equivalent to (2.5). Although this is not the case when the lattice prediction coefficients are adapting, the off-diagonal elements of the backward error covariance matrix will be converging towards zero, and hence, the effect of cross coupling between the taps in (2.15) should become negligible.

For both algorithms (2.5) and (2.9) the coefficient estimates shown in Figs. 2 and 4 contain significant bias. This is not surprising in view of the coefficient bias present in the lattice predictor. This bias is caused by filter coefficient variance, which alters the statistics driving other filter coefficients, and by correlations between filter coefficients and the input data. In the case of the second algorithm, (2.9), the estimate of  $k_n^{(x)}$  is unaffected by filter coefficients  $k_j^{(x)}$  and  $k_j$  for  $j > n$ . This is in contrast to the first algorithm, (2.5), in which  $f_{j|N}(i)$  depends upon *all* other filter coefficients. We would therefore suspect that the statistics which determine the values of  $f_{j|N}(i)$ ,  $1 \leq j \leq N$ , in (2.5) are more severely altered than when using (2.9). This would account for the fact that the coefficient estimates in Fig. 2 contain more bias than those in Fig. 4. This in turn implies that the asymptotic output MSE produced by the first algorithm is somewhat larger than that produced by the second. We have empirically observed, however, that this difference is slight.

A first-order analytic estimate of tap coefficient variance and output MSE using (2.5) or (2.9) can be obtained using the techniques in [1]. In particular, denoting  $E_\infty[X_i]$  as the asymptotic mean value of the sequence of random variables  $X_i$ , it is easily shown from (2.8) that

$$E_\infty[e_x^2(i|n+1)] \approx E_\infty[e_x^2(i|n)] - (k_{n,opt}^{(x)2} - \text{var } \infty k_n^{(x)}) \cdot E_\infty[e_b^2(i|n)] \quad (2.26)$$

where  $\text{var } \infty k_n^{(x)}(i) \equiv E_\infty[k_n^{(x)2}(i)] - \{E_\infty[k_n^{(x)}(i)]\}^2$ . Evaluation of  $\text{var } \infty k_n^{(x)}$ ,  $1 \leq n \leq N$ , can be accomplished in the same manner as the evaluation of  $\text{var } \infty k_n$  in [1]. Assuming  $E[e_b(i|n-1)e_b(i+m|n-1)] = E[e_x(i|n-1)e_b(i+m|n-1)] = 0$ ,  $m \neq 0$ , gives

$$\text{var } \infty k_n^{(x)} \approx \frac{\beta}{2 + \beta} \left[ \frac{E[e_x^2(i|n-1)]}{E[e_b^2(i|n-1)]} - k_{n,opt}^{(x)2} \right]. \quad (2.27)$$

Replacing  $k_n^{(x)}$  and  $k_{n,opt}^{(x)}$  by  $f_{n|N}$  and  $f_{n|N,opt}$ , respectively, gives the analogous results when using (2.5), assuming that the lattice predictor asymptotically eliminates cross coupling between the tap weights. In general, this assumption is not strictly true due to correlation effects and hence, small differences in coefficient variance and output MSE as produced by (2.5) and (2.9) will be observed.

### III. LS LATTICE SINGLE-STAGE CONVERGENCE ANALYSIS

The least squares lattice algorithm [12]-[16] adapts the PARCOR coefficients in a lattice structure so as to minimize the (exponentially weighted) sum of squares of output residuals  $e_f(i|n)$  and  $e_b(i|n)$ . It thus takes direct account of the nonstationarity of the input (although the model presented here assumes a stationary input). The algorithm is given as follows:

$$R_f(0|0) = R_b(0|0) = \delta > 0.$$

At each iteration  $i$ ,

$$e_f(i|0) = y_i, \quad e_b(i|0) = y_{i-1}, \quad \gamma(i|0) = 1$$

$$R_f(i|0) = wR_f(i-1|0) + y_i^2$$

$$e_f(i|n+1) = e_f(i|n) - k_{n+1}^{(b)}(i)e_b(i|n) \quad (3.1a)$$

$$e_b(i|n+1) = e_b(i-1|n) - k_{n+1}^{(f)}(i-1)e_f(i-1|n) \quad (3.1b)$$

$$R_b(i|n+1) = R_b(i-1|n) - \frac{\overline{k}_{n+1}^2(i-1)}{R_f(i-1|n)} \quad (3.2a)$$

$$R_f(i|n+1) = R_f(i|n) - \frac{\overline{k}_{n+1}^2(i)}{R_b(i|n)} \quad (3.2b)$$

$$\gamma(i|n+1) = \gamma(i|n) - \frac{e_b^2(i|n)}{R_b(i|n)} \quad (3.3)$$

$$\overline{k}_{n+1}(i) = w\overline{k}_{n+1}(i-1) + \frac{e_f(i|n)e_b(i|n)}{\gamma(i|n)} \quad (3.4)$$

$$R_f(i|n) = wR_f(i-1|n) + \frac{e_f^2(i|n)}{\gamma(i|n)} \quad (3.5a)$$

$$R_b(i|n) = wR_b(i-1|n) + \frac{e_b^2(i|n)}{\gamma(i|n)} \quad (3.5b)$$

where  $\gamma(i|n)$  has been interpreted as an optimal weighting factor [13],  $k_{n+1}^{(b)}(i) \equiv \overline{k}_{n+1}(i)/R_b(i|n)$  ("backward" PARCOR coefficient),  $k_{n+1}^{(f)}(i) \equiv \overline{k}_{n+1}(i)/R_f(i|n)$  ("forward" PARCOR coefficient), and in practice the update for  $R_b$  and  $R_f$  is given either by (3.2) or (3.5). The exponential weighting factor  $w$  is analogous to  $(1 - \beta)$  where  $\beta$  was the LMS adaptation constant used in [1].

The convergence properties of one stage of the adaptive lattice are first analyzed assuming the inputs to that stage are stationary. Taking expectations of (3.4) gives

$$E[\overline{k}_{n+1}(i)] = wE[\overline{k}_{n+1}(i-1)] + E\left[\frac{e_f(i|n)e_b(i|n)}{\gamma(i|n)}\right] \approx wE[\overline{k}_{n+1}(i-1)] + \frac{E[e_f(i|n)e_b(i|n)]}{E[\gamma(i|n)]}. \quad (3.6)$$

To evaluate  $E[\gamma(i|n)]$  we use (3.3) to write

$$E[\gamma(i|n)] \approx E[\gamma(i|n-1)] - \frac{E[e_b^2(i|n-1)]}{E[R_b(i|n-1)]}. \quad (3.7)$$

By assumption the previous stages have converged, and hence,  $E[R_b(i|n-1)] = E_\infty[R_b(i|n-1)]$ . Taking asymptotic expectations of (3.5b),

$$E_\infty[R_b(i|n-1)] \approx \frac{1}{1-w} \frac{E[e_b^2(i|n-1)]}{E[\gamma(i|n-1)]},$$

or

$$\frac{E[e_b^2(i|n-1)]}{E[R_b(i|n-1)]} \approx (1-w)E[\gamma(i|n-1)]. \quad (3.8)$$

Substituting (3.8) into (3.7) gives

$$E[\gamma(i|n)] \approx wE[\gamma(i|n-1)]. \quad (3.9)$$

Using the initial condition  $\gamma(i|0) = 1$ , (3.9) can be rewritten as

$$E[\gamma(i|n)] \approx w^n. \quad (3.10)$$

Substituting into (3.6) gives

$$\begin{aligned} E[\bar{k}_{n+1}(i)] &\approx wE[\bar{k}_{n+1}(i-1)] + w^{-n}E[e_f(i/n)e_b(i/n)] \\ &= w^i \left\{ k_n(0) - \frac{w^{-n}}{1-w} E[e_f(i/n)e_b(i/n)] \right\} \\ &\quad + \frac{w^{-n}}{1-w} E[e_f(i/n)e_b(i/n)]. \end{aligned} \quad (3.11)$$

Similarly,

$$\begin{aligned} E[R_b(i/n)] &\approx wE[R_b(i-1/n)] + w^{-n}E[e_b^2(i/n)] \\ &= w^i \left\{ R_b(0/n) - \frac{w^{-n}}{1-w} E[e_b^2(i/n)] \right\} \\ &\quad + \frac{w^{-n}}{1-w} E[e_b^2(i/n)]. \end{aligned} \quad (3.12)$$

If we assume

$$E[k_{n+1}^{(b)}(i)] \approx \frac{E[\bar{k}_{n+1}(i)]}{E[R_b(i/n)]}$$

as in [1], we can compute the time  $\tau_n^{(b)}$  it takes  $E[k_{n+1}^{(b)}(i)]$  to converge to  $k_n^{(b)}(0) + g(E_\infty[k_n(i)] - k_n^{(b)}(0))$  where  $0 < g < 1$ :

$$\tau_n^{(b)} \approx \frac{1}{\ln w} \ln \left[ w^n(1-w) \frac{g}{1-g} \frac{R_b(0/n)}{E[e_b^2(i/n)]} + 1 \right]. \quad (3.13)$$

Assuming  $E[e_b^2(i/n)] = E[e_f^2(i/n)]$ , the same formula applies to  $k_{n+1}^{(f)}(i)$ . The difference between (3.13) and the analogous formula for the LMS lattice derived in [1] is the added factor  $w^n$  in brackets. In general  $w$  is slightly less than unity so that  $\tau_n^{(b)}$  will approximately equal the analogous LMS "time constant." This is not surprising since the fundamental difference between the least squares lattice predictor and the LMS lattice predictor given by (2.1) and (2.2) is the added least-squares "likelihood variable"  $\gamma$ . If we assume  $\gamma(i/n)$  is constant as is approximately the case if the previous stages have converged, both algorithms are basically the same (see [3] for a more detailed comparison of LS and LMS lattice algorithms).

Single-stage output MSE is now approximated assuming both inputs are stationary. Our interest is in the asymptotic values of  $E[e_f^2(i/n+1)]$ , the signal in the lattice at time  $i$ , and  $E[e_f'^2(i/n+1)]$ , which denotes the causal mean squared prediction error. In particular,  $e_f(i/n)$  uses a regression coefficient based upon the data  $y_i, y_{i-1}, \dots, y_0$ . As the exponential weighting factor decreases, the more recent values of  $e_f^2(i/n+1)$  are weighted more heavily, and hence  $E_\infty[e_f^2(i/n+1)]$  decreases. For example, as  $w \rightarrow 0$ ,

$$\frac{\bar{k}_{n+1}(i)}{R_b(i/n)} \rightarrow \frac{e_f(i/n)e_b(i/n)}{e_b^2(i/n)},$$

and from (3.1a),  $E_\infty[e_f^2(i/n+1)] \rightarrow 0$ . In contrast,  $e_f'(i/n+1)$  uses regression coefficients calculated at time  $i-1$ , and hence, is the causal least squares prediction error obtained by estimating the value of  $y_i$  given  $y_{i-1}, y_{i-2}, \dots, y_0$ . As  $w$  decreases, the second-order statistics used to predict  $y_i$  are effectively

estimated from fewer samples, and hence,  $E_\infty[e_f'^2(i/n+1)]$  increases. It can be shown that [2]

$$e_f'(i/n) = \frac{e_f(i/n)}{\gamma(i/n)} \quad (3.14)$$

so that

$$E_\infty[e_f'^2(i/n)] \approx \frac{E_\infty[e_f^2(i/n)]}{E_\infty[\gamma^2(i/n)]} \approx \frac{1}{w^{2n}} E_\infty[e_f^2(i/n)]. \quad (3.15)$$

To approximate  $E_\infty[e_f^2(i/n+1)]$ , we take asymptotic expected values of (3.5a) to get

$$E_\infty[R_f(i/n)] = \frac{1}{1-w} E_\infty \left[ \frac{e_f^2(i/n)}{\gamma(i/n)} \right] \approx \frac{1}{w^n(1-w)} E[e_f^2(i/n)]. \quad (3.16a)$$

Similarly,

$$E_\infty[R_b(i/n)] \approx \frac{1}{w^n(1-w)} E[e_b^2(i/n)]. \quad (3.16b)$$

We now use (3.2b) to write

$$E_\infty[R_f(i/n+1)] \approx E_\infty[R_f(i/n)] - \frac{E_\infty[\bar{k}_{n+1}^2(i)]}{E_\infty[R_b(i/n)]}. \quad (3.17)$$

Substituting (3.16) into (3.17) and assuming  $E[e_f^2(i/n)] = E[e_b^2(i/n)]$ ,

$$\begin{aligned} E_\infty[e_f^2(i/n+1)] &\approx w \left\{ 1 - w^{2n}(1-w)^2 \frac{E_\infty[\bar{k}_{n+1}^2(i)]}{(E[e_b^2(i/n)])^2} \right\} \\ &\quad \cdot E[e_f^2(i/n)]. \end{aligned} \quad (3.18)$$

Squaring (3.4) and taking asymptotic expectations assuming  $\bar{k}_{n+1}(i)$  is uncorrelated with

$$\frac{e_f(i/n)e_b(i/n)}{\gamma(i/n)}$$

gives

$$\begin{aligned} E_\infty[\bar{k}_{n+1}^2(i+1)] &\approx w^2 E_\infty[\bar{k}_{n+1}^2(i)] + 2w E_\infty[\bar{k}_{n+1}(i)] \\ &\quad \cdot \frac{E[e_f(i/n)e_b(i/n)]}{E[\gamma(i/n)]} + \frac{E[e_f^2(i/n)e_b^2(i/n)]}{E[\gamma^2(i/n)]}. \end{aligned}$$

Substituting

$$E_\infty[\bar{k}_{n+1}(i)] \approx \frac{1}{w^n(1-w)} E[e_f(i/n)e_b(i/n)]$$

and

$$E_\infty[\gamma^2(i/n)] \approx w^{2n}$$

gives

$$\begin{aligned}
 E_\infty[\bar{k}_{n+1}^2(i)] &\approx \frac{2w}{w^{2n}(1-w)(1-w^2)} \\
 &\cdot \{E[e_f(i|n)e_b(i|n)]\}^2 \\
 &+ \frac{1}{w^{2n}(1-w^2)} E[e_f^2(i|n)e_b^2(i|n)].
 \end{aligned} \quad (3.19)$$

Substituting (3.19) into (3.18) and again assuming  $E[e_b^2(i|n)] = E[e_f^2(i|n)]$  gives

$$\begin{aligned}
 E_\infty[e_f^2(i|n+1)] &\approx w \left\{ 1 \frac{2w}{1+w} k_{n+1, opt}^2 - \left( \frac{1-w}{1+w} \right) \right. \\
 &\left. \frac{E[e_f^2(i|n)e_b^2(i|n)]}{(E[e_f^2(i|n)])^2} \right\} E[e_f^2(i|n)].
 \end{aligned} \quad (3.20)$$

If we further assume  $e_f(i|n)$  and  $e_b(i|n)$  to be jointly Gaussian, then the fourth-order statistics in (3.20) can be evaluated as follows,

$$E_\infty[e_f^2(i|n+1)] \approx \frac{2w}{1+w} (w - k_{n+1, opt}^2) E_\infty[e_f^2(i|n)]. \quad (3.21)$$

We have found (3.21) to be reasonably accurate for practical values of  $w$  using a Gaussian filter input sequence. If  $w$  is very small (i.e., less than 0.8), however,  $\bar{k}_{n+1}(i)$ ,  $R_f(i|n)$ , and  $R_b(i|n)$  start to track the immediate variations in  $e_f$  and  $e_b$  and hence the uncorrelated assumptions break down. Substitution of (3.21) into (3.15) yields the corresponding expression for the causal output MSE.

Also of interest is the asymptotic variance of the filter coefficients. This is approximated as

$$\text{var}_\infty k_{n+1}^{(b)} \approx \frac{E_\infty[\bar{k}_{n+1}^2(i)]}{E_\infty[R_b^2(i|n)]} - k_{n+1, opt}^2. \quad (3.22)$$

Squaring (3.5b) and taking asymptotic expectations assuming  $R_b(i|n)$  is uncorrelated with  $e_b^2(i|n)/\gamma(i|n)$  gives

$$\begin{aligned}
 E_\infty[R_b^2(i|n)] &\approx \frac{1}{1-w^2} \left\{ 2wE_\infty[R_b(i|n)] \frac{E[e_b^2(i|n)]}{E[\gamma(i|n)]} \right. \\
 &\left. + \frac{E[e_b^4(i|n)]}{E[\gamma^2(i|n)]} \right\}.
 \end{aligned}$$

Using (3.16b), (3.10), and assuming as a first approximation that  $e_b(i|n)$  is Gaussian,

$$E_\infty[R_b^2(i|n)] \approx \frac{3-w}{w^{2n}(1-w)(1-w^2)} \{E[e_b^2(i|n)]\}^2. \quad (3.23)$$

Similarly, assuming  $e_f$  and  $e_b$  are jointly Gaussian in (3.19),

$$\begin{aligned}
 E_\infty[\bar{k}_{n+1}^2(i)] &\approx \frac{1}{w^{2n}(1-w^2)} \left[ \frac{2}{1-w} k_{n+1, opt}^2 + 1 \right] \\
 &\cdot \{E[e_f^2(i|n)]\}^2.
 \end{aligned} \quad (3.24)$$

Substituting (3.24) and (3.23) into (3.22) gives

$$\text{var}_\infty k_{n+1}^{(b)} \approx \frac{1-w}{3-w} (1 - k_{n+1, opt}^2). \quad (3.25)$$

Notice that since  $w$  corresponds to  $(1-\beta)$  in [1], this formula is identical to the formula for coefficient variance using an LMS algorithm in [1]. Also, if the assumption that  $R_b(i|n)$  is uncorrelated with  $e_b^2(i|n)/\gamma(i|n)$  and that  $\bar{k}_{n+1}(i)$  is uncorrelated with  $e_f(i|n)e_b(i|n)/\gamma(i|n)$  is disregarded, the procedure outlined in [2] can be used to calculate a more accurate expression for  $\text{var}_\infty k_{n+1}^{(b)}$  analogous to (3.14) in [1]. The correspondence between simulated output MSE and coefficient variance using Gaussian inputs and calculated MSE and coefficient variance using (3.21), (3.15), and (3.25) has been found to be fairly close [2].

This completes the first-order description of single-stage convergence time and output MSE for the LS lattice. To summarize, for one stage of the lattice, the mean trajectory of the filter coefficients, and asymptotic variance of the filter coefficients and its affect upon the output MSE have been approximated assuming the inputs to the stage considered are stationary. Approximations used are 1) the filter coefficients are independent of the input data sequence, which is analogous to the independence assumption used to analyze the LMS transversal algorithm [19], 2) the average

$$E \begin{bmatrix} X_i \\ Y_i \end{bmatrix}$$

can be replaced by

$$\frac{E[X_i]}{E[Y_i]}$$

for the cases considered, and 3) the error sequences  $e_b$  and  $e_f$  are jointly Gaussian. The second assumption is reasonable since the random variables  $X_i$  and  $Y_i$  are, for the cases considered, time averages of second-order input statistics. The variances of  $X_i$  and  $Y_i$  therefore become relatively small as  $i$  increases. The third assumption was used to approximate fourth-order statistics by second-order statistics.

A comparison of the formulas in this section with their counterparts in [1] illustrates the similarities between the LS lattice and LMS lattice. In particular, calculated single-stage time constants, coefficient variance, and output MSE for both the LS and LMS lattice algorithms are nearly identical. The first-order techniques used in this paper therefore do not distinguish between the LS and LMS lattice algorithms. This is because the difference in performance between the two algorithms is primarily due to the added optimal weighting factors  $\gamma(i|n)$ ,  $1 \leq n \leq N$ , which tend to remain constant for stationary Gaussian inputs. The results in this section are consistent with the comparative LMS-LS simulations in [3] which indicate that the difference in performance between the two algorithms with Gaussian noise inputs is slight.

#### IV. MULTISTAGE LS LATTICE CONVERGENCE MODELS

Having described the convergence properties of one stage of the LS lattice, we now consider the effects which adapting

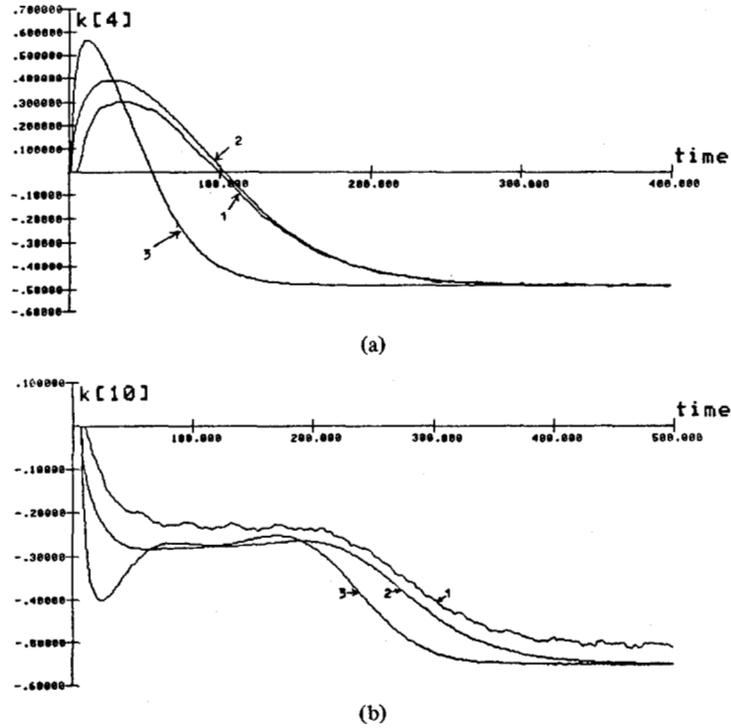


Fig. 5. Mean value trajectories of least squares backward PARCOR coefficients by 1) simulation and 2) from the convergence model. Curve 3 shows the trajectory of  $k_{n,opt}^{(b)}(i)$  computed by the model.

stages have on successive stages. In this section we therefore attempt to extend the convergence model in [1] for the LMS lattice to the least squares case.

#### A. LS Lattice Predictor Model

The convergence model for the LS lattice is again obtained by ignoring the effect of statistical fluctuations of the coefficients  $k_n^{(b)}(i)$  and  $k_n^{(f)}(i)$ ,  $1 \leq n \leq N$ . Given second-order information about the input sequence, we can then replace  $k_{n+1}^{(b)}(i)$  and  $k_{n+1}^{(f)}(i)$  by their mean value trajectories and obtain a set of simple deterministic iterative equations. In particular, noting that

$$e_f(i|n) = \bar{f}^T(i|n) \bar{y}_{i|n} \quad (4.1a)$$

and

$$e_b(i|n) = \bar{b}^T(i|n) \bar{y}_{i|n} \quad (4.1b)$$

where

$$\bar{y}_{i|n} = [y_i, y_{i-1}, \dots, y_{i-n}, \dots, y_{i-N+1}], \quad (4.2)$$

and  $\bar{f}(i|n)$  and  $\bar{b}(i|n)$  are defined by (2.19) and (2.22), respectively, the order recursions (3.1) can be rewritten with  $k_{n+1}^{(b)}$  and  $k_{n+1}^{(f)}$  replaced by their mean values as follows:

$$\bar{f}(i|n+1) = \bar{f}(i|n) - E[k_{n+1}^{(b)}(i)] \bar{b}(i|n) \quad (4.3a)$$

and

$$\bar{b}(i|n+1) = z^{-1} \bar{b}(i-1|n) - E[k_{n+1}^{(f)}(i)] z^{-1} \bar{f}(i-1|n). \quad (4.3b)$$

$z^{-1} \bar{b}(i-1|n-1)$  represents  $\bar{b}(i-1|n-1)$  shifted "down" one element, i.e.,

$$[z^{-1} \bar{b}(i-1|n)]_j = [\bar{b}(i-1|n)]_{j-1} \quad \text{for } 2 \leq j \leq N+2$$

and  $[z^{-1} \bar{b}(i-1|n)]_1 = 0$ .

In order to compute  $E[k_n^{(f)}(i)]$  and  $E[k_n^{(b)}(i)]$ , we assume

$$E[k_{n+1}^{(f)}(i)] \approx \frac{E[\bar{k}_{n+1}(i)]}{E[R_f(i|n)]}$$

and

$$E[k_{n+1}^{(b)}(i)] = \frac{E[\bar{k}_{n+1}(i)]}{E[R_b(i|n)]}. \quad (4.4)$$

Taking expected values of both sides of (3.3), (3.4), (3.5a), and (3.5b), using the second approximation discussed at the end of the last section, the values of  $E[\bar{k}_{n+1}(i)]$ ,  $E[R_f(i|n)]$ , and  $E[R_b(i|n)]$  can be computed at each iteration  $i$ . Second-order statistics are approximated as follows:

$$E[e_f(i|n)e_b(i|n)] \approx \bar{f}^T(i|n) E[\bar{y}_{i|n} \bar{y}_{i|n}^T] \bar{b}(i|n), \quad (4.5)$$

$$E[e_f^2(i|n)] \approx \bar{f}^T(i|n) E[\bar{y}_{i|n} \bar{y}_{i|n}^T] \bar{f}(i|n), \quad (4.6)$$

and

$$E[e_b^2(i|n)] \approx \bar{b}^T(i|n) E[\bar{y}_{i|n} \bar{y}_{i|n}^T] \bar{b}(i|n). \quad (4.7)$$

Equations (4.3)-(4.7) constitute the convergence model for the LS lattice filter. Fig. 5 compares the mean value trajectories of  $k_n^{(b)}(i)$ ,  $n=4$  and 10, obtained from the model and by simulation for the same input statistics used to illustrate the

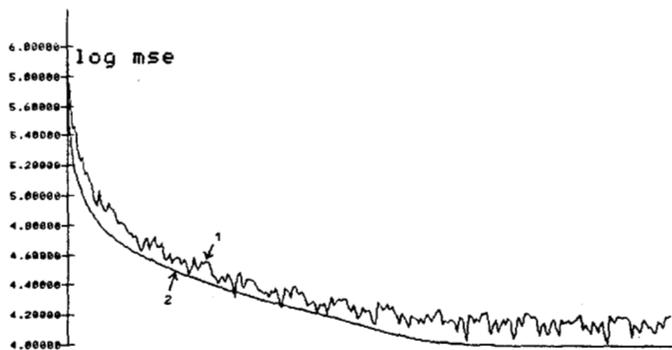


Fig. 6. Output MSE of a least-squares lattice predictor by 1) simulation and 2) from the convergence model.

accuracy of the LMS model in [1]. Also shown are the trajectories of

$$k_{n,opt}^{(b)} = \frac{E[e_f(i|n)e_b(i|n)]}{E[e_b^2(i|n)]}$$

Fig. 6 compares output MSE as generated by the model and by simulation. The adaptation constant was set to  $w = 1 - \beta$ , where  $\beta$  was the adaptation constant used for the simulations in [1]. The speeds of convergence for both the LMS and LS algorithms are therefore similar. A comparison of Figs. 5 and 6 with Figs. 3 and 4 in [1] indicates that the LS model for convergence is more accurate than the LMS model; however, additional results in [3] indicate that by using a somewhat different LMS algorithm from that simulated in [1], the LMS curves can be made nearly identical to the LS curves.

### B. LS Lattice Joint Process Estimator Model

In analogy with the LMS lattice, the convergence model for the LS lattice predictor is easily extended to the LS lattice joint process estimator. In addition to the predictor recursions in Section III, the LS lattice joint process estimator [15] uses the additional recursions

$$e_x(i|n+1) = e_x(i|n) - k_{n+1}^{(x)}(i)e_b(i|n) \quad (4.8)$$

and

$$\bar{k}_{n+1}^{(x)}(i+1) = w\bar{k}_{n+1}^{(x)}(i) + \frac{e_x(i|n)e_b(i|n)}{\gamma(i|n)} \quad (4.9)$$

where

$$k_{n+1}^{(x)}(i) = \frac{\bar{k}_{n+1}^{(x)}(i)}{R_b(i|n)}$$

As discussed in Section II,  $e_x(i|n)$  can be represented by the vector  $\bar{f}(i|n)$  defined by (2.19) and (4.8) can be rewritten with  $k_{n+1}^{(x)}(i)$  replaced by its mean value as (2.21). We then approximate

$$E[k_{n+1}^{(x)}(i)] \approx \frac{E[\bar{k}_{n+1}^{(x)}(i)]}{E[R_b(i|n)]} \quad (4.10)$$

and

$$E[\bar{k}_{n+1}^{(x)}(i+1)] \approx wE[\bar{k}_{n+1}^{(x)}(i)] + \frac{E[e_x(i|n)e_b(i|n)]}{E[\gamma(i|n)]} \quad (4.11)$$

where  $E[e_x(i|n)e_b(i|n)]$  is given by (2.25). Output MSE as a function of time is given by (2.24).

The LS lattice predictor model in Section IV-A combined with (4.10) and (4.11) completes the LS lattice joint process estimator model. Plots of mean coefficient trajectories and output MSE are nearly identical to the LMS cases discussed earlier and are therefore omitted.

### V. "FAST" KALMAN CONVERGENCE MODEL

The "fast" Kalman algorithm [9] is a computationally efficient version of the Kalman-Godard or least squares transversal algorithm presented in [8]. Both the "fast" Kalman and least-squares lattice algorithms minimize the (weighted) sum of the squares of the prediction residuals, and hence, in this sense are equivalent. The lattice structure, however, generates all of the least squares backward and forward residuals of orders one through  $N$  in addition to possessing other desirable characteristics [5]. On the other hand the "fast" Kalman algorithm requires somewhat less computation than the least squares lattice algorithm [16], [18]. Because the LS lattice and "fast" Kalman algorithms are equivalent (aside from finite word length effects), if output MSE is the only item of interest, the convergence model for the LS lattice algorithm would suffice for both algorithms. In addition, we point out that the formulas for asymptotic output MSE given in Section III should also apply to the LS transversal filter.

The algorithm is stated as follows:

$$e'_f(i) = y_i - f^T(i-1)y_{i-1|N} \quad (5.1)$$

$$f(i) = f(i-1) + e'_f(i)g(i) \quad (5.2)$$

$$e_f(i) = y_i - f^T(i)y_{i-1|N} \quad (5.3)$$

$$R_f(i) = wR_f(i-1) + e_f(i)e'_f(i) \quad (5.4)$$

$$\begin{bmatrix} \bar{g}_1(i+1) \\ \bar{g}_{2,N+1}(i+1) \end{bmatrix} = \begin{bmatrix} \frac{e_f(i)}{R_f(i)} \\ g(i) - \frac{e_f(i)}{R_f(i)}f(i) \end{bmatrix} \quad (5.5)$$

$$e'_b(i+1) = y_{i-N} - b^T(i)y_{i|N} \quad (5.6)$$

$$b(i+1) = \frac{b(i) + e'_b(i+1)\bar{g}_{1,N}(i+1)}{1 - e'_b(i+1)\bar{g}_{N+1}(i+1)} \quad (5.7)$$

$$g(i+1) = \bar{g}_{1,N}(i+1) + \bar{g}_{N+1}(i+1)b(i+1) \quad (5.8)$$

where

$$f(i) = [f_{1|N}(i), \dots, f_{N|N}(i)]; \quad (5.9a)$$

$$b(i) = [b_{1|N}(i), \dots, b_{N|N}(i)], \quad (5.9b)$$

$$y_{i|N} = [y_i, \dots, y_{i-N+1}]. \quad (5.10)$$

$g(i)$  is the Kalman gain, i.e.,

$$g(i) \equiv \Phi_{i|N}^{-1} \mathcal{Y}_{i-1|N} \quad (5.11)$$

and  $\Phi_{i|N}$  is the sample covariance matrix,

$$\Phi_{i|N} \equiv \sum_{j=1}^i w^{i-j} \mathcal{Y}_{j-1|N} \mathcal{Y}_{j-1|N}^T + w^i \delta \quad (5.12)$$

for some small initial value  $\delta$ .  $\bar{g}_{m,n}(i)$  denotes the vector formed by the  $m$ th through the  $n$ th element of the  $(N+1)$ -dimensional "extended Kalman gain vector"  $\bar{g}(i)$  given by

$$\bar{g}(i) = \Phi_{i+1|N+1}^{-1} \mathcal{Y}_{i|N+1}. \quad (5.13)$$

The residuals  $e_f'$  and  $e_b'$  are again the causal prediction errors as discussed in Section III. Note that (5.1) and (5.2) alone constitute the least squares transversal algorithm. Equations (5.3)-(5.8) merely evaluate  $g(i)$  given by (5.11).

To obtain a "fast" Kalman convergence model, we again represent the residuals  $e_f(i)$  and  $e_b(i)$  by their respective prediction coefficient vectors  $\bar{f}(i)$  and  $\bar{b}(i)$  defined as

$$\bar{f}^T(i) = [1 | -f^T(i) | 0] \quad (5.14a)$$

and

$$\bar{b}^T(i) = [0 | -b^T(i) | 1] \quad (5.14b)$$

where  $f(i)$  and  $b(i)$  are the forward and backward tap vectors defined by (5.9). By definition,  $e_f'(i)$  and  $e_b'(i)$  are represented by the vectors  $\bar{f}(i-1)$  and  $\bar{b}(i-1)$ . Since we assume the tap weights are evolving deterministically, the matrices  $\Phi_{i|N}^{-1}$  and  $\Phi_{i+1|N+1}^{-1}$  can also be regarded as deterministic. From (5.11) and (5.13) each element of  $g(i)$  and  $\bar{g}(i)$  is a linear combination of  $\mathcal{Y}_{i-1}, \mathcal{Y}_{i-2}, \dots, \mathcal{Y}_{i-N}$ , and  $\mathcal{Y}_i, \mathcal{Y}_{i-1}, \dots, \mathcal{Y}_{i-N}$ , respectively. We therefore represent  $[g(i)]_j$ ,  $1 \leq j \leq N$ , and  $[\bar{g}(i+1)]_j$ ,  $1 \leq j \leq N+1$ , by their respective coefficient vectors

$$g_j^T(i) = [0, g_{j1}, g_{j2}, \dots, g_{jN}, 0] \quad (5.15)$$

and

$$\bar{g}_j^T(i) = [\bar{g}_{j1}, \bar{g}_{j2}, \dots, \bar{g}_{j,N+1}, 0]. \quad (5.16)$$

The vectors  $g_j(i)$  and  $\bar{g}_j(i)$ , therefore, approximate the  $j$ th row of the matrices  $E[\Phi_{i|N}^{-1}]$  and  $E[\Phi_{i+1|N+1}^{-1}]$ , respectively. (The last "0" element is added to keep the dimension of  $g_j$  and  $\bar{g}_j$  the same as the dimension of  $\bar{f}$  and  $\bar{b}$ .)

The mean-forward tap update can be evaluated from (5.2) using second-order statistics as follows:

$$[f(i)]_j = [f(i-1)]_j + \sum_{l=1}^{N+1} \sum_{m=1}^N [f(i-1)]_l g_{jm}(i) R_{l-m-1}, \quad 1 \leq j \leq N. \quad (5.17)$$

The mean-covariance update is similarly given by

$$E[R_f(i)] \approx wE[R_f(i-1)] + \sum_{l=1}^{N+1} \sum_{m=1}^{N+1} [f(i)]_l [f(i-1)]_m R_{l-m}. \quad (5.18)$$

The coefficient vectors  $\bar{g}_j(i+1)$  are then updated by using (5.5) to write

$$\bar{g}_1(i+1) = \frac{\bar{f}(i)}{E[R_f(i)]} \quad (5.19a)$$

and

$$\bar{g}_j(i+1) = g_{j-1}(i) - [f(i)]_{j-1} \bar{g}_1(i+1) \quad (5.19b)$$

for  $2 \leq j \leq N+1$ .

In order to compute  $\bar{b}(i)$ , we use the fact that [2]

$$[\bar{g}(i+1)]_{N+1} = \frac{e_b(i+1)}{R_b(i+1)}, \quad (5.20)$$

and therefore

$$\bar{g}_{N+1}(i+1) = \frac{[z\bar{b}(i+1)]}{E[R_b(i+1)]} \quad (5.21)$$

where  $z\bar{b}(i+1)$  denotes  $\bar{b}(i+1)$  shifted "up" one element in analogy to the notation used in Section IV-A. Since  $[z\bar{b}(i+1)]_{N+1} = 1$ , it follows that

$$\bar{g}_{N+1,N+1}(i+1) = \frac{1}{E[R_b(i+1)]} \quad (5.22)$$

and

$$z\bar{b}(i+1) = \frac{\bar{g}_{N+1}(i+1)}{\bar{g}_{N+1,N+1}(i+1)}. \quad (5.23)$$

Finally, to update the vectors  $g_j$ ,  $1 \leq j \leq N$ , we use (5.8) to write

$$g_j(i+1) = \bar{g}_j(i+1) + [b(i+1)]_j \bar{g}_{N+1}(i+1). \quad (5.24)$$

Notice that since  $g_j(i+1)$  is a linear combination of  $\mathcal{Y}_i, \mathcal{Y}_{i-1}, \dots, \mathcal{Y}_{i-N+1}$ , we expect that  $[g_j(i+1)]_{N+1} = 0$ . This can be proven using (5.24) and (5.22). At the  $(i+1)$ st iteration of the algorithm,  $z^{-1}g_j(i+1)$  must be used in (5.17).

A convergence model for the least squares transversal filter is therefore given by (5.17), (5.18), (5.19), (5.23), and (5.24). Fig. 7 shows mean value trajectories of the tap weights  $f_{j|10}(i)$ ,  $j=4$  and  $10$ , obtained from the model and by simulation for the same input statistics used to illustrate the accuracy of the LS lattice model in Section IV-A. The adaptation constant  $w$  was also the same as that used in Section IV-A. Output MSE generated by the model and by simulation is nearly identical to the analogous LS lattice curves in Fig. 6 and is therefore not shown. Fig. 7 would seem to indicate that the model gives a reasonably accurate description of the convergence process.

## VI. CONCLUSIONS

This paper has discussed the convergence properties of the LMS and LS lattice joint process estimators, LS lattice predictor, and "fast" Kalman algorithm. Each case involved an extension of the techniques used in [1] to analyze the LMS lattice filter. The convergence models for both LMS lattice joint process estimation algorithms yield similar accuracy to that exhibited by the model presented in [1]. Results for the LMS lattice and LS lattice are similar, indicating that both algorithms exhibit similar performance in stationary environments. The behavior of LS and LMS lattice algorithms in non-

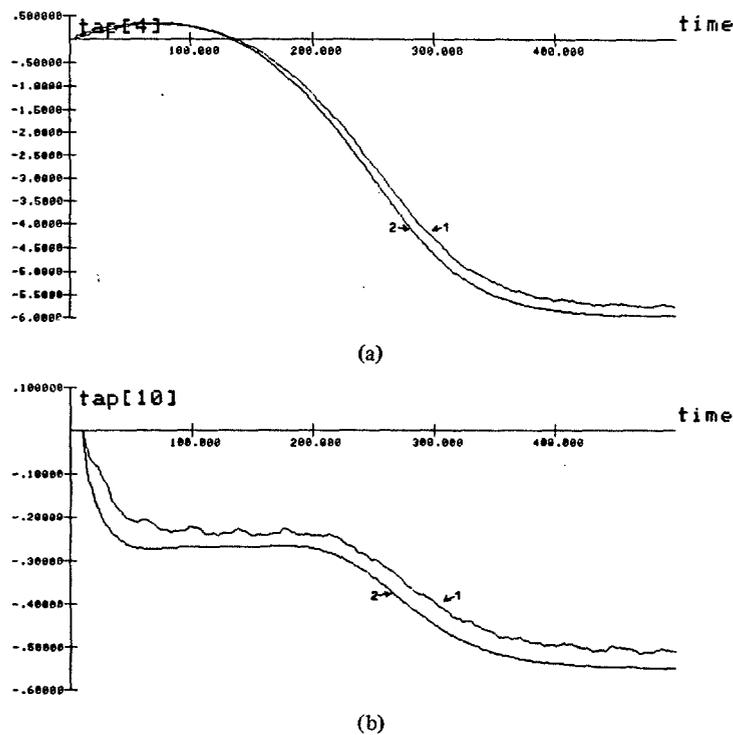


Fig. 7. Mean value trajectories of tap weights 4 and 10 in a "fast" Kalman tenth-order predictor by 1) simulation and 2) from the convergence model.

stationary environments is an interesting topic for future investigation.

ACKNOWLEDGMENT

The author wishes to thank Prof. D. G. Messerschmitt for his valuable guidance and suggestions concerning this work.

REFERENCES

[1] 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.  
 [2] M. L. Honig, "Performance of FIR adaptive filters using recursive algorithms," Ph.D. dissertation, Univ. California, Berkeley, Apr. 1981.  
 [3] M. L. Honig and D. G. Messerschmitt, "Comparison of LS and LMS lattice predictor algorithms using two performance criteria," submitted to *IEEE Trans. Acoust., Speech, Signal Processing*.  
 [4] L. J. Griffiths, "A continuously-adaptive filter implemented as a lattice structure," in *Proc. 1977 IEEE ICASSP*, Hartford, CT, May 1977, pp. 683-686.  
 [5] J. Makhoul, "A class of all-zero lattice digital filters: Properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 304-314, Aug. 1978.  
 [6] J. Makhoul and R. Viswanathan, "Adaptive lattice methods for linear prediction," in *Proc. 1978 IEEE ICASSP*, Tulsa, OK, pp. 83-86.  
 [7] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899-905, June 1979.  
 [8] D. Godard, "Channel equalization using Kalman filter for fast data transmission," *IBM J. Res. Develop.*, pp. 267-273, May 1974.  
 [9] 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.  
 [10] E. H. Satorius and M. J. Shensa, "On the application of recursive least squares methods to adaptive processing," presented at Int. Workshop Appl. Adaptive Contr., Yale Univ., New Haven, CT, Aug. 1979.  
 [11] Lennart Ljung, Martin Morf, and David Falconer, "Fast calcula-

tion of gain matrices for recursive estimation schemes," *Int. J. Contr.*, vol. 27, no. 1, pp. 1-19, 1978.  
 [12] M. Morf, D. T. Lee, J. R. Nickolls, and A. Vieira, "A classification of algorithms for ARMA models and ladder realizations," in *Proc. 1977 IEEE Conf. Acoust., Speech, Signal Processing*, Hartford, CT, April 1977, pp. 13-19.  
 [13] M. Morf and D. Lee, "Recursive least squares ladder forms for fast parameter tracking," in *Proc. 1978 IEEE Conf. Decision Contr.*, San Diego, CA, Jan. 12, 1979, pp. 1326-1367.  
 [14] M. Morf, A. Vieira, and D. T. Lee, "Ladder forms for identification and speech processing," in *Proc. 1977 IEEE Conf. Decision Contr.*, New Orleans, LA, Dec. 1977, pp. 1074-1078.  
 [15] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive square-root ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 627-641, June 1981.  
 [16] 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.  
 [17] R. S. Medaugh and L. J. Griffiths, "A comparison of two fast linear predictors," in *Proc. 1981 IEEE ICASSP*, Atlanta, GA, Apr. 1981.  
 [18] T. L. Lim and M. S. Mueller, "Rapid equalizer start-up using least squares algorithms," in *Proc. 1980 IEEE ICC*, Seattle, WA.  
 [19] J. E. Mazo, "On the independence theory of equalizer convergence," *Bell Syst. Tech. J.*, vol. 58, pp. 963-993, May-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.

Since July 1981, he has been a member of the Technical Staff at Bell Laboratories, Holmdel, NJ. His research interests include adaptive signal processing, computer networks, and data communications. Dr. Honig is a member of Tau Beta Pi and Phi Beta Kappa.