

# Bayesian Algorithms for Blind Equalization Using Parallel Adaptive Filtering

Ronald A. Iltis, *Senior Member, IEEE*, John J. Shynk, *Senior Member, IEEE*,  
and K. Giridhar, *Student Member, IEEE*

**Abstract**—A new blind equalization algorithm based on a suboptimum Bayesian symbol-by-symbol detector is presented. It is first shown that the maximum *a posteriori* (MAP) sequence probabilities can be approximated using the innovations likelihoods generated by a parallel bank of Kalman filters. These filters generate a set of channel estimates conditioned on the possible symbol subsequences contributing to the intersymbol interference. The conditional estimates and MAP symbol metrics are then combined using a suboptimum Bayesian formula. Two methods are considered to reduce the computational complexity of the algorithm. First, the technique of reduced-state sequence estimation is adopted to reduce the number of symbol subsequences considered in the channel estimation process and hence the number of parallel filters required. Second, it is shown that the Kalman filters can be replaced by simpler least-mean-square (LMS) adaptive filters. A computational complexity analysis of the LMS Bayesian equalizer demonstrates that its implementation in parallel programmable digital signal processing devices is feasible at 16 kbps. The performance of the resulting algorithms is evaluated through bit-error-rate simulations, which are compared to the performance bounds of the maximum-likelihood sequence estimator. It is shown that the Kalman filter and LMS-based algorithms achieve blind start-up and rapid convergence (typically within 200 iterations) for both BPSK and QPSK modulation formats.

**Keywords**—Blind equalization, Kalman filtering, Bayesian equalization, channel estimation

## I. INTRODUCTION

Blind equalization algorithms attempt to determine the transmitted symbol sequence in the presence of intersymbol interference (ISI) without prior knowledge of the channel impulse response. Most efforts in the development of blind equalizers have focused on "property restoral" algorithms in which a nonlinear function of the equalizer output is forced to a constant value [1],[2],[3],[4]. For example, in the constant modulus algorithm (CMA) [3] the error between the magnitude (modulus) of the equalizer output and a constant term is recursively minimized. The resulting gradient-descent method has a computational complexity similar to that of the LMS algorithm. The motivation for these methods is that by restoring the modulus of the

received signal, the channel impulse response is implicitly estimated and the ISI is removed. Since the cost functions of these algorithms are independent of the transmitted data sequence, they are capable of blind start-up.

While CMA-type algorithms have the advantage of computational simplicity, their convergence cannot be guaranteed. Specifically, it has been shown that CMA may converge to undesired local minima [5], and furthermore, its convergence rate is extremely slow compared to training-based equalizer algorithms. For wide bandwidth applications, such as digital microwave radio or high-speed switched telecommunication networks, property-restoral blind equalization algorithms may be the only alternative, despite the aforementioned drawbacks. However, with the advent of programmable digital signal processing (DSP) devices and special-purpose VLSI, more sophisticated blind equalization algorithms should be considered, especially for relatively narrowband systems such as HF digital radio and voiceband modems. The algorithms presented here, although seemingly complex, have a parallel structure that may prove to be well-suited for programmable DSP implementations and such narrowband communication systems.

In this paper, we present a new set of blind equalization algorithms that are approximations to the optimum MAP symbol-by-symbol detector for *a priori* unknown channels. The structure of the optimum MAP sequence estimator is first discussed, and it is shown that the sequence probabilities can be computed using the innovations derived from a bank of Kalman filters. The exact MAP sequence estimator requires a separate channel estimate for each possible symbol sequence and, thus, its computational complexity grows exponentially with time. To overcome this disadvantage, we develop a suboptimum Bayesian recursion for the MAP subsequence probabilities which maintains separate channel estimates for each of  $M^{N_b+1}$  subsequences, where  $M$  is the symbol alphabet size and  $N_b + 1$  is the estimated length of the channel impulse response. The symbol-by-symbol detector is then obtained by summing the appropriate MAP subsequence metrics, and the resulting algorithm consists of a bank of Kalman filters in which each filter maintains a channel estimate conditioned on one of the  $M^{N_b+1}$  subsequences.

In order to reduce the complexity of this method, the technique of reduced-state sequence estimation (RSSE) [6] is employed. In RSSE, the  $M^{N_b+1}$  subsequences are grouped into a coarser partition of  $N \leq M^{N_b+1}$  subsets, and the number of Kalman channel estimators is corresponding-

Paper approved by the Editor for Transmission Systems of the IEEE Communications Society. Manuscript received August 21, 1991; revised August 7, 1992 and February 25, 1993.

This work was sponsored in part by the University of California MICRO program, Sonatech, Inc., and Applied Signal Technology, Inc. This paper was presented in part at the 1991 Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA.

R. A. Iltis and J. J. Shynk are with the Dept. of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106.

K. Giridhar is with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305. He was with the Department of Electrical and Computer Engineering, University of California, Santa Barbara.

IEEE Log Number 9400999.

ly reduced. As a second simplification, the Kalman estimators are replaced by simpler LMS adaptive filters, thus yielding a set of algorithms which vary in complexity from a decision-feedback equalizer to the  $M^{N_b+1}$  parallel Kalman filter structure. A general analysis of the computational complexity of the LMS Bayesian equalizer is provided to evaluate its suitability for programmable DSP implementations.

The paper is organized as follows. The Bayesian recursion for the symbol probabilities is discussed in Section II and the suboptimum parallel Kalman estimator is derived in Section III. The RSSE version of the blind equalization algorithm is presented in Section IV and the LMS implementation is developed in Section V. The computational complexity analysis is presented in Section VI. Computer simulations and conclusions then follow in Sections VII and VIII.

## II. DEVELOPMENT OF THE OPTIMUM BLIND EQUALIZATION ALGORITHM

The following discrete-time channel and signal model is assumed

$$r(k) = \sum_{m=0}^{N_b} b_m(k)d(k-m) + n(k) \quad (1)$$

where  $r(k)$  is the output of a matched filter at time  $k$ ,  $d(k)$  is the current transmitted symbol, and  $\{b_m(k)\}$  represent the time-varying channel coefficients. For BPSK,  $d(k)$  is real, taking on the values  $\pm 1$ , and for QPSK,  $d(k)$  is complex with values  $\{\pm 1, \pm i\}$ . The channel coefficients  $\{b_m(k)\}$  represent the convolution of the actual ISI channel impulse response with that of a prewhitening filter, which is included to ensure that the additive noise samples  $\{n(k)\}$  are uncorrelated. These noise samples are drawn from a complex Gaussian distribution with zero mean and variance  $\sigma_n^2$ . Note that perfect synchronization, or equivalently, sampling of the matched filter output at the optimum times is implicitly assumed in (1). In the development of the Kalman filter channel estimator, it is assumed that the coefficients  $\{b_m(k)\}$  evolve according to the following complex Gaussian autoregressive (AR) process model:

$$\mathbf{b}(k+1) = \mathbf{F}\mathbf{b}(k) + \mathbf{w}(k) \quad (2)$$

where  $\mathbf{b}(k)$  is the coefficient (column) vector defined by

$$\mathbf{b}(k) = [b_0(k), b_1(k), \dots, b_{N_b}(k)]^T, \quad (3)$$

$\mathbf{F} \in \mathcal{C}^{(N_b+1) \times (N_b+1)}$  is the one-step transition matrix, and  $\mathbf{w}(k) \in \mathcal{C}^{N_b+1}$  is a zero-mean circular white Gaussian noise vector with covariance matrix  $\mathbf{Q} \in \mathcal{C}^{(N_b+1) \times (N_b+1)}$ .

For convenience in the derivation of the optimum MAP sequence estimator and blind equalizer, the following cumulative sequences are defined. The cumulative measurement sequence  $r^k$  represents the matched filter samples collected up to time  $k$ , and is given by

$$r^k = \{r(k), r(k-1), \dots, r(0)\}. \quad (4)$$

A cumulative data sequence is similarly defined as

$$d_i^k = \{d_i(k), d_i(k-1), \dots, d_i(0)\} \quad (5)$$

for the  $i^{\text{th}}$  of  $M^{k+1}$  possible sequences. Note that both of these sequences contain all samples. For the suboptimum Bayesian equalizer described in the next section, we consider only subsets of the full sequence  $d_i^k$ .

The optimum MAP sequence estimator is now reviewed for the assumed channel and signal models. The corresponding conditional probabilities can be written in the following recursive form:

$$p(d_i^k | r^k) = \frac{1}{c} p(r(k) | d_i^k, r^{k-1}) p(d_i^{k-1} | r^{k-1}), \quad (6)$$

for  $i = 1, 2, \dots, M^{k+1}$ , where  $p(d_i^k | r^k)$  represents the probability of the  $i^{\text{th}}$  possible data sequence given cumulative measurements  $r^k$ , and  $c$  is a normalization constant. Following the analysis in [7], it can be shown that the likelihood  $p(r(k) | d_i^k, r^{k-1})$  is determined by the Kalman filter estimate as follows:

$$p(r(k) | d_i^k, r^{k-1}) = \mathcal{N}(r(k); \hat{r}_i(k|k-1), \sigma_i^2(k|k-1)) \quad (7)$$

where  $\mathcal{N}(x; m_x, \sigma_x^2)$  denotes a univariate circular Gaussian density with mean  $m_x$  and variance  $\sigma_x^2$ . The estimated signal  $\hat{r}_i(k|k-1)$  is computed from the conditional channel estimates according to

$$\hat{r}_i(k|k-1) = \sum_{m=0}^{N_b} \hat{b}_{i,m}(k|k-1) d_i(k-m) \quad (8)$$

where  $\hat{b}_{i,m}(k|k-1)$  is exactly equal to the conditional mean of  $b_m(k)$ , under the AR process model in (2) when conditioned on data sequence  $d_i^k$ , i.e.,

$$\hat{b}_i(k|k-1) = E[\mathbf{b}(k) | d_i^k, r^{k-1}]. \quad (9)$$

This estimate is generated in a straightforward manner by Kalman filter equations similar to those to be discussed in Section III.

From (6) and (7), it is seen that the optimum MAP sequence estimator requires a bank of  $M^{k+1}$  Kalman filter channel estimators, each conditioned on a different sequence  $d_i^k$ . The MAP probabilities of the sequences are then obtained as a product of the corresponding likelihoods. Unfortunately, the number of channel estimators required increases exponentially with time so that the optimum sequence estimator is clearly impractical. In [7], a suboptimal approximation to the MAP estimator was proposed in which conditional channel estimates were propagated only for the survivor sequences in a Viterbi algorithm (VA). Thus, the number of channel estimators was fixed at the number of states in the trellis, which in turn was determined by the length of the channel impulse response. While the modified VA is computationally feasible for some low-rate data applications, we seek a further simplification to the optimum MAP sequence estimator discussed

here that does not require accumulation of the survivor sequences. The resulting blind equalization algorithm, while approximating the MAP estimator, also resembles a parallel decision-feedback equalizer structure, and hence is more computationally attractive.

### III. BLIND EQUALIZATION USING PARALLEL KALMAN CHANNEL ESTIMATORS

The suboptimum Bayesian equalizer and symbol-by-symbol detector is now derived. Observe that the probability density function  $p(\mathbf{b}(k)|d_i^k, r^{k-1})$  is exactly Gaussian with a mean and covariance determined by a Kalman channel estimator conditioned on the *entire* sequence  $d_i^k$ . Consider conditioning instead on the following *subsequence*:

$$d_i^{k, N_b} = \{d_i(k), d_i(k-1), \dots, d_i(k-N_b)\}, \quad (10)$$

which has the same number of terms as the channel impulse response. The corresponding density function  $p(\mathbf{b}(k)|d_i^{k, N_b}, r^{k-1})$  is not a Gaussian distribution, but is instead a weighted *sum* of Gaussian terms, as follows:

$$p(\mathbf{b}(k)|d_i^{k, N_b}, r^{k-1}) = \sum_{\{j: d_j^{k, N_b} = d_i^{k, N_b}\}} p(\mathbf{b}(k)|d_j^k, r^{k-1}) \frac{p(d_j^k | r^{k-1})}{p(d_i^{k, N_b} | r^{k-1})} \quad (11)$$

where the weights are given by the MAP sequence probabilities. In order to simplify the recursion in (6), we will condition on the subsequence  $d_i^{k, N_b}$  and approximate the above density for  $\mathbf{b}(k)$  with the following unimodal function:

$$p(\mathbf{b}(k)|d_i^{k, N_b}, r^{k-1}) \approx \mathcal{N}(\mathbf{b}(k); \hat{\mathbf{b}}_i(k|k-1), \mathbf{P}_i(k|k-1)) \quad (12)$$

where  $\mathcal{N}(\mathbf{x}; \mathbf{m}_\mathbf{x}, \mathbf{P}_\mathbf{x})$  represents a circular multivariate Gaussian density with mean vector  $\mathbf{m}_\mathbf{x}$  and covariance matrix  $\mathbf{P}_\mathbf{x}$ . It is next shown that the approximation in (12) yields a blind equalizer based on a parallel Kalman filter bank; i.e.,  $\hat{\mathbf{b}}_i(k|k-1)$  and  $\mathbf{P}_i(k|k-1)$  are updated by a set of Kalman filter equations.

The MAP estimate of subsequence  $d_i^{k, N_b}$  is determined by

$$p(d_i^{k, N_b} | r^k) = \frac{1}{c} p(r(k)|d_i^{k, N_b}, r^{k-1}) \sum_{\{j: d_j^{k-1, N_b} \in d_i^{k, N_b}\}} p(d_j^{k-1, N_b} | r^{k-1}) \quad (13)$$

where  $c$  is a normalization constant. In the above sum, subsequence  $d_j^{k-1, N_b} \in d_i^{k, N_b}$  implies that the *first*  $N_b$  symbols in subsequence  $d_j^{k-1, N_b}$  are identical to the *last*  $N_b$  symbols in  $d_i^{k, N_b}$ . For example, with  $N_b = 3$  and assuming BPSK modulation, if  $d_j^{k-1, N_b} = \{-1, 1, 1, -1\}$  and  $d_i^{k, N_b} = \{1, -1, 1, 1\}$ , then  $d_j^{k-1, N_b} \in d_i^{k, N_b}$ . We emphasize that (13) is exact since it follows directly from Bayes'

formula. Furthermore, under approximation (12), the likelihood  $p(r(k)|d_i^{k, N_b}, r^{k-1})$  is Gaussian with a mean given in terms of  $\hat{\mathbf{b}}_i(k|k-1)$ , i.e.,

$$p(r(k)|d_i^{k, N_b}, r^{k-1}) = \mathcal{N}(r(k); \hat{r}_i(k|k-1), \sigma_i^2(k|k-1)). \quad (14)$$

The mean  $\hat{r}_i(k|k-1)$  and innovations variance  $\sigma_i^2(k|k-1)$  are computed as

$$\hat{r}_i(k|k-1) = \mathbf{h}_i(k) \hat{\mathbf{b}}_i(k|k-1), \quad (15)$$

which is the same as (8), and

$$\sigma_i^2(k|k-1) = \mathbf{h}_i(k) \mathbf{P}_i(k|k-1) \mathbf{h}_i^H(k) + \sigma_n^2 \quad (16)$$

where  $\mathbf{h}_i(k)$  is the row vector

$$\mathbf{h}_i(k) = [d_i(k), d_i(k-1), \dots, d_i(k-N_b)]. \quad (17)$$

The one-step estimate  $\hat{\mathbf{b}}_i(k+1|k)$  is defined by the conditional mean

$$\hat{\mathbf{b}}_i(k+1|k) = E[\mathbf{b}(k+1)|d_i^{k, N_b}, r^k], \quad (18)$$

which follows from (9) such that the subsequence  $d_i^{k, N_b}$  is substituted for the full sequence  $d_i^k$ . Thus, this estimate and the covariance  $\mathbf{P}_i(k+1|k)$  can be computed from the *time* updates of the Kalman filter. The derivation of these quantities is somewhat involved, however, since  $p(\mathbf{b}(k+1)|d_i^{k+1, N_b}, r^k)$  is a Gaussian sum even though we have approximated  $p(\mathbf{b}(k)|d_i^{k, N_b}, r^{k-1})$  as a Gaussian density in (12). As shown in Appendix A for the AR process model in (2), the time-updated estimate and covariance are determined from the *measurement*-updated quantities  $\hat{\mathbf{b}}_i(k|k)$  and  $\mathbf{P}_i(k|k)$ , respectively, that specify  $p(\mathbf{b}(k)|d_i^{k, N_b}, r^k)$ , as follows:

$$\hat{\mathbf{b}}_i(k+1|k) = \sum_{\{j: d_j^{k, N_b} \in d_i^{k+1, N_b}\}} \mathbf{F} \hat{\mathbf{b}}_j(k|k) \frac{p(d_j^{k, N_b} | r^k)}{\sum_{\{m: d_m^{k, N_b} \in d_i^{k+1, N_b}\}} p(d_m^{k, N_b} | r^k)} \quad (19)$$

and

$$\mathbf{P}_i(k+1|k) = \sum_{\{j: d_j^{k, N_b} \in d_i^{k+1, N_b}\}} \left\{ \left( \mathbf{F} \mathbf{P}_j(k|k) \mathbf{F}^T + \mathbf{Q} + \nu_{j,i}(k) \nu_{j,i}^H(k) \right) \times \frac{p(d_j^{k, N_b} | r^{k-1})}{\sum_{\{m: d_m^{k, N_b} \in d_i^{k+1, N_b}\}} p(d_m^{k, N_b} | r^k)} \right\} \quad (20)$$

where

$$\nu_{j,i}(k) = [\hat{\mathbf{b}}_i(k+1|k) - \mathbf{F} \hat{\mathbf{b}}_j(k|k)] \quad (21)$$

determines the outer-product term in (20).

Finally, the measurement update is defined as

$$\hat{\mathbf{b}}_i(k|k) = E \left[ \mathbf{b}(k) | d_i^{k, N_b}, r^k \right], \quad (22)$$

which is conditioned on the subsequence  $d_i^{k, N_b}$  as in (18). The corresponding measurement updates are determined from the Kalman filter equations in a straightforward manner; these are given by

$$\hat{\mathbf{b}}_i(k|k) = \hat{\mathbf{b}}_i(k|k-1) + \frac{1}{\sigma_i^2(k|k-1)} \mathbf{P}_i(k|k-1) \mathbf{h}_i^H(k) [r(k) - \hat{r}_i(k|k-1)], \quad (23)$$

and

$$\mathbf{P}_i(k|k) = \left[ \mathbf{I} - \frac{1}{\sigma_i^2(k|k-1)} \mathbf{P}_i(k|k-1) \mathbf{h}_i^H(k) \mathbf{h}_i(k) \right] \mathbf{P}_i(k|k-1). \quad (24)$$

The structure of the blind equalizer is now evident from (13) and the Kalman measurement updates in (23) and (24). For each subsequence  $d_j^{k-1, N_b}$ , the *a posteriori* probability  $p(d_j^{k-1, N_b} | r^{k-1})$  is assumed to be available. Likewise, for each new subsequence  $d_i^{k, N_b}$ , the conditional channel estimate  $\hat{\mathbf{b}}_i(k|k-1)$  is also available. A conditional Kalman filter channel estimate is then computed for each possible subsequence, and the MAP probability of the subsequence is updated using the resulting innovations likelihood. The overall algorithm is summarized in Table 1. The Bayesian equalizer structure is thus seen to consist of a bank of  $M^{N_b+1}$  conditional channel estimators. The symbol decisions are made using the MAP probability metrics themselves, described below.

The optimum decision<sup>1</sup> on symbol  $d(k - N_b)$  can be performed by computing the following marginal probability for each possible symbol  $d(k - N_b)$ :

$$\hat{d}(k - N_b) = \arg \max_{d(k - N_b)} \sum_{\{j: d_j(k - N_b) = d(k - N_b)\}} p(d_j^{k, N_b} | r^k). \quad (25)$$

In practice, we have found that only one of the metrics  $p(d_i^{k, N_b} | r^k)$  converges near unity, so that one term contributes substantially to the summation in (25). The suboptimal decision used for the results presented later in this paper is based on the subsequence with the largest metric, as follows:

$$\hat{d}_i^{k, N_b} = \arg \max_{d_i^{k, N_b}} p(d_i^{k, N_b} | r^k) \quad (26)$$

$$\hat{d}(k - N_b) = \hat{d}_i(k - N_b). \quad (27)$$

We emphasize that both the Bayesian equalizer developed here and the algorithm of Abend and Fritchman [8] are *symbol-by-symbol* detectors, and hence minimize the probability of a *symbol* error. In contrast, the better known

<sup>1</sup>It should be noted that this MAP decision rule, combined with a Bayesian formula similar to (13), was also used in [8] for the case of a *priori known* channels. However, [8] did not consider the problem of channel estimation and, thus, their likelihood computation was much simpler than in the problem considered here.

<p>Define Observation Vectors  <math>\mathbf{h}_i(k) = [d_i(k), \dots, d_i(k - N_b)]</math></p> <p>Compute Conditional Innovations Covariances  <math>\sigma_i^2(k k-1) = \mathbf{h}_i(k) \mathbf{P}_i(k k-1) \mathbf{h}_i^H(k) + \sigma_a^2</math></p> <p>Compute Signal Estimates  <math>\hat{r}_i(k k-1) = \sum_{n=0}^{N_b} \hat{b}_{i,n}(k k-1) d_i(k-n)</math></p> <p>Update Conditional Measurement Estimates  <math>\hat{\mathbf{b}}_i(k k) = \hat{\mathbf{b}}_i(k k-1) + \frac{1}{\sigma_i^2(k k-1)} \mathbf{P}_i(k k-1) \mathbf{h}_i^H(k) [r(k) - \hat{r}_i(k k-1)]</math></p> <p>Update Conditional Error Covariances  <math>\mathbf{P}_i(k k) = \left[ \mathbf{I} - \frac{1}{\sigma_i^2(k k-1)} \mathbf{P}_i(k k-1) \mathbf{h}_i^H(k) \mathbf{h}_i(k) \right] \mathbf{P}_i(k k-1)</math></p> <p>Update Weighting Probabilities  <math>p(d_i^{k, N_b}   r^k) = \frac{1}{c} \mathcal{N}(r(k); \hat{r}_i(k k-1), \sigma_i^2(k k-1)) \sum_{\{j: d_j^{k-1, N_b} \in d_i^{k, N_b}\}} p(d_j^{k-1, N_b}   r^{k-1})</math></p> <p>Update One-Step Predictions  <math>\hat{\mathbf{b}}_i(k+1 k) = \sum_{\{j: d_j^{k, N_b} \in d_i^{k+1, N_b}\}} \mathbf{F} \hat{\mathbf{b}}_j(k k) \frac{p(d_j^{k, N_b}   r^k)}{\sum_{\{m: d_m^{k, N_b} \in d_i^{k+1, N_b}\}} p(d_m^{k, N_b}   r^k)}</math></p> <p>Compute Covariance Outer Product Vectors  <math>\nu_{j,i}(k) = [\hat{\mathbf{b}}_i(k+1 k) - \mathbf{F} \hat{\mathbf{b}}_j(k k)]</math></p> <p>Update One-Step Error Covariances  <math>\mathbf{P}_i(k+1 k) = \sum_{\{j: d_j^{k, N_b} \in d_i^{k+1, N_b}\}} (\mathbf{F} \mathbf{P}_j(k k) \mathbf{F}^T + \mathbf{Q} + \nu_{j,i}(k) \nu_{j,i}^H(k)) \frac{p(d_j^{k, N_b}   r^k)}{\sum_{\{m: d_m^{k, N_b} \in d_i^{k+1, N_b}\}} p(d_m^{k, N_b}   r^k)}</math></p>
--

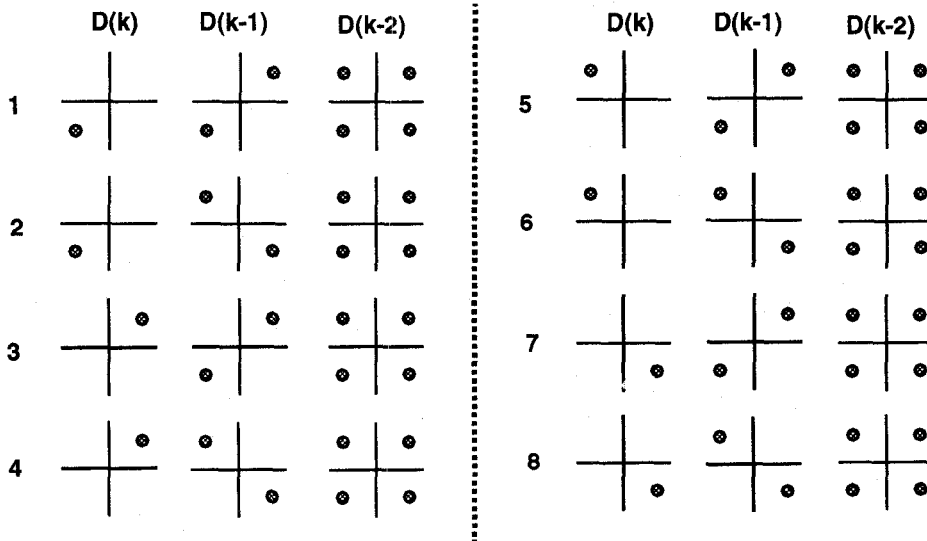
Table 1. Bayesian Blind Equalization Algorithm

maximum-likelihood sequence estimation (MLSE) algorithm of Forney [9] minimizes the probability of a *sequence* error. The symbol-by-symbol detector could be obtained by summing over all relevant sequence probabilities, but the MLSE algorithm maintains only those probabilities of the survivor sequences. MLSE is thus not equivalent to symbol-by-symbol detection. Hence, the optimal MAP symbol-by-symbol detector theoretically should yield a lower bit error rate than the optimum sequence estimator; furthermore, an upper bound on the MLSE symbol error rate should also serve as an upper bound on the performance of a symbol-by-symbol detector.

#### IV. SIMPLIFIED BLIND EQUALIZATION ALGORITHM USING RSSE

One possible drawback to the algorithm in Table 1 is that  $M^{N_b+1}$  Kalman measurement updates must be computed at each iteration, where  $M$  is the symbol alphabet size and  $N_b + 1$  is the length of the channel impulse response. To alleviate this problem, we consider a simplified algorithm using the concept of reduced-state sequence estimation (RSSE), first introduced in [6]. In RSSE, the symbol subsequences  $\{d_i^{k, N_b}\}$  are grouped into *reduced-state* subsequences,<sup>2</sup> denoted here by  $\{D_i^{k, N_b} = \{D_i(k), D_i(k-1), \dots, D_i(k - N_b)\}\}$ . Each symbol  $D_i(k - m)$  is actually a

<sup>2</sup>Although the  $\{D_i^{k, N_b}\}$  are not, strictly speaking, states in a VA trellis, each  $D_i^{k, N_b}$  corresponds to an individual Kalman channel estimator, and thus a reduction in the number of symbol subsequences can greatly simplify the algorithm.


 Fig. 1. Subsets for RSSE(421) (QPSK with  $N_b + 1 = 3$ ).

subset of the symbols  $\{d_i(k-m)\}$ , and has dimensionality  $M_m \leq M$ , satisfying  $M_{N_b} \leq M_{N_b-1} \leq \dots \leq M_0 \leq M$ . For example, for the case of QPSK signalling with  $N_b + 1 = 3$ , we could choose  $M_0 = 4$ ,  $M_1 = 2$ , and  $M_2 = 1$  (denoted by RSSE( $M_0, M_1, M_2$ )), which results in a dimensionality of 8 for the reduced subsequence  $D_i^{k, N_b}$ , as opposed to 64 for the original subsequence  $d_i^{k, N_b}$ . The eight subsets are shown in Fig. 1.

A recursion for the MAP metrics of the reduced-state subsequences is derived as follows. For the Ungerboeck partitioning described in [6], the last  $N_b$  elements of  $D_i^{k, N_b}$  are given by a union of the subsequences  $\{D_j^{k-1, N_b}\}$ . Thus, we have

$$p(D_i^{k, N_b} | r^k) = \frac{1}{c} p(r(k) | D_i^{k, N_b}, r^{k-1}) \sum_{\{j: D_j^{k-1, N_b} \in D_i^{k, N_b}\}} p(D_j^{k-1, N_b} | r^{k-1}), \quad (28)$$

which is similar to (13) for the original subsequences. The likelihood  $p(r(k) | D_i^{k, N_b}, r^{k-1})$  cannot be computed exactly from the MAP metric of  $D^{k-1, N_b}$  alone. As a result, we approximate this likelihood using the metric of the most likely subsequence  $d_i^{k, N_b}$  contained in  $D_i^{k, N_b}$ , i.e.,

$$p(r(k) | D_i^{k, N_b}, r^{k-1}) \approx p(r(k) | \hat{D}_i^{k, N_b}, r^{k-1}) \quad (29)$$

where

$$\hat{D}_i^{k, N_b} = \arg \max_{d_i^{k, N_b} \in D_i^{k, N_b}} p(r(k) | d_i^{k, N_b}, r^{k-1}). \quad (30)$$

The likelihood above, conditioned on the individual sequence  $d_i^{k, N_b}$ , can be computed using the Kalman filter one-step predictions according to

$$p(r(k) | \hat{D}_i^{k, N_b}, r^{k-1}) = \mathcal{N}(r(k); \hat{r}_i(k|k-1), \sigma_i^2(k|k-1)). \quad (31)$$

The estimated signal  $\hat{r}_i(k|k-1)$  and observation vector  $\mathbf{h}_i(k)$  are similar to those in (15) and (17), respectively, as follows:

$$\hat{r}_i(k|k-1) = \sum_{m=0}^{N_b} \hat{b}_{i,m}(k|k-1) \hat{D}_i(k-m) \quad (32)$$

$$\mathbf{h}_i(k) = [\hat{D}_i(k), \hat{D}_i(k-1), \dots, \hat{D}_i(k-N_b)]. \quad (33)$$

Finally, the required modifications to the rest of the algorithm in Table 1 simply involve replacing the original subsequences  $d_i^{k, N_b}$  with the reduced-state subsequences  $D_i^{k, N_b}$ . For example, the one-step prediction update for the channel estimate becomes

$$\hat{\mathbf{b}}_i(k+1|k) = \sum_{\{j: D_j^{k, N_b} \in D_i^{k+1, N_b}\}} \mathbf{F} \hat{\mathbf{b}}_j(k|k) \frac{p(D_j^{k, N_b} | r^k)}{\sum_{\{j: D_j^{k, N_b} \in D_i^{k+1, N_b}\}} p(D_j^{k, N_b} | r^k)}. \quad (34)$$

Again, it should be emphasized that for a proper partitioning of the symbol subsequences, the last  $N_b$  elements of  $D_i^{k+1, N_b}$  are expressed as a union of the first  $N_b$  elements of  $D_j^{k, N_b}$ .

## V. BLIND EQUALIZATION USING PARALLEL LMS ADAPTIVE FILTERS

In order to reduce the complexity even further, we propose using scalar *gradient* algorithms for the Kalman filter measurement updates. These algorithms are similar to the LMS algorithm and its normalized forms [10], and they do not require an underlying state-space model or any covariance matrix updates. Thus, the algorithm in Table 1 is simplified by approximating all covariance matrices with a scaled version of the identity matrix:

$\mathbf{P}_i(k|k) \approx \mathbf{P}_i(k|k-1) \approx \gamma \mathbf{I}$  (where  $\gamma > 0$ ). Furthermore, the underlying state-space model is "ignored," which is equivalent to setting the system matrix  $\mathbf{F}$  equal to the identity matrix, and the process noise covariance  $\mathbf{Q}$  equal to zero.

We will adopt a slightly different notation for clarity. The predicted estimates  $\{\mathbf{b}_i(k+1|k)\}$  are replaced by  $\{\mathbf{b}_i(k)\}$  and the filtered estimates  $\{\mathbf{b}_i(k|k)\}$  become  $\{\mathbf{b}_i^c(k)\}$ . We will refer to  $\{\mathbf{b}_i(k)\}$  and  $\{\mathbf{b}_i^c(k)\}$ , respectively, as the unconditional and conditional channel estimates. Fig. 2 shows the resulting efficient implementation of the LMS filter bank. Observe that there are  $N = M^{N_b+1}$  single-input, single-output adaptive finite impulse response (FIR) filters comprised of the unconditional estimates  $\hat{\mathbf{b}}_i(k-1)$ ,  $i = 1, \dots, M^{N_b+1}$ . The filter inputs are determined by all possible subsequences  $\{\mathbf{h}_i(k)\}$ , and each filter output  $\hat{r}_i(k)$  is generated according to the inner product

$$\hat{r}_i(k) = \mathbf{h}_i(k) \hat{\mathbf{b}}_i(k-1). \quad (35)$$

Thus, the output of the  $i^{\text{th}}$  FIR filter corresponds to an estimate of the current received symbol  $r(k)$ , assuming that the  $i^{\text{th}}$  subsequence was transmitted (i.e., conditioned on the  $i^{\text{th}}$  subsequence). The filter outputs are then compared to the received sample  $r(k)$  to generate a set of innovations or error signals,  $e_i(k) = r(k) - \hat{r}_i(k)$ ,  $i = 1, \dots, M^{N_b+1}$ .

The conditional innovations variance update in (16) becomes

$$\sigma_i^2(k) = \gamma \mathbf{h}_i(k) \mathbf{h}_i^H(k) + \sigma_n^2, \quad (36)$$

and the conditional measurement update in (23) reduces to

$$\hat{\mathbf{b}}_i^c(k) = \hat{\mathbf{b}}_i(k-1) + \frac{\alpha}{\sigma_i^2(k)} \mathbf{h}_i^H(k) [r(k) - \hat{r}_i(k)] \quad (37)$$

where  $\alpha$  is a constant usually chosen to be  $0 < \alpha < 2$  [10]. Table 2 outlines the simplified algorithm, which now requires only three parameters: (i) the step size  $\alpha$ , (ii) an estimate of the noise variance  $\sigma_n^2$ , and (iii) the variance factor  $\gamma$ .

The measurement updates have the same form as a normalized version of the LMS algorithm [10]. It is well known that the convergence properties of gradient algorithms are sensitive to changes in the power of the input signal. The conditional innovations variances  $\{\sigma_i^2(k)\}$  are estimates of this power for each of the possible subsequences, and they are used in the measurement update to compensate for any power variations. Thus, it is possible to have a more uniform convergence rate over a wide variation of the input signal power. In this context, the measurement noise variance  $\sigma_n^2$  can be viewed as a constant that is included primarily to ensure that the measurement update term is not exceedingly large when the inner product  $\gamma \mathbf{h}_i(k) \mathbf{h}_i^H(k)$  is small.

From Table 2, we see that the approximate innovations variance  $\sigma_i^2(k)$  affects not only the measurement updates, but also the probability metric updates. However, our simulations demonstrate that the overall algorithm performance is fairly insensitive to variations in  $\sigma_i^2(k)$ . Furthermore, the inner product  $\mathbf{h}_i(k) \mathbf{h}_i^H(k)$  of the observation

<p>Define Observation Vectors  <math>\mathbf{h}_i(k) = [d_i(k), \dots, d_i(k - N_b)]</math></p> <p>Compute Conditional Innovations Variances  <math>\sigma_i^2(k) = \gamma \mathbf{h}_i(k) \mathbf{h}_i^H(k) + \sigma_n^2</math></p> <p>Compute Signal Estimates  <math>\hat{r}_i(k) = \sum_{n=0}^{N_b} \hat{\mathbf{b}}_{i,n}(k-1) d_i(k-n)</math></p> <p>Update Conditional Estimates  <math>\hat{\mathbf{b}}_i^c(k) = \hat{\mathbf{b}}_i(k-1) + \frac{\alpha}{\sigma_i^2(k)} \mathbf{h}_i^H(k) (r(k) - \hat{r}_i(k))</math></p> <p>Update Weighting Probabilities  <math>p(d_i^{k, N_b}   r^k) = \frac{1}{c} \mathcal{N}(r(k); \hat{r}_i(k), \sigma_i^2(k)) \sum_{\{j: d_j^{k-1, N_b} \in d_i^{k, N_b}\}} p(d_j^{k-1, N_b}   r^{k-1})</math></p> <p>Update Unconditional Estimates  <math>\hat{\mathbf{b}}_i(k) = \sum_{\{j: d_j^{k, N_b} \in d_i^{k+1, N_b}\}} \hat{\mathbf{b}}_j^c(k) \frac{p(d_j^{k, N_b}   r^k)}{\sum_{\{m: d_m^{k, N_b} \in d_i^{k+1, N_b}\}} p(d_m^{k, N_b}   r^k)}</math></p>
---

Table 2. Simplified Blind Equalization Algorithm

vectors is invariant with respect to the data subsequences for any phase-modulated signal constellation (e.g., BPSK and QPSK). Thus, further simplification of the algorithm can be achieved by setting  $\sigma_i^2(k) = \sigma^2, \forall k, \forall i$ , where  $\sigma^2$  is a constant. The resulting measurement update is equivalent to the LMS algorithm with step size  $\mu = \alpha/\sigma^2$ . The metric probabilities are not sensitive to the actual value chosen for  $\sigma^2$ , provided that a "reasonable" value is chosen.<sup>3</sup> Since the parameters are fixed for this case, the conditional innovations variances in Table 2 are no longer computed. Thus, the simplified algorithm requires only (i) the conditional updates with parameter  $\mu$ , (ii) the weighting probability updates with parameter  $\sigma^2$ , and (iii) the unconditional updates based on the results of (i) and (ii). These unconditional updates remove the influence of the "oldest" symbol by summing over the  $M$  subsequences which differ only in the  $d_i(k - N_b)^{\text{th}}$  symbol.

## VI. COMPUTATIONAL COMPLEXITY OF THE LMS BAYESIAN EQUALIZER

Unlike the Kalman algorithm in Table 1, the LMS version of the algorithm outlined in Table 2 does not require matrix operations, and thus is better suited for implementation in parallel DSP devices. To investigate the computational complexity of the LMS Bayesian equalizer, the number of processor instruction cycles required for each input sample  $r(k)$  was determined. Table 3 provides the instruction count in terms of  $N$  (the number of subsequences  $d_i^{k, N_b}$ ),  $M$  (the size of the symbol alphabet), and  $N_b$  (the length of the ISI). Recall that  $N \leq M^{N_b+1}$ , depending

<sup>3</sup>The metrics are primarily influenced by  $\hat{r}_i(k)$ , which corresponds to the mean value of the Gaussian update.

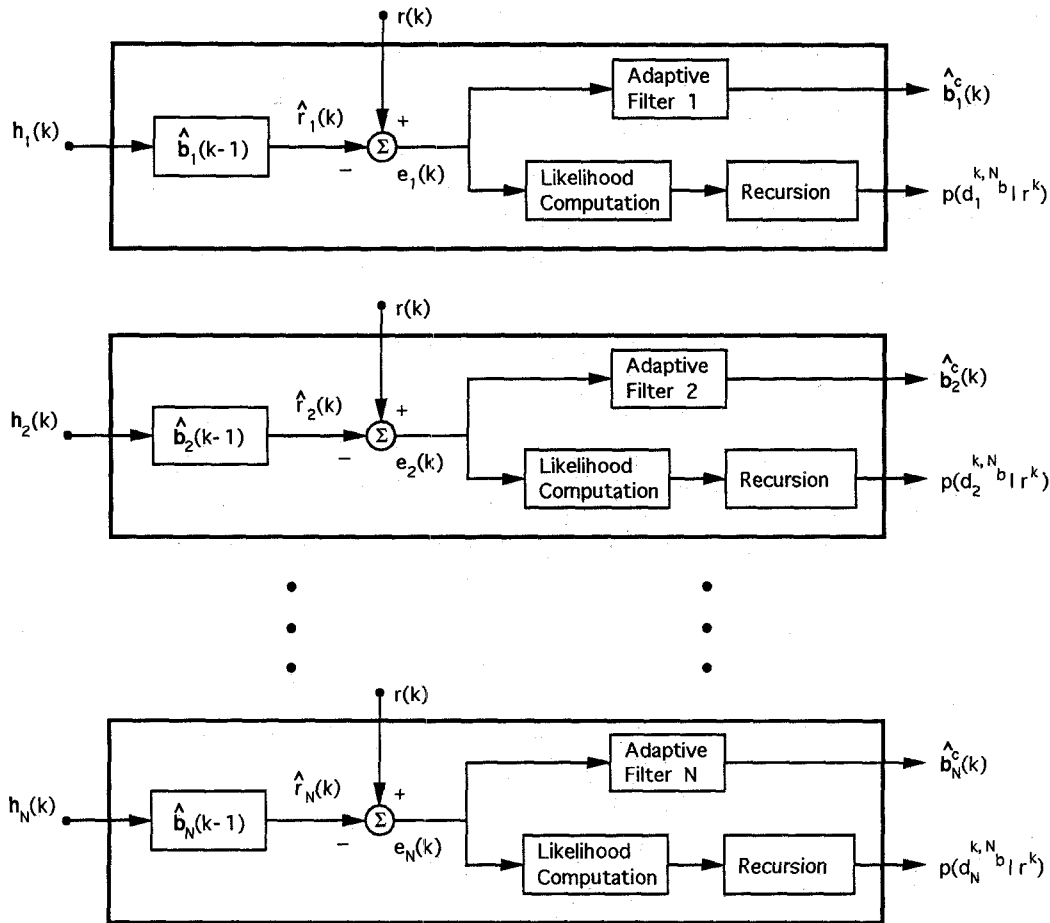


Fig. 2. Bayesian equalizer using parallel LMS adaptive filters.

Variable	Complex Adds	Complex Multiplies	Real Adds	Real Multiplies	Real Divides
$\hat{r}_i(k)$	$N \times N_b$	$N(N_b + 1)$	0	0	0
$\hat{\mathbf{b}}_i^c(k)$	$N(N_b + 2)$	$N(N_b + 2)$	0	0	0
$p'(d_i^{k,N_b}   r^k)$	0	0	$N(11 + (M - 1))$	$33N$	0
$p(d_i^{k,N_b}   r^k)$	0	0	$N - 1$	0	$N$
$\hat{\mathbf{b}}_i(k)$	0	0	$(N/M)(M - 1)(N_b + 2)$	$(N/M)2M(N_b + 1)$	$N/M$

Table 3. Instruction Count/Sample for the LMS Bayesian Equalizer

on the extent to which RSSE is employed.<sup>4</sup> The metrics  $\{p'(d_i^{k,N_b} | r^k)\}$  refer to the *unnormalized* MAP probabilities. The final metrics  $\{p(d_i^{k,N_b} | r^k)\}$  are generated by first summing the  $\{p'(d_i^{k,N_b} | r^k)\}$  to compute the normalization constant  $c$ , and then dividing each metric by  $c$ . It is assumed that either a fixed step size  $\mu$  or pre-computed gain sequence  $\mu(k)$  is used in the LMS update, and that  $\sigma_i^2(k)$  is approximated by the constant  $\sigma^2$ .

<sup>4</sup> A seventh-order series expansion was used for the computation of the exponentials in the likelihood update. The use of a look-up table with interpolation for this computation would reduce the overall number of instructions required per input symbol.

In order to compute the number of instructions required per second, the following assumptions were made, which are consistent with most programmable DSP devices.

- A multiply or add takes one machine instruction cycle.
- A divide requires 24 instruction cycles. (The number of cycles required for a divide operation is processor dependent; this figure applies to the Motorola 56000 DSP [11].)
- The symbol rate is 8000 symbols/second (baud), and QPSK modulation is employed (hence, the bit rate is 16 kbps).

- $N_b = 2$ .
- $N = 64$  (when RSSE is not employed), or  $N = 32$  (corresponding to RSSE with  $M_2 = 2$ ,  $M_1 = 4$ , and  $M_0 = 4$ ).
- The operations are partitioned among  $N_p = 4$  parallel Motorola 56000 DSP devices.

It should be stressed that the operations in Table 3 are of a highly parallel nature. In principle, all steps except for normalization of the probabilities  $p'(d_i^{k,N_b}|r^k)$  could be partitioned among  $N_p = N$  processors. Here, we assume that  $N_p$  is restricted to be 4. For  $N = 32$  (RSSE with  $M_0 = 4$ ,  $M_1 = 4$ ,  $M_2 = 2$ ), Table 3 yields  $1.38 \times 10^6$  instruction cycles per second. This corresponds to an instruction cycle duration of approximately 72 nsec. The Motorola 56000 DSP has a 50 nsec. instruction cycle, and hence the implementation for this example using four processors appears to be feasible, if the code can be optimized and overhead is kept to a minimum. If RSSE with  $N = 16$  subsequences is employed, Table 3 yields a required instruction cycle duration of 144 nsec., which is well within the capabilities of the Motorola 56000 DSP, even if considerable overhead is required.

These example calculations indicate that the LMS version of the Bayesian blind equalizer is a good candidate for implementation in programmable DSP devices. If special-purpose VLSI can be designed, then higher data rates and/or longer channels can be accommodated. Finally, it should be mentioned that the number of subsequences that needs to be considered, and hence the computational complexity, can be greatly reduced by incorporating a feedback channel estimator, as described in [12].

## VII. COMPUTER SIMULATIONS

The new blind equalization algorithm was simulated for BPSK and QPSK signals using the Kalman filter (KF) implementation as well as the simpler LMS version. The RSSE version of the algorithm was simulated for QPSK signalling and LMS adaptation. The following complex channel transfer function was used for all versions of the Bayesian equalizer:

$$H(z) = 0.444487 + (-0.488658 - j0.776700)z^{-1} + (-0.440101 + j0.0555976)z^{-2}, \quad (38)$$

which has the frequency response shown in Fig. 3. Observe that this channel has infinite nulls on the unit circle, which is difficult to equalize with an FIR equalizer. The eye patterns produced by this channel for SNR = 10 dB are shown in Fig. 4 for BPSK and QPSK signalling. Observe that for both signal formats, the eye is closed prior to equalization.

During demodulation, BPSK signals can be detected from either the in-phase or quadrature channel outputs alone. However, since we are also performing channel estimation, we have used a complex equalizer even for BPSK signalling. The SNR was defined in terms of the bit energy  $E_b$  and the noise power  $N_o$ , i.e.,  $\text{SNR} = 10 \log(E_b/N_o)$  dB.

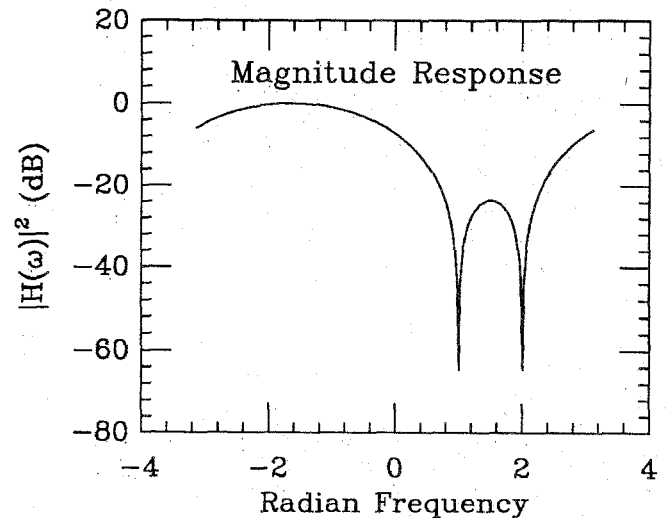


Fig. 3. Channel frequency response.

For convenience,  $E_b = 1$  in all simulations, and  $N_o$  was varied to evaluate the channel estimator for a variety of SNRs. In addition, the bit interval was set equal to one, i.e.,  $T_b = 1$ .

The results for BPSK signalling are shown in Figs. 5-8, while those for QPSK signalling are given in Figs. 9-12. The trajectories of the probability metrics are shown for one run of the algorithm, while the coefficient error plots were obtained by averaging 10 independent runs. Furthermore, in each of the 10 runs, a random initial coefficient estimate  $\hat{b}_i(0) - 1$  was chosen for each estimator in order to investigate the effects of initialization on the algorithm. Each coefficient estimate  $\hat{b}_{i,m}(0) - 1$  was chosen from a uniform distribution in  $[-0.5, 0.5]$ .

Since the channel is time-invariant, the algorithms were optimized to operate on stationary data. Thus, for the Kalman filter version, the state transition matrix was set to  $\mathbf{F} = \mathbf{I}$ , and the plant noise covariance  $\mathbf{Q}$  was set to zero. Also, to reduce the misadjustment error of the LMS version at steady state, the step size  $\mu$  was allowed to decay at a rate of  $\beta = 0.99$ , i.e.,  $\mu(k) = \beta^k \mu$  with  $\mu = 0.5$  for BPSK and  $\mu = 0.25$  for QPSK.

It should be emphasized that both the Kalman and LMS adaptive filters are equivalent in steady-state, in the sense that their gains both decay to zero. For the Kalman filter, this is seen as follows. At high SNR, one of the metrics  $p(d_i^{k,N_b}|r^k)$  is typically close to unity, with the rest near zero, and hence the outer product terms  $\nu_{j,i}(k)$  in the covariance update in (20) tend to zero. Thus,  $\mathbf{P}_i(k+1|k)$  is approximately equal to the ordinary Kalman filter covariance, which tends to zero for  $\mathbf{Q} = \mathbf{0}$ . The LMS gains tend to zero asymptotically due to the exponential decay factor  $\beta$ .

Fig. 5 shows the evolution of the probability metrics of the KF version for SNRs of 10 dB and 20 dB. Since the channel has three coefficients and  $M = 2$ , there are eight possible subsequences, and thus eight probability metrics. Observe that for the higher SNR, one of the metrics converges to unity in less than 20 iterations. Although the



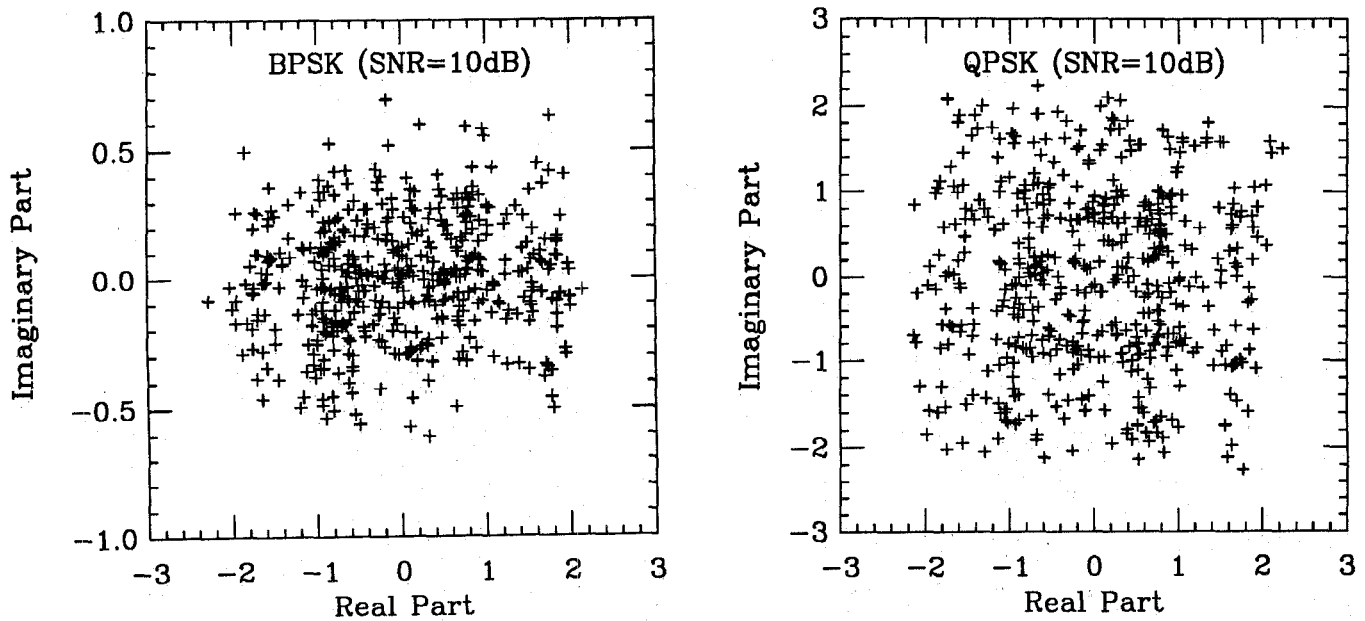


Fig. 4. Eye patterns.

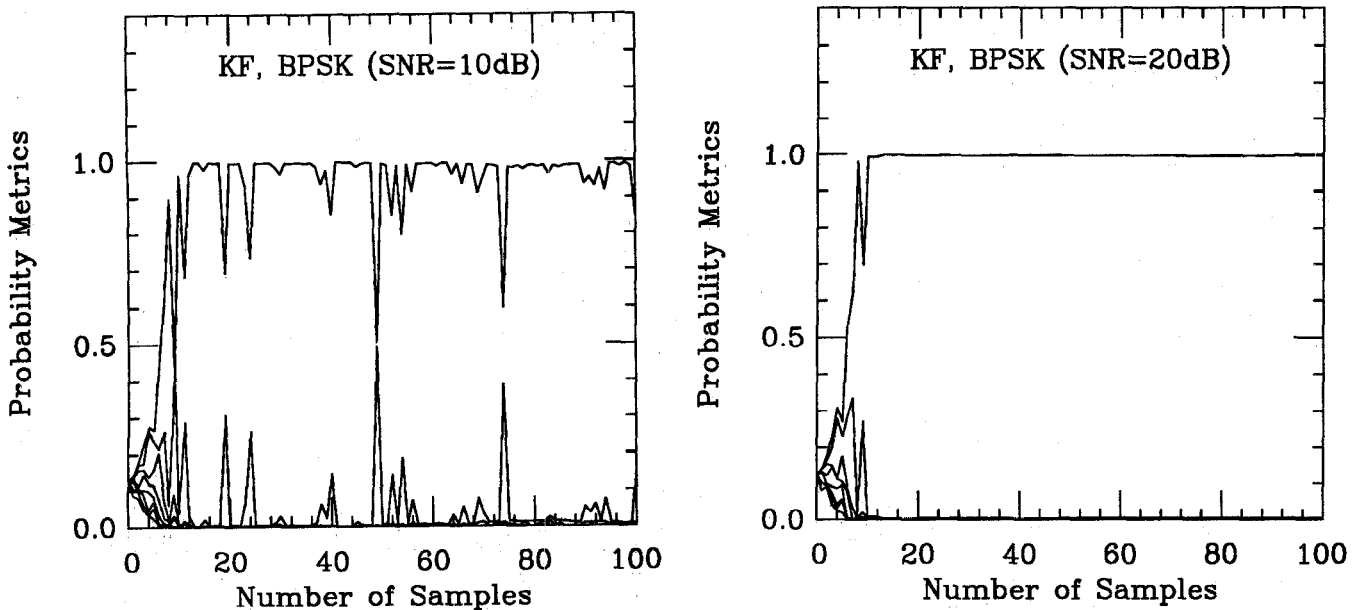


Fig. 5. Evolution of the probability metrics (KF, BPSK).

metric trajectories are noisier for the lower SNR, there is still only one metric that dominates after convergence.

Fig. 6 shows the trajectories for the corresponding ensemble averaged coefficient errors, which were obtained by averaging the squared errors between the actual channel coefficients and the conditional estimates, weighted by the probability metrics at each iteration. The channel coefficient squared-error at iteration  $k$  and averaged over  $N_r = 10$  runs is thus defined by

$$E(k) = \frac{1}{N_r} \sum_{n=1}^{N_r} \frac{1}{N_b + 1} \sum_{m=0}^{N_b} \sum_{i=1}^{2^{N_b+1}} \left\{ \left| \hat{d}_{i,m}^n(k|k-1) - b_m \right|^2 \right. \quad (39)$$

$$\left. \times p(d_i^{k,N_b} | r^{k-1}) \right\}.$$

Observe that for SNR = 20 dB,  $E(k)$  for the Kalman filter version is less than -30 dB by about 40 samples. Furthermore, this level of performance indicates that the algorithm is insensitive to a random initialization of the channel coefficient estimates.

Fig. 7 shows the corresponding metrics for the LMS version of the equalizer. Note that the convergence speed of these metrics is comparable to that of the KF algorithm. The corresponding channel coefficient squared error is shown in Fig. 8. Observe that for SNR = 20 dB, the LMS version takes about 300 samples to achieve -30 dB, which is about 7 times longer than for the KF algorithm. Again, the coefficient squared error was averaged over an

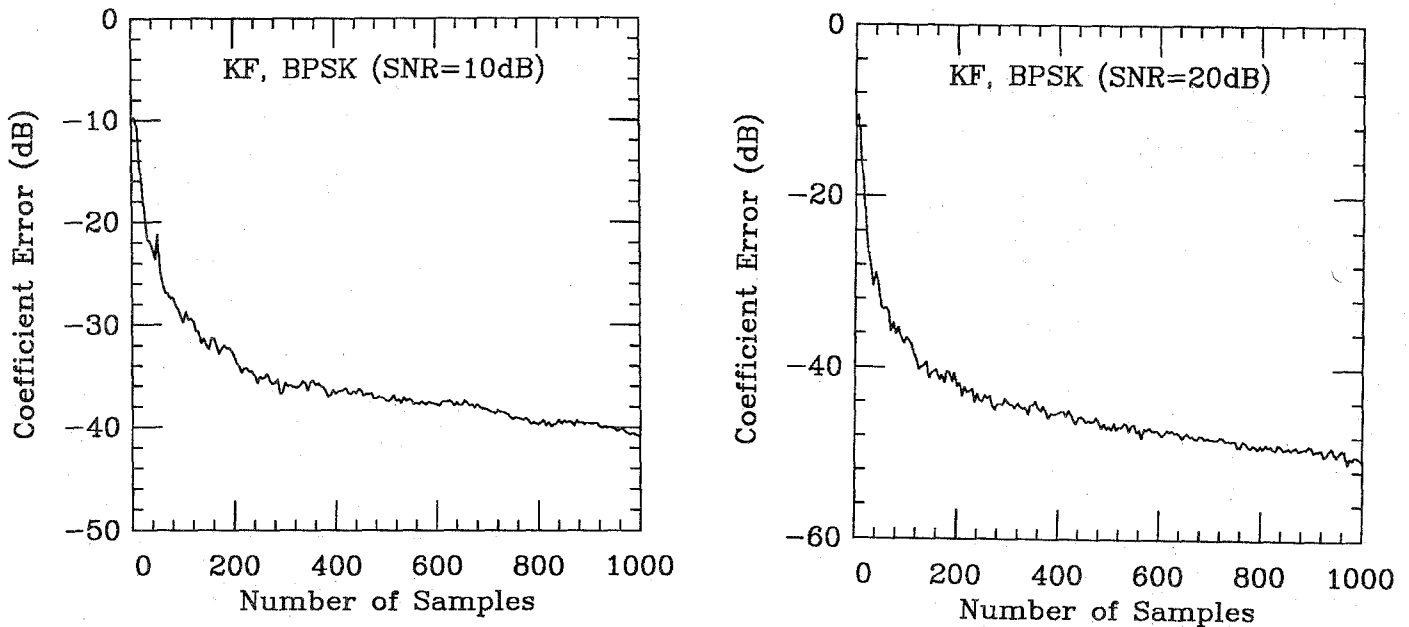


Fig. 6. Coefficient error trajectories (KF, BPSK).

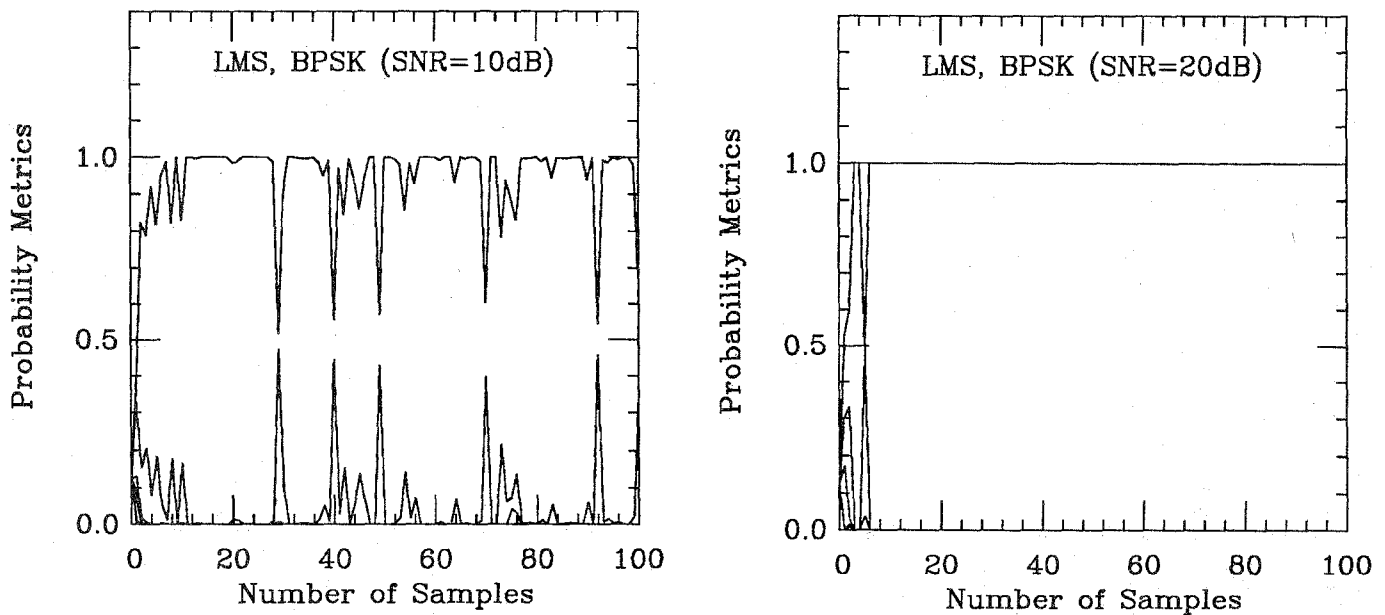


Fig. 7. Evolution of the probability metrics (LMS, BPSK).

ensemble of 10 runs, with the initial coefficient estimates randomly chosen from a uniform distribution.

The slow convergence of the channel estimates for the LMS algorithm can be explained as follows. Observe that the channel estimates are computed in parallel with the actual channel, in a configuration that resembles system identification. Since the transmitted data is assumed to be white, the eigenvalue spread of the (infinite data) autocorrelation matrix is unity. Thus, theoretically, we expect that the LMS algorithm would converge as rapidly as the KF algorithm or a recursive-least-squares (RLS) algorithm [10]. However, for the finite-data case, the autocorrelation matrix is not truly diagonal, and hence the convergence speed of the LMS is somewhat slower. Also, for PSK signals, the

simplified LMS version maintains only one common step size parameter  $\mu(k)$  for all adaptive filters in the parallel bank. This may also reduce the convergence speed of the LMS algorithm compared to the KF algorithm, where a separate error-covariance matrix is maintained for each channel estimate.

Figs. 9 through 12 show the corresponding results for QPSK signalling. In this case since  $M = 4$ , there are 64 possible subsequences, of which we plot only the eight largest metrics. Observe again that the metrics for the KF and LMS algorithms converge with comparable rates, while the coefficient error takes longer to reach steady state. Also, the channel coefficient estimates for QPSK signalling and the KF algorithm take more time to reach steady

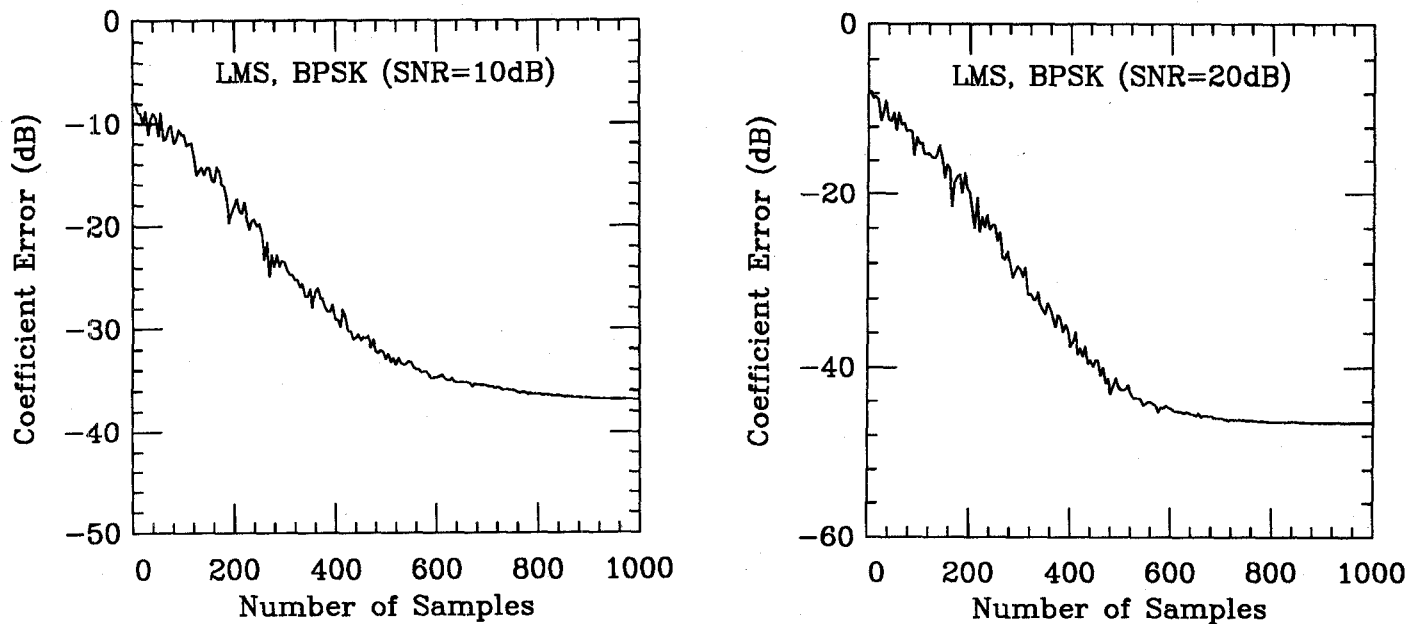


Fig. 8. Coefficient error trajectories (LMS, BPSK).

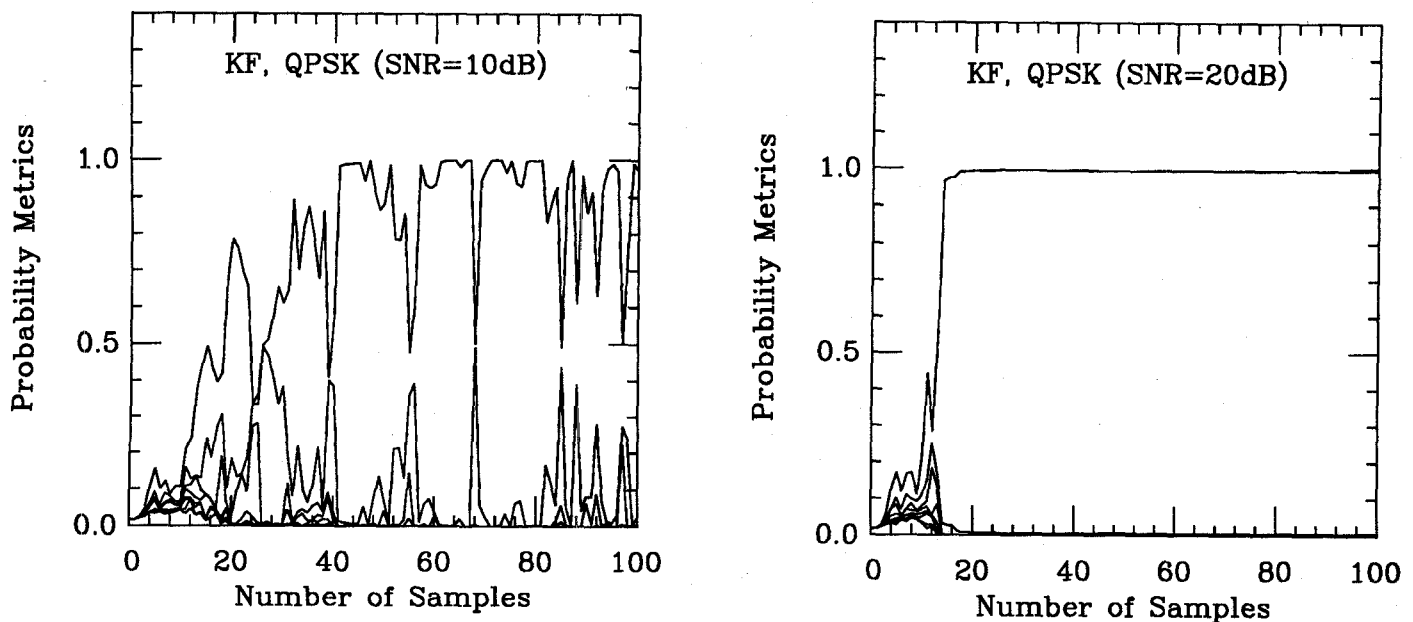


Fig. 9. Evolution of the probability metrics (KF, QPSK).

state, compared to that for BPSK signalling. Specifically, at SNR = 20 dB, the coefficient error for the KF algorithm takes approximately 100 iterations to reach -30 dB, whereas for the LMS algorithm, 300 iterations are required.

The bit-error-rate (BER) performance curves for BPSK are presented in Fig 13. Note that AF denotes the Abend and Fritchman algorithm [8]. The solid curve, corresponding to zero ISI, is the performance of binary signalling on an additive white Gaussian noise (AWGN) channel. To obtain an upper bound on the symbol error probability, we used the method in [9] and [13] developed for MLSE. It should be emphasized that the Bayesian equalizer presented here is an approximation to the symbol-by-symbol detector of [8]. As discussed in Section III, the optimum

symbol-by-symbol detector will perform at least as well as an optimum sequence estimator, in terms of minimizing the symbol error rate. Hence, the bound developed by Forney [9] for MLSE performance is a true upper bound on the performance of the Abend and Fritchman algorithm. This upper limit on the symbol error probability, assuming coherent detection, is given by [13]

$$P_e \approx \frac{1}{2} K_{dmin} \operatorname{erfc} \left( \sqrt{\frac{E_b}{N_o} d_{min}^2} \right) \quad (40)$$

where  $d_{min}^2$  is the minimum distance for worst-case ISI, and  $K_{dmin}$  is the corresponding weighting factor that is independent of  $N_o$ . The loss in SNR due to ISI can be approximated by  $10 \log_{10}(d_{min}^2)$ . For the three-coefficient FIR

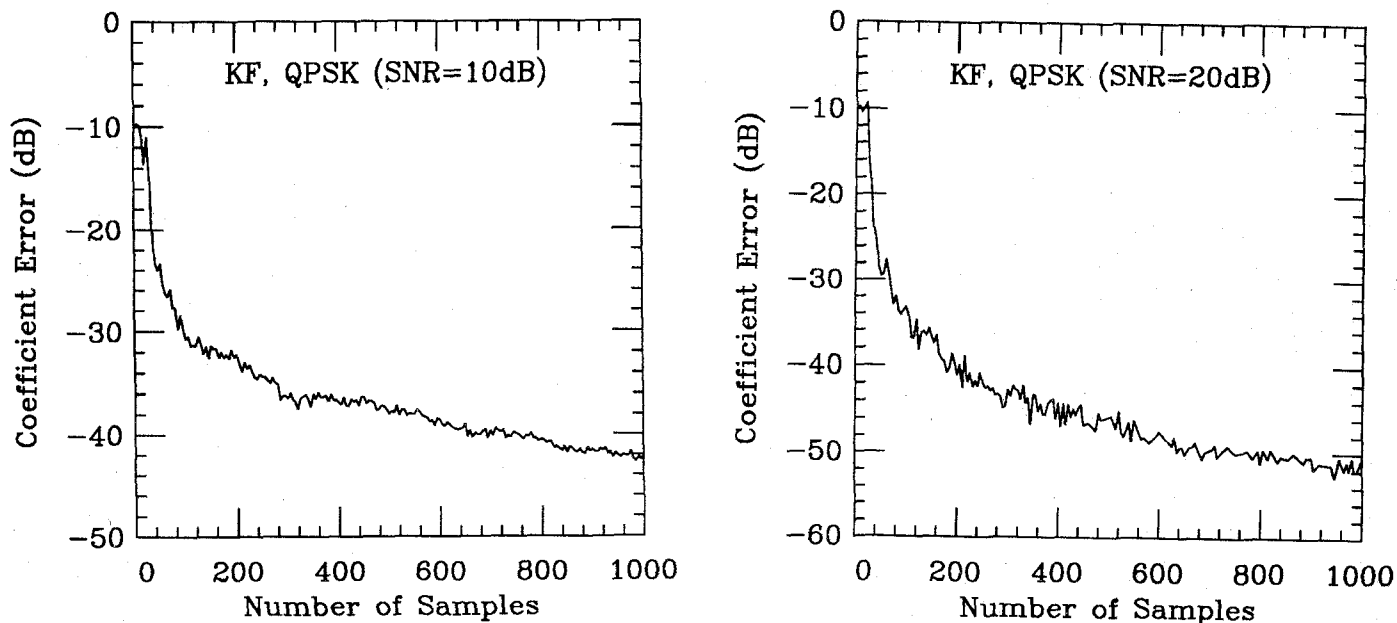


Fig. 10. Coefficient error trajectories (KF, QPSK).

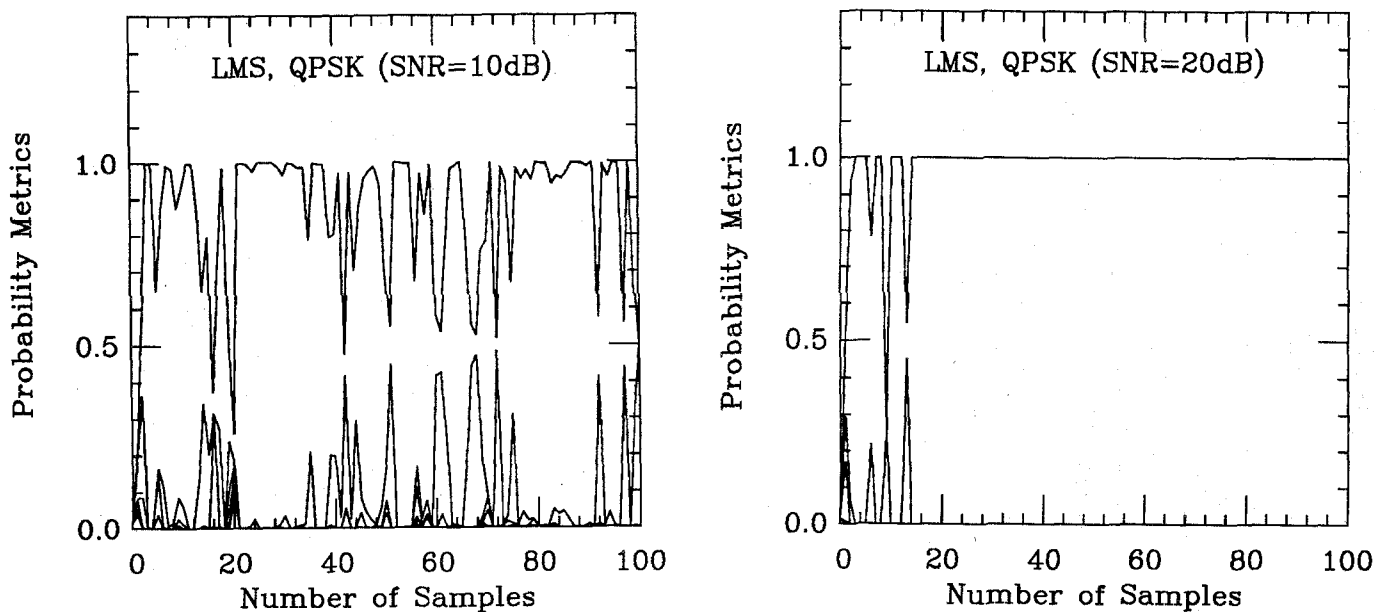


Fig. 11. Evolution of the probability metrics (LMS, QPSK).

channel, among all error events, an error event of length 2 gives the lowest value (worst case) for  $d_{min}^2$ , which is equal to  $2 - \sqrt{2}$ . Substituting this into the above expression with  $K_{dmin} = 2$ , we obtain the upper BER limit for MLSE in the presence of ISI, shown by the dashed line in Fig. 13.

The BER for the new algorithm was computed using coherent detection. The bit errors were counted by repeating the experiment many times, each with a different seed for the random number generator, and with a different (random) initial coefficient vector estimate. The BER was measured after reaching steady state, i.e., we discarded the initial 1000 samples before counting symbol errors. Since both the LMS and KF versions of the algorithm provided good-quality channel estimates after 1000 iterations,

we chose to evaluate the BER only for the LMS version. (The KF Bayesian equalizer should typically provide better BER performance only during initial convergence.) The standard deviation of the BER estimation error was kept to within 5% by repeating the experiment a sufficient number of times.

The performance of the optimum symbol-by-symbol detector, when the channel is assumed known *a priori*, provides a lower bound on the performance of the Bayesian equalizer. However, the error rate of the symbol-by-symbol detector of Abend and Fritchman [8] thus far has only been evaluated via simulation, and has not yet been bounded or derived analytically. We thus employ a simulation of the algorithm in [8], with perfect knowledge of the channel as-

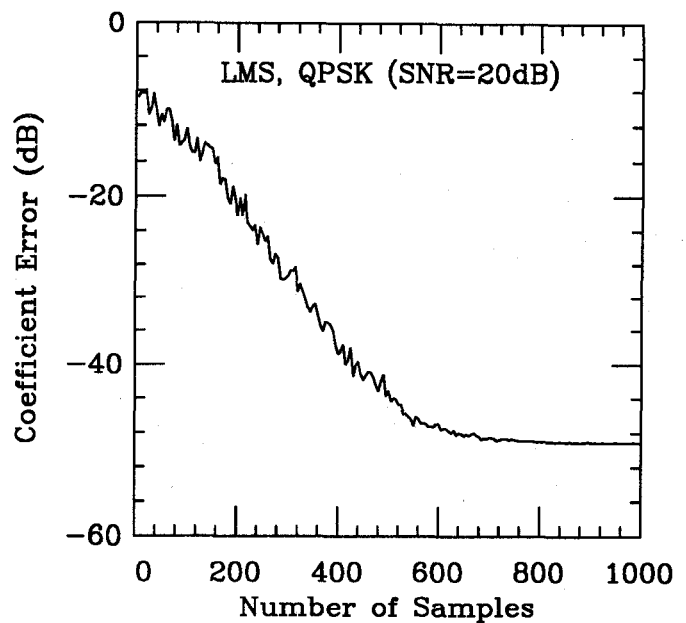
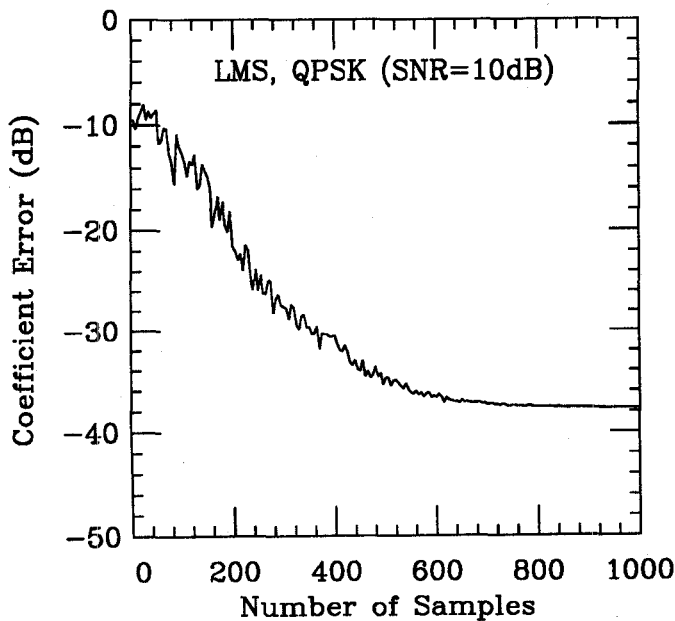


Fig. 12. Coefficient error trajectories (LMS, QPSK).

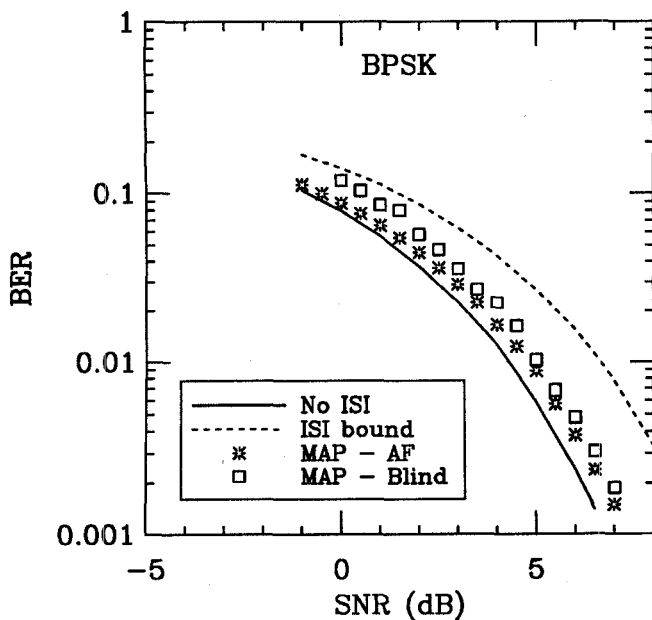


Fig. 13. BER performance curves (BPSK).

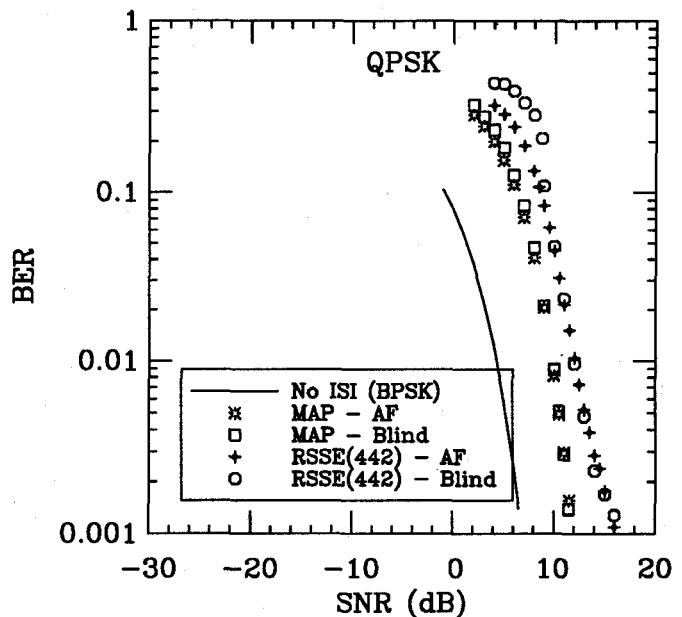


Fig. 14. BER performance curves (QPSK).

sumed, as an approximate lower bound on the performance of the adaptive Bayesian equalizer. This bound is shown for BPSK signalling by the \* symbol in Fig. 13. Also note from the figure that the LMS Bayesian equalizer ( $\square$  symbol) has nearly the same performance as that of the algorithm in [8]. Thus, as observed in Figs. 6 and 8, the channel estimates are extremely good at steady state, enabling the blind algorithm to decode the data as well as that in [8]. Fig. 14 shows the BER results of the LMS Bayesian algorithm for QPSK signalling. Observe again that the blind algorithm has a BER performance comparable to that of the algorithm in [8].

The reduced-state sequence estimator (RSSE) was also simulated in order to decrease the complexity of the filter

bank for QPSK signalling, and to observe its effect on performance. The subsets for the three channel coefficients were defined with  $M_0 = 4$ ,  $M_1 = 4$ , and  $M_2 = 2$ . Fig. 15 shows the trajectories of the probability metrics of the 32 subsets ( $M_0 M_1 M_2 = 32$ ). Again, the channel in (38) was used for both the RSSE and full-state versions of the algorithm. The corresponding coefficient error trajectory is also shown in Fig. 15. Note that even for this low-order channel, the RSSE version converges more slowly than the full version of the estimator. We expect that when the number of symbols  $M$  in the alphabet is large compared to the length of the channel, improved results will be achieved by choosing subsets with a greater intra-subset distance [6].

The bit-error rate of the RSSE version of the algorithm

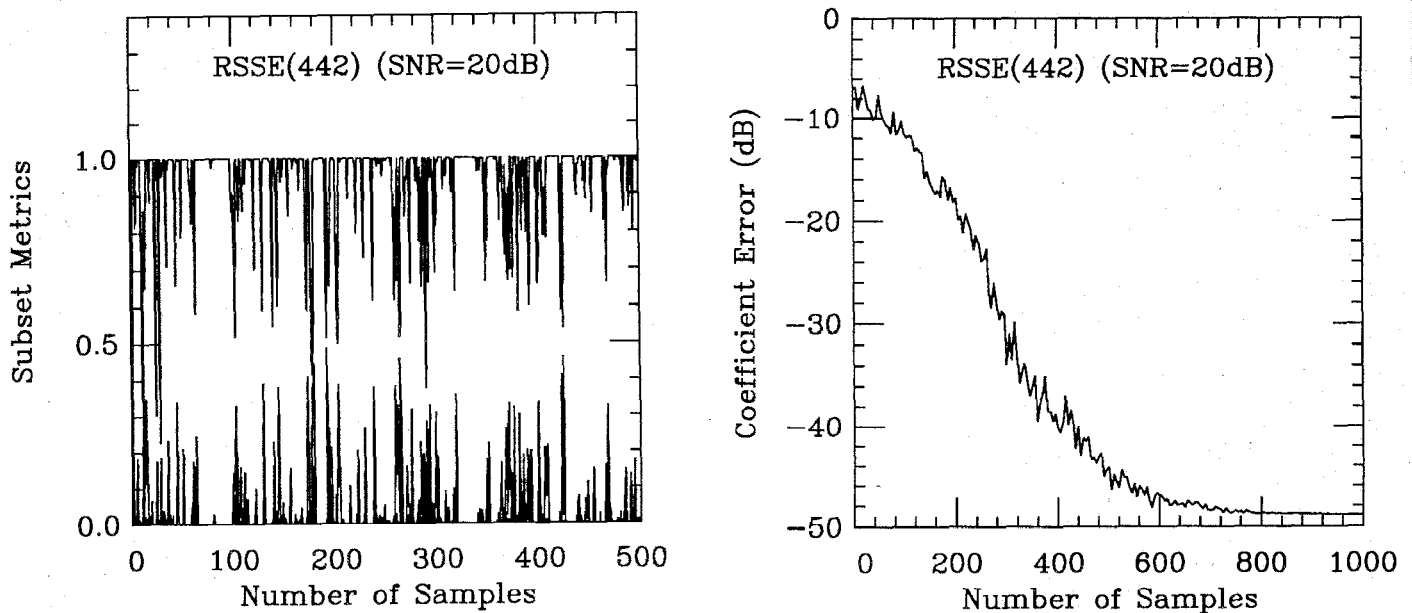


Fig. 15 Evolution of the probability metrics and coefficient error (KF, RSSE, QPSK).

using LMS adaptation is also shown in Fig. 14. For RSSE, a divergence test had to be performed by the blind equalizer, because the coefficient estimates diverged more frequently than for the full-state versions of the algorithm. The test statistic computed for LMS adaptation was

$$\bar{z} = \frac{1}{N} \sum_{k=1}^N \frac{|r(k) - \hat{r}_{i_{max}}(k)|^2}{\sigma_n^2} \quad (41)$$

where  $i_{max}$  represents the index corresponding to the largest metric. For small estimation errors,  $\hat{\mathbf{b}}(k) \approx \mathbf{b}$  and the expected value of  $\bar{z}$  is unity. When  $\bar{z}$  exceeded a threshold of 1.3 over the initial  $N = 1000$  samples of a given run, the blind equalizer declared an erasure and the simulation was restarted. It should be emphasized that this innovations test is itself "blind," in that it can be performed by the equalizer without access to either the transmitted data or true channel coefficients. Depending on the SNR, 10-15% of the runs were found to diverge for RSSE. Note that there is a 2 dB loss in performance relative to the full-state version of the algorithm.

## VIII. DISCUSSION AND CONCLUSION

A new set of blind Bayesian equalization algorithms has been presented that are approximations to the true MAP sequence estimator for *a priori* unknown channels. It was shown that the posterior density of the channel coefficients is a Gaussian sum when conditioned on the subsequence of data symbols contributing to ISI on the current symbol. A parallel Kalman filter algorithm for updating the channel estimates was derived using a unimodal Gaussian approximation for this posterior density. The algorithm consists of  $M^{N_b+1}$  conditional Kalman channel estimators whose innovations are used to update the MAP sequence probabilities. Simpler versions of the algorithm were also

considered in which the Kalman estimators are replaced by LMS adaptive filters, and reduced-state sequence estimation (RSSE) is used to reduce the number of symbol subsequences considered.

The Kalman filter equalizer provides excellent blind start-up performance, with the probability metrics and channel estimates both converging in about 40 iterations (SNR = 20 dB). For the LMS algorithm, the simulated bit-error rate (BER) was indistinguishable from that of the optimum MAP sequence estimator, in which exact knowledge of the channel is assumed. Thus, for the fixed channel, the LMS channel estimate corresponding to the largest probability metric converged almost exactly to the true channel coefficients. However, the LMS channel estimates usually converge more slowly than those of the Kalman filter version, which is to be expected since the LMS algorithm is a gradient-descent method.

The performance of the RSSE version of the algorithm was found to be inferior to that of the full-state version, with a 2 dB loss in BER performance, as well as occasional divergence of the channel estimates. While the RSSE algorithm could compute accurate channel estimates at high SNR (> 20 dB), it appears to be best suited for ISI-limited channels, as opposed to noise-limited channels. However, for high-dimension signal constellations, such as 16 QAM, RSSE offers a large computational savings, and it may have less performance loss than it does for low-dimension signal constellations (e.g., QPSK).

A computational complexity analysis was performed for the LMS Bayesian equalizer. For a 16 kbps application using QPSK, and assuming a channel duration of three coefficients, it was shown that an implementation using a programmable DSP was feasible with four parallel 50 nsec. devices. Although a detailed comparative discussion of implementation issues is beyond the scope of this paper, we suggest that the parallel structure of the algorithm, as

shown in Fig. 2, naturally lends itself to a hardware architecture employing parallel programmable DSP devices or special-purpose VLSI. Furthermore, the number of parallel estimators, and hence the amount of hardware required, can be reduced using the RSSE version of the algorithm. More recent results show that the complexity can be further reduced by combining a decision-feedback equalizer with the MAP estimator, and retaining only the  $N$  largest metrics  $p(d_i^{k,N_b}|r^k)$  at each iteration [12]. For example, the recent work in [14] indicates that only eight parallel LMS estimators are required for a QPSK application with a channel duration of  $N_b + 1 = 7$ . We can thus envision an implementation of the algorithm in Fig. 2 with the LMS estimators and likelihood computations partitioned among a relatively small number of parallel DSP devices.

To conclude, the Kalman and LMS versions of the algorithm, while computationally complex, provide extremely rapid start-up for blind equalization. Furthermore, since the Kalman algorithm assumes a time-varying channel, this type of blind equalizer may be better suited for HF modems, for example, in which large doppler spreads induce catastrophic error propagation and deteriorating channel estimates. For relatively narrowband applications, such as HF communications and voiceband data modems, the algorithms developed here may prove feasible for implementation in programmable DSP devices, and may provide significantly better performance than property-restoral blind equalization algorithms.

#### A. DERIVATION OF THE CHANNEL ESTIMATE ONE-STEP PREDICTION

In this Appendix, the one-step updates of the channel estimate and the associated error covariance matrix are derived. Recall that at time  $k$  the estimate

$$\hat{\mathbf{b}}_i(k|k-1) = E \left[ \mathbf{b}(k) | d_i^{k,N_b}, r^{k-1} \right] \quad (\text{A.1})$$

is approximated as a Gaussian vector with covariance  $\mathbf{P}_i(k|k-1)$ . The measurement updates  $\hat{\mathbf{b}}_i(k|k)$  are given by the ordinary Kalman filter equations in (23). The one-step prediction can then be updated using the following Bayesian formula:

$$\begin{aligned} \hat{\mathbf{b}}_i(k+1|k) &= E \left[ \mathbf{b}(k+1) | d_i^{k+1,N_b}, r^k \right] \\ &= \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} E \left[ \mathbf{b}(k+1) | d_j^{k,N_b}, r^k \right] \frac{p(d_j^{k,N_b}|r^k)}{M p(d_i^{k+1,N_b}|r^k)}. \end{aligned} \quad (\text{A.2})$$

The conditional expectation  $E \left[ \mathbf{b}(k+1) | d_j^{k,N_b}, r^k \right]$  is given by  $\mathbf{F}\hat{\mathbf{b}}_j(k|k)$ , which is available from the Kalman filter measurement update. The probability of subsequence  $d_i^{k+1,N_b}$  given  $r^k$  is

$$p(d_i^{k+1,N_b}|r^k) = \frac{1}{M} \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} p(d_j^{k,N_b}|r^k), \quad (\text{A.3})$$

so that the prediction update can be rewritten as

$$\hat{\mathbf{b}}_i(k+1|k) = \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} \mathbf{F}\hat{\mathbf{b}}_j(k|k) \frac{p(d_j^{k,N_b}|r^k)}{\sum_{\{m: d_m^{k,N_b} \in d_i^{k+1,N_b}\}} p(d_m^{k,N_b}|r^k)}, \quad (\text{A.4})$$

which is the result used in Table 1.

The covariance of the prediction  $\hat{\mathbf{b}}_i(k+1|k)$  is derived as follows. First, note that

$$\begin{aligned} E \left\{ \left[ \mathbf{b}(k+1) - \hat{\mathbf{b}}_i(k+1|k) \right] \times \right. \\ \left. \left[ \mathbf{b}(k+1) - \hat{\mathbf{b}}_i(k+1|k) \right]^H | d_i^{k+1,N_b}, r^k \right\} = \\ \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} E \left\{ \left[ \mathbf{b}(k+1) - \hat{\mathbf{b}}_i(k+1|k) \right] \times \right. \\ \left. \left[ \mathbf{b}(k+1) - \hat{\mathbf{b}}_i(k+1|k) \right]^H | d_j^{k,N_b}, r^k \right\} \times \\ \frac{p(d_j^{k,N_b}|r^k)}{M p(d_i^{k+1,N_b}|r^k)}. \end{aligned} \quad (\text{A.5})$$

The conditional covariances also depend on the Kalman filter measurement updates, e.g., the first term in the  $j^{\text{th}}$  conditional covariance is equal to

$$\begin{aligned} E \left[ \mathbf{b}(k+1) \mathbf{b}^H(k+1) | d_j^{k,N_b}, r^k \right] = \\ \mathbf{F}\mathbf{P}_j(k|k)\mathbf{F}^T + \mathbf{Q} + \mathbf{F}\hat{\mathbf{b}}_j(k|k)\hat{\mathbf{b}}_j^H(k|k)\mathbf{F}^T. \end{aligned} \quad (\text{A.6})$$

Since  $\hat{\mathbf{b}}_i(k+1|k)$  is a constant conditioned on  $d_j^{k,N_b}$ , we obtain for the cross terms:

$$E \left\{ \mathbf{b}(k+1) \hat{\mathbf{b}}_i^H(k+1|k) | d_j^{k,N_b}, r^k \right\} = \mathbf{F}\hat{\mathbf{b}}_j(k|k) \hat{\mathbf{b}}_i^H(k+1|k). \quad (\text{A.7})$$

Combining the above expressions yields the final form of the prediction error covariance matrix:

$$\begin{aligned} \mathbf{P}_i(k+1|k) = \\ \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} \left\{ \left( \mathbf{F}\mathbf{P}_j(k|k)\mathbf{F}^T + \mathbf{Q} \right) \times \right. \\ \left. \frac{p(d_j^{k,N_b}|r^{k-1})}{\sum_{\{m: d_m^{k,N_b} \in d_i^{k+1,N_b}\}} p(d_m^{k,N_b}|r^k)} \right\} \\ + \sum_{\{j: d_j^{k,N_b} \in d_i^{k+1,N_b}\}} \nu_{j,i}(k) \nu_{j,i}^H(k) \frac{p(d_j^{k,N_b}|r^{k-1})}{\sum_{\{m: d_m^{k,N_b} \in d_i^{k+1,N_b}\}} p(d_m^{k,N_b}|r^k)} \end{aligned} \quad (\text{A.8})$$

where the vectors  $\nu_{j,i}(k)$  are given by

$$\nu_{j,i}(k) = \left[ \hat{\mathbf{b}}_i(k+1|k) - \mathbf{F}\hat{\mathbf{b}}_j(k|k) \right]. \quad (\text{A.9})$$

These last two equations (slightly rewritten) are the final results used in Table 1.

## REFERENCES

- [1] Y. Sato, "A method of self-recovering equalization for multi-level multilevel amplitude-modulation systems," *IEEE Trans. Commun.*, vol. COM-23, pp. 679-682, June 1975.
- [2] D.N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communications systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867-1875, Nov. 1980.
- [3] J.R. Treichler and B.G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust. Speech and Sig. Proc.*, vol. ASSP-31, pp. 459-472, Apr. 1983.
- [4] A. Benveniste and M. Goursat, "Blind equalization," *IEEE Trans. Commun.*, vol. COM-32, pp. 871-883, Aug. 1984.
- [5] Z. Ding, R. A. Kennedy, B. D. O. Anderson, and C. R. Johnson, Jr., "Ill-convergence of Godard blind equalizers in data communication systems," *IEEE Trans. Commun.*, vol. 39, pp. 1313-1327, Sept. 1991.
- [6] V. M. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, pp. 13-20, Jan. 1988.
- [7] R. A. Iltis, "A Bayesian maximum-likelihood sequence estimation algorithm for a priori unknown channels and symbol timing," *IEEE Journal on Selected Areas in Commun.*, vol. 10, pp. 579-588, Apr. 1992.
- [8] K. Abend and B. D. Fritchman, "Statistical detection for communication channels with intersymbol interference," *Proc. IEEE*, vol. 58, pp. 779-785, May 1970.
- [9] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Information Theory*, vol. IT-18, pp. 363-378, May 1972.
- [10] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [11] Motorola, Inc., *DSP56000 Digital Signal Processor User's Manual*, 1986.
- [12] K. Giridhar, J. J. Shynk, and R. A. Iltis, "Bayesian/decision feedback algorithm for blind adaptive equalization," *Optical Engineering*, vol. 31, pp. 1211-1223, June 1992.
- [13] J. G. Proakis, *Digital Communications*, New York: McGraw-Hill, 1989.
- [14] K. Giridhar, J. J. Shynk, and R. A. Iltis, "Fast dual-mode adaptive algorithms for the equalization and timing recovery of TDMA signals," in *Proc. Twenty-Sixth Asilomar Conf. on Signals, Systems, and Computers*, (Pacific Grove, CA), pp. 339-345, Oct. 1992.

Ronald A. Iltis (M'86-SM'92) received the B.A. degree in Biophysics from The Johns Hopkins University in Baltimore, MD, in 1978, the M.Sc. degree in engineering from Brown University, Providence, RI, in 1980, and the Ph.D. degree in electrical engineering from the University of California, San Diego in 1984.

Since 1984, he has been with the faculty of the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, where he is currently an Associate Professor. His current research interests are in spread-spectrum communications, multisensor/multitarget tracking, and neural networks. He has also served as a consultant to government and industry in the areas of adaptive arrays and spread-spectrum communications.

Dr. Iltis was previously an Editor for the *IEEE Transactions on Communications*. He was also the recipient of the 1990 Fred W. Ellersick Award for best paper at the IEEE MILCOM Conference.

John J. Shynk (S'78-M'86-SM'91) received the B.S. degree in systems engineering from Boston University, Boston, MA, in 1979, the M.S. degree in electrical engineering and in statistics, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1980, 1985, and 1987, respectively.

From 1979 to 1982, he was a Member of Technical Staff in the Data Communications Performance Group at AT&T Bell Laboratories, Holmdel, NJ, where he formulated performance models for voiceband data communications. He was a Research Assistant from

1982 to 1986 in the Electrical Engineering Department at Stanford University where he worked on frequency-domain implementations of adaptive IIR filter algorithms. From 1985 to 1986, he was also an Instructor at Stanford teaching courses on digital signal processing and adaptive systems. Since 1987, he has been with the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, where he is currently an Associate Professor. His research interests include adaptive equalization, adaptive signal processing, neural networks, adaptive beamforming, and direction-of-arrival estimation. He served as an Associate Editor for adaptive filtering of the *IEEE Transactions on Signal Processing* and is currently an Editor for adaptive signal processing of the *International Journal of Adaptive Control and Signal Processing*.

Dr. Shynk is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.

K. Giridhar (S'90) received the B.Sc. degree in applied sciences from the P.S.G. College of Technology, Coimbatore, India, in 1985, the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1989, and the Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara, in 1993.

From June 1989 to February 1990, he was a Member of Research Staff at the Central Research Laboratory of Bharat Electronics, Ltd., Bangalore, where he worked on various performance evaluation issues of a digital sonar beamformer. From April 1990 to July 1993, he was a Research Assistant in the Department of Electrical and Computer Engineering at UCSB, where he worked on nonlinear channel equalization algorithms, and on blind adaptive estimation algorithms and their applications to mobile radio signal recovery and the joint recovery of narrowband cochannel signals. Since September 1993, he has been a Post-Doctoral Research Affiliate with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University. His current research interests include blind signal recovery algorithms, spread-spectrum cellular communications, and applications of antenna arrays to digital wireless systems.