

Transactions Papers

A Reduced-Complexity Algorithm for Combined Equalization and Decoding

Dan Raphaeli, *Member, IEEE*, and Tal Kaitz

Abstract—This paper presents a new application of a sub-optimal trellis decoding algorithm for combined equalization and decoding. The proposed algorithm can outperform the reduced-state sequence estimator (RSSE) of the same order of complexity. The algorithm, termed *estimated future decision-feedback algorithm* (EFDFA), was originally proposed for the problem of noncoherent decoding with multiple-symbol overlapped observations and is now reformulated for the problem of intersymbol interference inflicted channels. The EFDFA uses the RSSE as a building block. The performance improvement is achieved by using *estimated future* symbols in the decision process. The estimated future symbols are obtained by RSSE decoding time-reversed blocks of the input. The same technique can be used to greatly enhance the performance of the conventional decision-feedback equalizer. An analysis of the performance of the EFDFA based on the performance of the RSSE is described. The EFDFA can be configured as an adaptive equalizer capable of operating in a time-varying environment, and is shown to perform well in fading conditions. With only minor additional complexity, the EFDFA is also capable of producing soft outputs.

Index Terms—Adaptive equalizers, decision-feedback equalizers, maximum-likelihood decoding, trellis-coded modulation.

I. INTRODUCTION

IN high-speed digital transmission over band-limited channels, error performance is degraded by additive noise and intersymbol interference (ISI). The type of detection technique used to combat the latter has a great impact on error probability. It is well established that maximum-likelihood sequence estimation (MLSE), implemented by the Viterbi algorithm (VA) [1], can provide optimal performance in terms of error event probability. This technique, however, has a large computational complexity, which hinders it from use even for relatively simple channels. It is highly desirable to reduce the complexity of the detection technique while retaining near-optimal performance. One of the most powerful techniques for doing so is called reduced-state sequence estimation (RSSE) [2]. A less general form is presented in [3], and an extension to the coded case is given in [4] and [5]. The RSSE operates by searching the

path of minimal metric in the reduced-state trellis, in which several states of the original trellis are fused together. In the branch-metric computation, ISI terms not represented by the trellis states are taken from survivor history.

Although powerful, the RSSE algorithm needs a large number of states in order to obtain near-optimal performance on severely ISI-limited channels. The degradation in performance has two causes. The first is the loss of Euclidean distance at the decision point, which increases the first error-event probability. The second is error propagation (EP) related to the inherent decision-feedback mechanism of the RSSE.

The estimated future decision-feedback algorithm (EFDFA) [6] has been developed for efficient and almost lossless sub-optimal decoding of noncoherent trellis-coded modulation (NTCM) with multiple symbols overlapping observations. The EFDFA was very successful also in the noncoherent decoding of continuous phase modulation (CPM) [7]. This algorithm is based on a novel concept called *estimated future* (to be explained). For the case of NTCM, EFDFA has shown to provide much improved performance relative to the basic decision-feedback algorithm (BDFA) and to also almost eliminate EP. Motivated by this success, and since the BDFA is based on the same concept as the RSSE, it was observed that the EFDFA can use the RSSE as its decision-feedback algorithm replacing the BDFA.

The EFDFA uses RSSE as a building block. The overall algorithm outperforms the RSSE part by addressing the two causes of suboptimality mentioned above. The performance improvement is achieved by using *estimated future* symbols in the decision process. The estimated future symbols are obtained by RSSE decoding time-reversed blocks of the input. The EFDFA is about 4–5 times more computationally complex than the original RSSE. It outperforms RSSE with the equivalent degree of complexity. The gain can be translated to complexity reduction since for the same degradation the number of states in the RSSE can be reduced when using the EFDFA. The EFDFA gain is especially noted when the channel impulse response is long and its last taps are of substantial magnitude. In such cases, RSSE will need a large number of states in order to use the energy of these taps in the decision process. The same technique can be used to greatly enhance the performance of the conventional decision-feedback equalizer (DFE), which is a special case of the RSSE when using only one state.

The EFDFA can be configured as an adaptive equalizer capable of operating in a time-varying environment, and is shown

Paper approved by E. Eleftheriou, the Editor for Equalization and Coding of the IEEE Communications Society. Manuscript received November 13, 1998; revised March 15, 2000. This paper was presented in part at the IEEE Global Telecommunications Conference (GLOBECOM'95), Singapore, November 1995.

D. Raphaeli is with the Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel.

T. Kaitz is with BreezeCOM Ltd., Tel Aviv 61131, Israel.

Publisher Item Identifier S 0090-6778(00)09872-X.

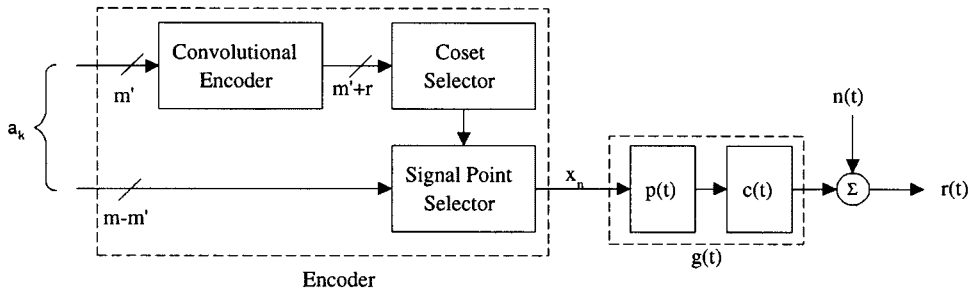


Fig. 1. The system model used.

to perform well in fading conditions. With only minor additional complexity, the EFDFA is also capable of producing soft outputs.

This paper is organized as follows. In Section II, the system model is presented. In Section III, an overview of RSSE is given. Section IV describes the proposed algorithm. Section V presents a soft output version of the EFDFA. Section VI presents an adaptive implementation, and Section VII presents some simulation results.

II. SYSTEM DESCRIPTION

A. The Transmitter and Channel

Consider the (possibly coded) quadrature amplitude modulation (QAM) transmission system shown in Fig. 1.

The transmitter side is composed of an optional convolutional encoder, a subset selector, a signal point selector, and a pulse shape filter $p(t)$. Let us denote by a_k the m -bit words that enter the transmitter, among which only m' enter the convolutional encoder. The encoder has N_{enc} states and q output symbols per branch. The symbol mapper selects the output symbols x_n from an alphabet of 2^{m+r} symbols. The symbol is shaped by the pulse shaped filter $p(t)$. In what follows, k will be used to index branch-time intervals and n for symbol time intervals. Obviously, $k = \lfloor n/q \rfloor$.

The channel is modeled as a linear filter $c(t)$ and additive white Gaussian noise (AWGN) $n(t)$ with double-sided power spectral density N_0 . The signal at the input of the receiver is

$$y(t) = \sum_{k=-\infty}^{\infty} x_k g(t - kT) + n(t) \quad (1)$$

where $g(t) \triangleq p(t) \star c(t)$ and \star denotes convolution.

B. Whitened Matched Filter (WMF)

As was shown in [1], the MLSE can be implemented by using the outputs of the WMF as sufficient statistics. The WMF shall be briefly reviewed here for purposes of future reference. First, we begin by application of a matched filter $g^*(-t)$ to the signal in (1) and sampling at $t = nT$. The resulting discrete time signal is [8, pp. 551–554]

$$z_n = \sum_{i=-S}^S h_i x_{n-i} + \nu_n \quad (2)$$

where $h_i = \int_{-\infty}^{\infty} g(t)g^*(t+iT) dt$. Here, we have assumed that $g(t)$ is such that $\{h_i\} = 0$ for $|i| > S$. Let us denote by $H(z)$ the \mathcal{Z} transform of $\{h_i\}$. The additive Gaussian noise ν_n has the power spectrum given by $P_{\nu\nu}(z) = N_0 H(z)$. Since $H(z)$ has the property $H(z) = H^*(z^{-1})$, we can always factor $H(z)$ according to

$$H(z) = F(z)F^*(z^{-1}) \quad (3)$$

where $F(z)$ is minimum phase and casual and $F^*(z^{-1})$ is maximum phase and anticausal.¹ Thus, we can apply the filter $W_{\text{frw}}(z) = 1/F^*(z^{-1})$ as a *forward whitening filter*. The reason for the term “forward” will become clear later on. The resulting channel model is given by

$$r_n = \sum_{i=0}^S f_i x_{n-i} + w_n \quad (4)$$

where r_n is the signal at the WMF output, f_i are the tap coefficients of $F(z)$, and w_n is a white Gaussian noise sequence with variance N_0 .

III. RSSE DECODING

A. Definition of the RSSE Trellis

MLSE [1] is performed by finding the input sequence $\{a_k\}_{-\infty}^{\infty}$, which minimizes the cumulative metric

$$\eta(\mathbf{x}) = \sum_{k=-\infty}^{\infty} \beta_k(\mathbf{x}) = \sum_{k=-\infty}^{\infty} \sum_{n=kq}^{(k+1)q-1} \left| r_n - \sum_{i=0}^S f_i x_{n-i} \right|^2 \quad (5)$$

where $\{x_n\}$ is the symbol sequence related to $\{a_k\}$ and $\beta_k(\mathbf{x})$ are the branch metrics. The application of the VA for implementing MLSE can be accomplished if we consider the combination of the encoder and channel as a finite-state machine having the states

$$\phi_k = \{\sigma_k, x_{n-1}, x_{n-2}, \dots, x_{n-S}\}. \quad (6)$$

Here, σ_k is the encoder state at time k , and x_{n-1} is the last symbol of the trellis branch at time $k-1$. The number of trellis states in (6) is $N_{\text{ml}} = N_{\text{enc}} 2^{mS_q}$, where N_{enc} is the number of encoder states and $S_q = \lceil S/q \rceil$ is the channel length in branches.

¹Here we have assumed that $H(z)$ has no poles on the unit circle. This restriction will be relaxed in Section VI.

In order to reduce the number of trellis states, several ϕ_k states are fused together to form the RSSE trellis. We shall now use the coset language of [9] to describe how the RSSE trellis is formed. For simplicity, we shall from now on assume that $q = 1$, unless otherwise noted.

Let us assume that the symbols $\{x_n\}$ form a subset of the group Λ . For rectangular QAM signals, $\Lambda = \mathcal{Z}^2$ where \mathcal{Z} is the integer lattice. For M -ary phase-shift keying (M-PSK) signals, the symbols are labeled by elements of the group $\Lambda = \{0, 2\pi/M, \dots, 2\pi(M-1)/M\}$ with modulo 2π addition.

Any subgroup Λ' of Λ defines the partition set Λ/Λ' into $|\Lambda/\Lambda'|$ cosets. For the definition of RSSE states, we can use the following partition chain: $\Lambda/\Lambda(S)/\Lambda(S-1)/\dots/\Lambda(\ell)/\dots/\Lambda(1)$. For each $1 \leq \ell \leq S$, the partition $\Lambda/\Lambda(\ell)$ defines $1 \leq J_\ell \leq 2^{m+r}$ cosets. We assume that the number of symbols x_n in each coset is the same. The cosets of $\Lambda/\Lambda(\ell)$ are used to label the symbol $x_{n-\ell}$ of the maximum-likelihood (ML) state given in (6). Let $c_{n-\ell}$ denote the coset to which $x_{n-\ell}$ belongs. The reduced state can then be written as

$$\mu_k = \{\sigma_k, c_{n-1}, c_{n-2}, \dots, c_{n-S}\}. \quad (7)$$

Due to the nested nature of the partition chain, (7) corresponds to a well-defined trellis. An important point to note is that in order to compute the branch metrics the information needed to complete the reduced representation of (7) to that of (6) is taken from the survivor history. Thus, a decision-feedback mechanism is introduced into the RSSE. Furthermore, whenever $J_1 < 2^{m+r}$, the trellis contains parallel transitions. The decisions between the parallel transitions are performed per branch in a DFE fashion.

It is convenient to introduce the following operator \mathbf{R} . For rectangular QAM, we define \mathbf{R} as rotation by $\pi/4$ and scaling by $\sqrt{2}$. For the M-PSK case (where M is a power of 2), \mathbf{R} is defined as modulo 2π multiplication by 2. Thus, the partition chain can be written as

$$\Lambda/\mathbf{R}^{j_S}\Lambda/\mathbf{R}^{j_{S-1}}\Lambda/\dots/\mathbf{R}^{j_\ell}\Lambda/\dots/\mathbf{R}^{j_1}\Lambda \quad (8)$$

under the provision that $j_S \leq j_{S-1} \leq \dots \leq j_1$. Since the number of cosets in the partition $\Lambda/\mathbf{R}^{j_\ell}\Lambda$ is 2^{j_ℓ} , $J_\ell = 2^{j_\ell}$.

In the uncoded case, the number of states in the RSSE trellis N_{RS} is determined solely by the partitions sets according to $N_{\text{RS}} = 2^{\sum_{\ell=1}^S j_\ell}$. In the coded case, however, there is mutual information between the encoder state and the channel state and the number of reduced trellis states is not solely determined by $\{j_\ell\}$ (see [4]).

B. Suboptimality of RSSE

Let us consider the probability for an error event $\mathbf{e} = \{e_k\}$, where $e_k = \hat{x}_k - x_k$, and where $\hat{\mathbf{x}} = \{\hat{x}_k\}$ and $\mathbf{x} = \{x_k\}$ are the decoded and transmitted symbols respectively. Let $\{\hat{\phi}_k\}$ and $\{\phi_k\}$ denote the sequence of ML states relating to $\{\hat{x}_k\}$ and $\{x_k\}$, respectively.

Let us assume that in the ML trellis \mathbf{e} ends at time k_1 . That is, $\hat{\phi}_k = \phi_k$ for $k \geq k_1$. At node k_1 , the decision between $\{x_k\}$, $\{\hat{x}_k\}$ and any other survivors are performed by comparing the metrics

$$\eta_{k_1}(\mathbf{x}) = \sum_{k=-\infty}^{k_1} \beta_k(\mathbf{x}) \quad (9)$$

to $\eta_{k_1}(\hat{\mathbf{x}})$. The Euclidean distance of the error event $\mathbf{x} \rightarrow \hat{\mathbf{x}}$ in MLSE decoding is given by

$$d_{\text{ML}}^2(\mathbf{e}) = \sum_{k=-\infty}^{k=k_1} \left| \sum_{i=0}^S f_i c_{k-i} \right|^2. \quad (10)$$

Let us now consider the same error event in the RSSE trellis. Let $\{\mu_k\}$ and $\{\hat{\mu}_k\}$ denote the RSSE states of \mathbf{x} and $\hat{\mathbf{x}}$, respectively. Because several ML states are fused into one RSSE state, the RSSE error event ends at time $k_2 \leq k_1$. Since the RSSE decision might occur prior to the optimal MLSE decision, it is therefore termed a *premature decision*.

Let us define the decision depth of the error event \mathbf{e} to be $L(\mathbf{e}) \stackrel{\text{def}}{=} k_1 - k_2$, and the (overall) decision depth of the RSSE trellis as

$$L \stackrel{\text{def}}{=} \max_{\mathbf{e} \in \mathbf{E}} L(\mathbf{e}) \quad (11)$$

where \mathbf{E} is the set of all possible error events of the trellis. The decision depth signifies how many symbols are taken from path history, rather than from state information, hence it is related to the RSSE suboptimality. As shall be shown, the value of $L(\mathbf{e})$ (together with the channel response) determines the amount of degradation in the Euclidean distance of \mathbf{e} .

Both $L(\mathbf{e})$ and L depend upon the structure of the trellis. For instance, if we take some $K < S$ and $j_1 = j_2 = \dots = j_K = m+r$ and $j_{K+1} = \dots = j_S = 0$, then we have $L = S - K$. This is the case of delayed decision feedback, as outlined in [3].

The RSSE decision at time k_2 is performed by comparing the metric

$$\eta_{k_2}(\mathbf{x}) = \sum_{k=-\infty}^{k_2} \beta_k(\mathbf{x}) \quad (12)$$

to $\eta_{k_2}(\hat{\mathbf{x}})$. Comparing (9) to (12) reveals that the RSSE decision process ignores the inputs samples $r_{k_2+1} \dots r_{k_1}$. These samples carry some of the dispersed energy of the symbols that are being decided on. Ignoring this part of the energy results in suboptimal decisions.

The Euclidean distance of the error event for the RSSE case is given by $d_{\text{RS}}^2(\mathbf{e}) = \sum_{k=-\infty}^{k=k_2} \left| \sum_{i=0}^S f_i c_{k-i} \right|^2$, and the degradation relative to the MLSE is given by

$$d_{\text{ML}}^2(\mathbf{e}) - d_{\text{RS}}^2(\mathbf{e}) = \sum_{\ell=1}^{L(\mathbf{e})} \left| \sum_{i=S-L(\mathbf{e})+\ell}^S f_i c_{k_2+\ell-i} \right|^2. \quad (13)$$

We have used the fact that $c_k = 0$ for $k = k_1 - S + 1 \dots k_1$. It can be concluded that the degradation is related to the energy embedded in the *last taps of the impulse response*, namely in $\{f_i\}_{i=S-L(\mathbf{e})+1}^S$.

The premature decisions and decision feedback also cause EP. To illustrate let us consider three Paths A, B, and C (refer to Fig. 2). Let us assume that A is the ML path. Suppose that B merges with A at time k_{1B} in the ML trellis and at time $k_{2B} < k_{1B}$ in the reduced trellis. Path C merges with Path A at times k_{1C} and k_{2C} . Suppose that at time k_{2B} , B wins over A. If the comparison would have been made at time k_{1B} , then A would surely win for it is the ML path. Now let us turn to time k_{2C} .

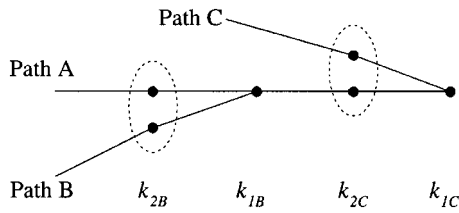


Fig. 2. Explanation of EP.

Path C might win Path B even though it would have lost against Path A. Thus, the error continues.

IV. PROPOSED ALGORITHM

A. Basic Principles

The algorithm uses a novel concept called *estimated future*, in which estimated future symbols are used to improve the RSSE decision process. To demonstrate, let us focus on the case illustrated in the previous section and consider the RSSE decision at time k_2 . Suppose that at the decision time k_2 , the L symbols $x_{k_2+1} \dots x_{k_2+L}$ belonging to ML path formed by $\mu_{k_2+1} \dots \mu_{k_2+L}$ are known. For now, we assume that node μ_{k_2} is on the ML path.

An optimal decision can be made by considering all candidates reaching μ_{k_2} and selecting the one minimizing

$$\eta_{k_2+L}(\mathbf{x}) = \sum_{k=-\infty}^{k_2+L} \beta_k(\mathbf{x}) \quad (14)$$

where the known future symbols are used for x_k for $k > k_2$. Since the decision described above involves future symbols, we term it a *with future* (WF) decision. Equation (14) can be also written as

$$\eta_{k_2+L}(\mathbf{x}) = \sum_{k=-\infty}^{k_1} \beta_k(\mathbf{x}) + \sum_{k=k_1+1}^{k_2+L} \beta_k(\mathbf{x}). \quad (15)$$

The second element in (15) is zero if $L = k_1 - k_2$ or else contains elements of \mathbf{x} which are common to all the candidates. Comparing (15) with (9) reveals that the WF decision is optimal.

The conclusion from the above discussion is that if a state is on the ML path and correct future symbols are available, then the WF decision is optimal. If, however, the state is not on the ML path, then the WF decision may not be optimal. As a result, paths with lower likelihood would survive at the states which are not on the ML path. These survivors would anyway lose when compared with the ML path in future WF decisions, and so the optimality of the final decisions would not be affected. The use of a correct known future would mitigate the effect of EP, since by optimality of previous decisions, the ML path is always present for the current one.

How can one possibly know the future? The approach is to save a block of the input signal in memory and perform the RSSE backward, starting from the end of the block. After the backward process (BP) ends, we have the survivor paths belonging to each state at each time in the trellis within the block. These paths will be used as future estimates.

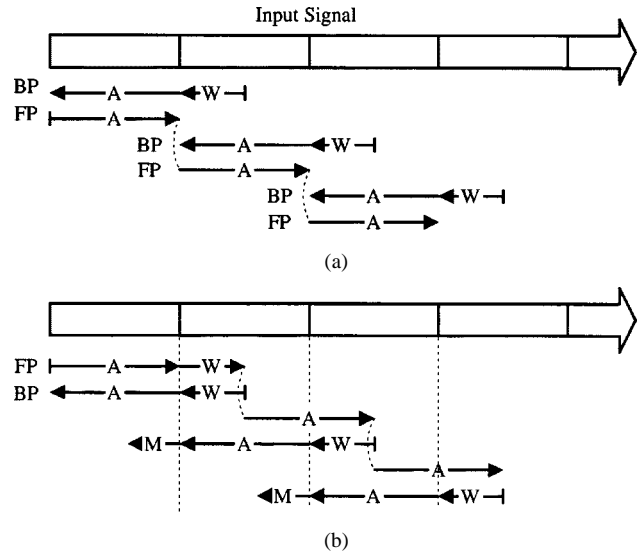


Fig. 3. Input stream processing flow. (a) Known channel mode. (b) Adaptive mode.

B. Algorithm Description

The algorithm is composed of two high-level processes. The first is the BP in which the future estimates are produced. The second is the forward process (FP) in which the future estimates are used to produce the final output of the algorithm.

The input stream is divided into overlapping blocks, each block having a length of $A+W$ input symbols and starts A symbols after the beginning of the previous block [refer to Fig. 3(a)].

The BP operates first and processes $A+W$ symbols. The BP operates in a manner similar to the RSSE. The BP produce future estimate symbols used by the WF.

The FP process the first A symbols of the block, and produces the final decisions. The FP is continuous from block to block. The blocking is intended only for the operation of the BP. The section of length W is intended for letting the BP converge from the initial conditions, in which no particular state is used as a beginning state. This convergence region can be eliminated by inserting $\log_2 N_{ml}$ known input bits at the end of each block of mA input bits and, in that way, reach a known state at the end of each block.

The FP is composed of two low-level, RSSE-like processes termed the WF process and the no-future (NF) process.

The WF operates by maintaining, for every time k and every state s , survivor and candidate lists. Each WF candidate reaching state s at time k is the concatenation of past WF decisions and the L symbols x_{k+1}, \dots, x_{k+L} taken from the BP survivor reaching state s . Thus, each WF candidate contains enough information to calculate the metric given in (15). If the backward survivor is the ML path, then the WF decision is optimal.

The NF process maintains survivor and candidate list which are composed of past decisions. At each time k , and for every state s , the NF decision involves candidates which are extended from previous NF decisions and the WF survivors from time $k-L$ whose future estimate projects on state s . The final outputs of the algorithm are derived by backtracking the NF survivor list. The decision process is such that the algorithm does not

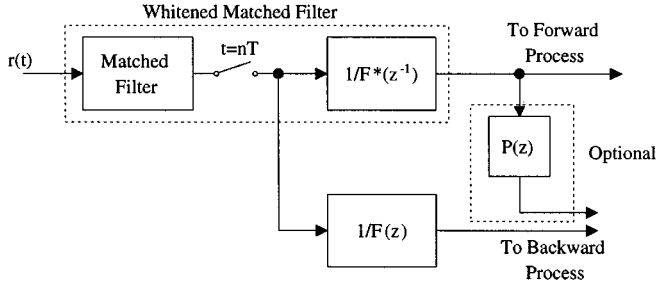


Fig. 4. Linear preprocessing for EFDFEA.

produce the optimal path, only when both the future estimate and the NF are in error.

In the following, a detailed description of the algorithm is provided. The description closely follows the notation of [6] but is extended here to allow parallel transitions. For a detailed explanation, the reader is referred to [6] and [11]. In the following, we shall denote by R the number of trellis branches reaching a state and by P the number of parallel transitions, so $RP = 2^m$ where m is the number of input bits per branch.

1) *Backward Process*: The BP is composed of an RSSE process operating backward on the block from time $A + W$ to time 1. A crucial point to note is that $F(z)$ is minimum phase and the energy of the first tap weights is maximized. Due to time reversal in the BP, it is the energy embedded in the *last* taps that is used in the decision process. The BP will have poor performance relative to the FP. The solution is to use the filter $W_{\text{back}}(z) = 1/F(z)$ of (3) as a whitening filter for the BP. The resulting impulse response will be anticausal and maximum phase. In this case, the backward WMF model is given by

$$r_n^b = \sum_{i=0}^S f_i^* x_{n+i} + v_n \quad (16)$$

where r_n^b is the signal at the BP WMF output, and v_n is a white Gaussian noise sequence with variance N_0 . Hence, the BP uses the time-reversed conjugated response of a forward RSSE. Alternatively, if only the outputs of the forward WMF are available, we can pass them through the realizable all-pass filter (refer to Fig. 4)

$$P(z) = \frac{1}{f_0} \cdot \frac{f_0^* + f_1^* z^1 + \dots + f_S^* z^S}{1 + f_1/f_0 z^{-1} + \dots + f_S/f_0 z^{-S}} \quad (17)$$

to obtain the backward WMF outputs.

The branch metrics used in the BP are given by

$$\zeta_n = \left| r_n^b - \sum_{i=0}^S f_i^* x_{n+i} \right|^2. \quad (18)$$

For each state s and for every time k , the BP keeps track of the survivors list

$$Q_k^s = \{q_{k,A+W}^s, q_{k,A+W-1}^s, \dots, q_{k,k+1}^s, q_{k,k}^s\}$$

where $q_{k,k}^s \equiv s$. The list Q_k^s would serve as a future estimate in the FP. If the trellis contains parallel transitions, then the list Q_k^s does not fully specify the associated symbols. For the explanation below, we assume that the symbols list $X_k^s = \{x_{k,A+W}^s, x_{k,A+W-1}^s, \dots, x_{k,k+1}^s, x_{k,k}^s\}$ is kept along with the

states lists, for every state in the trellis. In practical implementations, however, only the corresponding input bits should be kept. The associated accumulated metric of Q_k^s is given by

$$E_k^s = \sum_{\ell=k}^{A+W} \zeta_{\ell}. \quad (19)$$

2) *The Forward Process*: The FP operates on the input stream in a continuous manner. The FP is composed of two RSSE-like processes. The first is the WF decision process. The WF process utilizes the future estimates produced by the BP. The WF survivor at time k reaching state s is denoted by $F_k^s = \{\dots, f_{k,k-2}^s, f_{k,k-1}^s, f_{k,k}^s, f_{k,k+1}^s, \dots, f_{k,k+L}^s\}$, where $f_{k,k}^s \equiv s$ and $f_{k,k+i}^s \equiv q_{k,k+i}^s$ for $i = 1 \dots L$. The symbol list associated with the WF list is denoted by $Y_k^s = \{\dots, y_{k,k-2}^s, y_{k,k-1}^s, y_{k,k}^s, y_{k,k+1}^s, \dots, y_{k,k+L}^s\}$. The associated metric related to F_k^s is denoted by

$$H_k^s = \eta(Y_k^s). \quad (20)$$

The second process is the NF process which contains the best guess at the ML path. The NF survivor at time k is given by $C_k^s = \{\dots, c_{k,k-2}^s, c_{k,k-1}^s, c_{k,k}^s\}$ with $c_{k,k}^s \equiv s$. The symbol list associated with C_k^s is denoted by $Z_k^s = \{\dots, z_{k,k-2}^s, z_{k,k-1}^s, z_{k,k}^s\}$. The associated metric related to C_k^s is denoted by

$$G_k^s = \eta(Z_k^s). \quad (21)$$

The FP works recursively as follows. At time k , the following steps are taken.

- Step 1) For each state s , form the WF candidate $\hat{F}_{k+1}^{s,m}$ and the WF candidate symbols list $\hat{Y}_{k+1}^{s,m}$, where $m = q_{k,k+1}^s$. The candidates are formed by appending the state $q_{k,k+L+1}^s$ and the associated symbol \hat{x}_{k+L+1}^s to F_k^s and to Y_k^s , respectively.
- Step 2) Compute the metrics of the WF candidates constructed in Step 1.

$$\hat{H}_{k+1}^{s,m} = \eta(\hat{Y}_{k+1}^{s,m}) = H_k^s + \beta_{k+L+1}(\hat{Y}_{k+1}^{s,m}) \quad (22)$$

where β_{k+L+1} was defined in (5).

- Step 3) For each state s , find the next state $n_i = \text{next}(s, i)$ $i = 1 \dots R$ and form the WF candidate path \hat{F}_{k+1}^{s,n_i} by appending $\{n_i, q_{k+1,k+2}^{n_i}, q_{k+1,k+2}^{n_i}, \dots, q_{k+1,k+L}^{n_i}\}$ to C_k^s .
- Step 4) For each transition $s \rightarrow n_i$ and for all associated parallel transitions $p = 1 \dots P$, form the candidate symbols list $\hat{Y}_k^{s,n_i,p}$ by appending $\{x_n^{n_i}\}_{n=k+1}^{k+L}$ to Y_k^s .

For every transition $s \rightarrow n_i$, find the best parallel transition and set the WF candidate metric

$$\hat{H}_k^{s,n_i} = G_k^s + \min_P \sum_{\ell=1}^{L+1} \beta_{k+\ell}(\hat{Y}_k^{s,n_i,p}). \quad (23)$$

For the case of $n_i = m$, the minimization should exclude the transition for which the symbol $y_{k+1}^{s,m}$ is equal to the corresponding symbol in X_k^s .

- Step 5) For each state w , there are a total of R values $\hat{H}_{k+1}^{s_i,w}$. The index i for which $\hat{H}_{k+1}^{s_i,w}$ is minimized is used

B), no error will occur. The algorithm can fail (i.e., produce the path that is not ML) only when the NF was in error and the future estimate was wrong so the WF could not correct the NF decision.

D. Evaluation of the EFDFA Performance

In [6], an error performance analysis of the EFDFA was performed for the case of noncoherent decoding with independent overlapped near MLSE. In the following section, we shall demonstrate the applicability of this analysis for the ISI case. Here, we shall cite only several results.

In [6], the analysis was based on two probabilistic models for the dynamics of the EP. The first model, termed the *worst case* model, assumes that EP, once started, continues until the end of the block. The second model, termed the *refined* model, assumes that the EP length can be modeled as a random variable with geometrical distribution. More specifically, the probability that the length of the an EP in a forward RSSE is equal to k is given by

$$p_f(k) = \lambda_f(1 - \lambda_f)^{k-1} \quad (27)$$

where $1/\lambda_f$ is the average lifetime of the forward EP. Similarly, in a backward RSSE, we have $p_b(k) = \lambda_b(1 - \lambda_b)^{k-1}$ with $1/\lambda_b$ is the average lifetime of a backward RSSE.

Here, we shall be concerned only with the refined model. The validity of this model for the ISI case was demonstrated in [12], where the EP length distribution, derived from computer simulations, was shown to follow (27).

Now, let us introduce several variables. Let p_e and N_e denote the probability per symbol and the average number of bits in error, respectively, of an error event of the MLSE. Let p_f denote the first event-error probability of the forward RSSE (or the NF decisions) and p_b the first event-error probability of the backward RSSE. p'_f is the probability of a correctable forward error event, i.e., this error can be corrected by using a correct future estimate. Errors are uncorrectable only when the same errors would also be made by the optimal decisions. From the union bound, $p'_f \geq p_f - p_e$. Let γ_b denote the conditional probability that EP occurs in the BP given the first error occurred. γ'_f denotes the EP conditional probability given a correctable forward error event occurred. Let N'_f be the average number of bit errors in a correctable forward first error event.

As was shown in [6], the error event probability P_{seq} can be approximated by

$$P_{\text{seq}} \cong p_e + \frac{p'_f p_b \gamma_b}{\lambda_b}. \quad (28)$$

The bit-error probability is given by

$$P_{\text{bit}} \cong \frac{p_e N_e}{m} + \frac{p'_f \gamma'_f p_b \gamma_b}{\lambda_b (\lambda_b + \lambda_f)}. \quad (29)$$

It should be noted, however, that the above approximation does not serve as an upper or lower bound, and therefore the actual error rates may be higher or lower than the values predicted by (28) and (29).

E. Reduced-Complexity EFDFA

A further complexity reduction can be achieved using fewer symbols in the future estimate. Let us consider using $L' < L$ future symbols at each WF decision step and examine the effect on the probability of error. In the following, we shall consider the error event \mathbf{e} outlined in Section III-B. At time k_2 , we use the future estimate $x_{k_2}, \dots, x_{k_2+L'}$ to produce the metric [compare with (14)]

$$\eta_{k_2+L'}(\mathbf{x}) = \sum_{k=-\infty}^{k_2+L'} \beta_k. \quad (30)$$

Assuming the future estimate is correct the distance of the error event \mathbf{e} is

$$d_{L'}^2(\mathbf{e}) = \sum_{k=-\infty}^{k=k_2+L'} \left| \sum_{i=0}^S f_i e_{k-i} \right|^2. \quad (31)$$

We shall now consider a trellis that is equivalent to the reduced-complexity EFDFA. First, we note that $d_{L'}^2$ is the distance of the error \mathbf{e} if the decision was made at time $k_2 + L'$. There exists a trellis for which for all possible \mathbf{e} the decision is made at time $k_2 + L'$ and is given by

$$\Gamma_{L'} = \Lambda / \mathbf{R}^{j_{S-L'}} \Lambda / \dots / \mathbf{R}^{j_1} \Lambda / \overbrace{\mathbf{R}^{m+r} \Lambda / \dots / \mathbf{R}^{m+r} \Lambda}^{L'} \quad (32)$$

where $j_1 \dots j_S$ define the original RSSE trellis [see (8)]. We shall refer to $\Gamma_{L'}$ as the *equivalent trellis*, and denote the error probability of an RSSE defined using $\Gamma_{L'}$ by $p_{L'}$. We can conclude that the performance of the EFDFA with L' future symbols approaches the performance of that RSSE according to (28) or (29), where p_e is replaced by $p_{L'}$. Note that since $p_{L'} > p_e$, the second terms in (28) or (29) will vanish faster. By comparing (8) with (32), we have that the number of states in the equivalent trellis is given by

$$N_{\text{eq}} = N_{\text{rs}} 2^{\sum_{\ell=S-L'+1}^S m+r-j_\ell}. \quad (33)$$

If one is concerned only with asymptotical error rate that is primarily determined by the minimum distance event, then there may be a simpler trellis than $\Gamma_{L'}$, which asymptotically produces the same error rate as $\Gamma_{L'}$.

V. SOFT-OUTPUT EFDFA

The algorithm is capable of producing soft output in a way similar to the operation of the max-log-map algorithm (MLMA) [13]. The objective is to produce, for every decoded bit b_m , the log-likelihood ratio (LLR) given by

$$\Gamma(b_m) = \log \frac{\Pr(b_m = 1 | \mathbf{r})}{\Pr(b_m = 0 | \mathbf{r})} \quad (34)$$

where $\mathbf{r} = \{r_n\}$ is the received sequence. The optimal solution is given by the Bahl *et al.* algorithm [14], which is relatively

complex. In the MLMA, complexity reduction is achieved by using the approximation

$$\begin{aligned} \Pr(b_m = p | \mathbf{r}) &= \sum_{\mathbf{b}, b_m=p} \Pr(b_m = p | \mathbf{r}) \\ &\approx \max_{\mathbf{b}, b_m=p} \Pr(b_m = p | \mathbf{r}) \end{aligned} \quad (35)$$

where \mathbf{b} is the sequence of bits and p is either 0 or 1. The maximization in (35) is performed over all possible bit sequences, or equivalently over all possible paths in the trellis. Hence, the LLR can be approximated by

$$\Gamma(b_m) \approx \frac{1}{N_0} (\eta_{m,1}^{\min} - \eta_{m,0}^{\min}) \quad (36)$$

where $\eta_{m,p}^{\min}$ is the minimum accumulated metric among all the paths whose m 'th bit is p .

In the MLMA, the evaluation of $\eta_{m,p}^{\min}$ is performed as follows. Consider the ML path $\phi = \{\dots, \phi_{k-2}, \phi_{k-1}, s, \phi_{k+1}, \dots\}$ that passes through the ML state $\phi_k = s$ at time k . This path is found by performing a forward VA to find the section $\{\phi_\ell\}_{\ell=-\infty}^k$ and a backward VA to find the section $\{\phi_\ell\}_{\ell=k}^{\infty}$. The accumulated metric of ϕ is given by

$$\eta(\phi) = \eta_{-\infty}^k + \eta_k^{\infty} \quad (37)$$

where $\eta_{-\infty}^k$ and η_k^{∞} denote the metrics of the forward and backward sections, respectively. Once all the paths of time k are established and the associated metrics are computed, $\eta_{m,p}^{\min}$ can be evaluated by selecting the paths with the minimum metrics.

The EFDFA can be readily used to closely approximate the value of $\eta_{m,p}^{\min}$ by operating on a reduced trellis. This is performed by substituting the section $\{\phi_\ell\}_{\ell=-\infty}^k$ with the NF path reaching $\phi_k = \ell$ and the section $\{\phi_\ell\}_{\ell=k}^{\infty}$ with the BP path reaching the same state. The accumulated metric is approximated by

$$\eta(\phi) \approx G_k^s + E_k^s \quad (38)$$

where G_k^s is the NF metric defined by (21) and E_k^s is the BP metric defined by (19). Once $\eta(\phi)$ are computed by the EFDFA, the bit likelihoods $\Gamma(b_m)$ are computed as in the MLMA.

We should note that the approximation in (38) is threefold as follows.

- 1) The NF path is only an approximation to the optimum path reaching from the past. However, this approximation is justified since the EFDFA is nearly optimal.
- 2) The BP path is an approximation to the optimal path from the future.
- 3) Even when the BP path reaching a node is the optimum path, then E_k^s is merely an approximation to η_k^{∞} due to the minimum to maximum phase transformation. Better results are obtained when the metrics in BP are recomputed using the forward channel for a short section until the paths merge.

Nevertheless, the soft output EFDFA can provide reliability levels close to the MLMA algorithm even when it operates on a reduced trellis.

VI. ADAPTIVE IMPLEMENTATION

The equalizer structure presented in Fig. 4 has several practical limitations. First, the whitening filter $W_{\text{fwh}}(z)$ and $W_{\text{bck}}(z)$, and the allpass $P(z)$ exists only if $H(z)$ has no zeros on the unit circle. Even if this is the case, one should be concerned with the stability of these filters. Moreover, in situations where the channel is unknown or slowly varying, it is desired that both the matched filter and the whitening filters would be made adaptive.

The proposed solution is to use a predictive form equalizer structure and to use the framework of minimum mean square error (MMSE) equalization. The incoming signal $r(t)$ is processed by an MMSE linear equalizer (LE). As in many practical situations (see, for instance, [16]), the MMSE-LE is composed of an analog front end (AFE) unit and a fractional spaced feedforward equalizer. The output of the LE is given by $y_n = x_n + \xi_n$, where ξ_n is the noise sequence at the output of the MMSE. Let $R_{\xi\xi}(z)$ denote the autocorrelation function of ξ_n . The signal for the FP is derived by applying a forward linear predictor (FLP), matched to $R_{\xi\xi}(z)$, to the output of the LE. The FLP whitens the noise sequence ξ_n and introduces ISI. Let us consider the finite-length FLP given by

$$G(z) = \sum_{i=0}^{S'} g_i z^{-i} \quad (39)$$

where S' is the number of coefficients and $g_0 = 1$. The output of the FLP, denoted by r_n , is given by $r_n = \sum_{i=0}^{S'} g_i x_{n-i} + \nu_n$, where ν_n is a noise sequence composed of additive noise and residual ISI. By linear prediction theory (see, for instance, [15]), the FLP is always minimum phase.

The signal for the BP is derived by applying a backward linear predictor (BLP). By linear prediction theory, the BLP is given by $G^*(z^{-1})$. The signal at the output of the BLP is denoted by r_n^b and given by $r_n^b = \sum_{i=0}^{S'} g_i^* x_{n+i} + v_n$, where v_n is the backward noise sequence. Since $G(z)$ is minimum phase, $G^*(z^{-1})$ is maximum phase and r_n^b is well suited for the BP.

We now consider the adaptive form of the receiver. We should note that the version given in Section IV is not appropriate for adaptive operation. This is due to the fact that we must first apply the BP. If the channel varies significantly in the duration of one block, then the BP will be mismatched to the channel characteristics. The solution is to switch the roles of the forward and BPs. The algorithm works as follows [refer to Fig. 3(b)].

- 1) The FP operates on the received signal in overlapping blocks from time 1 to time $A + W$ in the first block and from time W to time $A + W$ in all other blocks.
- 2) In the FP, the symbols are decoded by an adaptive RSSE decoder, as will be described below. The equalizer adapts the tap weights of the MMSE-LE and the FLP. The outputs of the MMSE-LE and survivor paths in each nodes are saved.
- 3) After completion of the FP, the BLP tap weights are initiated according to taps of the FLP at time $A + W$.
- 4) The BP is composed of the WF and NF decisions operating as described in Section IV but in reversed direction. The future estimates are based on FP decisions. The BP

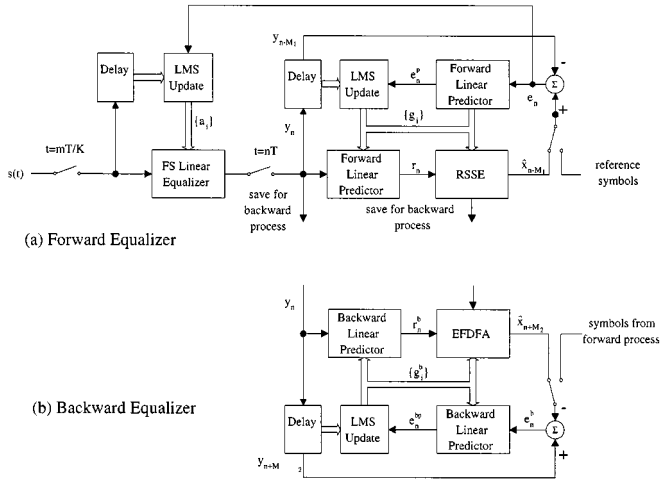


Fig. 6. Forward and backward adaptive equalizers.

operates from time $A + W$ to time $-M$, where M is the backtrack depth of the algorithm. The input to the BP is the result of the operation of the BLP on the saved LE outputs.

- 5) The BLP is updated using decisions from either the FP or BP.

The structure of the forward equalizer is given in Fig. 6(a). The LE is a $1/K$ spaced equalizer with $2N_f + 1$ taps denoted by $\{a_i\}$, whose operation is given by $y_n = \sum_{i=-N_f}^{N_f} a_i s_{nK-i}$, where $s_m = s(t)|_{t=mT/K}$. The FLP operation is given by $r_n = y_n + \sum_{i=1}^{S'} g_i y_{n-i}$.

The error signal for adaptation is derived from either known symbols in training mode or from tentative decisions in tracking mode. In the latter case, at time n , the symbol \hat{x}_{n-M_1} will be available, M_1 being the decision delay. The tap weights of the LE are adjusted so that the error $e_n = y_{n-M_1} - \hat{x}_{n-M_1}$ is minimized in mean square sense. This is performed using the following LMS update equations

$$a_i^{(n+1)} = a_i^{(n)} - \mu_1 e_n s_{(n-M_1)K-i}^* \quad (40)$$

with $-N_f \leq i \leq N_f$. For the adaptation of the FLP, we use the error signal $e_n^p = e_n + \sum_{i=1}^{S'} g_i e_{n-i}$. The update equations are

$$g_i^{(n+1)} = g_i^{(n)} - \mu_2 e_n^p y_{n-M_1-i}^*, \quad 1 \leq i \leq S'. \quad (41)$$

The step-size μ_2 is usually taken to be $\mu_2 \ll \mu_1$ since the energy of e_n is much smaller than that of s_m .

The structure of the backward equalizer is given in Fig. 6(b). The operation of the BLP is given by $r_n^b = y_n + \sum_{i=1}^{S'} g_i^b y_{n+i}$, where $\{g_i^b\}$ denote the tap weights of the BLP. The error signal for adaptation of the BLP is $e_n^{pb} = e_n^b + \sum_{i=1}^{S'} g_i^b e_{n+i}^b$, where $e_n^b = \hat{x}_{n+M_2} - y_{n+M_2}$ and M_2 is the decoding delay. The LMS update equations for the BLP are

$$g_i^{b(n+1)} = g_i^{b(n)} - \mu_3 e_n^{pb} y_{n+M_2+i}^*. \quad (42)$$

The symbols \hat{x}_{n+M_2} can be taken from NF decisions or as zero delay decisions from the FP. Since the NF decision are much more reliable than the forward decisions, one can trade reliability with adaptation loop delay.

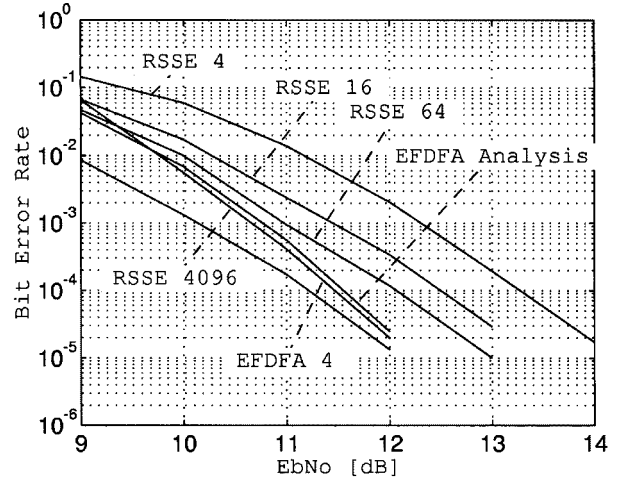


Fig. 7. BER performance of 4-state TCM.

VII. SIMULATIONS RESULTS

We present some simulation experiments to demonstrate the algorithm and analysis presented in previous sections.

A. Trellis Coded Modulation

A four-state trellis code was used in conjunction with an 8PSK modulation. We use a randomly selected channel of 11 taps: $\{f_i\} = \{1, -0.38 + 0.23j, 0.17 + 0.15j, -0.21 + 0.14j, -0.07 - 0.04j, 0.12 + 0.22j, 0.14 - 0.176j, -0.27 + 0.43j, 0.09 + 0.1j, -0.21 + 0.26j, 0.12 - 0.24j\}$. The performance of the four-state EFDFA was compared with the following algorithms: four-state RSSE based on the encoder trellis ($j_1 = 1$); 16-state RSSE ($j_1 = j_2 = 2, j_3 = 1$); 64 states RSSE ($j_1 = \dots = j_4 = 2, j_5 = 1$); and MLSE approximated by a 4096-state RSSE.

The simulation results are shown in Fig. 7. It can be observed that at $P_b = 10^{-4}$, EFDFA has about 1-dB improvement over 16-state RSSE and 0.7 dB over 64-states RSSE. The EFDFA loss relative to MLSE is less than 0.3 dB. It was observed that the average EP length for FP and BP was much less than the block length. Under this condition, the refined model is valid, and (28) and (29) can be used to obtain performance estimations. The estimated bit-error rates (BERs) are also shown in Fig. 7.

The simulation results of the reduced-complexity EFDFA with $L' = 2$ and $L' = 4$ are shown in Fig. 8.

It was observed that performance was slightly better than the performance of the 16-state RSSE and 64-state RSSE, respectively. We have found out that these same RSSE schemes are equivalent to the reduced-complexity EFDFA in the minimum distance sense for the above channel. The equivalent RSSE defined in (32) has 64 and 256 states, respectively, and has slightly better performance (not shown). The performance curves of the full-complexity EFDFA with $L' = L = 10$ are plotted for reference.

B. Uncoded 16 QAM

Here, we demonstrate the operation of a one-state EFDFA for an uncoded 16-QAM system. The channel under study is randomly selected and is $\{f_i\} = \{1, -0.36 + 0.015j, -0.22 -$

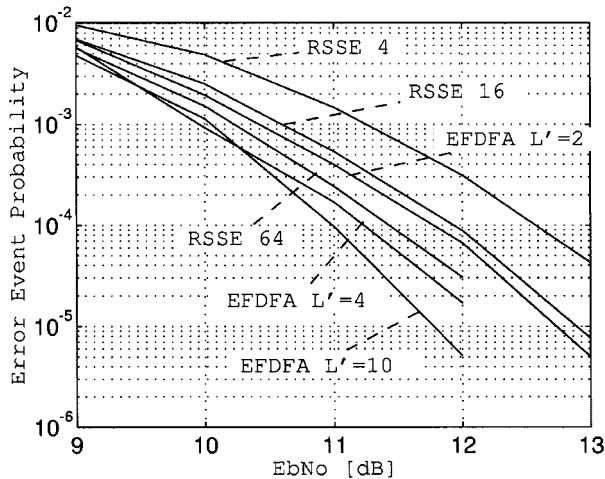


Fig. 8. Performance of the reduced-complexity EFDFA.

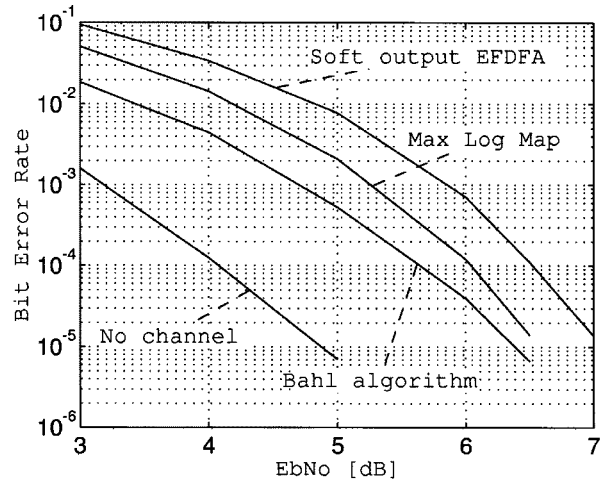


Fig. 10. Performance of BA, MLMA, and soft-output EFDFA.

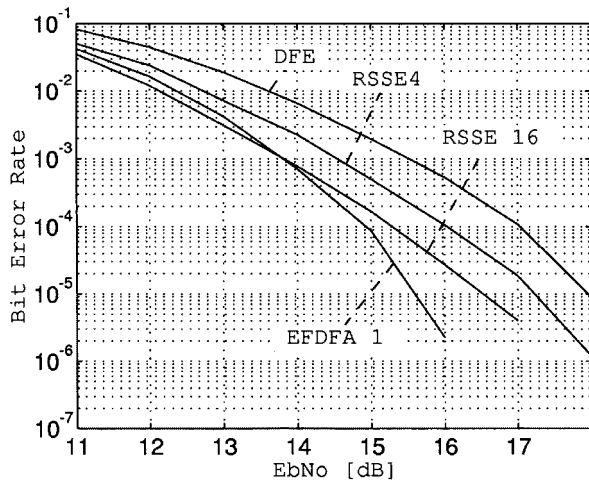


Fig. 9. BER performance of one-state EFDFA.

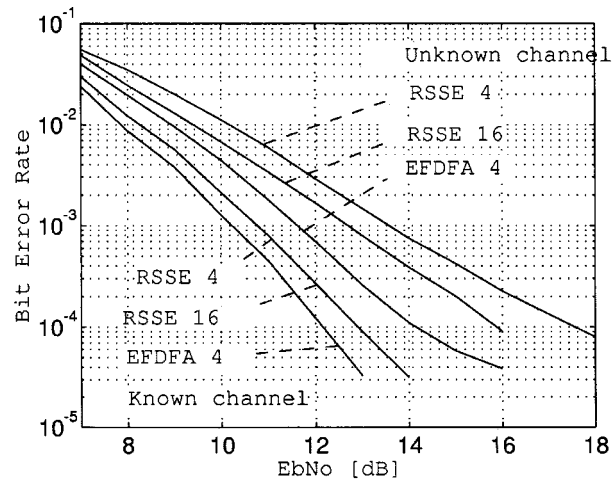


Fig. 11. Performance in adaptive mode.

$0.045j, -0.15 - 0.04j, 0.12 + 0.3j, -0.085 - 0.16j, 0.27 - 0.12j, -0.25 - 0.14j, 0.04 + 0.18j\}$.

The BER performance curves of the conventional DFE and the one state EFDFA with 16-QAM transmission are given in Fig. 9. For reference, the performance curves of the 4- and 16-state RSSE are also included.

At BER levels of 10^{-5} , it can be observed that the EFDFA outperforms the DFE by about 2.1 dB and the 4- and 16-state RSSE by about 1.5 dB and 0.8 dB, respectively.

C. Soft-Output EFDFA

The performance of the soft-output EFDFA was demonstrated on the following system. The transmitter side was composed of a 32-states, rate-1/2 convolutional encoder with generator polynomials $\{23, 35\}$ in octal, an interleaver, and a QPSK modulator. The receiver side was composed of a soft-output equalizer, a deinterleaver, and a soft-input VA decoder. The channel simulated was $\{1, 0.26 - 0.61j, -0.25 + 0.4j, 0.255 - 0.15j\}$.

The four-state soft-output EFDFA using the trellis $j_1 = 2$ was compared with a 64-state MLMA using the MLSE trellis.

The results are presented in Fig. 10. It can be observed that the EFDFA loss is about 0.5 dB. In that figure, the BERs of the Bahl algorithm are also shown, along with the performance under flat channel conditions.

D. Adaptive EFDFA

Here, the adaptive form of the EFDFA was simulated over a slowly-varying fading channel. The channel was modeled by a six-ray model. Each ray was independently produced by a white Gaussian noise sequence filter by a second-order Butterworth low-pass filter with 3-dB point at $2 \cdot 10^{-4}$. Here, $N_f = 16$ and $S' = 8$. Adaptation step-sizes were set at $\mu_1 = 5 \cdot 10^{-4}$ and $\mu_2 = \mu_3 = 10^{-3}$. Backtrack depth is $M_1 = M_2 = 10$. The four-state EFDFA was compared with 4- and 16-state RSSE. The results are shown in Fig. 11. To investigate the effect of channel tracking, the three algorithms were also simulated with the channel and noise variance assumed known to the receiver. In this case, the exact MMSE equations were solved for every step of the algorithms. It can be observed that EFDFA was about 1.6 dB better than the 16-state RSSE in the adaptive case and about 1 dB better in the known channel case.

VIII. CONCLUSION

We show that the EFDFA can be applied to the combined decoding and equalization of channels with ISI. The EFDFA, which is built around the RSSE, has much better performance than the RSSE of equivalent complexity. Convenient analytic expressions, based on the RSSE parameters, can accurately estimate the performance of the EFDFA. A reduced-complexity version of the EFDFA is shown to be equivalent with a high-complexity RSSE. The EFDFA is capable of producing soft-output information at reliability levels close to those of the MLMA. An adaptive version of the algorithm was proposed, and it outperforms the RSSE on a fading multipath channel.

REFERENCES

- [1] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.
- [2] M. V. Eyuboğlu 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.
- [3] A. Duel-Hallen and C. Heegard, "Delayed decision feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–435, May 1989.
- [4] M. V. Eyuboğlu and S. U. H. Qureshi, "Reduced state sequence estimation for coded modulation on intersymbol interference channels," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 989–995, Aug. 1989.
- [5] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 37, pp. 669–676, July 1989.
- [6] D. Raphaeli, "Decoding algorithms for the noncoherent coded modulation," *IEEE Trans. Commun.*, vol. 44, pp. 312–323, Mar. 1996.
- [7] D. Raphaeli and D. Divsalar, "Noncoherent detection of continuous phase modulation using overlapped observations," in *Proc. IEEE Communication Theory Mini-Conf. (GLOBECOM'94)*, San Francisco, CA, Nov. 1994, pp. 191–195.
- [8] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989.
- [9] G. D. Forney, "Coset codes—Part I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123–1151, Sept. 1988.
- [10] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–66, Jan. 1982.
- [11] D. Raphaeli, "Noncoherent Coded Modulation," Ph.D. dissertation, California Inst. Technol., Pasadena, 1994.
- [12] T. Kaitz, "A Reduced Complexity Algorithm for Combined Equalization and Decoding," M.Sc. thesis, Tel Aviv University, Tel Aviv, Israel, 1998.
- [13] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *Proc. GLOBECOM'91*, pp. 1001–1005.
- [14] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [15] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [16] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1386, Sept. 1985.



Dan Raphaeli (S'93–M'95–SM'99) was born in Israel in 1967. He received the B.Sc. degree in electrical and computer engineering from Ben Gurion University, Israel, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1992 and 1994, respectively.

From 1986 to 1991, he was a Research Member at the Electronic Research Institute of the Israel Defense Ministry, where he was involved in the development of many advanced communication and signal processing projects. From 1992 to 1994, he was at the Jet Propulsion Laboratory, Pasadena, CA, where he was involved in research on communication systems for future spacecrafts. Since 1994, he has been an Assistant Professor at the Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv, Israel. He is also a Professional Consultant to industry in modem design for wireless and twisted-pair high-speed communications and DSP. His research subjects include modulation/demodulation, turbo codes, coding and decoding algorithms, spread spectrum, mobile communication, synchronization, equalization, and digital signal processing.



Tal Kaitz was born in Tel Aviv, Israel, in December 1966. He received the B.Sc. degree in electrical engineering from the Technion Technical Institute, Haifa, Israel, in 1984, and the M.Sc. degree in electrical engineering from the Tel Aviv University, Tel Aviv, Israel, in 1998.

He is with BreezeCOM Wireless Communications Ltd., Tel Aviv, Israel. His research interests include equalization techniques, error control coding, modulation techniques for wireless LAN's, and VLSI structures for signal processing.