

Robust Iterative Noncoherent Reception of Coded FSK over Block Fading Channels

Shi Cheng, *Student Member, IEEE*, Matthew C. Valenti, *Member, IEEE*, and Don Torrieri, *Senior Member, IEEE*

Abstract—An iterative noncoherent receiver is developed for bit-interleaved coded orthogonal multiple frequency-shift keying (FSK) over block fading channels. The receiver uses the Expectation Maximization (EM) algorithm to jointly estimate the received amplitude and noise spectral density of each block. The received symbols and their corresponding channel estimates are passed through a soft-output demapper, deinterleaved, and decoded. Soft-outputs from the decoder are passed back to the channel estimator and demapper to refine estimates of the channel and bit likelihoods, respectively. Several techniques for reducing the estimator's complexity are discussed, and the performance is assessed through simulation.

Index Terms—EM (expectation maximization) algorithm, ML (maximum likelihood) estimation, NFSK (noncoherent frequency shift keying), turbo code.

I. INTRODUCTION

NONCOHERENT FSK is an attractive modulation when the phase changes too quickly to be tracked. To improve performance, higher-order orthogonal FSK signal sets can be used along with channel coding, though this comes at the expense of spectral efficiency. While modulation and coding could be combined into a single operation, a more pragmatic approach is to use a *binary* encoder followed by a bitwise interleaver prior to performing FSK modulation. Such an approach, known as *bit interleaved coded modulation* (BICM), can offer performance benefits in fading channels [1]. The performance of a BICM system can be improved by feeding *a priori* information from the decoder back to the demodulator, as proposed in [2] where it was termed *BICM with iterative decoding* (BICM-ID). In [3], BICM-ID was applied to systems with FSK modulation and turbo coding, and gains of up to 1 dB were observed relative to BICM.

The receiver developed in [3] relied on the availability of perfect estimates of the noise variance and the received signal amplitude. In practice, this information is not known *a priori* and must be estimated at the receiver. In an iterative receiver, a reasonable approach is to feed back extrinsic information from the decoder back to a channel estimator [4]. For a good overview of iterative decoding and channel estimation, see [5]

Manuscript received February 16, 2006; revised May 20, 2006; accepted June 30, 2006. The associate editor coordinating the review of this letter and approving it for publication was E. Serpedin. This paper was presented in part at the 2005 IEEE Military Communications Conference (Milcom). Matthew Valenti was supported in part by the Xenotran Corporation under phase-II SBIR contract number FA8750-04-C-0274.

S. Cheng and M. Valenti are with the Lane Department of CSEE, West Virginia University, Morgantown, WV 26506 USA (e-mail: {mvalenti, shic}@csee.wvu.edu).

Don Torrieri is with the Army Research Laboratory, Adelphi, MD 20783 USA (email: dtorr@arl.army.mil).

Digital Object Identifier 10.1109/TWC.2007.06020004.

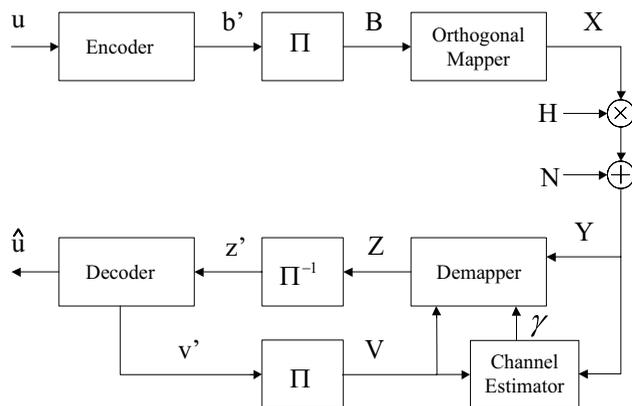


Fig. 1. Discrete-time system model. Π represents the BICM interleaver and Π^{-1} the deinterleaver.

and the references therein.

This paper extends [3] by including an iterative channel estimator in the receiver structure. To facilitate the development of a pragmatic estimator, it is assumed that the channel experiences block fading. Blocks of N consecutive FSK symbols are attenuated by the same channel gain (though they could experience different phase shifts) and are corrupted by noise that is stationary for the duration of the block. Aside from this block fading condition, the estimator makes no assumptions regarding the statistics of the channel. Both the received fading amplitude and the noise spectral density are estimated because either one or both could change from block-to-block due to jamming, interference, or other environmental conditions. The estimator itself is derived using the expectation maximization (EM) algorithm [6], which iteratively finds the maximum likelihood (ML) estimate even though an explicit form is not readily achievable when extrinsic information is fed back to the estimator from the decoder.

After presenting the system model in Section II and summarizing the receiver implementation in Section III, an EM-based estimator is derived in Section IV. Complexity reduction techniques are discussed in Section V and simulation results given in VI. After a discussion of algorithmic complexity in Section VII, the paper concludes in Section VIII.

II. SYSTEM MODEL

In the discrete-time system model shown in Fig. 1, a vector $\mathbf{u} \in \{0, 1\}^{N_u}$ of message bits is passed through a binary encoder to produce a codeword $\mathbf{b}' \in \{0, 1\}^{N_b}$. The codeword is passed through an interleaver, which permutes the order of the code bits and reshapes the permuted vector into a matrix \mathbf{B} with $\mu = \log_2 M$ rows and N_d columns, where M is the symbol alphabet size. \mathbf{B} is then transformed into a

length- N_d row vector \mathbf{d} , with elements $d_i \in \{0, 1, \dots, M-1\}$. Element d_i of \mathbf{d} is found from column \mathbf{b}_i of \mathbf{B} through a mapping function $d(\mathbf{b})$. Because the signaling is orthogonal, the mapping function is irrelevant. Each element d_i is used to select the tone sent during the i^{th} signaling interval. This is accomplished by transforming the vector \mathbf{d} into the M by N_d matrix $\mathbf{X} = [\mathbf{x}_0, \dots, \mathbf{x}_{N_d-1}]$, where the column vector \mathbf{x}_i is equal to $\sqrt{\mathcal{E}_s}$ in position d_i and has zeros elsewhere.

The modulated signal passes through a frequency-nonselective fading channel with additive Gaussian noise. The receiver front-end downconverts the signal and passes it through a bank of $2M$ matched filters, a quadrature pair for each of the M possible transmitted tones [7]. The output of the matched filters are sampled at the symbol rate (assuming perfect synchronization) and each quadrature pair is represented as a complex scalar value. The complex samples are then placed into an $M \times N_d$ matrix \mathbf{Y} whose i^{th} column represents the outputs of the matched filters corresponding to the i^{th} received symbol.

The channel estimator is derived under the assumption of a *block-fading* channel. More specifically, it is assumed that blocks of N contiguous symbols experience the same fading amplitude, though the symbols in the block could experience different phase shifts. An appropriate choice for N is to equate it to the coherence time of the channel [8]. Because the estimator works on a block-by-block basis, the correlations among blocks are irrelevant. Furthermore, it is assumed that while the noise spectral density is constant for the duration of a block, it could vary from one block to the next in an arbitrary manner. If there are N symbols per fading block, then there will be $L = \lceil N_d/N \rceil$ blocks per codeword. The matrix \mathbf{Y} can be partitioned according to $\mathbf{Y} = [\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_{L-1}]$, where the M by N submatrix \mathbf{Y}_ℓ corresponds to the ℓ^{th} fading block.

During the ℓ^{th} block, the channel is represented by the $N \times N$ diagonal matrix $\mathbf{H}_\ell = a_\ell \text{diag}(e^{j\theta_0}, \dots, e^{j\theta_{N-1}})$, where a_ℓ is the (real-valued) fading amplitude during the ℓ^{th} block, and the θ_i 's are independent and identically distributed (i.i.d.) over the range $[0, 2\pi)$. The ℓ^{th} block at the output of the receiver front-end is then $\mathbf{Y}_\ell = \mathbf{X}_\ell \mathbf{H}_\ell + \mathbf{N}_\ell$, where \mathbf{X}_ℓ consists of the corresponding columns of \mathbf{X} and \mathbf{N}_ℓ is a $M \times N$ noise matrix whose elements are i.i.d. circularly symmetric complex Gaussian variables with zero mean and variance $N_0^{(\ell)}$.

Following [7], we can represent the conditional probability density function (pdf) of the $(k, i)^{\text{th}}$ entry of \mathbf{Y} given that the transmitted symbol is $d_i = j$, the symbol energy is \mathcal{E}_s , the fading amplitude is a , and the noise spectral density is N_0 as

$$p(y_{k,i} | d_i = j, \mathcal{E}_s, a, N_0) = \frac{1}{\pi N_0} \exp\left(-\frac{|y_{k,i}|^2 + a^2 \mathcal{E}_s \delta_{k,j}}{N_0}\right) I_0\left(\frac{2a\sqrt{\mathcal{E}_s} |y_{k,i}| \delta_{k,j}}{N_0}\right) \quad (1)$$

where $\delta_{k,j}$ is the Kronecker delta function ($\delta_{k,j} = 1$ if $k = j$, otherwise $\delta_{k,j} = 0$) and I_ν is the modified Bessel function of the first kind and order ν .

Let $B_\ell = 2a_\ell \sqrt{\mathcal{E}_s}$ and $A_\ell = N_0^{(\ell)}$. It follows from (1) that the conditional pdf of block \mathbf{Y}_ℓ given A_ℓ , B_ℓ , and the corresponding set of transmitted symbols $\mathbf{d}_\ell = [d_{\ell N}, \dots, d_{(\ell+1)N-1}]$

is

$$p(\mathbf{Y}_\ell | \mathbf{d}_\ell, A_\ell, B_\ell) = \left(\frac{1}{\pi A_\ell}\right)^{NM} \exp(\psi_\ell - \xi_\ell) \quad (2)$$

where

$$\psi_\ell = \sum_{i=\ell N}^{(\ell+1)N-1} \log I_0\left(\frac{B_\ell |y_{d_i,i}|}{A_\ell}\right) \quad (3)$$

$$\xi_\ell = \frac{1}{A_\ell} \left[C_\ell + \frac{NB_\ell^2}{4} \right] \quad (4)$$

$$C_\ell = \sum_{i=\ell N}^{(\ell+1)N-1} \sum_{k=0}^{M-1} |y_{k,i}|^2. \quad (5)$$

III. RECEIVER OVERVIEW

The channel observation matrix \mathbf{Y} is passed to the receiver back-end, which comprises three main processing modules: a channel estimator, a demapper, and a soft-input/soft-output (SISO) decoder. The channel estimator uses \mathbf{Y} and *a priori* information fed back to it from the decoder to produce the ratio $\gamma_\ell = \hat{B}_\ell / \hat{A}_\ell$ of channel estimates for the ℓ^{th} block. A full description of the estimator is given in Section IV.

The demapper and decoder exchange extrinsic information in a turbo-processing loop. Following [3], the demapper output is a μ by N_d matrix \mathbf{Z} whose $(k, i)^{\text{th}}$ element is

$$z_{k,i} = \log \frac{p(b_{k,i} = 1 | \mathbf{y}_i, \gamma_{\lfloor i/N \rfloor}, \mathbf{v}_i \setminus v_{k,i})}{p(b_{k,i} = 0 | \mathbf{y}_i, \gamma_{\lfloor i/N \rfloor}, \mathbf{v}_i \setminus v_{k,i})}, \quad (6)$$

where \mathbf{v}_i is the i^{th} column of \mathbf{V} , a μ by N_d matrix output by the SISO decoder. The conditioning in $\mathbf{v}_i \setminus v_{k,i}$ implies that the extrinsic information for bit $b_{k,i}$ is produced without using $v_{k,i}$. The $(k, i)^{\text{th}}$ element of \mathbf{V} is

$$v_{k,i} = \log \frac{p(b_{k,i} = 1 | \mathbf{Z} \setminus z_{k,i})}{p(b_{k,i} = 0 | \mathbf{Z} \setminus z_{k,i})}, \quad (7)$$

which is derived for SISO decoders in [9]. The derivation of (6) for noncoherent FSK is given in [3] and can be expressed as

$$z_{k,i} = \log \frac{\sum_{d \in \mathcal{D}_k^{(1)}} I_0(\gamma_{\lfloor i/N \rfloor} |y_{d,i}|) \prod_{\substack{j=0 \\ j \neq k}}^{\mu-1} \exp(b_j(d) v_{j,i})}{\sum_{d \in \mathcal{D}_k^{(0)}} I_0(\gamma_{\lfloor i/N \rfloor} |y_{d,i}|) \prod_{\substack{j=0 \\ j \neq k}}^{\mu-1} \exp(b_j(d) v_{j,i})}, \quad (8)$$

where $\mathcal{D}_k^{(b)}$ is the set of all symbols labelled with $b_k = b$, and $b_j(d)$ is the value of the j^{th} bit in the labelling of symbol d .

IV. CHANNEL ESTIMATOR

To compute $\gamma_\ell = \hat{B}_\ell / \hat{A}_\ell$, the estimator directly uses the channel observation for the ℓ^{th} block, \mathbf{Y}_ℓ , while the observations of the other blocks are used indirectly through feedback of extrinsic information from the decoder. Since the form of the estimation algorithm is the same for each block, in the following discussion we can drop the dependence on ℓ . Thus in this section, \mathbf{Y} is a generic $M \times N$ received block,

$\mathbf{d} = [d_0, \dots, d_{N-1}]$ is the corresponding set of transmitted symbols, and $\{\hat{A}, \hat{B}\}$ are the corresponding channel estimates.

Although a direct maximum-likelihood estimation is impractical, the expectation-maximization (EM) algorithm is an appropriate iterative approach to estimating $\{\hat{A}, \hat{B}\}$ [6]. Let $\{\mathbf{Y}, \mathbf{d}\}$ denote the *complete* data set, which using (2) has log-likelihood

$$\begin{aligned} L(A, B) &= \log p(\mathbf{Y}, \mathbf{d} | A, B) \\ &= \log p(\mathbf{Y} | A, B, \mathbf{d}) + \log p(\mathbf{d}) \\ &\sim -MN \log A - \frac{C}{A} - \frac{NB^2}{4A} \\ &\quad + \sum_{i=0}^{N-1} \log I_0 \left(\frac{B |y_{d_i, i}|}{A} \right), \end{aligned} \quad (9)$$

where \sim is used to indicate that the quantities are equal up to irrelevant quantities that do not affect the maximization, namely the terms $-NM \log \pi$ and $\log p(\mathbf{d})$.

Let q denote the EM iteration and $\hat{A}^{(q)}, \hat{B}^{(q)}$ denote the estimates of A, B after the q^{th} iteration. Iteration q starts with the *E-step*

$$Q(A, B) = E_{\mathbf{d} | \mathbf{Y}, \hat{A}^{(q-1)}, \hat{B}^{(q-1)}} [L(A, B)] \quad (10)$$

where the expectation is taken with respect to the unknown symbols \mathbf{d} conditioned on \mathbf{Y} and the estimates $\hat{A}^{(q-1)}, \hat{B}^{(q-1)}$ from the last EM iteration. Substituting the likelihood function (9) into (10) yields

$$\begin{aligned} Q(A, B) &= -MN \log A - \frac{C}{A} - \frac{NB^2}{4A} + \\ &\quad \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} p_{k,i}^{(q-1)} \log I_0 \left(\frac{B |y_{k,i}|}{A} \right) \end{aligned} \quad (11)$$

where

$$\begin{aligned} p_{k,i}^{(q-1)} &= p(d_i = k | \mathbf{y}_i, \hat{A}^{(q-1)}, \hat{B}^{(q-1)}) \\ &= \frac{p(\mathbf{y}_i | d_i = k, \hat{A}^{(q-1)}, \hat{B}^{(q-1)}) p(d_i = k)}{p(\mathbf{y}_i | \hat{A}^{(q-1)}, \hat{B}^{(q-1)})}. \end{aligned} \quad (12)$$

The last step uses the fact that d is independent of A and B . Applying (1), we obtain

$$p_{k,i}^{(q-1)} = \alpha_i^{(q-1)} I_0 \left(\frac{\hat{B}^{(q-1)} |y_{k,i}|}{\hat{A}^{(q-1)}} \right) p(d_i = k) \quad (13)$$

where $\alpha_i^{(q-1)}$ is the normalization factor forcing $\sum_{k=0}^{M-1} p_{k,i}^{(q-1)} = 1$, i.e.

$$\alpha_i^{(q-1)} = \frac{1}{\sum_{k=0}^{M-1} I_0 \left(\frac{\hat{B}^{(q-1)} |y_{k,i}|}{\hat{A}^{(q-1)}} \right) p(d_i = k)} \quad (14)$$

and $p(d_i = k)$ is found from the a priori input \mathbf{v}_i using [10]

$$p(d_i | \mathbf{v}_i) = \prod_{j=0}^{\mu-1} \frac{e^{v_{j,i} b_j(d_i)}}{1 + e^{v_{j,i}}}. \quad (15)$$

The *M-step* is

$$\hat{A}^{(q)}, \hat{B}^{(q)} = \arg \max_{A, B} Q(A, B) \quad (16)$$

which can be found by setting the derivatives of the function $Q(A, B)$ with respect to A and B to zero. The solution to the corresponding system of equations is

$$\begin{aligned} \hat{A}^{(q)} &= \frac{1}{MN} \left(C - \frac{N(\hat{B}^{(q)})^2}{4} \right) \\ \hat{B}^{(q)} &= \frac{2}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} p_{k,i}^{(q-1)} |y_{k,i}| F \left(\frac{4MN\hat{B}^{(q)} |y_{k,i}|}{4C - N(\hat{B}^{(q)})^2} \right), \end{aligned} \quad (17)$$

$$(18)$$

where $F(x) = I_1(x)/I_0(x)$. While a closed form solution to (18) is difficult to obtain, it can be found recursively [11].

To select an initial estimate for B prior to the first BICM-ID iteration, consider that in the absence of noise, $y_{k,i} = a\sqrt{\mathcal{E}_s} \delta_{k,d_i} e^{j\theta_i}$, which has a magnitude of either $|y_{k,i}| = a\sqrt{\mathcal{E}_s}$ (when $k = d_i$) or $|y_{k,i}| = 0$ (otherwise). Thus, an estimate for $a\sqrt{\mathcal{E}_s} = B/2$ can be achieved by taking the maximum $|y_{k,i}|$ over any column of \mathbf{Y} . To account for noise, the average could be taken across all columns in the block, resulting in

$$\hat{B}^{(0)} = \frac{2}{N} \sum_{i=0}^{N-1} \max_k |y_{k,i}|. \quad (19)$$

The initial estimate of A is found from $\hat{B}^{(0)}$ by evaluating (17) for $q = 0$. After the initial values $\hat{A}^{(0)}$ and $\hat{B}^{(0)}$ are calculated, the initial probabilities $\{p_{k,i}^{(0)}\}$ are calculated using (13) with $p(d_i = k) = 1/M$ for all i and k . Next, $\hat{B}^{(1)}$ is found by recursively solving (18). Once the recursion is complete, $\hat{A}^{(1)}$ can be directly found from (17), which finalizes the first EM iteration. The second EM iteration then starts by calculating $p_{k,i}^{(1)}$ using (13) with $p(d_i = k) = 1/M$ and the newly acquired $\hat{A}^{(1)}$ and $\hat{B}^{(1)}$, and the remaining steps are identical to the first EM iteration. The EM estimator will continue to iterate until some stopping criterion is reached. In our simulations, we halted the EM algorithm when the value of the estimate of B changed less than 10%, when the estimate of B became very close to zero, or when a maximum number of 20 iterations was reached. After the first BICM-ID iteration, the final value of $\hat{B}^{(q)}$ from the previous BICM-ID iteration can be used as the initial estimate of B , and the value of $p(d_i = k)$ in (13) is found from the decoder output using (15).

V. REDUCED COMPLEXITY ESTIMATION

A major drawback of the proposed EM-based estimator is its complexity. In this section, two techniques are proposed for reducing the complexity of the algorithm. One involves a linear approximation to the $F(\cdot)$ function, while the other involves the hard limiting of $p_{k,i}$.

A. Linear Approximation of $F(\cdot)$

During each iteration of the full-complexity EM algorithm, $\hat{B}^{(q)}$ is found by recursively solving (18). For each step in the recursion, the nonlinear function $F(x) = I_1(x)/I_0(x)$ must be evaluated for each of the MN entries in the \mathbf{Y} matrix, presumably by a table look-up. The number of required table look-ups can be drastically reduced by performing a first-order Taylor series expansion of $F(x)$ about the point $x = t$,

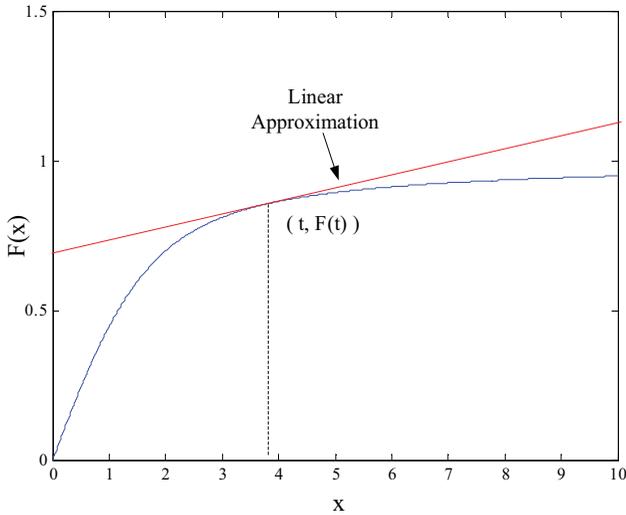


Fig. 2. $F(x) = I_1(x)/I_0(x)$ and its linear approximation.

resulting in $F(x) = F(t) + F'(t)(x - t)$. The expansion point t is the approximate maximum value of the argument of $F(\cdot)$ in (18). Setting $|y_{k,i}| \approx a\sqrt{\mathcal{E}_s}$ and $C \approx N(a^2\mathcal{E}_s + MN_0)$, we obtain $t \approx 2a^2\mathcal{E}_s/N_0 = B^2/(2A)$.

The linear approximation of the $F(\cdot)$ function is illustrated in Fig. 2. As shown, $F(x)$ is a monotonically increasing function with respect to x and is concave, approaching 1 when $x \rightarrow \infty$. Because the curve becomes flat when x is reasonably large, such a linear approximation is reasonable. Assuming $4C \gg N(\hat{B}^{(q)})^2$ and substituting the linear expansion of $F(\cdot)$ about the point $t = (\hat{B}^{(q-1)})^2/(2\hat{A}^{(q-1)})$ into (18) yields

$$\hat{B}^{(q)} \approx \frac{[F(t) - tF'(t)] \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} p_{k,i}^{(q-1)} |y_{k,i}|}{N \left(\frac{1}{2} - \frac{M}{C} F'(t) \sum_{i=0}^{N-1} \sum_{k=0}^{M-1} p_{k,i}^{(q-1)} |y_{k,i}|^2 \right)} \quad (20)$$

where $F'(t) = 1 - \frac{F(t)}{t} - F^2(t)$, as implied by equation (8.486) of [12].

With this approximation, (18) is replaced with (20), and now only a single table look-up is required per EM iteration, instead of the NM look-ups in (18). Due to the linearization, $\hat{B}^{(q)}$ can be found directly from (20) without requiring a recursion, which greatly simplifies the algorithm. Notice, however, that the expansion point $t^{(q-1)}$ must be changed after each EM iteration.

The linear approximation of $F(x)$ is tight when the expansion point is sufficiently large and the argument of $F(\cdot)$ in the original EM equation (18) is close to the expansion point. Since the expansion point is proportional to the estimated SNR, the approximation gets worse with decreasing SNR. Because of the concavity of the $F(\cdot)$ function, the approximation will overestimate its value, leading to an overestimation of B . However, overestimating B is better than underestimating it, which agrees with observations made in [13] that the SNR can be overestimated in an AWGN channel by as much as 3 dB without significantly harming the performance of a turbo code. Even when the expansion point is sufficiently high, the approximation will be loose when the arguments in the linearized $F(\cdot)$ function are small, which occurs for those

values of $|y_{k,i}|$ that are small. Small values of $|y_{k,i}|$ occur more frequently at high SNR, since the $M-1$ entries of each vector y_i that do not pertain to the transmitted symbol would all be small. While the linear approximation is indeed poor for these small values of $|y_{k,i}|$, this problem is mitigated by the fact that every $|y_{k,i}|$ is weighted by its corresponding probability $p_{k,i}$, which will also be small. Thus, the contribution of the small values of $|y_{k,i}|$ to the overall estimate is negligible, and the poor approximation at these values does not seriously harm overall performance.

B. Hard Limiting of $p_{k,i}$

During the q^{th} iteration of the full-complexity EM algorithm, each $p_{k,i}$ must be evaluated using (13). For each symbol, the normalization factor α_i must also be calculated to assure that $\sum_{k=0}^{M-1} p_{k,i} = 1$. The normalization factor can be avoided by setting $p_{k,i} = 1$ for one particular value of k , denoted k_0 , and setting $p_{k,i} = 0$ for all $k \neq k_0$. The index k_0 should be the value of k that maximizes (13). Taking the logarithm of (13), which does not change the maximization, and using (15) for $p(d_i = k)$ results in

$$k_0 = \arg \max_k \log \left[I_0 \left(\frac{\hat{B}^{(q-1)} |y_{k,i}|}{\hat{A}^{(q-1)}} \right) \right] + \sum_{j=0}^{\mu-1} v_{j,i} b_j(k). \quad (21)$$

In addition to eliminating the need for computing the normalization factor α_i , this approximation has the additional benefit of eliminating the exponential functions in (15). Complexity is further reduced when \hat{B} is calculated with either (18) or (20) because those terms for which $p_{k,i} = 0$ do not need to be considered, and therefore the summations over k are eliminated. Another benefit of this method is that it provides a natural stopping criterion for the EM algorithm, which should halt once the $p_{k,i}$'s no longer change from one iteration to the next.

While (21) is a very coarse approximation to (13) in the normal EM algorithm, it still uses both the decoder's *a priori* information as well as the channel likelihood based on the current estimates. This approximation tends to make (18) overestimate the value of B , but the performance loss due to this approximation is small, as will be demonstrated in the next section.

VI. SIMULATION RESULTS

To illustrate the performance of the proposed estimators, a set of simulations were run. The simulated system uses the turbo code from the cdma2000 specification [14] and 16-FSK modulation. The specific turbo code that was selected is a rate-1/2 code with $N_u = 1530$ input bits. As the cdma2000 standard requires 12 coded tail bits, the length of each code word is actually $N_b = 2(1530) + 12 = 3072$ bits or $N_d = 768$ FSK symbols. The receiver executed up to 20 BICM-ID iterations. A perfect CRC check was assumed in the simulations, so that the iterations would stop once the data is correctly decoded.

Fig. 3 shows the bit error rate (BER) performance of five systems over a Rayleigh block fading channel with $N = 4$

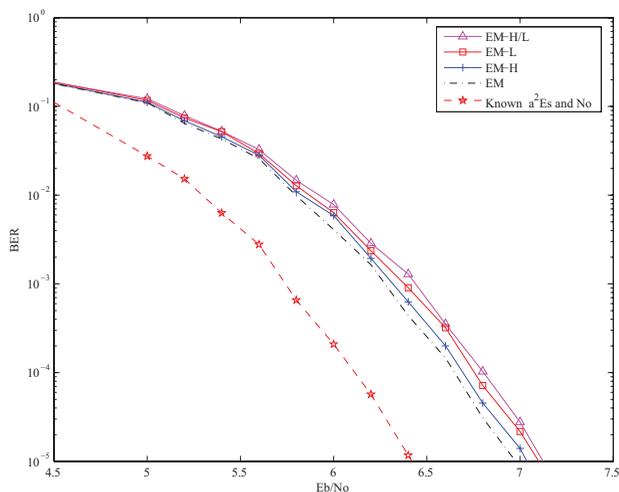


Fig. 3. BER comparison of the different estimators in block Rayleigh fading with $N = 4$ symbols per block. The system uses 16-FSK modulation and the rate 1/2 cdma2000 turbo code ($N_u = 1530$ input bits). Shown from left to right is performance with: (1) $a\sqrt{\mathcal{E}_s}$ and N_0 known for each block; (2) The full-complexity EM estimator; (3) Estimator *EM-H*, which makes hard decisions on $p_{k,i}$; (4) Estimator *EM-L*, which uses a linear approximation to the $F(\cdot)$ function; and (5) Estimator *EM-H/L*, which makes hard decisions on $p_{k,i}$ and uses a linear approximation to $F(\cdot)$.

symbols per block. The curve with the best performance corresponds to the case that $a\sqrt{\mathcal{E}_s}$ and N_0 are known by the receiver. While not possible in practice, this curve serves as a benchmark. The other curves correspond to four implementations of the proposed estimator. The best performing estimator is the full-complexity EM-based estimator described Section IV (EM estimator). The other curves correspond to the reduced complexity techniques described in Section V. In order from best-to-worst performing, the curves use the following complexity reduction techniques: (1) Hard limiting of $p_{k,i}$ (EM-H); (2) Linear approximation of the $F(\cdot)$ function (EM-L); and (3) *Both* Hard limiting of $p_{k,i}$ and a linear approximation of $F(\cdot)$ (EM-H/L). For this example, the full-complexity EM estimator has a 0.55 dB loss relative to the system with known $a\sqrt{\mathcal{E}_s}$ and N_0 . The additional loss due to the complexity reduction techniques is about 0.05 dB for EM-H, 0.1 dB for EM-L, and 0.15 dB for EM-H/L.

Fig. 4 shows BER results in Rayleigh block fading for several values of block length N . For each value of N , two curves are shown. The curve on the left (dashed line) is for the case that $a\sqrt{\mathcal{E}_s}$ and N_0 are known by the receiver, while the curve on the right shows performance of Estimator EM-H/L. As the value of N decreases, performance of both systems improves due to increasing diversity. However, the gap between the two curves widens with decreasing N due to increasing estimation error. Results were also produced for $N = 1$ (not shown to keep the plot uncluttered), but the performance of the EM-H/L estimator with $N = 1$ is about 0.5 dB worse than when $N = 4$ and nearly 2 dB worse than when $a\sqrt{\mathcal{E}_s}$ and N_0 are known.

To better illuminate the effect of block length on estimator performance, Fig. 5 shows simulation results for the same cdma2000 turbo code and 16-FSK in an unfaded, AWGN channel. While the fading is a constant $a = 1$, the estimator

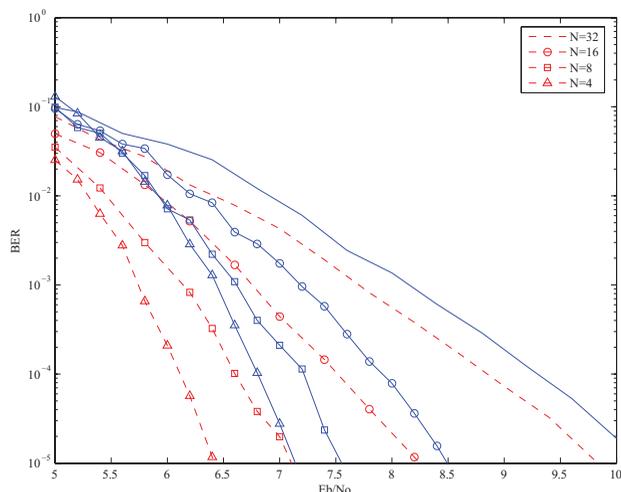


Fig. 4. Influence of the block length N on the BER performance in block Rayleigh fading. For each value of $N = \{4, 8, 16, 32\}$, two curves are shown. The left curve (dashed line) shows performance when $a\sqrt{\mathcal{E}_s}$ and N_0 are known for each block; the right curve (solid line) shows performance with Estimator EM-H/L. The system uses 16-FSK modulation and the rate 1/2 cdma2000 turbo code ($N_u = 1530$ input bits).

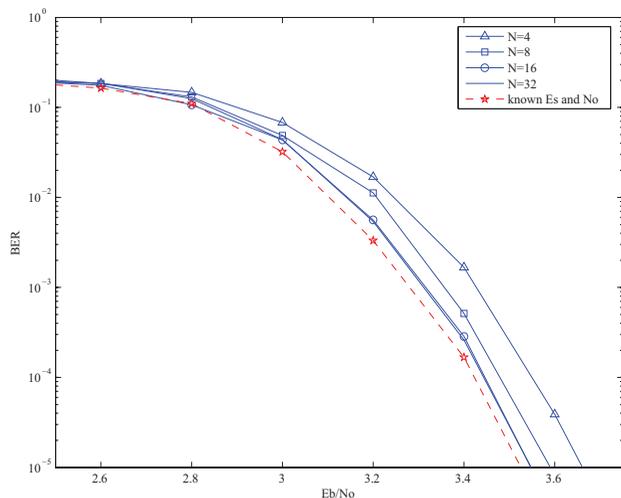


Fig. 5. Performance in AWGN as a function of block length N . The performance with known \mathcal{E}_s and N_0 (dashed lines) is compared against the performance with Estimator EM-H/L. Modulation is 16-FSK. The code is the rate 1/2 cdma2000 turbo code with $N_u = 1530$.

runs assuming a block length of N symbols. When $N = 4$, the performance of the estimator is about 0.3 dB away from when \mathcal{E}_s and N_0 are known. The performance improves with increasing N , and when $N = 32$ it is only 0.03 dB away from the performance with known \mathcal{E}_s and N_0 .

VII. COMPLEXITY COMPARISON

Table I shows the number of operations required for the four versions of the proposed estimator that were used to generate the results shown in Fig. 3. As all four estimators use (17) to compute A , they differ only in how $p_{k,i}$ and B are computed. Estimator EM-L benefits from not having to perform a table look-up for each received symbol and by not requiring a recursion on (18). Estimator EM-H benefits from not needing to compute the normalization factor (14),

TABLE I

NUMBER OF OPERATIONS REQUIRED FOR EACH TYPE OF ESTIMATOR TO EXECUTE ONE EM ITERATION PER BLOCK OF N SYMBOLS. M IS THE MODULATION ORDER AND R IS THE NUMBER OF RECURSIONS USED TO SOLVE (18).

(a) Operations required to compute $p_{k,i}$

Algorithm	Additions	Multiplications	Look-Ups
EM	$N(M-1)$	$3NM$	NM
EM-L	$N(M-1)$	$3NM$	NM
EM-H	NM	NM	NM
EM-L/H	NM	NM	NM

(b) Operations required to compute B

Algorithm	Additions	Multiplications	Look-Ups
EM	RNM	$NM + R(2NM + 5)$	RNM
EM-L	$2(NM-1) + 4$	$2NM + 7$	1
EM-H	RN	$R(2N + 5)$	RM
EM-L/H	$2(N-1) + 4$	$3N + 7$	1

by computing (15) in the log-domain, and not needing to sum over k in (18). EM-L/H combines the benefits of EM-L and EM-H.

The overall complexity also depends on the average number of EM iterations per BICM-ID iteration. For the simulation that produced the BER results shown in Fig. 3, the average number of full EM iterations (per BICM-ID iteration) was approximately 1.1 for EM-H, 1.4 for both EM and EM-L/H, and 1.5 for EM-L. These values are small primarily as a consequence of the loose stopping criterion for the EM algorithm (if B changes less than 10%, it will halt). A tighter stopping criterion (e.g. halting when B changes less than 1%) will induce more EM iterations (about 3 for the EM estimator), but will not significantly improve the BER performance. Longer blocks generally required fewer iterations, on average. The higher value for EM-L suggests that the approximation for $F(\cdot)$ caused it to converge more slowly.

In addition to counting operations, another way to assess complexity is to count CPU cycles in an actual implementation. We did this for the four estimators (implemented in the C language) during the simulation that produced Fig. 3. As expected, the original EM estimator required the most CPU cycles, and in fact required more than that used for the turbo decoder. Estimator EM-L had a complexity of about 1/4 that of Estimator EM, making it only a little more complex than the demapper. Estimator EM-L/H is 1/3 the complexity of Estimator EM-L, making its complexity negligible compared to the decoder and demapper. Given the slight loss in performance, Estimator EM-L/H is an attractive solution.

VIII. CONCLUSION

The proposed robust noncoherent system has been shown to withstand the severe channel conditions of fast fading, unknown fading attenuation, unknown fading statistics, and unknown noise-power spectral density. The channel-state estimator is based on the Expectation Maximization algorithm and exploits extrinsic information produced after each decoding iteration of the turbo code. Each updated channel-state estimate is applied to the next decoder iteration. Simulation results indicate that if the fading coherence time exceeds four channel symbols, then the performance is close to what could be obtained with perfect channel-state information. Although the estimator using the exact EM algorithm has a high complexity, the linear approximation of $F(\cdot)$ and the hard limiting of $p_{k,i}$ can be applied to reduce the complexity with minor loss in BER performance.

REFERENCES

- [1] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.
- [2] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, pp. 169–171, Nov. 1997.
- [3] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded M -ary noncoherent orthogonal modulation," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 1738–1747, Sep. 2005.
- [4] M. Sandell, C. Lusch, P. Strauch, and R. Yan, "Iterative channel estimation using soft decision feedback," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 1998, pp. 3728–3733.
- [5] M. Nissilä and S. Pasupathy, "Adaptive Bayesian and EM-based detectors for frequency-selective fading channels," *IEEE Trans. Commun.*, vol. 51, pp. 1325–1336, Aug. 2003.
- [6] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
- [7] D. Torrieri, *Principles of Spread-Spectrum Communication Systems*. New York: Springer, 2005.
- [8] R. Knopp and P. A. Humblet, "On coding for block fading channels," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 189–205, Jan. 2000.
- [9] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Commun. Lett.*, vol. 1, pp. 22–24, Jan. 1997.
- [10] M. C. Valenti, E. Hueffmeier, B. Bogusch, and J. Fryer, "Towards the capacity of noncoherent orthogonal modulation: BICM-ID for turbo coded NFSK," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Nov. 2004, pp. 1549–1555.
- [11] S. C. Chapra and R. Canale, *Numerical Methods for Engineers*, 4th ed. New York: McGraw-Hill, 2002.
- [12] I. Gradshteyn, I. Ryzhik, A. Jeffrey, and D. Zwillinger, *Tables of Integrals, Series and Products*, 6th ed. Burlington, MA: Academic Press.
- [13] M. Jordan and R. Nichols, "The effects of channel characteristics on turbo code performance," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Oct. 1996, pp. 17–21.
- [14] Third Generation Partnership Project 2 (3GPP2), "Physical layer standard for cdma2000 spread spectrum systems, revision D," 3GPP2 C.S0002-D Version 2.0, pp. 115–122, Sep. 2005.