Iterative Decoding Based on Error Pattern Correction

Hakim Alhussien, Jihoon Park, and Jaekyun Moon

Communications and Data Storage Laboratory, Department of Electrical and Computer Engineering,
University of Minnesota, Minneapolis, MN 55455 USA

The error-pattern correction code is a code specialized to correct dominant error patterns observed at the channel detector output in heavy intersymbol interference channels. In this paper, we consider a turbo-equalizer system that utilizes the error-pattern correcting code (EPCC) as a building component. A soft-input soft-output (SISO) decoder for the EPCC is described, and the performance of the proposed turbo equalizer is compared with those of the conventional turbo equalizer and a low density parity check (LDPC) code system.

***Index Terms*—Error pattern correction code, list decoding, turbo equalization.**

I. Introduction

THIS work builds upon the previous work on error-pattern correction coding as described in [1] and [2]. In particular, we describe a soft-input soft-output (SISO) decoder for the error-pattern correction code, and consider the application of the code as a building block of a turbo-equalization system. Turbo equalization has been considered previously for magnetic recording applications. See, for example, [3]–[5]. The error-pattern correcting code (EPCC) is motivated by the well-known observation that the error rate at the channel detector output of an intersymbol-interference (ISI) channel is dominated by a few specific known error cluster patterns. This is due to the fact that the channel output energies associated with these error patterns are smaller than those of other patterns.

The cyclic codes described in [2] are based on construction of a generator polynomial $g(x)$ that gives rise to distinct syndrome sets for all targeted dominant error patterns. It has been shown that such a $g(x)$ can be obtained from the irreducible factors making up the polynomial representations of the dominant error patterns. The code can be further improved by introducing another factor in $g(x)$, namely, a primitive polynomial that is not already a factor of $g(x)$ [1]. The results are an increased code rate, improved single-error-pattern correction accuracy (via reduced miss-correction probability), and capability to correct some important multiple-pattern events based on an increased number of distinct syndrome patterns. The work of [1] describes a combined syndrome-mapping and channel-reliability-aided decoding that is capable of correcting double error patterns. While the performance gain of this decoding scheme is significant over the case of single-pattern correction only, decoder complexity also becomes large as the size of the syndrome table increases to account for the large number of double error patterns that are possible within the fairly large codeword length.

The recent work described in [6] discusses a more efficient decoding method based on a list decoding strategy, wherein a list of tentative decoder input words are constructed by perturbing the original channel detector output. While the concept of running parallel decoders on such test vectors to find a number of possible codewords is well-known [7], [8], the novel feature of the decoding method presented in [6] lies in the way the test decoder input vectors are constructed. The method of [6] uses reliability measures of local bit patterns in the original channel detector output, rather than the usual bit level reliability measure, to flip bit patterns at different positions to generate likely word errors, with respect to the original channel detector output. The highly accurate and relatively simple single-pattern correcting decoder then acts on each test word. A number of valid codewords are typically produced at the outputs of the parallel decoders, and the most probable codeword is chosen and released as the final decision. This method provides a considerably improved ability to correct multiple-pattern error events. The decoder is also a soft-input decoder in the sense that the soft information out of the channel detector is also utilized.

In this paper, we present a modification of this list decoder that can work in the medium noise environment. We also introduce a soft-decision generator stage that produces bit-level soft decisions based on the list of candidate codewords made available at the output of the single-pattern correcting parallel decoders. Deriving the bit-level soft decisions is based on the existing approach of finding probability measures of each candidate codeword, converting them to bit-level reliability measures by grouping the candidate codewords according to the binary value of the given bit position, and then performing group-wise averaging of the codeword probabilities. What is unique in our approach, however, is that we estimate the candidate codeword probability based on the reliability measures of the local error patterns in the candidate codeword, rather than on the usual Euclidean distance of the codeword from the detector output. In estimating the reliability of a local error pattern, we make use of maximum *a posteriori* probability (MAP)-based, finite-window correlators matched to the dominant error patterns.

We finally consider the application of the EPCC in a turbo-equalizer setup, where the channel-matched EPCC decoder exchanges soft information iteratively with the convolutional code decoder. We evaluate the performance of this "EPCC-enhanced" turbo equalizer (TE), and comment on the potential performance gain after concatenation with an outermost Reed–Solomon (RS) code in hopes of removing the TE error floor. A performance comparison with conventional TE as well as an LDPC-coded system is also presented.

Digital Object Identifier 10.1109/TMAG.2007.912943

II. A SISO Decoder for EPCC

In our TE setup, the EPCC is matched to the ISI channel and serves as an inner code for an outer interleaved convolutional code. Since the EPCC maintains substantial error correction power while having a high code rate that is close to 1, the hope is that the redistribution of redundancy between the EPCC and the outer code in a TE setup would improve overall system performance. In SISO decoding of EPCC, EPCC utilizes the output extrinsic information coming from the channel detector as *a priori* information in calculating error pattern *a posteriori* probabilities. Then, after generating a list of the most probable candidate codewords, the decoder uses the list to calculate the output bit-level decision reliabilities that serve as the *a priori* information for other SISO elements in the soft iterative system.

The list decoding scheme employed here has been discussed in [6]; in this paper, we formulate the channel-matched decoder as a SISO decoder and modify it to handle medium-noise environments. The list decoder of [6] is philosophically similar to Chase decoding [7] in the sense of generating test vectors at the parallel decoder input. The difference is that in the setup of [6], during the test word construction stage the EPCC decoder flips multibit error patterns, not the individual bits, to form a set of test error words that are decoded by an array of single-pattern correcting decoders. The parallel decoder outputs form the final list of valid candidate codewords.

In summary, the list-decoding/soft-output generation process we consider has three phases.

- The test error word list is generated by inserting the most probable combination of local error patterns to the channel detector output.
- An array of parallel single-pattern correcting decoders decode the test words to produce a list of valid codewords.
- The list of candidate codewords is used to generate bit-level decisions along with their reliabilities.

We now discuss each phase in detail.

A. Generation of the Test Error Word List

First, consider an event that the channel detector output word, which we will call the maximum likelihood (ML) word (assuming the channel detector outputs the ML word or its close approximation), is corrupted by a dominant pattern $e_i(x)$ starting at bit location k_i. We can think of a local error pattern (with respect to the ML word) denoted by $x^{k_i}e_i(x)$ of type i and starting location k_i. A test word may contain one or more such local error patterns. We wish to produce a number of highly probable test words. The probability measure of a given test word with a particular combination of local patterns can be determined by estimating the probability of dominant patterns at a given starting position. This is done in Section IV. Specifically, those type/location pairs that are most probable in the sense of maximizing the correlator function of Section IV are used to construct test words. The requirement to have M-error-pattern-correction capability using the single-pattern correcting decoders, dictates that test words must include up to $M-1$ local error patterns. Starting from the m most probable such local error patterns $x^{k_i}e_i(x)$'s (i.e., corresponding to the m most probable pairings of (i,k_i)), one can think of $\binom{m}{1}+\binom{m}{2}+\ldots+\binom{m}{M-1}$ ways of corrupting the ML word with up to $M-1$ local error patterns. From this large set of potential combinations, a relatively small subset of most probable combinations needs to be chosen to maintain reasonable complexity. One can think of many different ways of effectively constructing such a list [6], based on the probable local error patterns that have been identified. In this work, we limit the size of the test word list to $L=25$ with each word having up to two local error patterns.

B. Parallel Algebraic Decoding

The list of test error words generated above is delivered to an array of single-error-pattern correcting decoders that work in parallel to generate the candidate codeword list. The number of parallel decoders L is identical to the size of the test word list, and is a crucial parameter that controls the EPCC decoder's complexity/performance tradeoff. In the decoding process, each decoder searches the space in the proximity of its input test word for valid codewords with zero syndrome. Since it is a single-pattern correcting code, the space of interest around the test word is the union of the neighborhoods centered around the test word each having radius chosen solely from the list of most probable local error patterns. In the event of finding a valid codeword in the search space, the decoder releases it as its decoded output. Otherwise, with no codeword found, the decoder does not contribute to the final list. As such, the size of the generated final list is variable.

C. Generation of Soft Output

The candidate codeword list constructed by our "pattern-level" list decoder is used to evaluate the more familiar bit-level reliabilities that constitute the output soft information supplied by the EPCC SISO decoder. Typically, the candidate codeword list forms a reduced space for the maximum likelihood decoder (MLD) to search for the maximum-likelihood codeword given the observed word. When the observed word is the output of an additive white Gaussian noise (AWGN) channel, the metric that specifies the likelihood of the codeword is the Euclidean distance to the observed word. In our list soft-decoder formulation, the resolution of the space of possible codewords is measured in units of "error-patterns" rather than error bits. As a result, we measure the probability of a candidate codeword given the observed word by the product of the probabilities of each "local" error pattern forming the candidate word. Specifically, let $\mathbf{c}$ represent a candidate codeword with, say, K error-pattern corruption with respect to the ML word $\hat{\mathbf{c}}$. Then, the *a posteriori* probability of this particular test word, $Pr(\mathbf{c}/\hat{\mathbf{c}},\mathbf{r})$, is estimated by multiplying the probability estimates of the K local patterns, given the channel observation $\mathbf{r}$ at the detector input. The probability of $e_i(x)$ starting at bit location k_i, i.e., the local error pattern $x^{k_i}e_i(x)$, can be computed via (4) to be discussed later in Section IV. Also, the actual probability computation is done in the log-domain using the metrics (9) for each constituent error-pattern. The probability measures of candidate codewords are further differentiated apart by the utility of the "pattern-level" *a priori* probability term in (9), which is available by grouping the constituent "bit-level" *a priori* probabilities of the bits forming the error patterns, as reflected by (7) of Section IV, for each constituent error pattern of the candidate word.

Given the list of codewords and their accompanying *a posteriori* probabilities, the reliability λ_k of the coded bit c_k is evaluated as

$$\lambda_k = \frac{\sum_{\mathbf{c}\in\mathbf{S}_k^+} Pr(\mathbf{c}/\hat{\mathbf{c}},\mathbf{r})}{\sum_{\mathbf{c}\in\mathbf{S}_k^-} Pr(\mathbf{c}/\hat{\mathbf{c}},\mathbf{r})} \tag{1}$$

where $\mathbf{S}_k^+$ is the set of candidate codewords where $c_k = +1$, and $\mathbf{S}_k^-$ is the set of candidate codewords where $c_k = -1$. The quantity in (1) is utilized when the candidate codewords do not all agree on the bit decision for location k. In the event that all codewords do agree on the decision for c_k, a method used by [9] is adopted for generating soft information as follows:

$$\lambda_k = \beta^{\text{iter}} \times \lambda_{\max} \times \hat{d}_k \tag{2}$$

where $\hat{d}_k$ is the bipolar representation of the agreed-upon decision, $\lambda_{\max}$ is a preset value for the maximum reliability at convergence of turbo performance, and the multiplier $\beta^{\text{iter}} < 1$ will prove useful when incorporating the EPCC SISO decoder in an iterative system. Note that in an iterative system the level of confidence in bit decisions is lower at the initial iterations, and thus multiplying the generated log likelihood ratios by the back-off factor β^{iter} reduces the risk of error propagation. It will also be necessary to use (2) to generate soft information if the reliability threshold check activates only one algebraic single error pattern correction decoder rather than the list decoder, as discussed in the next section.

III. A TE Incorporating EPCC SISO Decoder

The proposed EPCC SISO decoder is now ready to be used as a building block in turbo systems. Since the EPCC is matched to the ISI channel, no interleaving should be present between EPCC and the channel. On the other hand, an interleaver is essential between the EPCC and the outer recursive systematic convolutional code (RSC). A legitimate question to be posed is whether the EPCC and TE would benefit from working together. A major step in answering this question is the following observation. We see that the set of most dominant error patterns produced by an ISI channel is not a sensitive function of its operating signal-to-noise ratio (SNR). This means that iteratively-improved *a priori* information fed to the channel detector, which can be viewed as providing a boost in SNR, would still give the same set of dominant error patterns that can be further corrected by the EPCC, resulting eventually in an improved *a priori* input to the channel detector and the same dominant error set, provided no serious miss-correction is present, and so on. This is a strong motivation for the integration of the two systems, since then, gradual iterative improvement would be achieved. This behavior is in contrast to the performance saturation phenomena in a conventional turbo system within a few channel iterations. The proposed setup is shown in Fig. 1. The Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [10] incorporating pattern-dependent noise-prediction (PDNP) [11], [12] generates extrinsic log-likelihood ratios (LLR) λ_k^e that can be used by the EPCC along with channel observation r_k to estimate the probability of single error patterns. Using the list of candidate codewords, the EPCC SISO decoder calculates extrinsic bit-level reliabilities $\tilde{\lambda}_k^e$ that are fed as channel observations to the RSC decoder after deinterleaving. The RSC BCJR-based decoder in turn generates its own extrinsic information that is fed back to the BCJR/PDNP detector after interleaving. In the EPCC decoder, a decision is first made on whether or not the decoder input contains a single error pattern via the syndrome check. If the initial syndrome check indicates either an error-free input, or else a single error pattern with the reliability threshold qualification also satisfied [6], then the corrected output is released accordingly, and (2) is used to generate the decision reliabilities for each of the hard bits making up the corrected codeword. If not, the list decoder is activated that involves computing correlator-based reliability estimates for local patterns in the ML word, and generation of decision reliabilities using (1). Simulations show that the aforementioned strategy of moving between list-decoding and algebraic single pattern decoding results in improved performance compared to running list-decoding all the time, since at later turbo iterations single error-pattern occurrences are more likely, and syndrome-decoding is more robust in such scenarios. One drawback of the EPCC-TE setup involving the serial concatenation of an interleaved RSC and EPCC is that the parity bits introduced by the addition of the innermost EPCC are not protected by the outer RSC. Thus, the *a priori* information of these bits are not updated by turbo channel iterations. Although error events in those bits do not count towards the final bit error rate (BER), since EPCC parity bits get discarded prior to deinterleaving, they can still pair up with error events elsewhere in the codeword to form multiple-error-pattern occurrences that are not always resolvable by the EPCC-turbo action. One solution around this problem would be to design a turbo product code (TPC) in which the EPCC is the row encoder, in which case its parity bits are protected by the column encoder. An interesting design problem for TPC there would be to ensure that the column encoder does not break up dominant error-patterns into unrecognized error events that are not targeted by EPCC of the next inner TPC iteration.

IV. Computation of Correlator-Based Reliability Measures

Error pattern reliability measures are computed by the maximum *a posteriori* (MAP)-based error-pattern correlator shown in Fig. 2. The correlator design discussed here is an extension of the AWGN-environment correlator presented in [6] to the media noise dominated environment, in which pattern-dependent noise prediction is incorporated in the sliding-window correlator metric. The modified pattern-dependent correlator scans the ML word, estimating the data-dependent noise in the process without requiring any trellis expansion.

Let r_k be the channel detector input sequence $r_k = c_k * h_k + n_k^e + n_k^j(c_k)$, where c_k is the bipolar representation of the transmitted codeword sequence, h_k is the partial response maximum likelihood (PRML) target of length l_h, n_k^e is the zero-mean AWGN electronic noise with variance σ_e^2, and $n_k^j(c_k)$ is the zero-mean correlated data-dependent media noise with variance σ_j^2. Also, let $q_k = r_k - (\hat{c}_k * h_k) = (c_k - \hat{c}_k) * h_k + n_k$ be the channel detector's output error sequence (with the channel detector utilizing PDNP via an expanded trellis [13]), and $n_k =$

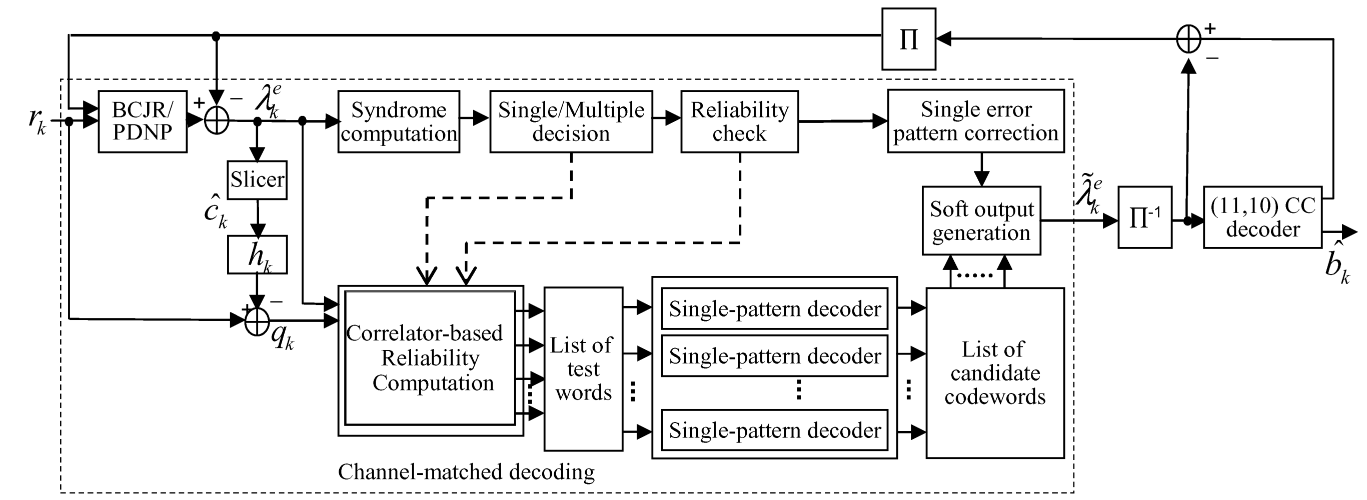
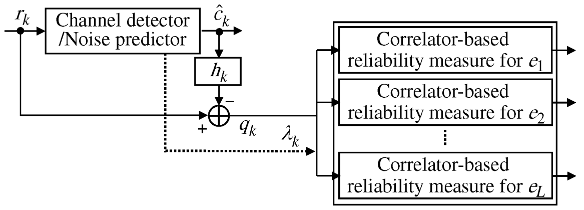


Fig. 1. EPCC-TE block diagram.



r_k
Channel detector /Noise predictor
$\hat{c}_k$
h_k
q_k
λ_k
Correlator-based reliability measure for e_1
Correlator-based reliability measure for e_2
Correlator-based reliability measure for e_L

Fig. 2. Local error-pattern correlators.

$n_k^e + n_k^j(c_k)$. If a target error pattern sequence $e_k^{(i)}$ occurs at positions from $k = j$ to $k = j + l_i - 1$, then this q_k can be written as

$$\begin{aligned} q_k &= (c_k - \hat{c}_k) * h_k + n_k \\ &= \left[\mathbf{e}^{(i)}\right]_j^{j+l_i-1} * h_k + n_k \\ &= \left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h} + n_k \end{aligned} \tag{3}$$

where the notation $[\mathbf{x}]_i^j$ denotes a local segment $[x_i, x_{i+1}, \ldots, x_j]$ of the sequence x_k, $s_k^{(i)}$ is the noise-free error signal given by $s_k^{(i)} = e_k^{(i)} * h_k$, and $l_i^h = l_i + l_h - 2$.

The reliability measure for each possible error starting position $\rho_m^{(i)}$ can be computed by the local *a posteriori* probabilities

$$P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h} |[\mathbf{r}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) = P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h} |[\mathbf{q}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right)$$

where $l^{p,h} = l_p + l_h - 1, j \in \{\rho_1^{(i)}, \rho_2^{(i)}, \cdots, \rho_M^{(i)}\}$, and l_p PDNP taps are employed. Using Bayes' theorem, the *a posteriori* probability can be rewritten as

$$P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h} |[\mathbf{q}]_{j-l_p}^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) = \frac{P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} | \left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h}, [\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h}\right)}{P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} |[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right)}. \tag{4}$$

Here, $P([\mathbf{q}]_{j-l_p}^{j+l_i^h} |[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h})$ can be approximated by

$$\begin{aligned} &P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} |[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}\right) \\ &\approx P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} |[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}, \left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h}\right) \cdot P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h}\right) \\ &+ P\left([\mathbf{q}]_{j-l_p}^{j+l_i^h} |[\hat{\mathbf{c}}]_{j-l^{p,h}}^{j+l_i^h}, \left[\tilde{\mathbf{s}}^{(i)}\right]_j^{j+l_i^h}\right) \cdot P\left(\left[\tilde{\mathbf{s}}^{(i)}\right]_j^{j+l_i^h}\right) \end{aligned} \tag{5}$$

where $[\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}$ corresponds to the most probable competing error pattern, and by definition, this is the all-zero error pattern with respect to the ML detector's decision sequence (i.e., the most probable competing local pattern is that associated with the ML word itself). Examining (3), and modeling the media noise using an l_p-tap PDNP, q_k can be represented as a sequence of statistically independent Gaussian random variables with mean $\hat{n}_k^p([\hat{\mathbf{c}}]_{k-l^{p,h}}^k)$ and variance $\sigma_k^2([\hat{\mathbf{c}}]_{k-l^{p,h}}^k)$, which reflect the predictor noise estimate and noise prediction error variance, respectively. Thus, the likelihood probabilities in the RHS of (5) are given by (with sequence upper and lower indices dropped for decluttering the notations)

$$\begin{aligned} P\left(\mathbf{q}|\mathbf{s}^{(i)}, \hat{\mathbf{c}}\right) &= \frac{(2\pi)^{-l_i^h/2} e^{-\sum_{k=j}^{j+l_i^h} \frac{\left(q_k - s_k^{(i)} - \hat{n}_k^p(-\hat{\mathbf{c}}_k)\right)^2}{2\sigma_k^2(-\hat{\mathbf{c}}_k)}}}{\prod_{k=j}^{j+l_i^h} \sqrt{\sigma_k^2(-\hat{\mathbf{c}}_k)}} \\ P\left(\mathbf{q}|\tilde{\mathbf{s}}^{(i)}, \hat{\mathbf{c}}\right) &= \frac{(2\pi)^{-l_i^h/2} e^{-\sum_{k=j}^{j+l_i^h} \frac{\left(q_k - \hat{n}_k^p(\hat{\mathbf{c}}_k)\right)^2}{2\sigma_k^2(\hat{\mathbf{c}}_k)}}}{\prod_{k=j}^{j+l_i^h} \sqrt{\sigma_k^2(\hat{\mathbf{c}}_k)}}. \end{aligned} \tag{6}$$

The two *a priori* probability terms in the RHS of (5) are given by

$$\begin{aligned} P\left(\left[\mathbf{s}^{(i)}\right]_j^{j+l_i^h}\right) &= \prod_{k=0}^{l_i-1} P(c_{j+k} = (-1)^{\left|e_k^{(i)}/2\right|} \hat{c}_{j+k}) \\ P\left(\left[\tilde{\mathbf{s}}^{(i)}\right]_j^{j+l_i^h}\right) &= \prod_{k=0}^{l_i-1} P(c_{j+k} = \hat{c}_{j+k}). \end{aligned} \tag{7}$$

For an equally-probable c_k, i.e., $P(c_k = \pm 1) = 1/2$, we have $P([\mathbf{s}^{(i)}]_j^{j+l_i^h}) = P([\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h})$. However, if *a priori* information λ_k is available through another detector/decoder stage so that

$$P(c_k = +1) = \frac{e^{\lambda_k}}{1 + e^{\lambda_k}}, \text{ and } P(c_k = -1) = \frac{1}{1 + e^{\lambda_k}} \quad (8)$$

where $\lambda_k = \log(P(c_k = +1)/P(c_k = -1))$, then $P([\mathbf{s}^{(i)}]_j^{j+l_i^h}) \neq P([\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h})$.

Utilizing (6) and (5), taking the log-ratio between the *a posteriori* probability (4) and its counterpart *a posteriori* probability for $[\tilde{\mathbf{s}}^{(i)}]_j^{j+l_i^h}$ gives

$$C\left(e_j^{(i)}\right) = \sum_{k=j}^{j+l_i^h} \frac{1}{2} \left[\frac{(q_k - \hat{n}_k^p(\hat{\mathbf{c}}_k))^2}{\sigma_k^2(\hat{\mathbf{c}}_k)} - \frac{\left(q_k - s_k^{(i)} - \hat{n}_k^p(-\hat{\mathbf{c}}_k)\right)^2}{\sigma_k^2(-\hat{\mathbf{c}}_k)} + \log \frac{\sigma_k^2(\hat{\mathbf{c}}_k)}{\sigma_k^2(-\hat{\mathbf{c}}_k)} \right] - \log \frac{P\left(\left[\tilde{s}^{(i)}\right]_j^{j+l_i^h}\right)}{P\left(\left[s^{(i)}\right]_j^{j+l_i^h}\right)}. \quad (9)$$

Equation (9) represents the "local" error-pattern-dependent noise-predictive correlator output in the sense that it essentially describes the correlator operation between q_k and the channel output version of the dominant error pattern $e_j^{(i)}$ within the local region $[j, j + l_i^h]$, while accounting for the data-dependent correlated noise and any available side information.

V. Performance Evaluation

We investigate the performance of the proposed EPCC-TE scheme in a perpendicular magnetic recording channel (PMR) environment. The PMR read channel is modeled by a hyperbolic tangent transition response with normalized channel density D_s [14], with rate penalty proportional to $1/R^2$, measured in decibel scale, for a system running at a rate R. The normalized channel density is defined as the ratio of the width over -50% to 50% of the transition response's saturation level to the user bit period. The PMR channel is equalized to the partial response target $1 + 0.9D$, which was found to be among the best targets for the indicated environment that minimize the number of PDNP taps required to model the data-dependent noise memory. The noise mixture is assumed to be 10% AWGN and 90% jitter noise. For these channel parameters, the 10 most dominant error events are the polarity alternating error sequences of length 1 to 10 of the form $\pm[2, -2, 2, -2, \ldots]$. The SNR has been defined as the energy of the first derivative of the transition response E_{dt} to the noise spectral density N_{90}, which corresponds to 90% jitter noise [15]. The channel detector is BCJR with 1 PDNP tap running on an expanded trellis of 4 states.

1) Conventional TE: An outer 4-state $(7,5)_{\text{oct}}$ RSC code having punctured rate 8/9 is serially concatenated to the channel through a 4401 bit interleaver. The normalized user density is $D_u = 1.2$, and the normalized channel density is $D_s = 1.35$.

2) EPCC-TE: The EPCC-TE operates at the same rate as the conventional TE but divides the correction power between a punctured 10/11 outer $(7,5)_{\text{oct}}$ RSC and an inner 616/630 EPCC. The encoded output of the outer RSC is bit-interleaved by a 4401 bit interleaver and encoded into seven $(630, 616)$ EPCC codewords forming one sector that is passed to the channel without interleaving. The normalized user density is again $D_u = 1.2$, and the normalized channel density for EPCC-TE is also $D_s = 1.35$. The pattern-dependent correlator incorporates 4 PDNP taps, without requiring trellis expansion, to estimate the media noise given the input hard ML codeword, where the ML codeword is the output of the BCJR channel detector employing 1 PDNP tap. A total of $L = 25$ test words were constructed with each containing up to 2 corruptions, which allows the correction of up to $M = 3$ multiple occurrences. In the soft output generation stage of the EPCC SISO decoder, the value of the back-off factor was increased gradually each turbo iteration as $\beta^{\text{iter}} = [0.4, 0.4, 0.5, 0.6, 0.6, 0.8, 0.8, 0.9, 0.9, 1, 1, 1, 1, 1, 1]$, and $\lambda_{\max}$ in (2) was increased from 7 to 15 as SNR increased. Details on the design of the EPCC single pattern correcting code can be found in [1].

3) Quasi-Cyclic LDPC: The benchmark LDPC code was designed using the algebraic structured technique in [16], [17] to have performance that rivals the best comparable random codes at the design block length of 4699 and $R = 0.81$ with considerably lower encoding complexity. The LDPC parity check matrix is made up of a 7×37 array of circulant submatrices, where the (i, j)th submatrix is a 127×127 identity matrix with its rows cyclically shifted to the left by $(18^i \times 16^j)_{\text{mod } 127}$. The constructed LDPC code as such is a (7,37) regular code, that is quasi-cyclic (QC) with period 37. In the decoding process of this QC-LDPC code, 15 LDPC message-passing decoder iterations were performed each channel iteration.

4) Channel Capacity: To have an insight into the significance of the performance gap between the compared systems, their BER performance is compared to the capacity of the PMR channel constrained to have uniformly and independently distributed (UID) binary inputs, where the UID capacity is identical to the channel capacity at high code rates. The UID capacity of a media noise dominated channel is evaluated numerically exploiting the forward recursion of the BCJR/PDNP algorithm [18]. For a PMR channel with normalized channel density $D_s = 1.48$, and 90% media noise, the minimum SNR required to achieve a reliable rate $R = 0.81$ was computed to be 9.7 dB.

The BER comparison is shown in Fig. 3. Since it is hard to estimate the error rate performance involving an outer RS code, the BERs of TE systems were simply simulated for $R = 8/9$ and $D_s = 1.35$ without RS coding, although we envision TEs to be used with RS outer coding in practice. The actual gain for using an additional RS code of, say, $t = 20$ (lowering the overall code rate to 0.81) would depend on its correction capability in the presence of certain per-sector probability distribution of symbol errors, after considering the rate penalty associated with it. For the outer $t = 20$ RS code of rate $R_{\text{rs}} = 0.91$, the rate penalty is reasonably estimated as $10 \times \log_{10}(1/R_{\text{rs}}^2) = 0.82$ dB.

Simulations have shown that the BER of conventional TE saturates by the second iteration while the EPCC-TE performance gradually improves, and does not show a sign of saturation before the ninth iteration. Fig. 3 shows that EPCC-TE with

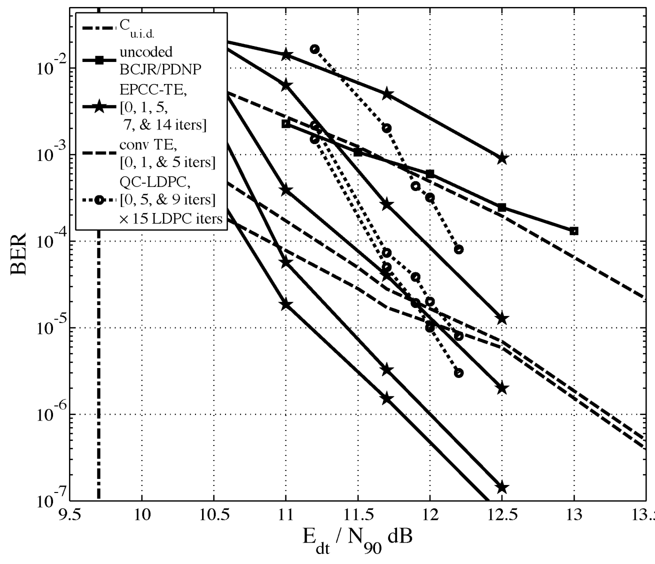


Fig. 3. BER comparison of conventional TE, EPCC-TE, QC-LDPC, and uncoded BCJR/PDNP, compared to UID capacity at a user density of 1.2, differenct channel densities, and 90% media noise.

25 test error words has a gain of 0.8 dB compared to conventional TE at a target BER of 10^{-5}. Moreover, in the worst case scenario of an outer $t = 20$ RS offering no coding gain after rate penalty, EPCC-TE would still be as good as QC-LDPC, and is 2.3 dB away from UID capacity. The EPCC-TE BER gains in this low-to-medium SNR region are beneficial if the per-sector symbol error burst statistics are not severe, since then, an outer RS code can drive the systems sector error rate (SER) down to a satisfactory low SER required in commercial disk drives. Simulations are ongoing to better understand the per-sector symbol error probability distribution and SER performance of EPCC-TE.

Still, with the limited simulation results we have thus far, we could observe that EPCC-TE does exhibit somewhat worse error propagation over the conventional TE system. This additional burstiness can be attributed in part to errors in the unprotected parity bits of EPCC teaming up with errors elsewhere to form multiple error occurrences that resist correction, with interleaving making the situation worse by spreading bit errors over several output symbol errors. To clarify further, for the inner EPCC parity no *a priori* information is supplied by the outer convolutional code, thus the EPCC correction power for those bits does not improve with more channel iterations, as is observed during simulation. On the other hand, our simulations show that the featured QC-LDPC [16], [17] generates a large number of erroneous symbols per sector on the onset of a decoding failure. If this QC-LDPC is concatenated to an outer RS code, most failed sectors it generates cannot be handled by the outer RS for any reasonable symbol-correction power.

VI. Conclusion

We enhanced the performance of TE by reallocating a small portion of the outer convolutional code redundancy to an inner high rate EPCC code that is tailored to the channel. The inner EPCC code attempts to correct any single dominant error pattern and a considerable portion of their multiple occurrences. While the burstiness of the errors in failed sectors seems worse than the conventional TE, pointing to a need for further investigation, the proposed EPCC-based turbo equalizer does show a BER advantage over conventional TE in the BER region of interest. Nevertheless, the BER performance of the proposed EPCC enhanced TE is comparable to that of LDPC, while the number of corrupted symbols is much smaller than in LDPC when sectors fail.

References

[1] J. Park and J. Moon, “A new class of error-pattern-correcting codes capable of handling multiple error occurrences,” *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 2268–2270, Jun. 2007.

[2] J. Park and J. Moon, “High-rate error correction codes targeting dominant error patterns,” *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2573–2575, Oct. 2006.

[3] M. Öberg and P. H. Siegel, “Performance analysis of turbo-equalized partial response channels,” *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 436–444, Mar. 2001.

[4] K. R. Narayanan, “Effect of precoding on the convergence of turbo equalization for partial response channels,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pt. 2, pp. 686–698, Apr. 2001.

[5] W. Ryan, “Performance of high rate turbo codes on a PR4-equalized magnetic recording channel,” in *Proc. Int. Conf. Communications*, Jun. 1998, vol. 2, pp. 947–951.

[6] J. Park and J. Moon, “Error-pattern-correcting cyclic codes tailored to a prescribed set of error cluster patterns,” *IEEE Trans. Inf. Theory*, submitted for publication.

[7] D. Chase, “Class of algorithms for decoding block codes with channel measurement information,” *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

[8] G. D. Forney, Jr., “Generalized minimum distance decoding,” *IEEE Trans. Inf. Theory*, vol. IT-12, no. 2, pp. 125–131, Apr. 1966.

[9] R. M. Pyndiah, “Near-optimum decoding of product codes: Block turbo codes,” *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 248–287, Mar. 1974.

[11] J. Caroselli, S. A. Altekar, P. McEwen, and J. K. Wolf, “Improved detection for magnetic recording systems with media noise,” *IEEE Trans. Magn.*, vol. 33, no. 5, pp. 2779–2781, Sep. 1997.

[12] J. Moon and J. Park, “Pattern-dependent noise prediction in signal-dependent noise,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr. 2001.

[13] A. Kavčić and J. M. F. Moura, “Correlation-sensitive adaptive sequence detection,” *IEEE Trans. Magn.*, vol. 34, no. 3, pp. 763–771, May 1998.

[14] M. Madden, M. Öberg, Z. Wu, and R. He, “Read channel for perpendicular magnetic recording,” *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 241–246, Jan. 2004.

[15] J. Moon and J. Park, “Detection of prescribed error events: Application to perpendicular recording,” in *Proc. IEEE ICC*, May 2005, vol. 3, pp. 2057–2062.

[16] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., “LDPC block and convolutional codes based on circulant matrices,” *IEEE Trans. Inf. Theory*, vol. IT-50, no. 12, pp. 2966–2984, Dec. 2004.

[17] D. Sridhara, T. Fuja, and R. M. Tanner, “Low density parity check codes from permutation matrices,” presented at the 2001 Conf. Information Sciences and Systems, The Johns Hopkins University, Mar. 21–23, 2001.

[18] Z. Zhang, T. M. Duman, and E. M. Kurtas, “Information rates of binary input intersymbol interference channels with signal-dependent media noise,” *IEEE Trans. Magn.*, vol. 39, no. 1, pp. 599–607, Jan. 2003.

Manuscript received May 18, 2007. Corresponding author: H. Alhussien (e-mail: hakimh@umn.edu).