PARITY CHECK CODES FOR PARTIAL RESPONSE CHANNELS

Mats Öberg and Paul H. Siegel, *Fellow, IEEE*
Dept. of Electrical and Computer Engineering, University of California, San Diego
La Jolla, California 92093-0407, {moberg, psiegel} @ucsd.edu

Abstract— **The use of turbo coding and decoding techniques in digital magnetic recording is now being given serious consideration. Major technical concerns are implementation complexity and decoding delay. In this paper, we present and analyze the performance of a simple, serial concatenation scheme comprising an outer parity check code, interleaver, and a precoded partial response channel. We apply an iterative decoding procedure incorporating separate *a posteriori probability* (APP) detectors for the code and precoded channel. Simulation results for a dicode channel show a bit-error-rate (BER) of 10^{-5} at rate-normalized signal-to-noise ratio SNR = 6.7 dB for a rate 8/9 code, representing a gain of about 3.5 dB over the uncoded channel. We also present simulation results for higher-rate codes and other partial response channels, confirming the performance benefits of the new scheme.**

I. Introduction

Turbo codes were introduced by Berrou *et al.* [1] in 1993 as a parallel concatenation of two or more recursive systematic encoders connected via interleavers, utilizing an iterative decoding procedure. These codes have been demonstrated to operate near Shannon's theoretical capacity on additive white Gaussian noise (AWGN) channels. The iterative decoding procedure has subsequently come to be referred to as turbo decoding. For the magnetic recording channel, several approaches related to the application of concatenated codes and iterative decoding have been explored. Ryan [2], Ryan, *et al.* [3], Heegard [4], and Pusch, *et al.* [5] have applied parallel-concatenated turbo codes to partial response channels of interest in digital recording. Reed and Schlegel [6] have evaluated the benefits of turbo-equalization for rate 1/2, convolutionally-coded, partial response channels. More recently, Souvignier, *et al.* [7], McPheters, *et al.* [8], and Öberg and Siegel [9] have investigated the performance of a serial concatenation of a high rate convolutional code, interleaver, and partial response channel, with iterative decoding. This simple scheme was found to perform as well as the more complex turbo-coded system down to a bit-error rate (BER) of about 10^{-5} [7],[8],[9].

In this paper we will propose a simple serial concatenation scheme for partial response channels, where the outer code is a concatenation of single parity check codes. The performance will be analyzed using the maximum likelihood union bound and simulation results will be shown. Section II will describe the components of the proposed system. In Section III, the analysis of the system performance will be summarized. Simulation results are presented in Section IV. In Section V, we propose a method to implement a runlenght constraint in the system. Finally, in Section VI, concluding remarks are provided.

This work was supported in part by the UC Micro Project 98-140 in cooperation with Marvell Semiconductor Incorporated and NSF grant NCR-9612802.

II. System Description

The proposed system is shown in Fig. 1 and consists of a parity check encoder, an interleaver, a precoded partial response channel and an iterative decoder structure.

A. Encoder

The parity-check encoder accepts N words $u_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,n-1}),\ i = 1, \ldots, N$ of $n-1$ information bits each. The encoder output consists of N words $c_i = (c_{i,1}, c_{i,2}, \cdots, c_{i,n}),\ i = 1, \ldots, N$ of n bits each, defined as follows:

$$c_{i,j} = \begin{cases} u_{i,j} & 1 \leq j < n \\ \sum_{k=1}^{n-1} u_{i,k} + 1 \mod 2 & j = n. \end{cases} \quad (1)$$

Thus, a bit is appended to each input word to ensure odd parity. The encoder implementation requires very little hardware, in principle only an XOR gate. As we will see later, the corresponding decoder is also very simple.

B. Interleaver

The interleaver performs a permutation of the Nn output bits from the encoder. Three types of interleavers will be discussed in this paper. The *pseudo-random* interleaver is just a randomly generated permutation of the encoder output. The *S-random* interleaver [10] is random as well, but mappings of bits that are closer than S in distance at the input cannot be closer than S at the output. The third type of interleaver is not a true permuter, but rather a probabilistic device. It is the average over all possible interleavers and will be referred to as a *uniform interleaver* [11].

This type of interleaver is more amenable to theoretical analysis of code performance.

C. Precoded Partial Response Channel

A linear channel with additive white Gaussian noise (AWGN) is assumed. Several partial response targets are considered in this paper. The first is the dicode channel $h(D) = (1 - D)$, which is also the simplest model and therefore used in the analysis. For this target, the precoder is $g(D) = 1/(1 \oplus D)$, where $\oplus$ denotes modulo-2 addition. The precoded dicode channel can be interleaved to model the precoded class-4 (PR4) partial response channel.

The other targets considered are "extended PR4" (EPR4) and E^2PR4 with transfer polynomials $h(D) = 1 + D - D^2 - D^3$ and $h(D) = 1 + 2D - 2D^3 - D^4$, respectively. For those targets, several precoders have been considered, all of the form $1/(1 \oplus D^{p_1} \oplus \ldots \oplus D^{p_k})$.

The transmission power is normalized so that the energy per code symbol $E_s = 1$. The signal to noise ratio (SNR) is defined as $SNR = 10 \log E_b/N_0$, where we set $E_b = E_s/R = 1/R$. The one-sided power spectral density $N_0 = 2\sigma^2$. Since the rate $R = (n-1)/n$, we have $E_b = n/(n-1)$ and the noise variance is $\sigma^2 = n/(2(n-1)10^{SNR/10})$. The noise is added at the output of the partial response channel.

D. Decoder

The turbo decoding is performed by two soft-in soft-out (SISO) decoders that pass information between each other via an interleaver/deinterleaver. The SISO's are matched to the precoded channel and the parity check encoder, respectively. Each SISO is an *a posteriori* probability (APP) detector, which computes the *a posteriori* probability of the corresponding encoder input and/or output symbol, using *a priori* information. Fig. 2 depicts a general APP detector block. The

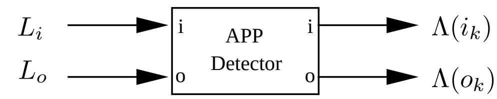

Fig. 2. General APP detector.

symbols corresponding to the encoder input and output are denoted as i and o, respectively. The inputs L_i and L_o denote *a priori* information for encoder input and output symbols. The $\Lambda(i_k)$ and $\Lambda(o_k)$ denote *a posteriori* probabilities corresponding to encoder inputs and outputs, respectively. For a symbol u, drawn from some finite alphabet of size l, $\mathcal{A} = \{a_1, a_2, \ldots, a_l\}$, the general *a priori* and *a posteriori* probabilities are used to form log-APP ratios as follows:

$$L(u = a_j) = \log \frac{Pr(u = a_j)}{Pr(u \neq a_j)} \tag{2}$$

$$\Lambda(u = a_j) = \log \frac{Pr(u = a_j | \underline{L}_i, \underline{L}_o)}{Pr(u \neq a_j | \underline{L}_i, \underline{L}_o)} \tag{3}$$

where $\underline{L}_i$ is a vector containing all *a priori* information regarding encoder inputs, and $\underline{L}_o$ is a vector containing all *a priori* information regarding encoder outputs.

In the case of the binary alphabet $\mathcal{A} = \{0, 1\}$, we use the shorthand notations $L(u) \overset{\text{def}}{=} L(u = 1)$ and $\Lambda(u) \overset{\text{def}}{=} \Lambda(u = 1)$. Note that then $L(u = 0) = -L(u)$ and $\Lambda(u = 0) = -\Lambda(u)$. The channel APP is matched to the precoded partial response channel. The number of detector trellis states for the dicode, PR4, EPR4, and E^2PR4, are 2, 4, 8, and 16, respectively.

For the simulations, we based the APP detector for the block parity-check code on a two-state trellis representation of the constituent parity-check encoder. A simpler, yet equivalent implementation, is the algo-

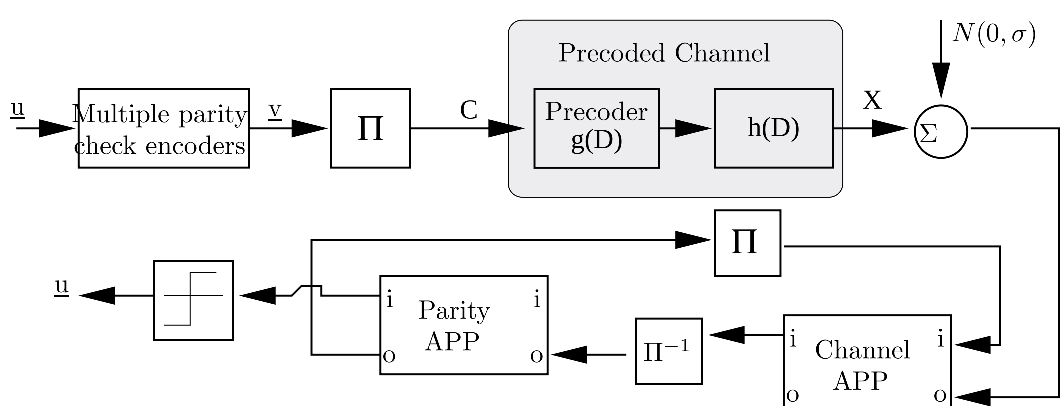

Fig. 1. Proposed system.

rithm given in [12] and Gallager's parity-check decoder [13]. Due to the independence between the parity-check codewords, the decoder can use a window equal to the codeword length n. The short window length opens up possibilities for parallel implementations to improve the speed of the detector, although this direction is not further pursued in this paper.

III. Analysis

We have analyzed the performance of the proposed system by computing a maximum likelihood union bound for the probability of word error. Although the decoder does not implement maximum likelihood sequence estimation (MLSE), the performance of the iterative decoding structure has been shown to be close to that of MLSE. The maximum–likelihood (ML) union bound on word error rate (WER) for a block-coded, additive white Gaussian noise (AWGN) channel can be expressed as [14]

$$P_w \leq \sum_{d_E=d_{min}}^{\infty} \overline{T}(d_E) Q\left(\frac{d_E}{2\sigma}\right), \qquad (4)$$

where d_E denotes Euclidean distance between two channel output words, σ^2 denotes the noise variance on the channel and $\overline{T}(d_E)$ denotes the average Euclidean weight enumerator, which is the average number of codewords whose channel outputs have Euclidean distance d_E from the output of a given codeword. The corresponding bit error rate (BER) bound [14] is

$$P_b \leq \sum_{d_E=d_{min}}^{\infty} \frac{\overline{T}(d_E)\overline{w}(d_E)}{K} Q\left(\frac{d_E}{2\sigma}\right), \qquad (5)$$

where K denotes the number of information bits in a codeword and $\overline{w}(d_E)$ denotes the average information Hamming distance between codewords whose channel outputs have Euclidean distance d_E.

For an exact analysis, the full compound error-event characterization for a code interleaved and concatenated with the partial response channel must be determined. The complexity of this computation is often prohibitively high. To overcome this difficulty, we use a technique introduced in [9] for computing an approximation to the average weight enumerator for a high-rate, coded partial response channel. For completeness, we briefly describe the application of this approximation in this setting.

Fig. 3 shows a trellis section for the dicode channel with precoder $g(D) = 1/(1 \oplus D)$. The branch labels are of the form c_i/x_i, where c_i is the input to the precoder at time i, and x_i is the corresponding channel output. Referring to Fig. 3, it can be seen that an error word f

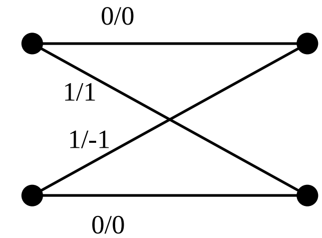

Fig. 3. Trellis section for the precoded dicode channel.

may be decomposed into a sequence of $m = \lceil d_H(f)/2 \rceil$ simple error sub-events f_i, $i = 1, \ldots, m$. For $1 \leq i \leq m-1$, each sub-event is closed, sub-event f_m may be either closed or open. The length of the sub-event f_i is denoted l_i, and the Hamming weight of a sub-event satisfies

$$d_H(f_i) = \begin{cases} 2 & i = 1, \ldots, m-1 \\ 2 & i = m \text{ and } d_H(f) \text{ even} \\ 1 & i = m \text{ and } d_H(f) \text{ odd.} \end{cases} \qquad (6)$$

Let j_i^0 denote the bit position in the word where error sub-event f_i begins. For a closed sub-event, let j_i^1 denote the bit position where it terminates. Then $l_i = j_i^1 - j_i^0 + 1$ for all closed sub-events. If f_m is open, we define $j_m^1 = N + 1$, and $l_m = j_m^1 - j_m^0$. Finally we define $L = \sum_{i=1}^{m} l_i$.

The error word f has total squared Euclidean distance

$$d_E^2(f) = \sum_{i=1}^{m} d_E^2(f_i) = d_H(f) + 4 \sum_{i=1}^{m} \sum_{k=j_i^0+1}^{j_i^1-1} c_k. \qquad (7)$$

The approximation is based upon the assumption that the code bit values in the error events may be treated as samples of independent, equiprobable binary random variables. Under this "i.i.d. assumption," the contribution of an error word f to the average weight enumerator is given by the distribution

$$Pr(d_E^2(f) = z | d_H(f) = d, L) = \binom{L-d}{(z-d)/4} 0.5^{L-d}. \qquad (8)$$

The i.i.d. assumption is justified by the action of the uniform interleaver for error words corresponding to short error event duration. On the other hand, when the duration of error events is long, the contribution to the dominant terms of the Euclidean error weight enumerator will be negligible, due to the low probability of such an error word generating small Euclidean distance. For a general linear block code, the accuracy of the i.i.d. assumption can be measured by reference to the weight enumerator of the dual code. In this instance, we are interested in the dual code of the N-fold concatenation of even parity-check codes, which is simply the N-fold concatenation of $(n,1)$ repetition codes.

For example, consider the rate 8/9 system consisting of $N = 128$ concatenated parity-check codes with an interleaver of length 1152. The minimum distance of the dual code is 9, with multiplicity 128. Therefore, any 8 bits at the interleaver output are linearly independent, and the probability of choosing 9 linearly dependent bits is $128/\binom{1152}{9}$. These remarks apply also to the concatenation of odd parity-check codes; moreover, in any set of 9 dependent code bits, at least one of the bits must be a 1. In fact, there will be at least one symbol 1 in any set of linearly dependent code symbols at the interleaver output. This leads us to a conjecture that the i.i.d. assumption upper bounds the ML union bound for the system proposed in this paper for all rates below the computational cut-off rate. This conjecture will, however, be addressed elsewhere.

In [9], the distribution of the total length L of error words f generated by the action of a uniform interleaver upon an error word e of Hamming weight d was shown to be

$$Pr(L|d) = \frac{\binom{N-L+\lfloor d/2\rfloor}{\lfloor d/2\rfloor}\binom{L-1-\lceil (d-1)/2\rceil}{\lfloor (d-1)/2\rfloor}}{\binom{N}{d}}. \qquad (9)$$

The approximation of the Euclidean weight enumerator depends only upon the input-output Hamming weight enumerator of the outer code $A(d) = \sum_{i=0}^{K} A(d,i)$, where $A(d,i)$ denotes the number of error words of Hamming output weight d and input weight i. It can be computed by substituting (8) and (9) into

$$\overline{T}(d_E) = \sum_{k=1}^{N} A(k) \sum_{L=k}^{N-k} Pr\left(d_E|k,L\right) Pr\left(L|k\right). \qquad (10)$$

Similarly, the approximate average input error weight enumerator may be obtained from

$$\overline{w}(d_E) = \frac{1}{\overline{T}(d_E)} \sum_{k=1}^{N} A(k)\overline{W}(k) \sum_{L=k}^{N-k} Pr\left(d_E|k,L\right) Pr\left(L|k\right),$$

where $\overline{W}(d)$ is the average input weight for output weight d.

For the concatenation of N $(n,n-1)$ even parity-check codes, the Hamming weight enumerating function $IOWEF(D,I)$ is the product of N weight enumerating functions for a single $(n,n-1)$ even parity-check code

$$IOWEF(D,I) = \sum_{i\geq 0, d\geq 0} A(d,i) D^d I^i = \left[\sum_{j=0}^{n-1} \binom{n-1}{j} D^{2\lceil j/2\rceil} I^j\right]^N. \qquad (11)$$

Since the odd parity-check code is a coset of the even parity-check code, the weight enumerating function for the even parity-check code can be used to enumerate the weights of error words for the odd parity-check code.

IV. Analytical and simulated results

We computed an estimate of the word-error-rate (WER) upper bound for the rate 8/9 system on the precoded dicode ($h(D) = 1-D$) channel with $N = 128$ and a uniform interleaver, as outlined in the previous section. The estimate is shown in Fig. 4 together with simulation results. We have also plotted simulation results for different interleavers at $E_b/N_0 = 8.0$ dB. Note how the corresponding points are located on both sides of the estimated bound, consistent with the fact that the analysis assumes a uniform interleaver. The agreement is quite good in all cases.

In Fig. 5, the simulated bit-error-rate (BER) performance for the rate 8/9 system with $N = 512$ and a randomly-generated interleaver is compared to that of a system using an S-random interleaver, with $S = 30$. [10]. Clearly the S-random interleaver improves the performance of the system. The better performance with the S-random interleaver can be explained by analyzing the effects of (8) and (9) on (10). This analysis depends upon the particular S-random interleaver, but a heuristic understanding follows from the following observations. First, note that the value of (8) increases as L increases. For a parity check code with $n < S$, the use of an S-random interleaver implies

$$Pr(L|2) = 0 \text{ for } L \leq S, \qquad (12)$$

because the S-random interleaver cannot map two bits from the same parity check codeword to positions closer than S. Hence the non-zero contributions from $Pr(L|k)$ in (10) for $k = 2$ must correspond to values of L greater than S. For $S > \log_2(N)$, the value of $\overline{T}(\sqrt{2})$, (10) will be smaller for the S-random interleaver than for the uniform interleaver.

Fig. 6 shows simulation results for rate 8/9, 16/17, and 24/25 parity check codes on the dicode channel using S-random interleavers. Included in the graph,

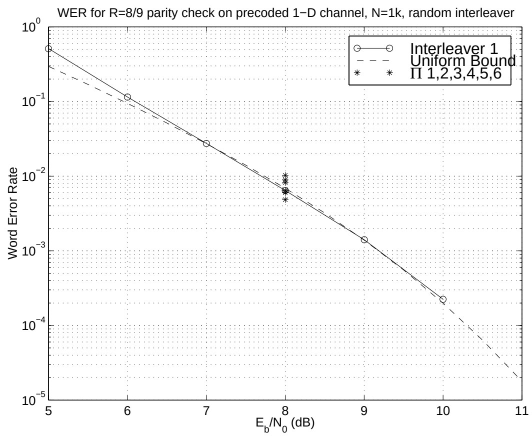

Fig. 4. Bound and simulation results for rate 8/9 system of length 1k

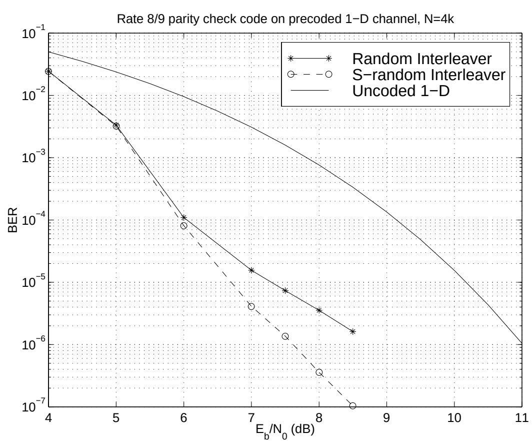

Fig. 5. Simulation results for random and S-random interleaver of length 4k

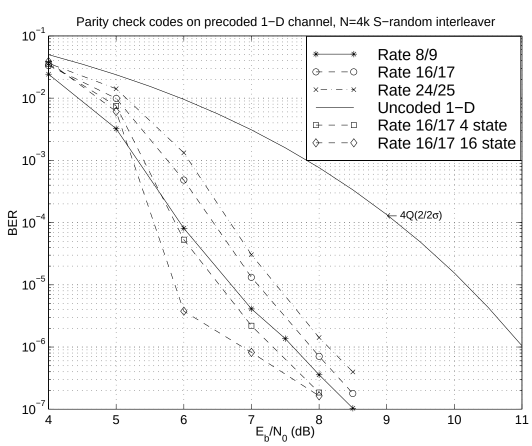

Fig. 6. Simulation results for different rates with S-random interleaver of length 4k

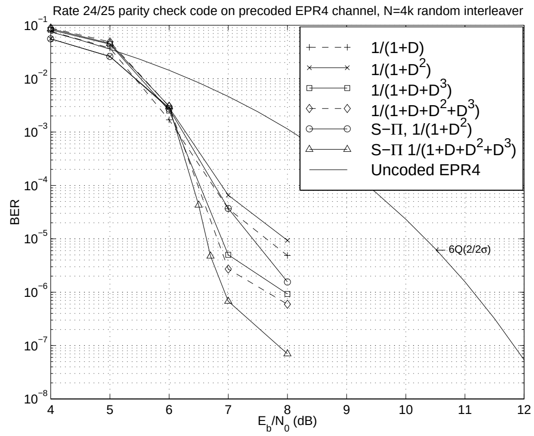

Fig. 7. Simulation results for different precoders on EPR4 channel

for comparison purposes, are performance curves corresponding to 4-state and 16-state recursive systematic convolutional (RSC) outer codes, using an S-random interleaver. The outer codes were rate 1/2, with encoder polynomials $(1, 5/7)_{octal}$ and $(1, 33/31)_{octal}$, punctured to rate 16/17. These are the outer codes used in [7]and [9], although the results reported therein were for a random interleaver. The system with the 16-state RSC outer code outperforms the system with parity check codes by more than 1 dB at BER 10^{-5}, but at BER 10^{-7} the difference is only about 0.5 dB. The performance of the system with the 4-state RSC outer code is also better than that achieved with the parity check code, but only by about 0.5 dB, even at BER 10^{-5}.

The results for higher order channels are similar. For example, Fig. 7 shows simulation results for a rate 24/25 parity-check code on an EPR4 channel, using a pseudo-random interleaver. Results were obtained for four different precoders: $1/(1 \oplus D)$, $1/(1 \oplus D^2)$, $1/(1 \oplus D \oplus D^3)$ and $1/(1 \oplus D \oplus D^2 \oplus D^3)$. The poorer performance of the first two precoders can be attributed, in part, to the fact that weight-1 sequences can be generated at their output by certain weight-2 input sequences, namely $(1 \oplus D)$ and $(1 \oplus D^2)$, respectively. The figure also shows the performance for two of these precoders when an S-random interleaver with $S = 30$ was used. Although not shown, when the proposed scheme is applied to the E^2PR4 channel, the coding gains relative to the uncoded channel are similar.

V. Imposing a Runlength Constraint

In Fig. 8 we show how a runlength limitation (RLL) can be imposed on the system. The RLL-encoder generates code sequences with no more than $\Delta - 1$ zeros in a row. The interleaver permutes the output from

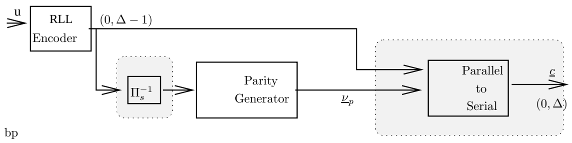

Fig. 8. Runlength constrained serial-concatenated system.

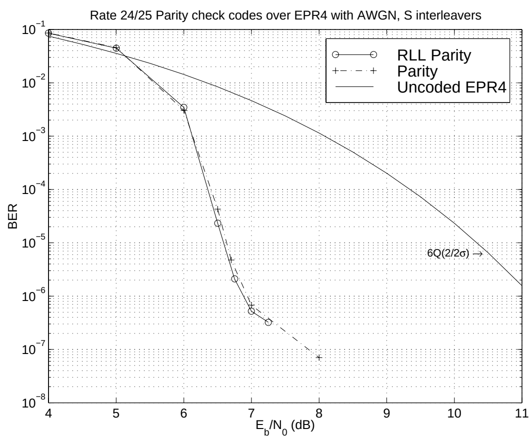

Fig. 9. Comparison of systems with and without RLL constraint.

the RLL-encoder, before feeding it to the parity bit generator. The parity generator outputs the parity bits corresponding to a parity check encoder with odd-parity. The parity bits are then inserted periodically into the sequence from the RLL-encoder. If the code rate $R \geq \Delta/(\Delta+1)$, then no more than Δ zeros in a row will be fed into the precoded partial response channel. That is, we have a $(0, \Delta)$ RLL constraint on the sequence that is fed to the precoded channel. We have simulated the proposed system with a rate $16/17$, $(0, 6)$ constraint RLL encoder on the EPR4 channel with precoder $1/(1 \oplus D \oplus D^2 \oplus D^3)$. The parity check code is of rate $24/25$. This guarantees that we do not see more than 7 consecutive zeros at the input to the precoder. An S-random interleaver of length 4080 is used.

In Fig. 9 the BER performance of this RLL-encoded system is compared to the corresponding rate $24/25$ system without the RLL constraint. The two systems display nearly identical performance.

VI. Conclusions

A simple, serial concatenated, precoded partial response system with an outer parity check encoder, interleaver, and an iterative (turbo) decoding technique has been presented for magnetic recording applications. The low complexity of the system is attractive for hardware implementations.

Analytical and simulation results show that this is an attractive approach to increase the capacity in magnetic storage devices. The performance in terms of bit error rate (BER) for a rate $16/17$ system on the dicode channel is 10^{-5} at $E_b/N_0 = 7.1$ dB. This is only about 1.7 dB worse than a corresponding system with a 16-state outer convolutional code, and about 3 dB better than an uncoded system. At BER of 10^{-7} the performance difference is only about 0.5 dB. With a 4-state convolutional outer code, the difference is about 0.5 dB at most bit error rates.

We have also shown how to add a runlength limitation to this system, and that it performs comparably to the system without an outer RLL code.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[2] W. E. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *Proc. IEEE Int. Conf. Commun.*, (Atlanta, GA), pp. 947–951, June 1998.

[3] W. E. Ryan, L. L. McPheters, and S. W. McLaughlin, "Combined turbo coding and turbo equalization for PR4-equalized Lorentzian channels," in *Proc. Conf. on Inform. Sciences and Systems*, March 1998.

[4] C. Heegard, "Turbo coding for magnetic recording," in *Proc. IEEE Inform. Theory Workshop*, (San Diego, CA), pp. 18–19, Feb. 1998.

[5] W. Pusch, D. Weinrichter, and M. Taferner, "Turbo-codes matched to the $1 - D^2$ partial response channel," in *Proc. IEEE Int. Symposium on Inform. Theory*, (Cambridge, MA), p. 62, Aug. 1998.

[6] M. C. Reed and C. B. Schlegel, "An iterative receiver for the partial response channel," in *Proc. IEEE Int. Symposium on Inform. Theory*, (Cambridge, MA), p. 63, Aug. 1998.

[7] T. Souvignier, A. Friedman, M. Öberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo codes for PR4: Parallel versus serial concatenation," in *Proc. IEEE Int. Conf. Commun.*, (Vancouver, BC, Canada), pp. 1638–1642, June 1999.

[8] L. L. McPheters, S. W. McLaughlin, and K. R. Narayanan, "Precoded PRML, serial concatenation and iterative (turbo) decoding for digital magnetic recording," in *Proc. Int. Magn. Conf.*, (Kyongju, Korea), May 1999. To appear.

[9] M. Öberg and P. H. Siegel, "Performance analysis of turbo-equalized dicode partial-response channel," in *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, (Monticello, IL, USA), pp. 230–239, Sept. 1998.

[10] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. IEEE Int. Conf. Commun.*, (Seattle, WA), pp. 54–59, June 1995.

[11] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–428, March 1996.

[12] T. Johansson and K. Zigangirov, "A simple one-sweep algorithm for optimal APP symbol decoding of linear block codes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 3124–3128, Nov. 1998.

[13] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, jan 1962.

[14] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 18, pp. 363–378, May 1972.