

Turbo Decoding for Partial Response Channels

Tom V. Souvignier, *Member, IEEE*, Mats Öberg, *Student Member, IEEE*, Paul H. Siegel, *Fellow, IEEE*, Robert E. Swanson, and Jack K. Wolf, *Life Fellow, IEEE*

Abstract—The partial response channel can be viewed as a rate-1 encoder in which the output alphabet differs from the input alphabet. In serially concatenated coding schemes, the partial response channel can serve as the inner encoder. Recent work on the application of turbo decoding techniques to partial response channels has focused on using a parallel concatenation of convolutional encoders as the outer code and the partial response channel as the inner code. This system requires three *a posteriori* probability (APP) detectors—one matched to the channel and two matched to the constituent encoders. A simplified system will be presented that uses as its outer code a single convolutional code and as its inner code the partial response channel. The simplified system requires only two APP detectors, offering significant savings in complexity and computation time. This single convolutional code system will be shown to perform as well as the more complicated system, offering substantial gains over uncoded systems. Simulation results for three magnetic recording channel models will be presented: a partial response channel with additive white Gaussian noise, an equalized Lorentzian channel model, and a media noise model called the microtrack model. Since the use of an outer Reed–Solomon code is anticipated in an actual system, the burst-error statistics are investigated. System performance with various interleaver designs and precoders is also investigated.

Index Terms—Concatenated coding, iterative methods, partial response signaling, turbo codes.

I. INTRODUCTION

TURBO codes were introduced by Berrou *et al.* [1] in 1993 as two or more parallel concatenated convolutional codes (PCCC's) connected with an interleaver and decoded using an iterative technique. These codes are capable of operating near Shannon capacity on an additive white Gaussian noise (AWGN) channel. The term “turbo decoding” has subsequently come to refer to this iterative decoding process. Therefore, in this paper,

Paper approved by S. S. Pietrobon, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received April 27, 1999; revised July 22, 1999 and December 1, 1999. This work was supported in part by the National Science Foundation under Grant NCR-9405008 and Grant NCR-9612802, in part by a UC MICRO Grant 98-140 in cooperation with Marvell Semiconductor, Inc., in part by the Center for Magnetic Recording Research at the University of California–San Diego, and in part by a Seagate Technology Fellowship.

T. V. Souvignier is with Texas Instruments, Incorporated, Longmont, CO 80503 USA (e-mail: tom.souvignier@ti.com).

M. Öberg is with Marvell Semiconductors, Inc., Sunnyvale, CA 94085 USA (e-mail: moberg@marvell.com).

P. H. Siegel is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: psiegel@cwc.ucsd.edu).

R. E. Swanson is with Seagate Technologies Corporation, Fremont, CA 94538 USA (e-mail: bob_swanson@notes.seagate.com).

J. K. Wolf is with the Center for Magnetic Recording Research, University of California at San Diego, La Jolla, CA 92039-0401 USA (e-mail: jwolf@ucsd.edu).

Publisher Item Identifier S 0090-6778(00)07099-9.

turbo decoding will refer to the iterative decoding process for both parallel and serial concatenation systems.

Considerable work has been done recently by Heegard [2], McPheters *et al.* [3], Reed and Schlegel [4], Ryan [5], Ryan *et al.* [6], Souvignier *et al.* [7]–[9], and others investigating the application of turbo decoding to partial response channels. Related work includes [10]–[14]. The technique in [5] involves using an *a posteriori* probability (APP) detector matched to the partial response channel, followed by the standard turbo system of two APP detectors matched to the constituent PCCC's. A simpler architecture, where the inner encoder is the precoded partial response channel and the outer encoder is a single convolutional code (SCC), will be shown to perform comparably to the structure utilizing an outer PCCC. A similar architecture is proposed in [11], but the application is very different. The effects of precoding on both architectures will be investigated. In particular, the choice of precoders will be shown to significantly affect performance.

The APP detectors of the above schemes assume uncorrelated Gaussian noise. The performance on more realistic channel models, with correlated and media-dependent noise, remains largely uninvestigated. In [5], Ryan colored the noise for the PR4 parallel structure by passing AWGN through an equalization filter. The systems in this paper will be investigated using a simple white noise channel model, a colored noise channel model, and a media noise channel model. A common characteristic of concatenated coding systems is that their performance curves [error probability versus signal-to-noise ratio (SNR)] generally fall off steeply (in the “cliff region”) and then level off (in the “floor region”). Since an actual system would operate in the floor region to ensure robustness to SNR variation, and since the use of an outer Reed–Solomon (RS) byte-error correcting code is anticipated, the byte-error burst statistics on the error floor will be investigated. Also, the choice of precoders will be shown to significantly affect the error floor location. Finally, designing the interleaver to avoid certain permutations will be shown to improve average bit-error rate (BER) performance [15].

Section II provides some necessary background, including a brief review of the components of the systems being considered. Section III includes a review of the concatenated encoding architectures for partial response systems and their corresponding decoders. Simulation results are provided in Section IV, and concluding remarks in Section V.

II. SYSTEM COMPONENTS

The various components of the systems shown in Figs. 1 and 2 will be described in this section.

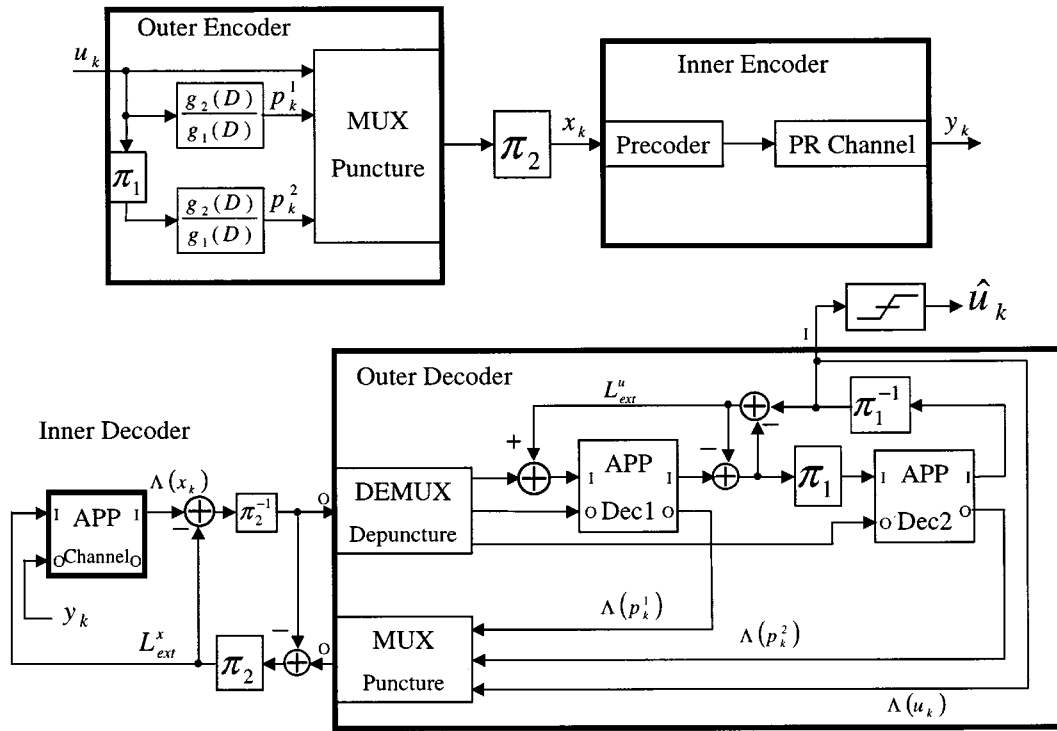


Fig. 1. PCCC as outer encoder with full turbo decoder.

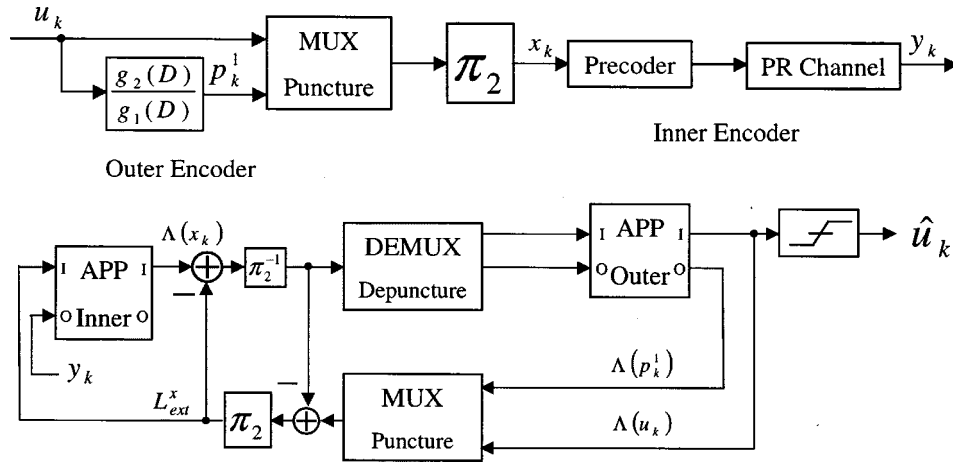


Fig. 2. SCC as outer encoder with iterative decoder.

A. APP Detector

It may be beneficial to contrast the well-known Viterbi algorithm (VA) with the APP algorithm. The VA finds the symbol sequence \mathbf{x} that maximizes the likelihood function $P(\mathbf{r}|\mathbf{x})$, where \mathbf{r} is the received sequence. The APP algorithm calculates the APP of each transmitted symbol given the received sequence and *a priori* information [16], [17]. The two APP detector outputs are logarithms of APP ratios (LAPPR's) computed as

$$\Lambda_i(w_k = a_j) = \log \frac{P(w_k = a_j | R_1^N, L_{i,1}^N, L_{o,1}^N)}{P(w_k \neq a_j | R_1^N, L_{i,1}^N, L_{o,1}^N)} \quad (1)$$

and

$$\Lambda_o(z_k = b_j) = \log \frac{P(z_k = b_j | R_1^N, L_{i,1}^N, L_{o,1}^N)}{P(z_k \neq b_j | R_1^N, L_{i,1}^N, L_{o,1}^N)}. \quad (2)$$

The i and o indicate the corresponding encoder input and encoder output, respectively. Here, the encoder input symbol w is drawn from some finite alphabet $A = \{a_1, a_2, \dots, a_i\}$ and the encoder output symbol z is drawn from some finite alphabet $B = \{b_1, b_2, \dots, b_m\}$. The APP $P(w_k = a_j | R_1^N, L_{i,1}^N, L_{o,1}^N)$ is conditioned on the received channel sequence R_1^N on *a priori* information regarding the encoder input $L_{i,1}^N$ and on *a priori* information regarding the encoder output $L_{o,1}^N$. $L_{i,1}^N$ and $L_{o,1}^N$ are logarithms of the ratio of the *a priori* probabilities of the encoder input and output computed as

$$L_i(w_k = a_j) = \log \frac{P(w_k = a_j)}{P(w_k \neq a_j)} \quad (3)$$

and

$$L_o(z_k = b_j) = \log \frac{P(z_k = b_j)}{P(z_k \neq b_j)}. \quad (4)$$

$P(w_k = a_j)$ is the *a priori* probability of the encoder input and $P(z_k = b_j)$ is the *a priori* probability of the encoder output. Equations (1) and (2) are general expressions; the algorithm can be modified for situations where one or more of the conditions are not available. It should be noted that (1) and (2) are often referred to as log-likelihood ratios. Since the numerator and denominator are not likelihoods, but APP's, we have chosen to refer to these quantities as log-APP ratios [18].

B. RSC Encoder

The rate-1/2 recursive systematic convolutional (RSC) encoders are identified in Figs. 1 and 2 by $g_2(D)/g_1(D)$, where $g_1(D)$ is the feedback polynomial and $g_2(D)$ is the feedforward polynomial.

C. MUX-Puncture

The puncturing is accomplished by omitting as many code symbols as necessary to achieve the desired rate. For the simulation results presented, no data symbols were punctured, and parity symbols were punctured in the following manner. For the rate $k/(k+1)$ architecture utilizing an outer PCCC, the data symbols were transmitted, then every $(2k)$ th parity symbol was transmitted from each parity sequence. For the rate $k/(k+1)$ architecture utilizing an outer SCC, the data symbols were transmitted, then every k th parity symbol was transmitted. The multiplexing function creates a sequence consisting of all of the data symbols followed by all of the unpunctured parity symbols. The DEMUX-depuncture block simply reverses the process of the MUX-puncture block, converting a serial sequence to parallel sequences, and placing erasures in locations of the punctured symbols.

D. Interleaver

The block interleaver π is a device that permutes a block of N symbols. The corresponding deinterleaver π^{-1} simply reverses the process. Four different block interleavers will be discussed briefly. The first, the random interleaver, pseudorandomly permutes a block of N symbols. A second type of block interleaver is the S -random interleaver [15], which has the constraint that all symbols spaced S or fewer positions apart in the original sequence are spaced more than S positions apart in the permuted sequence. A third interleaver, a modification of the S -random interleaver termed the ST -random interleaver and presented in [19], has the additional constraint that all symbols spaced T or fewer multiples of L positions apart ($L, 2L, \dots, TL$) in the original sequence are not spaced T or fewer multiples of L positions apart in the permuted sequence. The fourth interleaver, also a modification of the S -random interleaver, is presented here and termed the ST_s -random interleaver. The ST_s -random interleaver has, in addition to the S constraint, the constraint that all symbols spaced T_s or fewer multiples of L positions apart in the original sequence are spaced more than S positions apart in the permuted sequence. Both the ST -random interleaver with $T = 0$ and the ST_s -random interleaver with $T_s = 0$ are equivalent to the S -random interleaver.

The S constraint ensures that error bursts shorter than S are dispersed by the interleaver. Practical values of S , in terms of

constructing the interleaver, are $S < \sqrt{N/2}$ [15]. The importance of the T_s constraint and the value of the L parameter are explained as follows. Define L to be the smallest integer such that feedback polynomial $g_1(D)$ is a divisor, in $\text{GF}(2)[D]$, of $1+D^L$. Then, $g_1(D)$ also divides $1+D^{pL}$ for any integer $p > 0$ (since $1+D^L$ is a factor of $1+D^{pL}$). Therefore, an encoder input sequence $1+D^L$ (1 followed by $L-1$ zeros followed by 1) is an input error event for an RSC encoder with feedback polynomial $g_1(D)$. Any input sequence $1+D^{pL}$, $p > 0$, is also an error event. For a heavily punctured code, the output Hamming weight of such an error event might equal the input Hamming weight for relatively small p . The T_s constraint ensures that the two errors in such an event are separated by at least S in the permuted sequence, and therefore the output Euclidean distance is likely to be higher [20]. If $T_s \leq LS$, the T_s constraint is not beneficial.

As an example, consider the RSC encoder described by generator polynomials, in octal form, $(g_1, g_2) = (31, 33)$ used in a system with interleaver size $N = 4096$. The shortest weight-2 error event divisible by $g_1(D) = 1 + D + D^4$ (31 in octal) is $1 + D^{15}$. When the code is punctured to rate-16/17 as described in the previous section, the longest input weight-2 event of the form $1 + D^{pL}$ generating output weight-2 codewords is $1 + D^{60}$ ($L = 15, p = 4$). The largest practical value of S is $S = \lfloor \sqrt{N/2} \rfloor$. Since $S = 45$ is smaller than 60, the $T_s = 4$ constraint ensures that two errors in the error event are not mapped close to each other.

E. Partial Response Channel

Channels with discrete-time transfer functions of the form

$$F(D) = \sum_{i=0}^N f_i D^i, \quad f_0 = 1; f_i \text{ integer} \quad (5)$$

where D is the unit-delay operator corresponding to one modulation interval T of the channel, termed partial response channels [17]. Of particular interest in magnetic recording are partial response polynomials of the form

$$F(D) = (1 - D)(1 + D)^n, \quad n = 0, 1, 2, \dots \quad (6)$$

For $n = 0$, the channel is the 1-D or "dicode" partial response channel. The channel for $n = 1$ is termed the class-4 partial response (PR4) channel and the channel for $n = 2$ is termed the extended class-4 partial response (EPR4) channel [21]. Higher values of n result in more intersymbol interference (ISI) and permit higher recording densities.

F. Precoder

Precoding the channel data with a rate-1 encoder has been shown to be beneficial when turbo decoding partial response channels [7], [22]. For the PR4 systems considered in this paper, the precoder used was $1/(1 \oplus D^2)$, where \oplus indicates modulo-2 addition. For the EPR4 systems considered in this paper, the precoder used was either $1/(1 \oplus D^2)$ or $1/(1 \oplus D \oplus D^2 \oplus D^3)$. Since the PR4 polynomial is $1 - D^2$ and the EPR4 polynomial is $1 + D - D^2 - D^3$, the precoders chosen do not increase the decoder complexity.

G. White Noise Channel Model

A very simple discrete-time model of a digital magnetic recording channel subject to ISI assumes a partial response channel polynomial as in (6), followed by the discrete-time equivalent of AWGN.

H. Equalized Lorentzian Channel Model

In digital magnetic recording systems using nonreturn to zero (NRZ) coding, binary information is stored in tracks on the medium by magnetizing the medium completely in one direction to represent a 0 and completely in the other direction to represent a 1. Transitions in magnetization are permitted to occur only at multiples of a fundamental linear spacing corresponding to the channel bit duration T . The transition sequence represents nonreturn to zero inverted (NRZI) coding of the data. During playback, the read head detects the magnetic field emanating from the medium. The response of the read head to a transition in magnetization along the track, denoted $s(t)$, is termed the step response or transition response. Here, we will model this step response with a Lorentzian pulse

$$s(t) = \frac{1}{1 + (2t/PW_{50})^2} \quad (7)$$

where the only parameter is PW_{50} , the width of the pulse at 50% of its peak value. Since $s(t)$ is a response to NRZI data, the NRZ data are convolved with the continuous-time dibit response, $h(t) = (1/2)(s(t) - s(t - T))$. Electronic noise in the system is modeled as a stationary AWGN process with uniform two-sided spectral density $N_0/2$. The noisy dibit response is passed through an ideal low-pass filter with cutoff frequency $1/(2T)$ Hz and then sampled every T s to form the discrete time sequence \mathbf{w} , where $w_k = w(kT)$.

The linear density of the transmitted signal, the channel density, is defined as PW_{50}/T . As the channel density increases, the energy in the dibit response $h(t)$ decreases. Therefore, reliable detection at higher channel recording densities is more difficult.

The discrete-time model, a close approximation to the continuous-time model, assumes a discrete-time dibit response \mathbf{h} , where $h_k = h(kT)$, and an independent and identically distributed Gaussian noise sequence \mathbf{n} . The (precoded) recorded data sequence $\tilde{\mathbf{x}}$ is convolved with the dibit response \mathbf{h} . The addition of the noise sequence \mathbf{n} results in the equalizer input

$$w_k = (\tilde{\mathbf{x}} * \mathbf{h})_k + n_k \quad (8)$$

where “*” denotes linear convolution. This sequence is passed through the equalizer with impulse response \mathbf{f} . The objective of this equalizer is to shape the channel response to the desired partial response target. Passing white noise through the equalizer results in colored noise at the equalizer output.

I. Microtrack Channel Model

The pertinent characteristics of the microtrack model are discussed here. For a more thorough explanation, see [23] and [24].

In the equalized Lorentzian model, transitions in magnetization are assumed to be ideal. That is, the polarity of the media grains is assumed to reverse exactly at the bit cell boundary

across the entire track. A more realistic model assumes each transition will have a random zigzag shape [25]. The microtrack model approximates the randomness of the zigzag transition shape, producing the effects of transition noise including position jitter and pulse widening. In this model, the track along which the signal is written is thought to be divided into M thinner tracks of equal width called microtracks. When a transition is written, an instantaneous magnetization reversal is thought to occur on each microtrack with its position being independent from microtrack to microtrack. For the i th microtrack, the displacement of this position τ_i around the ideal center of the transition is characterized by the probability density function

$$g(\tau_i) = \frac{1}{\pi a} \operatorname{sech}^2(2\tau_i/\pi a) \quad (9)$$

where sech is the hyperbolic secant function. The a parameter specifies the width of the transition and will be a determinant of the amount of transition noise. The response of an ideal transition on a single microtrack at the ideal center is $\tilde{h}(t)/M$. The response for a single microtrack displaced from the ideal center by τ_i is $\tilde{h}(t - \tau_i)/M$. Summing the M microtrack responses forms the overall channel response $h(t)$

$$h(t) = \frac{1}{M} \sum_{i=1}^M \tilde{h}(t - \tau_i). \quad (10)$$

The statistical average of $h(t)$ is equal to the convolution of $\tilde{h}(t)$ with $g(t)$. The microtrack channel model with an infinite number of microtracks is equivalent to the equalized Lorentzian channel model since in this case the sum approaches the statistical average.

III. TURBO DECODING

A. PCCC as Outer Encoder

The first system considered, shown in Fig. 1 [5], consists of two rate-1/2 RSC encoders connected by a pseudorandom interleaver π_1 of length N . The input data sequence \mathbf{u} is permuted by the interleaver before entering the second convolutional encoder. Each input bit u_k results in three output bits u_k, p_k^1 , and p_k^2 , resulting in a rate-1/3 code. A higher rate code is obtained by puncturing (omitting some parity bits) as described in Section II-C. The punctured sequence—data symbols followed by parity symbols—is then passed through a second interleaver, π_2 , resulting in the channel input sequence \mathbf{x} .

The first APP detector, denoted APP-channel, is matched to the channel and computes LAPPR's of the channel input \mathbf{x} based on the received noisy channel output \mathbf{y} . The DEMUX-depuncture block reverses the procedure of the MUX-puncture block, creating the sequences $\hat{\mathbf{u}}, \hat{\mathbf{p}}^1$, and $\hat{\mathbf{p}}^2$, which act as prior information for the subsequent decoders. The remaining blocks are the standard turbo decoder for parallel concatenated encoders [1], where APP-Dec1 is an APP detector matched to the first convolutional encoder, APP-Dec2 is an APP detector matched to the second convolutional encoder, and π_1 is an interleaver placed between the two. The output of APP-Dec2 is the log-APP ratio of the interleaved sequence \mathbf{u} . For subsequent iterations,

the soft information L_{ext}^u is added to the input of APP-Dec1 as shown. Also, the log-APP ratios from the two decoders are used to form the channel APP extrinsic information sequence L_{ext}^x as shown. After the final iteration, the output of APP-Dec2 is passed through a deinterleaver and a threshold-0 slicer to form the estimated information sequence $\hat{\mathbf{u}}$. As shown in [6], incorporating the channel APP detector into the iterative decoding cycle (termed full turbo decoding in [7]) offers approximately 0.5-dB performance improvement.

B. Serial Concatenation

Although the convolutional encoders for the full turbo system are concatenated in parallel, the overall system can be viewed as a serial structure. Blocks have been added to Fig. 1 to identify the outer and inner encoders and the corresponding outer and inner decoders. Here, the partial response channel is viewed as the inner encoder and the APP-channel is the corresponding inner decoder.

Removing the second convolutional encoder and interleaver in the outer code results in the far simpler serial system shown in Fig. 2. The corresponding decoder is shown in the lower portion of Fig. 2. Note that the decoder complexity has decreased significantly: the two M -state APP detectors matched to the convolutional encoders have been replaced by a single M -state APP detector. Also, since each APP detector operates on a block of N symbols before outputting a LAPP sequence of length N , the computation time has been substantially reduced.

IV. SIMULATION RESULTS

A. SNR Definition

For simulations using the white noise channel model, $E_c = R \cdot E_b$, where E_c is the code bit energy, E_b is the user bit energy, and the code rate R is the ratio of the number of user bits to code bits. SNR is defined as

$$\begin{aligned} \text{SNR} &= 10 \cdot \log \left(\frac{E_b}{N_o} \right) \\ &= 10 \cdot \log \left(\frac{E_b}{2 \cdot \sigma^2} \right) \\ &= 10 \cdot \log \left(\frac{E_c}{2 \cdot R \cdot \sigma^2} \right) \end{aligned} \quad (11)$$

where N_o is the one-sided power spectral density and σ^2 is the noise variance. Scaling the code bit energy in this manner permits the use of a constant user bit energy in all simulations.

For both the equalized Lorentzian channel model and the microtrack channel model, SNR is defined as

$$\text{SNR} = 10 \cdot \log \left(\frac{S_{\text{ms}}}{N_{\text{ms}}} \right) \quad (12)$$

where S_{ms} is the mean-square signal value measured at the input to the equalizer and N_{ms} is the mean-square noise value measured at the input to the equalizer. The amount of media noise is determined by the number of microtracks; fewer microtracks results in more media noise.

B. Parameters

Unless noted otherwise, the simulation results presented here use the RSC encoders described by generator polynomials, in octal form, $(g_1, g_2) = (31, 33)$, where g_1 is the feedback polynomial and g_2 is the feedforward polynomial. When two RSC encoders were concatenated in parallel, both were $(31, 33)$. The $(31, 33)$ encoder has four memory elements, necessitating a 16-state APP detector. Except where noted otherwise, an ordinary pseudorandom block interleaver was used in all simulations.

It has been well established that codes perform better with large block lengths [26]. The segmentation of disks into sectors may prevent large block lengths, making it most appropriate to operate on one 512-byte sector at a time. Therefore, unless otherwise noted, all simulations used a block size of $N = 4$ kbits. All turbo simulations used a maximum of ten decoding iterations; decoding terminated with fewer than ten iterations when no errors were detected in a block. Note that this method of termination could not be used in a real system since it assumes the decoder knows the correct sequence. However, comparisons were done with other termination methods—iterating ten times on every block and iterating until three consecutive output sequences were identical—with no significant performance difference. The puncturing scheme used is explained in Section II-C.

For the microtrack channel model, an a parameter of 30 nm was used with a bit cell length of 102 nm. The number of microtracks M was varied from 200 (virtually no media noise) to 2 (a large amount of media noise).

In the captions of Figs. 3–10, R indicates code rate, N indicates interleaver block size, PCCC refers to the PCCC outer code system, SCC refers to the single convolutional code outer code system, and h is the channel response polynomial. The partial response target is indicated by the numerator polynomial, while the precoder is indicated by the denominator polynomial. A “1” in the numerator of h indicates a full response target while a “1” in the denominator indicates no precoder was used.

C. Results

The effect of varying the block length in the full turbo decoder of Fig. 1 is plotted in Fig. 3. The straight line indicates an upper capacity bound (lower SNR bound) for the channel [27], the same bound as used in [5]. Results are plotted for PR4 at rate 8/9 (except, of course, for the uncoded plot) and with the precoder $1/(1 \oplus D^2)$. Note that a minor performance loss occurs when going from a block size of $N = 10$ k to $N = 4$ k. Gains on the order of 2 dB over the uncoded system are still observed when the block size is decreased to $N = 100$. Also shown is the performance of the full response system, “ $R = 8/9$, $N = 4$ k PCCC $h = 1/(1 + D^2)$,” where the inner encoder of Fig. 1 consists of just the precoder.

Fig. 4 shows the effects of removing the precoder for the EPR4 white noise channel model. The curves labeled “ $R = 8/9$, $N = 4$ k, PCCC, $h = (1 + D - D^2 - D^3)/(1 + D^2)$ ” and “ $R = 8/9$, $N = 4$ k, PCCC, $h = 1 + D - D^2 - D^3$ ” indicate the performance of the full turbo system with and without the precoder, respectively. A gain of approximately 0.3 dB on the cliff region of the curve is achieved by removing the precoder.

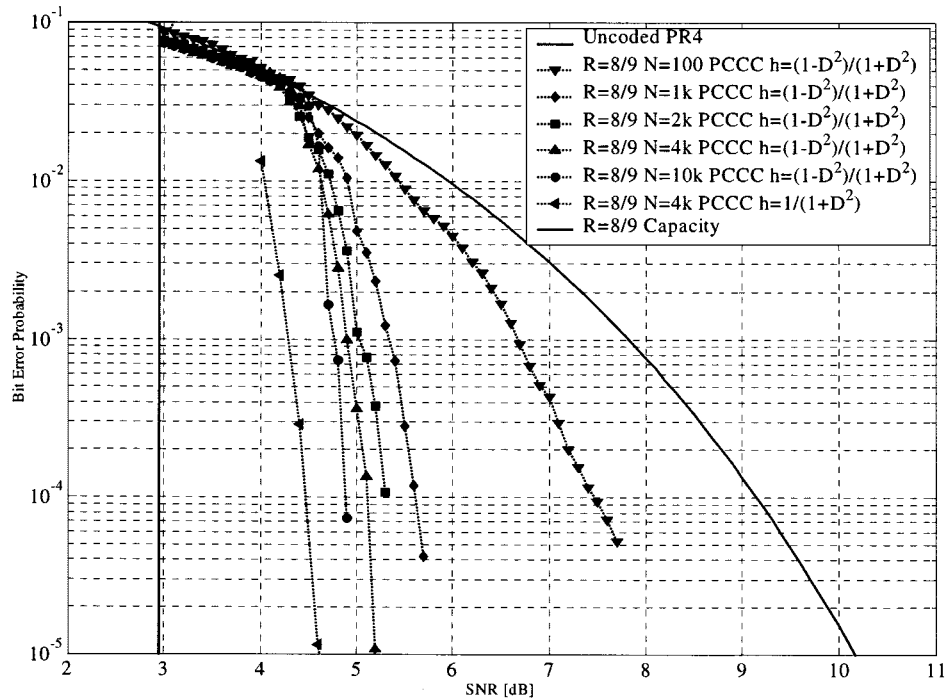


Fig. 3. Block length comparison for PR4 white noise channel model precoded with $1/(1 + D^2)$, PCCC outer code with full turbo decoding.

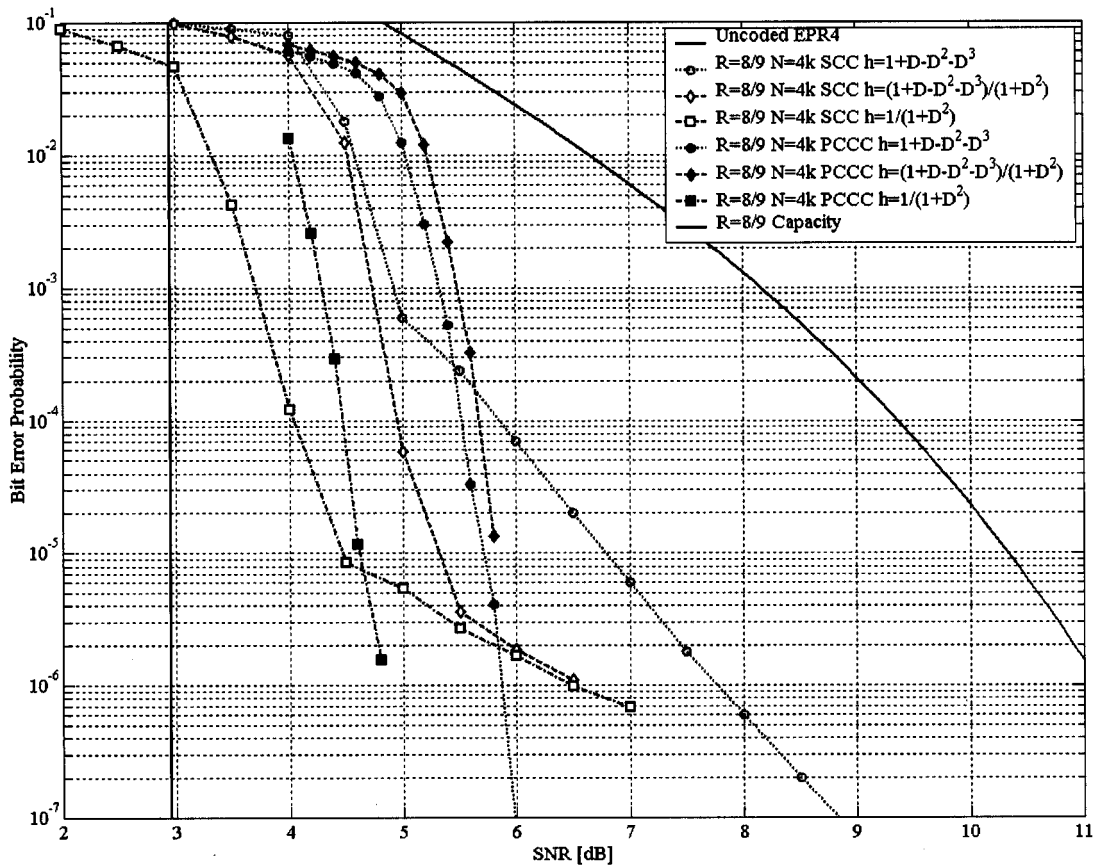


Fig. 4. EPR4 white noise channel model. PCCC—parallel concatenated convolution code as outer code. SCC—single convolutional code as outer code.

Though not shown, the nonprecoded curve reaches the error floor before the precoded curve, indicating precoding would probably be beneficial in an actual system.

Simulation results for the system of Fig. 2, where the outer encoder is a single convolutional encoder and the inner code is the EPR4 channel, are also plotted in Fig. 4 and labeled “ $R =$

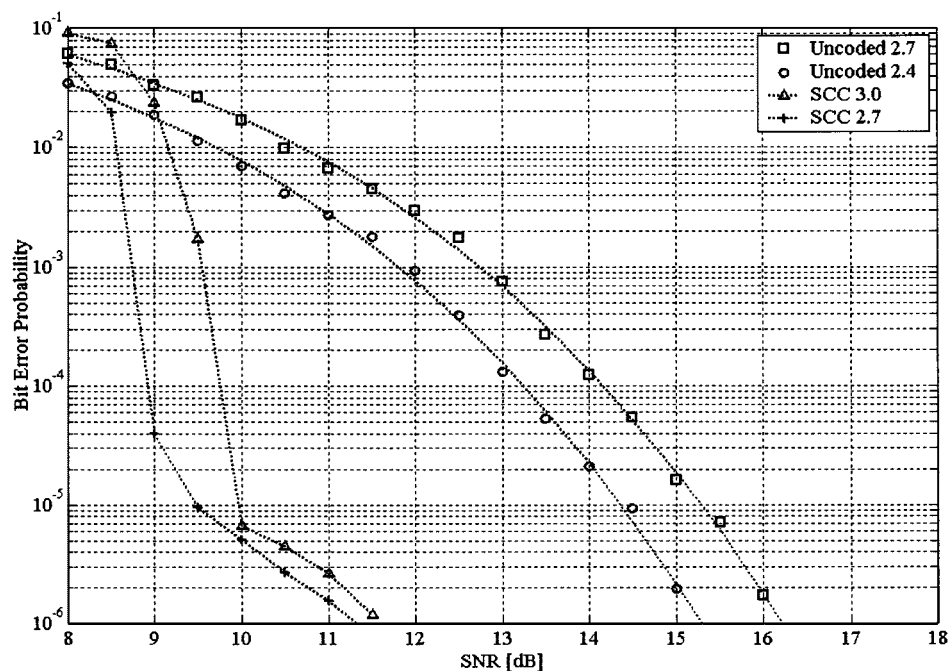


Fig. 5. EPR4 equalized Lorentzian channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$.

$8/9$, $N = 4k$, SCC, $h = (1 + D - D^2 - D^3)/(1 + D^2)$ and “ $R = 8/9$, $N = 4k$, SCC, $h = 1 + D - D^2 - D^3$.” In this system, the precoded system outperforms the nonprecoded system at higher SNR. Note that this simple precoded system offers approximately a 5-dB gain over uncoded EPR4 at a bit-error probability of 10^{-5} .

Also plotted are the corresponding full response systems. For the systems examined, the full response system provides a performance limit for the partial response systems. This was also observed in [10].

Similar results were obtained for the PR4 channel (not shown). For both the PCCC and SCC structures, the precoded PR4 and EPR4 systems performed worse than the nonprecoded systems at low SNR, but then performed better at higher SNR. This crossing of the curves occurred at much lower SNR for the SCC structure than for the PCCC structure. The fact that precoding is beneficial at higher SNR can be attributed to enhanced distance spectrum properties of the precoded systems and is discussed in [22].

The fact that the SCC system performs comparably to the significantly more complex PCCC system can be explained in part by observing Figs. 1 and 2. Both the PCCC encoder and the SCC encoder can be viewed as outer codes serially concatenated with an inner encoder, where, in both cases, the inner encoder is the precoded partial response channel. The outer decoder for the PCCC system contains two APP decoders iterating with each other. This will likely perform worse than using a single APP matched to the encoder. Therefore, while the outer encoder for the PCCC system might be better than the outer encoder for the SCC system, the outer decoder for the PCCC system is undoubtedly worse than the outer decoder for the SCC system.

In [28], Benedetto showed that the inner convolutional encoder should be recursive when serially concatenating convolutional encoders. In our scheme, the precoder is providing the recursive element to the inner encoder.

Though not shown, the SCC system precoded with $1/(1 \oplus D \oplus D^2 \oplus D^3)$ has a significantly lower error floor than the same system precoded with $1/(1 \oplus D^2)$. The cliff region, however, is shifted slightly to the right. In most applications, the benefit of a lower error floor would outweigh the cost of a slightly shifted cliff region, suggesting the $1/(1 \oplus D \oplus D^2 \oplus D^3)$ precoder should be considered.

The SCC system of Fig. 2 with the precoder $1/(1 \oplus D^2)$ was used for the simulations considered in Figs. 5–10. Simulation results for the SCC system on the equalized Lorentzian channel model are plotted in Fig. 5. Since a rate- $8/9$ code is used, the channel density must be increased for the coded systems. An uncoded channel density of 2.4 should be compared to a coded channel density of $2.4 * 9/8 = 2.7$ and an uncoded channel density of 2.7 should be compared to a coded channel density of $2.7 * 9/8 \approx 3.0$. At both densities considered, turbo decoding offers gains of approximately 5 dB at a BER of 10^{-5} .

Simulation results for the SCC system using the equalized Lorentzian channel model are plotted for a variety of outer RSC encoders in Fig. 6. Both the $(g_1, g_2) = (7, 5)$ encoder and the $(5, 7)$ encoder have two memory elements, necessitating a 4-state APP detector. The $(11, 17)$ encoder has three memory elements (8-state detector), the $(31, 33)$ encoder has four memory elements (16-state detector), and the $(75, 53)$ encoder has five memory elements (32-state detector). The system using the $(7, 5)$ encoder significantly outperforms the system using the $(5, 7)$ encoder. In fact, the 4-state $(7, 5)$ encoder performs

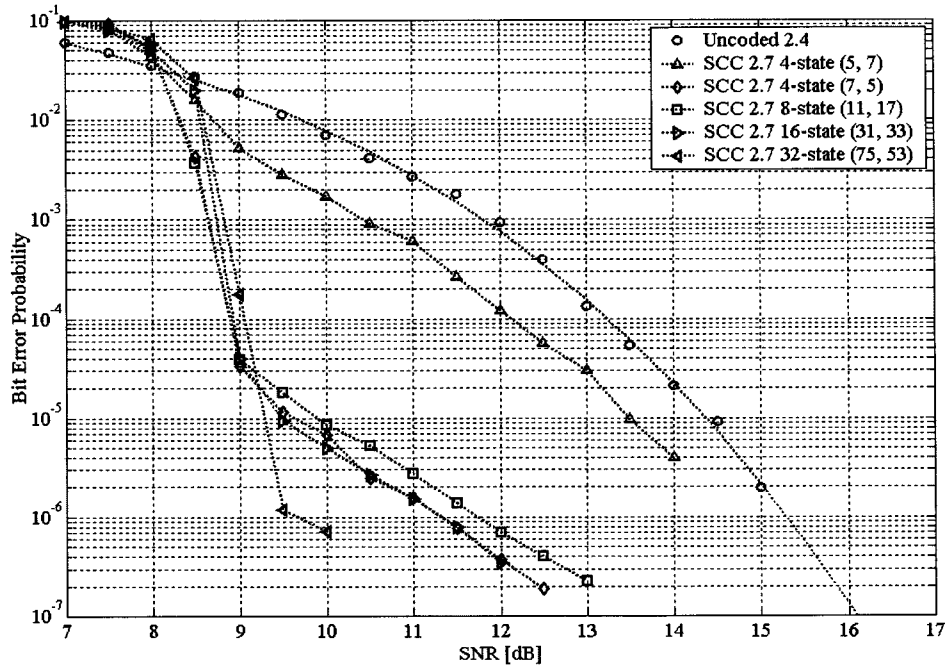


Fig. 6. Outer code comparison for EPR4 equalized Lorentzian channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$.

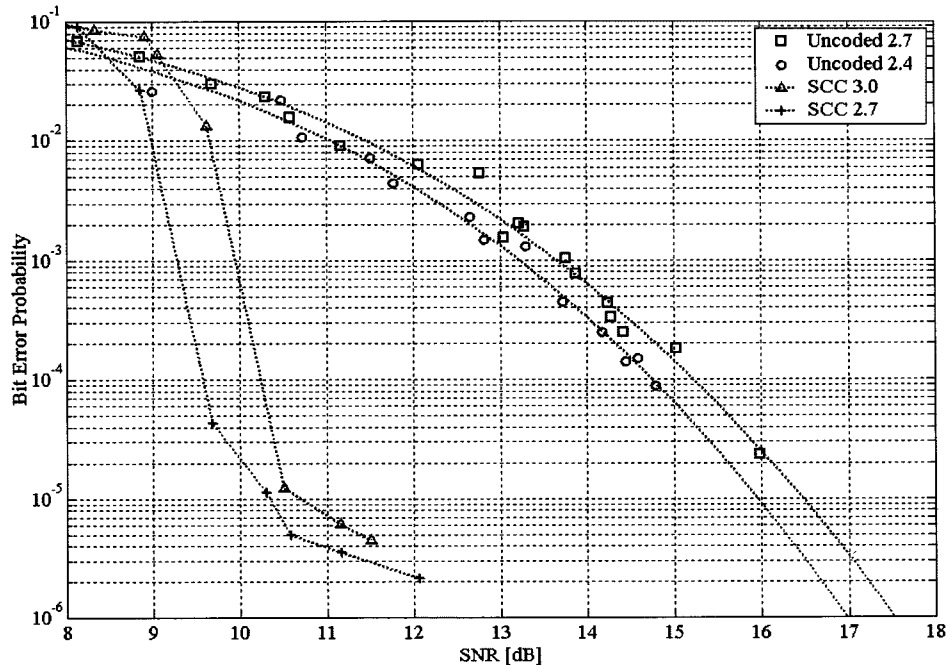


Fig. 7. EPR4 50–50 noise ratio microtrack channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$.

as well as the 8-state (11, 17) encoder and the 16-state (31, 33) encoder. A significant lowering of the error floor was achieved when using the 32-state (75, 53) encoder.

The relatively poor performance of the system using the (5, 7) encoder, punctured to rate $R = k/(k + 1) = 8/9$, can be explained as follows. Weight-2 input sequences result in weight-2 output sequences N times more frequently, where N is the interleaver size, with the (5, 7) code than with the (7, 5) code when puncturing to rate $8/9$ and transmitting every k th parity bit. This can be seen by observing the parity sequences and calculating the frequency that only zeros are selected as the parity bits.

Thus, the (5, 7) code will produce N times more low-weight codewords, resulting in a much weaker code. This argument holds only when every k th parity bit for a rate- $k/(k + 1)$ code is transmitted. If the parity bits are chosen in a different manner, performance may improve.

Simulation results for the SCC system on the microtrack channel model using a 50–50 ratio of AWGN to media noise are plotted in Fig. 7. Note that the uncoded performance curves for channel densities 2.4 and 2.7 are separated by about 0.5 dB, as opposed to about 1-dB separation for the 100% AWGN case. Here, turbo decoding offers gains of approximately 5.5 dB at a

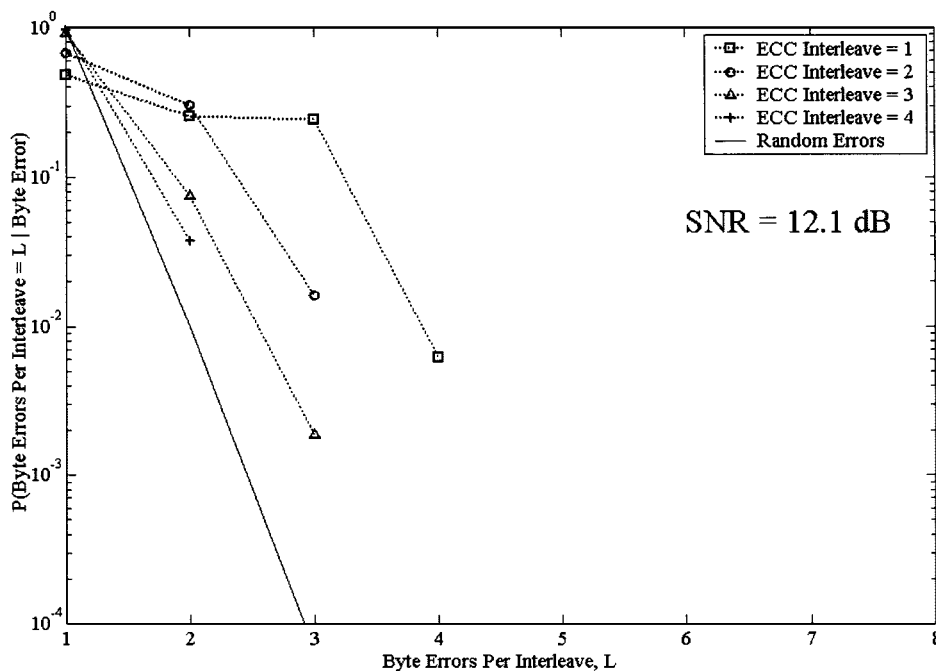


Fig. 8. Byte-error distribution for EPR4 50–50 noise ratio microtrack channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$, density = 2.7, SNR = 12.1 dB.

BER of 10^{-5} at both densities considered. For the SCC system with 100% media noise, there was essentially no difference for uncoded performance and gains of almost 6 dB were achieved. The results for the system with 100% media noise are not shown. The fact that the uncoded performance does not worsen as density increases for the 100% media noise system can be attributed to the fact that the media noise spectrum scales with density, similar to the signal spectrum. Therefore, media noise enhancement due to equalization does not worsen at higher densities.

As noted previously, the use of an RS byte-error correcting code is anticipated in an actual system. Thus, in Figs. 1 and 2, the user data would first be encoded by an RS encoder operating on 8-bit bytes and then passed through an error correction code (ECC) byte interleaver. This sequence \mathbf{u} would then be input to the outer PCCC for Fig. 1 or the outer SCC for Fig. 2. In the decoder, the slicer output sequence $\hat{\mathbf{u}}$ is passed through an ECC byte deinterleaver and then through an RS decoder. The ECC byte interleaver reads in the bytes in i rows and reads out the bytes in $\lceil N/(8i) \rceil$ columns, where N is the block size in bits and 8-bit bytes are assumed. Thus, each RS codeword consists of $n = \lceil N/(8i) \rceil$ bytes. Using $i = 1$ corresponds to no ECC byte interleaving, $i = 2$ corresponds to forming a sequence of every other byte, etc. The performance of the complete system is a function of both the average number of bit errors and of the byte-error distribution for various levels of ECC interleaving at the input to the RS decoder.

The probability distribution for the number of byte errors per ECC interleave in an erroneous block on the error floor for the SCC system with the microtrack channel model using a 50–50 ratio of AWGN to media noise is shown in Fig. 8. One thousand bit errors were observed for density 2.7 at SNR = 12.1 dB. Byte errors are counted for each block containing one or more bit

errors. The conditional probability of byte error is shown for no ECC interleaving (ECCinterleave = 1), two ECC interleaves, three ECC interleaves, and four ECC interleaves.

The objective is to determine the error correction coding needed to guarantee extremely low bit-error probabilities, on the order of 10^{-14} . This is not practical with computer simulations. As a reference, we can compare our results to the random error case in which the byte errors have an ideal binomial distribution

$$P(k \text{ byte errors in } n \text{ bytes}) = \binom{n}{k} p^k (1-p)^{n-k} \quad (13)$$

where p is the byte-error probability and n is the number of symbols in the ECC frame. For the case of random errors and $k > 0$, the conditional probability of k errors in n bytes given that $k > 0$ is

$$P(k \text{ byte errors in } n \text{ bytes} | k > 0) = \frac{P(k \text{ errors in } n \text{ bytes})}{P(k > 0)} \quad (14)$$

where $P(k > 0) = 1 - (1-p)^n$. The ideal binomial distribution of (14), with three ECC byte interleaves, is plotted as a solid line in Fig. 8. The ideal distribution plots for one, two, and four ECC byte interleaves are shifted slightly and have been omitted for clarity. Since the conditional probability of byte error of the system with a 50–50 noise ratio is greater than the binomial distribution, the errors coming out of the turbo decoder are not statistically independent. Similar statistics were observed, though not plotted, for the equalized Lorentzian channel model and the all media noise microtrack channel model.

The effects of varying the parameters of the interleavers of Section II-D (the pseudorandom block interleavers, not the

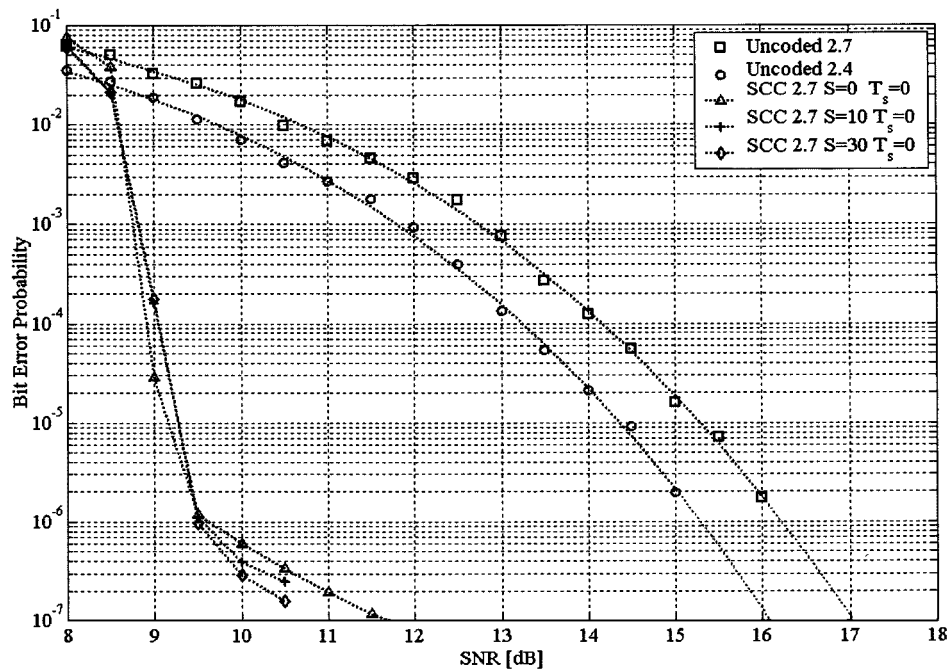


Fig. 9. Various S -interleavers for EPR4 equalized Lorentzian channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$, coded density is 2.7.

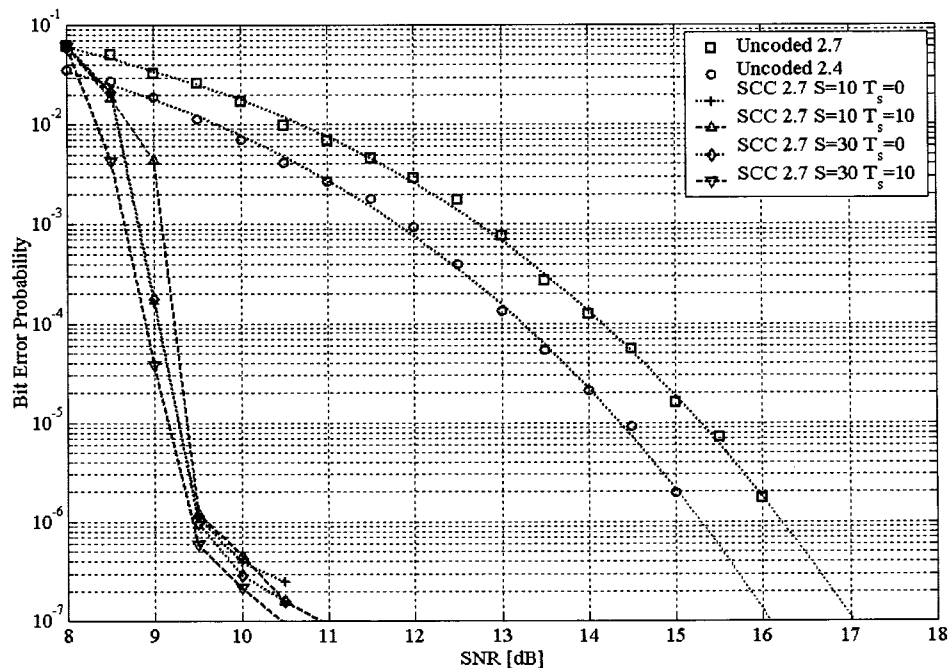


Fig. 10. Various ST_s -interleavers for EPR4 equalized Lorentzian channel model, precoded with $1/(1 + D^2)$, SCC outer code, $R = 8/9$, coded density is 2.7.

ECC byte interleavers) are shown in Fig. 9, where the coded system is precoded EPR4 on the equalized Lorentzian channel. The system performance using the ST_s -random interleaver with $T_s = 0$ and $S = 0, 10$, and 30 is shown. As the S -constraint is increased, the error floor is lowered slightly. The effect of adding a T_s -constraint ($L = 15$) is shown in Fig. 10. Comparing the curves denoted “SCC 2.7 $S = 10, T_s = 0$ ” with “SCC 2.7 $S = 10, T_s = 10$ ” shows that the addition of the $T_s = 10$ constraint results in a slight lowering of the error floor. A similar performance improvement is shown for the

system when a $T_s = 10$ constraint is added to the $S = 30$ constraint.

V. CONCLUSIONS

Two concatenated encoding architectures for partial response channels have been compared. In the first architecture, the precoded partial response channel is viewed as the inner encoder while a PCCC is the outer code. In the second architecture, the outer code is replaced with an SCC. Iterative decoding for the

systems using an outer SCC has been shown to be far simpler than iterative decoding for systems using an outer PCCC. The simpler system has been shown to offer approximately a 5-dB gain over an uncoded system at a bit-error probability of 10^{-5} for both the PR4 and EPR4 white noise channel models. This 5-dB gain was maintained when colored noise was introduced by using the equalized Lorentzian channel model. Media noise was introduced with the microtrack channel model and a gain of approximately 5.5 dB was observed when a 50–50 ratio of AWGN to media noise was used. This gain increased to approximately 6 dB for the 100% media noise case. These results suggest that, unlike conventional coding schemes, turbo decoders are quite insensitive to the type of noise in the system. Also, the byte-error distributions were measured on the error floor for the various systems considered. The byte errors were found to be not statistically independent, suggesting a strong error correcting code (perhaps with ECC interleaving) may be required in a real system. Use of the precoder $1/(1 \oplus D \oplus D^2 \oplus D^3)$ instead of $1/(1 \oplus D^2)$ was shown to significantly lower the error floor for the EPR4 SCC system. Finally, modifications of the pseudorandom interleaver were shown to lower the error floor, offering a simple way to improve average bit-error performance.

ACKNOWLEDGMENT

The authors are indebted to A. Friedmann and A. Dati for their many helpful suggestions, comments, and discussions. Also, the authors are grateful to the Editor and the reviewers for providing insightful comments and suggestions, resulting in a much improved paper.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] C. Heegard, "Turbo coding for magnetic recording," in *Proc. IEEE Information Theory Workshop*, San Diego, CA, Feb. 1998, pp. 18–19.
- [3] L. McPheters, S. McLaughlin, and E. Hirsch, "Turbo codes for PR4 and EPR4 magnetic recording," in *Proc. 32nd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1998, pp. 1778–1782.
- [4] M. Reed and C. Schlegel, "An iterative receiver for the partial response channel," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 63.
- [5] W. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *Proc. IEEE Int. Conf. Communications*, Atlanta, GA, June 1998, pp. 947–951.
- [6] W. Ryan, L. McPheters, and S. McLaughlin, "Combined turbo coding and turbo equalization for PR4-equalized Lorentzian channels," in *Proc. Conf. Information Sciences and Systems (CISS'98)*, Princeton, NJ, Mar. 1998, pp. 489–493.
- [7] T. Souvignier, A. Friedmann, M. Öberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo decoding for PR4: Parallel versus serial concatenation," in *Proc. IEEE Int. Conf. Communications*, Vancouver, BC, Canada, June 1999, pp. 1638–1642.
- [8] T. Souvignier, A. Dati, and J. K. Wolf, "Turbo decoding for partial response channels with colored noise," *IEEE Trans. Magn.*, vol. 35, pp. 2322–2324, Sept. 1999.
- [9] T. Souvignier, J. Caroselli, A. Dati, and J. K. Wolf, "Turbo decoding for partial response channels with media noise," in *Proc. IEEE Global Telecommunications Conf. (GLOBECOM'99)*, Rio de Janeiro, Brazil, Dec. 1999, pp. 739–743.

- [10] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sep./Oct. 1995.
- [11] M. Reed and C. Schlegel, "Iterative multiuser detection for CDMA with FEC: Near single user performance," *IEEE Trans. Commun.*, vol. 46, pp. 1693–1699, Dec. 1998.
- [12] M. Moher, "Turbo-based multiuser detection," in *Proc. IEEE Int. Symp. Information Theory*, Ulm, Germany, June 1997, p. 195.
- [13] —, "An iterative multiuser decoder for asynchronous BPSK users," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 423.
- [14] —, "An iterative multiuser decoder for near-capacity communications," *IEEE Trans. Commun.*, vol. 46, pp. 870–880, July 1998.
- [15] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. IEEE Int. Conf. Communications*, Seattle, WA, June 1995, pp. 54–59.
- [16] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [17] E. Lee and D. Messerschmitt, *Digital Communication*, 2nd ed. Norwell, MA: Kluwer, 1994.
- [18] C. Helstrom, *Statistical Theory of Signal Detection*. New York: Pergamon, 1960.
- [19] M. Öberg, A. Vityaev, and P. H. Siegel, "The effect of puncturing in turbo encoders," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 1997, pp. 184–187.
- [20] M. Öberg and P. H. Siegel, "Application of distance spectrum analysis to turbo code performance improvement," in *Proc. 35th Allerton Conf. Communication, Control and Computing*, Monticello, IL, Sept. 1997, pp. 701–710.
- [21] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, pp. 921–934, Sept. 1975.
- [22] M. Öberg and P. H. Siegel, "Performance analysis of turbo-equalized dicode partial-response channel," in *Proc. 36th Allerton Conf. Communication, Control and Computing*, Monticello, IL, Sept. 1998, pp. 230–239.
- [23] J. Caroselli and J. K. Wolf, "A new model for media noise in thin-film magnetic recording media," in *Proc. 1995 SPIE Int. Symp. Voice, Video, and Data Communications*, vol. 2605, Philadelphia, PA, Oct. 1995, pp. 29–38.
- [24] —, "Applications of a new simulation model for media noise limited magnetic recording," *IEEE Trans. Magn.*, vol. 32, pp. 3917–3919, Sept. 1996.
- [25] T. C. Arnoldussen and L. L. Nunnally, *Noise in Digital Magnetic Recording*. Singapore: World Scientific, 1992.
- [26] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of blocksize," Jet Propulsion Lab., Pasadena, CA, Telecommunications and Mission Operations Progress Rep. 42-133, Jan.–Mar. 1998, May 15, 1998.
- [27] S. Shamai, L. Ozarow, and A. Wyner, "Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1527–1539, Nov. 1991.
- [28] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 909–926, May 1998.



Tom V. Souvignier (S'98–M'99) was born in Mt. Clemens, MI, and grew up primarily in Texas and California. He received the B.S.E.E. degree from the University of California at Davis in 1989 and the M.S.E.E. degree from the University of California at San Diego, La Jolla, in 1996. In 1999, he received the Ph.D. degree in electrical engineering, in the field of communication theory and systems, from the University of California at San Diego.

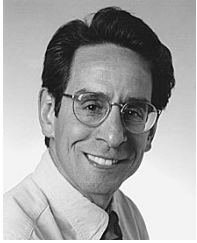
Since graduating, he has been with Texas Instruments, Incorporated, Longmont, CO. His research interests include signal processing and iterative decoding for communications and magnetic recording channels.



Mats Öberg (S'00) was born in Pitea, Sweden, in 1969. He received the M.S. degree in electrical engineering from Lund University, Lund, Sweden, in 1995. He received the Ph.D. degree in electrical engineering, communication theory and systems from the University of California at San Diego, La Jolla, in 2000.

He is a Staff Design Engineer at Marvell Semiconductor, Inc., Sunnyvale, CA. His research interests include loading and signal processing for wireless communications and magnetic storage applications.

Dr. Öberg is a member of the IEEE Communications Society and the IEEE Information Theory Society.



Paul H. Siegel (M'82–SM'90–F'97) received the S.B. degree in mathematics in 1975 and the Ph.D. degree in mathematics in 1979, both from the Massachusetts Institute of Technology, Cambridge. He held a Chaim Weizmann Postdoctoral Fellowship at the Courant Institute, New York University.

He was with IBM Research in San Jose, CA, from 1980 to 1995, where he conducted and managed research on modulation, coding, and signal processing, primarily for data storage applications. He was a Visiting Associate Professor at the University of California (UCSD), San Diego, La Jolla, while at the Center for Magnetic Recording Research during the 1989–1990 academic year. He is currently Professor of Electrical and Computer Engineering at UCSD, and is affiliated with the Center for Magnetic Recording Research as well as the Center for Wireless Communications. His primary research interests lie in the areas of information theory and communications, particularly coding and modulation techniques, with applications to digital data storage and transmission. He holds 17 patents in the area of coding and detection, and was named a Master Inventor at IBM Research in 1994.

Dr. Siegel was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996. He served as Co-Guest Editor of the May 1991 Special Issue on Coding for Storage Devices of the IEEE TRANSACTIONS ON INFORMATION THEORY, and served as Associate Editor for Coding Techniques from 1992 to 1995. He was co-recipient, with R. Karabed, of the 1992 IEEE Information Theory Society Paper Award and shared the 1993 IEEE Communications Society Leonard G. Abraham Prize Paper Award with B. Marcus and J. K. Wolf. He is a member of Phi Beta Kappa.

Dr. Siegel was a member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996. He served as Co-Guest Editor of the May 1991 Special Issue on Coding for Storage Devices of the IEEE TRANSACTIONS ON INFORMATION THEORY, and served as Associate Editor for Coding Techniques from 1992 to 1995. He was co-recipient, with R. Karabed, of the 1992 IEEE Information Theory Society Paper Award and shared the 1993 IEEE Communications Society Leonard G. Abraham Prize Paper Award with B. Marcus and J. K. Wolf. He is a member of Phi Beta Kappa.



Robert E. Swanson received the B.S. degree from Valparaiso University, Valparaiso, IN, in 1967, the M.S. degree from Stanford University, Stanford, CA, in 1968, and the Ph.D. degree from the University of Arizona, Tucson, in 1978, all in electrical engineering.

He is a Senior Director in the Advanced Concepts Group at Seagate Technology, Fremont, CA, responsible for advanced channel development. He was a Motorola Merit Scholar during graduate studies at the University of Arizona. He has been on the faculty of the Department of Electrical Engineering at both Valparaiso University and The University of Arizona. He has worked extensively as a Contract Engineer in computer design and developed several error correction and modulation coding schemes. He holds seven patents in the field and has published numerous articles and technical reports on computer architecture, systems, and signal processing.



Jack K. Wolf (S'54–M'60–F'73–LF'97) was born on March 14, 1935, in Newark, NJ. He received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1956, and the M.S.E., M.A., and Ph.D. degrees from Princeton University, Princeton, NJ, in 1957, 1958, and 1960, respectively. He has been a University Professor for over 40 years and is presently the Steven O. Rice Professor in the Department of Electrical and Computer Engineering at the University of California, San Diego, La Jolla. He is also a part-time employee at Qualcomm, Inc.

His research interests include coding and signal processing for communications and storage systems.

Dr. Wolf was a Guggenheim Fellow and is a member of the National Academy of Engineering.