Turbo Decoding for PR4: Parallel Versus Serial Concatenation

Tom Souvignier[*], Arnon Friedmann[β], Mats Öberg[*], Paul H. Siegel[*], Robert E. Swanson[γ], and Jack K. Wolf[*]
[*]University of California – San Diego
[β]Quantum Corporation
[γ]Seagate Technology

Abstract **-- Recent work on the application of turbo decoding techniques to partial response class 4 (PR4) channels has focused on parallel concatenation systems that require three APP detectors. A simplified serial concatenation system will be presented that uses as its outer code a single convolutional code and as its inner code the partial response channel. An extension of this serial concatenation system will also be presented that combines a second code with the channel, forming a more powerful inner code. Both proposed systems require only two APP detectors, offering significant savings in complexity and computation time. These serial concatenation systems will be shown to perform as well as the more complicated parallel concatenation systems, offering substantial gains over uncoded systems. Additionally, the effect of precoding will be investigated. Simulation results comparing the parallel and serial concatenation systems will be presented.**

I. INTRODUCTION

Turbo codes were introduced by Berrou, et al [1] in 1993 as two or more parallel concatenated convolutional codes connected with an interleaver and decoded using an iterative technique. These codes are capable of operating near Shannon capacity on additive white Gaussian noise (AWGN) channels. The term turbo decoding has subsequently come to refer to this iterative decoding process. Therefore, in this paper, turbo decoding will refer to the iterative decoding process for both parallel and serial concatenation systems. Considerable work has been done recently by Ryan [2], Ryan, et al [3], Heegard [4], and others investigating the application of turbo decoding to partial response channels. Ryan's technique involves using an additional APP detector matched to the partial response channel, followed by the standard turbo system of two APP detectors matched to the constituent convolutional encoders. Simulation results suggest that high rate turbo codes offer substantial gain over uncoded systems. In Ryan's work, the precoder $\frac{1}{1 \oplus D^2}$, where $\oplus$ indicates modulo-2 addition, was added to the $1 - D^2$ (PR4) channel.

This paper will (1) investigate the effects of removing the precoder, (2) propose a simple serial system where the inner code is just the PR4 channel, and (3) propose a slightly more complex serial system in which the inner code is the concatenation of a convolutional code and the channel. 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 system proposed by Ryan and shows its relationship to the proposed serially concatenated systems. Simulation results are provided in Section IV, and concluding remarks in Section V.

II. SYSTEM COMPONENTS

The various components of the systems shown in the block diagrams will be described in this section.

A. APP Detector

While a Viterbi detector chooses the most probable data sequence given the received sequence, an *a posteriori probability* (APP) detector calculates the *a posteriori* probability of each transmitted bit given the received sequence [5], [6]. A general APP detector module is shown in Fig. 1. The *i* and *o* indicate the corresponding **encoder** input and **encoder** output, respectively. The inputs to the APP detector, L_i and L_o, are *a priori* probabilities for the encoder input and output symbols. The APP detector computes the log-likelihood ratios (LLRs) $\Lambda(i_k)$ and $\Lambda(o_k)$ as

$$\Lambda(i_k) = \log \frac{\Pr(i_k = 1 \mid L_i, L_o)}{\Pr(i_k = 0 \mid L_i, L_o)} \qquad (1)$$

and

$$\Lambda(o_k) = \log \frac{\Pr(o_k = 1 \mid L_i, L_o)}{\Pr(o_k = 0 \mid L_i, L_o)} \quad , \qquad (2)$$

where $Pr(x_k=u|L_i,L_o)$ denotes the probability that symbol $x_k=u$ conditioned on the APP input sequences L_i and L_o.

Note that not all four ports to the general APP detector need be

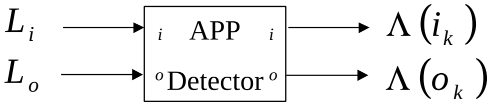

Fig. 1 General APP detector.

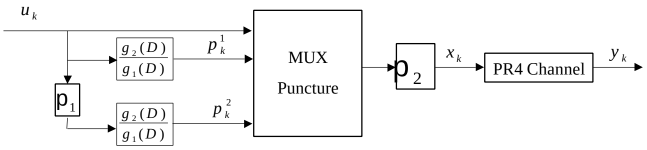

Fig. 2 Turbo encoder.

This work was supported in part by the National Science Foundation under Grant NCR-9405008, by the Center for Magnetic Recording Research at the University of California – San Diego, and by a Seagate Technology fellowship.

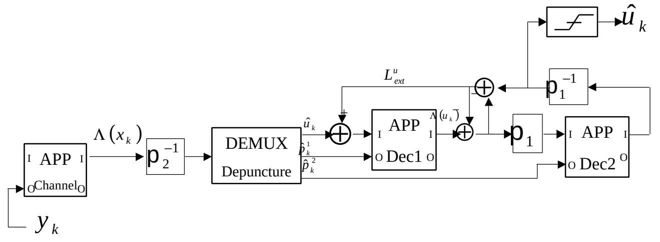

Fig. 3 Partial turbo decoder.

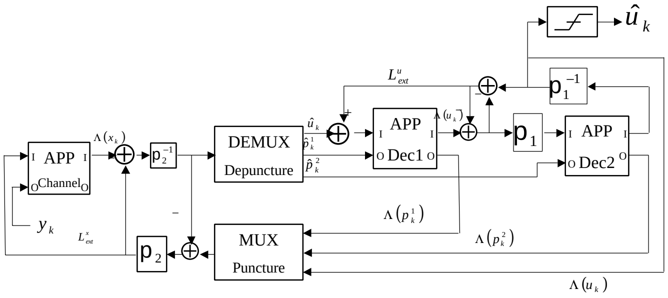

Fig. 4 Full turbo decoder.

used – some applications may use only one input port and one output port.

B. Interleaver

The interleaver, π, takes a block of N symbols and pseudo-randomly permutes them. The de-interleaver, π^{-1}, simply reverses the process.

C. Channel Model

A linear channel with additive white Gaussian noise is assumed. For this paper, the desired target is PR4 ($1-D^2$). For simplicity, we will assume perfect equalization and uncorrelated noise at the channel output, resulting in an equivalent discrete-time model $y_k = x_k - x_{k-2} + n_k$. When a precoder is added to the system, the overall equivalent discrete-time model is $\dfrac{x_k - x_{k-2}}{x_k \oplus x_{k-2}} + n_k$, where $\oplus$ indicates modulo-2 addition.

D. MUX-Puncture

The MUX function of the MUX-Puncture block converts two or more parallel sequences to a single serial sequence. The puncturing is accomplished by omitting as many parity bits as is necessary to achieve the desired rate. The DEMUX-Depuncture block simply reverses the process of the MUX-Puncture block, converting a serial sequence to parallel sequences, and placing 0's in locations of the punctured symbols.

E. RSC Encoder

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

III. TURBO DECODING FOR PR4

A. Parallel Concatenation – Partial Turbo

The first system considered, shown in Fig. 2 [2], consists of two rate 1/2 RSC encoders connected by a pseudo-random interleaver, π_1, of length N. The input data sequence, **u,** is permuted by the interleaver before entering the second convolutional encoder. The data sequence and the two parity sequences result in a rate 1/3 code which is punctured to the desired rate by omitting parity bits. This sequence is then passed through a second interleaver, π_2, resulting in the channel input sequence **x**.

A decoder for the system of Fig. 2 is shown in Fig. 3. The first APP detector, APP-Channel, is matched to the channel and computes LLRs of the channel input **x** based on the received noisy channel output **y**. The DEMUX-Depuncture block reverses the procedure of the MUX-Puncture block, creating the log-likelihood sequences $\Lambda(u_k)$, $\Lambda(p_k^1)$, and $\Lambda(p_k^2)$. The remaining blocks are the standard turbo decoder, 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-likelihood ratio of the interleaved sequence **u**. Therefore, passing the output through a de-interleaver and a threshold-0 slicer results in the estimated information sequence $\hat{\mathbf{u}}$. For subsequent iterations, the soft information L_{ext}^u is added to the input of APP-Dec1 as

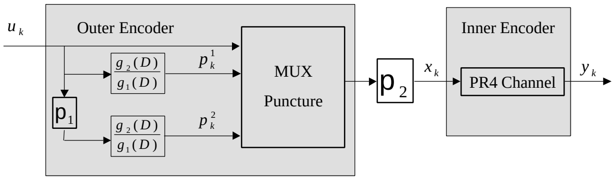

Fig. 5 Turbo encoder viewed as serial encoder.

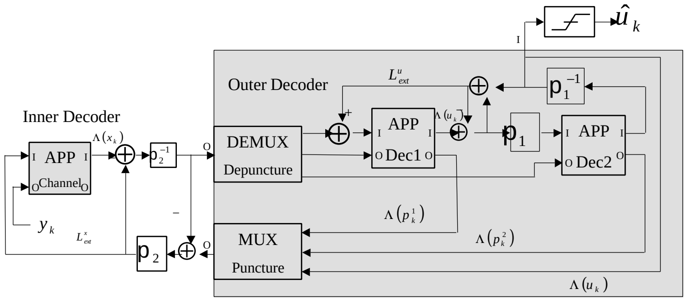

Fig. 6 Full turbo decoder viewed as serial decoder.

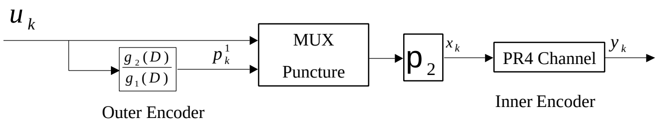

Fig. 7 Serial encoder with channel as inner code.

shown.

B. Parallel Concatenation – Full Turbo

The outputs from the channel detector, APP-Channel, can be improved by using *a priori* information from APP-Dec1 and APP-Dec2 [3]. The information fed back consists of the LLR for the first parity sequence, $\Lambda\left(p_k^1\right)$, the LLR for the second parity sequence $\Lambda\left(p_k^2\right)$, and the LLR for the systematic sequence, $\Lambda(u_k)$. The interleaver, p_2, and the corresponding de-interleaver, p_2^{-1}, are placed at the input and output to APP-Channel. The full turbo decoder is shown in Fig. 4.

C. Serial Concatenation

Although the convolutional codes for the full turbo system are concatenated in parallel, the overall system can be viewed as a serial scheme. This is evident in Figs. 5 and 6, where shaded blocks have been added to Figs. 2 and 4 to identify the outer and inner encoders and the corresponding outer and inner decoders. Here the PR4 channel is viewed as the inner encoder and APP-Channel is the corresponding inner decoder.

A far simpler serial system can be created by removing the second convolutional encoder, resulting in the system shown in Fig. 7. The corresponding decoder is shown in Fig. 8. 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 LLR sequence of length N, the computation time has been substantially reduced.

To improve the performance of the system of Fig. 7, a second encoder can be concatenated with the channel. The corresponding decoder remains as shown in Fig. 8, except that the inner APP detector is matched to the concatenation of the second encoder and the channel. When an encoder is concatenated with the channel, the system code rate will be decreased by the rate of the inner encoder. It is possible to choose this inner encoder to force a (d,k) constraint on the output sequence.

IV. SIMULATION RESULTS

A. SNR Definition

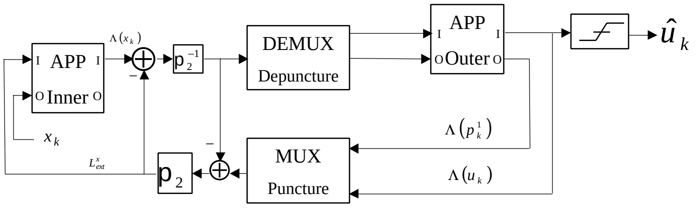

Fig. 8 Serial decoder.

In all simulations, $E_c = R \cdot E_b$,where E_c is the code bit energy, E_b is the user bit energy, and R is the code rate

$$R = \frac{number\ of\ user\ bits}{number\ of\ code\ bits} \tag{3}$$

SNR is defined as

$$\begin{aligned} SNR &= 10 \cdot \log\left(\frac{E_b}{N_o}\right) = 10 \cdot \log\left(\frac{E_b}{2 \cdot s^2}\right) \\ &= 10 \cdot \log\left(\frac{E_c}{2 \cdot R \cdot s^2}\right) \end{aligned}, \tag{4}$$

where N_o is the one-sided power spectral density and s^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. It is important to note that in a real magnetic recording system the performance is further degraded by misequalization, media noise, and other factors not predicted by the white noise model being considered.

B. Block Length, N

It has been well established that block codes perform better with large block lengths [7]. However, use in a magnetic recording channel may prevent large block lengths. For example, it may be most appropriate to operate on one sector at a time. Thus a 512 byte sector would limit N to 512 * 8 = 4096 bits. Therefore, unless otherwise noted, all simulations were done with N = 4000.

C. Results

The results presented were simulated using 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). When a convolutional code was combined with the channel to form the inner code, the inner RSC encoder used was $(g_1, g_2) = (7,5)$. The (31,33) encoder has four memory elements, necessitating a 16 state APP detector. All turbo simulations used 10 decoding iterations. Several puncturing schemes were investigated and the performance effects observed were minimal. No data bits were punctured, and parity bits were punctured in a systematic fashion. For example, for the rate 8/9 parallel structure, 8 data bits were transmitted, then the 8th bit from the first parity sequence, then 8 data bits, then the 16th bit from the second parity sequence, etc.

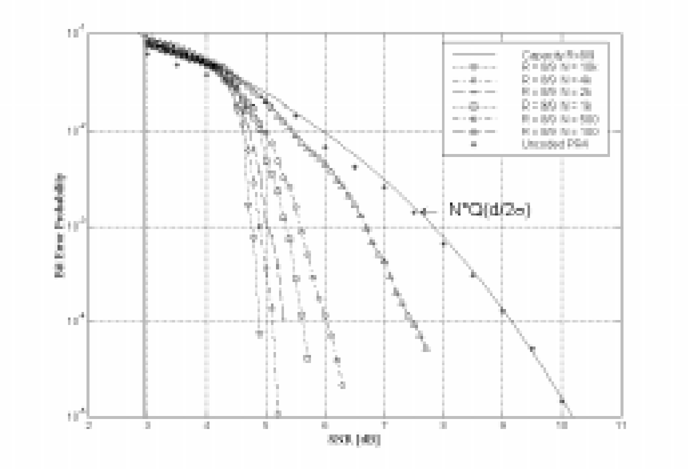

Fig. 9 Block length comparison for parallel concatenation – full turbo.

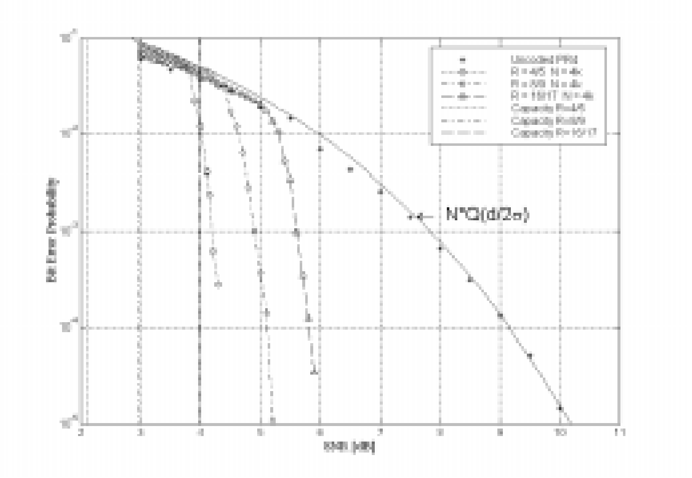

Fig. 10 Rate comparison for parallel concatenation – full turbo.

The effect of varying the block length on the parallel concatenated encoder of Fig. 2 with the full turbo decoder of Fig. 4 is plotted in Fig. 9. The straight lines indicate a lower capacity bound for an ISI free channel [8]. Similar bounds for a partial response channel would be higher (shift to the right) so the system is actually performing closer to the limit than indicated. Results are plotted for rate 8/9 (except, of course, for the uncoded plot) and with the precoder $\frac{1}{1 \oplus D^2}$. Note that a minor performance loss occurs when going from a block size of N = 10,000 to N = 4,000. Small gains over the uncoded system are still observed when the block size is decreased to N = 100.

The same system punctured to rates of 4/5, 8/9, and 16/17, with block size N = 4,000 are plotted in Fig. 10. The effects of varying the block size for rates 4/5 and 16/17 were observed to be similar to the effect of varying the block size for rate 8/9. These results are not shown.

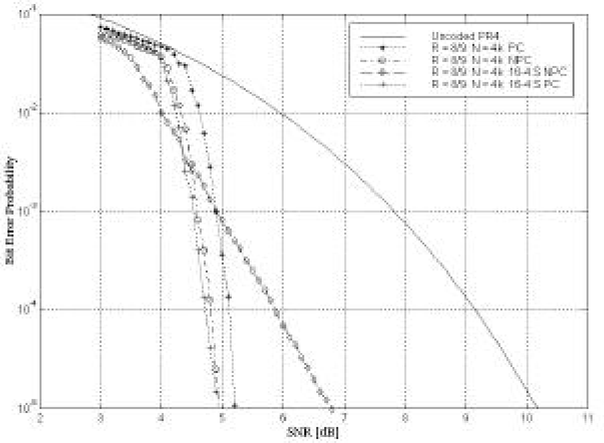

Fig. 11 Parallel and serial, precoded versus non-precoded.

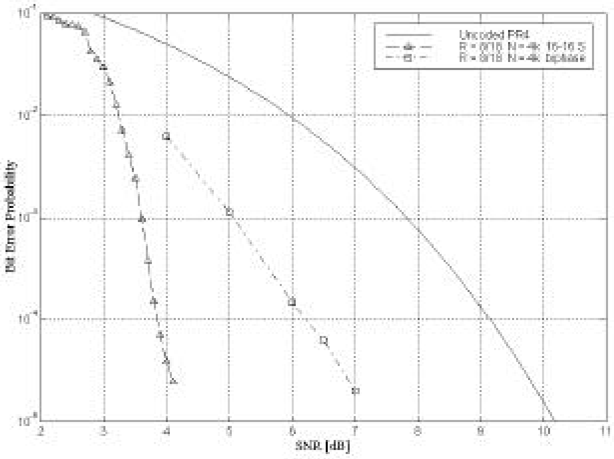

Fig. 12 Concatenated inner code and bi-phase code.

Fig. 11 shows the effects of removing the precoder on the parallel concatenation with full turbo decoding system. The curves labeled "R=8/9, N=4k, PC" and "R=8/9, N=4k, NPC" indicate the performance of the full turbo system with and without the precoder. Several tenths of a dB gain is achieved by removing the precoder. Though not plotted, similar gains were observed at rate 4/5 and rate 16/17. The gain shown is for the cliff region of the curve. Though not shown, the non-precoded curve floors out before the precoded curve, so the precoded system outperforms the non-precoded system in the floor region. Since an actual system would operate in this floor region, precoding is beneficial.

Simulation results for the serial system of Fig. 7, where the inner code is just the PR4 channel, are also plotted in Fig. 11 and labeled "R=8/9, N=4k, 16-4 S, PC" and "R=8/9, N=4k, 16-4 S, NPC". In this serial system, the precoded system outperforms the non-precoded system at higher SNR. Note that this simple system offers over 5 dB gain over uncoded PR4 at a bit error probability of 10^{-5}.

For both the parallel and serial structures, precoded performed worse than non-precoded at low SNR, but then performed better at higher SNR. This crossing of the curves occurred at much lower SNR for the serial structure than for the parallel 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 [9].

Simulation results for the slightly more complex serial system, where the inner code is a combination of the PR4 channel and a convolutional encoder, are plotted in Fig. 12 and labeled "R=8/18, N=4k, 16-16 S". Here the same 16-state RSC encoder was used as the outer code, while a 4-state RSC encoder concatenated with the 4-state channel formed the 16-state inner code. This preliminary result shows the feasibility of concatenating a convolutional code with the channel and using an APP detector matched to this concatenation. The simple (4-state) low rate (R = 1/2) inner code resulted in an overall rate of R = 8/18. For comparison, a rate 8/18 bi-phase coded system was also simulated. The bi-phase system employed as its inner code the simple rate 1/2 bi-phase code concatenated with the PR4 channel, where the bi-phase code maps '0' to '01' and '1' to '10'.

V. CONCLUSION

A serial concatenation system with iterative decoding has been introduced that is far simpler than parallel concatenation systems currently being considered for use with partial response channels. This simpler system has been shown to offer approximately 5 dB gain over uncoded systems at a bit error probability of 10^{-5} for the PR4 channel. This remarkable performance suggests serial concatenation with iterative decoding may provide a viable alternative to current coded partial response systems. Also, simulation results were presented that indicate precoding is beneficial at higher SNR, but detrimental at lower SNR. Finally, simulation results were presented for a more complicated serial system, where the inner code was a combination of a second code and the channel. Results were presented for systems using as this second code both a convolutional code and a bi-phase code. The system using the convolutional code was shown to significantly outperform the bi-phase code of the same rate.

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

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

[4] C. Heegard, "Turbo coding for magnetic recording," *Proc., Information Theory Workshop* (San Diego, CA, February 1998), pp. 18-19.

[5] 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.

[6] E. Lee and D. Messerschmitt, *Digital Communication, Second Edition*. Kluwer Academic Publishers, 1994.

[7] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of blocksize", *The Telecommunications and Mission Operations Progress Report 42-133, January-March 1998*, Jet Propulsion Laboratory, Pasadena, California, pp 1-23, May 15, 1998.

[8] 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*, pp. 1527-1539, Nov. 1991.

[9] M. Öberg and P. H. Siegel, "Performance analysis of turbo-equalized dicode partial-response channel," *Proc., 36th Allerton Conference on Communication, Control and Computing*, (Monticello, IL, September 1998).