

Iterative Tri-stage Decoding for Turbo Codes in Partial Response Channels

Meng-guang Tsai, Kuen-suey Hou, and Hen-wai Tsao
 Graduate Institute of Communication Engineering
 National Taiwan University
 Taipei, Taiwan

Abstract—Combination of turbo codes with partial response (PR) channels was comprehensively studied in magnetic recording systems. In this paper, we investigate a decoding scheme called ‘iterative tri-stage decoding’ for the turbo code as the outer code and the precoded PR channel as the inner code. This decoding scheme reduces the computational complexities by reducing the number of iterations without degrading the coding gain.

I. INTRODUCTION

Turbo codes were introduced by Berrou, *et al.* [1] in 1993 which are capable of operating near the Shannon capacity on additive white Gaussian noise (AWGN) channels. Unlike conventional error control codes, turbo codes increase the Euclidean distances between coded sequences by increasing the interleaving size [2]. In magnetic recording systems, the intersymbol interferences (ISI) hamper the intended coding gain. The application of turbo equalization and turbo decoding for signal detection on partial response (PR) equalized channels has been extensively investigated [3-6]. Ryan [3], Ryan *et al.* [4] first investigated this scheme, and reported a gain of 4-6 dB on ideal PR4 and EPR4 channels. Souvigner *et al.* [5,6] further enhanced the decoding scheme called ‘full turbo decoding’ with better decoding gain. Souvigner *et al.* [5,6] also emphasized another scheme called ‘serial concatenation code’ which consists of only one single high rate convolutional code with lower decoding computational complexity at the expense of higher error floor.

In this paper, we investigate another decoding scheme called ‘iterative tri-stage decoding’. The corresponding encoder is the same as the one investigated by Ryan [3]. The decoding gain is almost the same as full turbo decoding [5], but with much lower computational complexity.

This paper is organized as the following. Section II introduces various turbo encoder and decoder schemes on PR equalized channels. Simulation results are presented in Section III. The comparison of computational complexity between the iterative tri-stage decoding scheme and the serial concatenation coding scheme is discussed in Section IV, and this paper is concluded in Section V.

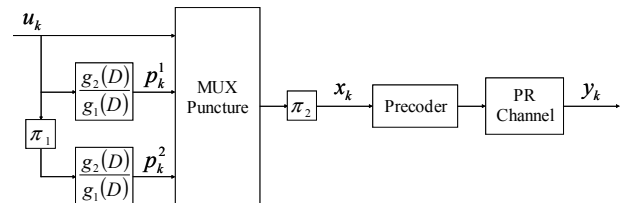


Figure 1. Turbo encoder with precoded PR channel.

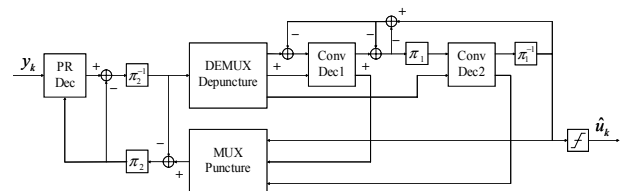


Figure 2. Full turbo decoder.

II. SYSTEM DESCRIPTION

The studies in this paper are based on the ‘standard concatenation’ arrangement where the outer code is the parallel concatenated convolutional code (PCCC), which is also known as the conventional turbo code, and the inner code is the precoded PR equalized channel. Fig. 1 shows the standard concatenated turbo encoder block schematic [3], which consists of two convolutional encoders parallel concatenated with an interleaver, π_1 , of length N . The user data sequence, \mathbf{u} , is interleaved by π_1 before entering the second convolutional encoder. The two parity sequences, \mathbf{p}^1 and \mathbf{p}^2 , are punctured to the desired coding rate, and multiplexed with the user data sequence. The multiplexed sequence passes through an interleaver, π_2 , resulting in the input sequence, \mathbf{x} , for the precoded PR equalized channel.

There are several decoding schemes for this encoder. Ryan [3] investigated a scheme called ‘partial turbo decoding’ (PTD), which the received sequence is first fed into a soft-output decoder matched to the precoded PR equalized channel, and then the calculated soft sequence is deinterleaved by π_2 and fed into the turbo decoder to

This work was supported in part by Ministry of Economic Affairs, Taiwan, R.O.C., under grant 93-EC-17-A-08-S1-0006.

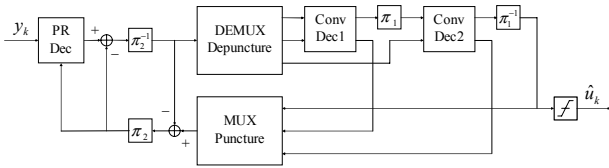


Figure 3. Iterative tri-stage decoder.

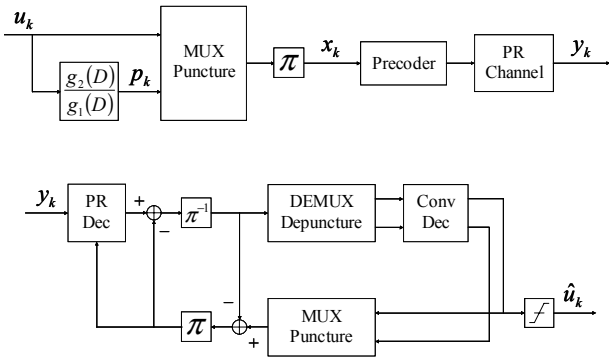


Figure 4. SCC encoder and decoder.

determine the user bit sequence. The decoding gain can be further improved by a scheme called ‘full turbo decoding’ (FTD) investigated by Souvignier *et al.* [5] by further feeding the extrinsic soft information decoded by the turbo decoder back to the soft-output decoder matched to the precoded PR equalized channel. This decoding scheme is shown in fig. 2.

In this paper, we introduce a new decoding scheme called ‘iterative tri-stage decoding’ (I3D), as shown in fig 3. The PR-Dec block is the soft-output decoder matched to the precoded PR equalized channel, and the Conv-Dec1 and Conv-Dec2 blocks are the soft-output decoders matched to the two convolutional encoders, respectively. The PR-Dec, Conv-Dec1, and Conv-Dec2 blocks can be implemented using the soft-output Viterbi algorithm (SOVA) [7] or the BCJR algorithm [8].

The received sequence is first fed into the PR-Dec block, and then the calculated *a posteriori* log-likelihood ratio (LLR) sequence of \mathbf{x} is deinterleaved by π_2^{-1} , demultiplexed and depunctured into the LLR sequences of \mathbf{u} , \mathbf{p}^1 , and \mathbf{p}^2 . These LLR sequences are fed into the Conv-Dec1 and Conv-Dec2 blocks, and then the calculated *a posteriori* LLR sequences of \mathbf{u} , \mathbf{p}^1 , and \mathbf{p}^2 are punctured and multiplexed. The multiplexed sequence is interleaved by π_2 , and provides the extrinsic information that is fed into the PR-Dec block for the next iteration. The term ‘tri-stage’ stands for that each iteration consists of three decoding stages: PR-Dec, Conv-Dec1, and Conv-Dec2.

The main difference between the proposed I3D scheme and the FTD scheme is that the former consists of only one single iterative loop, but the later consists of nested two-level loops. The computational complexity can be substantially reduced by decreasing the number of iterations, especially

for the loop consisting of the Conv-Dec1 and Conv-Dec2 blocks. In the full turbo decoding scheme, the lower-level loop formed by the Conv-Dec1 and Conv-Dec2 blocks iterates several times for each upper-level iteration which consists of the PR-Dec block. Thus if the upper-level loop iterates M times, and the lower-level loop iterates N times per upper-level iteration, the total number of times the Conv-Dec1 and Conv-Dec2 blocks being invoked is MN in the full turbo decoding scheme. Thus we can expect that the I3D scheme reduces the computational complexity due to the single iterative loop.

Souvignier [5] also investigate the serial concatenated coding (SCC) scheme which the outer code consists of only one convolutional code. Fig. 4 shows the encoder and decoder block schematic of the SCC scheme. The decoding computational complexity in the SCC scheme is much lower than the full turbo decoding. At high coding rate, i.e. 8/9, the coding gain in the SCC scheme is close to that of the PCCC scheme with FTD. But the drawback of the SCC scheme is that the error floor is higher than the PCCC scheme. We can see later that the PCCC scheme with the proposed I3D scheme avoids these drawbacks.

III. SIMULATION RESULTS

In the following simulations, the noise is assumed to be AWGN. The signal-to-noise ratio (SNR) is defined as

$$\begin{aligned} \text{SNR (dB)} &= 10 \cdot \log \left(\frac{E_b}{N_0} \right) \\ &= 10 \cdot \log \left(\frac{1}{2R} \cdot \frac{E_c}{\sigma^2} \right), \end{aligned} \quad (1)$$

where E_b is the signal energy per user bit, N_0 is the one-sided noise power spectral density, E_c is the signal energy per coded bit, σ^2 is the noise variance measured on the channel, and R is the coding rate. In all simulations, we can easily adjust the SNR by adjusting the ratio E_c/σ^2 .

Although the noise is assumed to be AWGN, it is worth to note that the performance may be further degraded by various reasons, such as misequalization, asymmetry, media noise, and other noise that is not modeled as AWGN. These conditions are not discussed in the paper.

In the following simulations, the convolutional encoders are described by generator polynomials, in octal form, $(g_1, g_2) = (31, 33)$, where g_1 is the feed back polynomial, and g_2 is the feed forward polynomial. The block size of packing user data bits is 4096. No user data bit is punctured, and the parity bits are punctured in a systematic fashion. For example, at coding rate 8/9, only the parity bits of both convolutional encoders located at multiples of 16 are passed in the PCCC coding scheme, and only parity bits located at multiples of 8 are passed in the SCC scheme [5]. The precoder combined with the PR equalized channel is $1/(1 \oplus D^2)$.

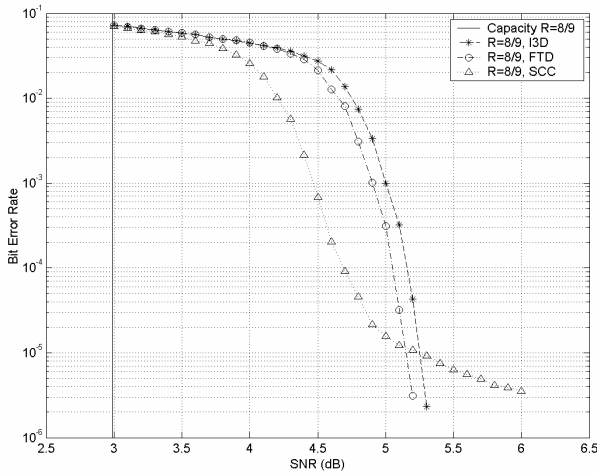


Figure 5. Bit error rate v.s. SNR for the I3D, FTD, and SCC schemes at coding rate 8/9.

When simulating with the I3D and SCC schemes, the decoding loop iterates 10 times. When using the FTD schemes, the upper-level loop iterates 10 times, and the lower-level loop iterates 10 times per upper-level iteration.

Fig. 5 shows the performance of the I3D scheme, the FTD scheme, and the SCC scheme in PR4 channels. The straight lines indicate the upper capacity bound (the lower SNR bound) of the PR4 channel [9]. Note that the coding gains of the I3D and FTD schemes are almost the same.

In fig. 5, we also find that in the SCC scheme, the error floor is higher than in the I3D and FTD schemes. Thus we can conclude that the I3D scheme avoids the problem of high error floor of the SCC scheme.

To reduce the decoding computational complexity, we adopt the ‘stop condition’ for whether the iteration should be stopped or not. The stop condition we used is that if the two consequent decoded user data sequences are the same, we stop iterating, until 10 iterations are encountered. Although not shown, adopting this stop condition incurs negligible performance loss with less than hundredth of dB.

Fig. 6 shows the average number of required iterations versus SNR in the FTD and I3D schemes when adopting the stop condition. The curves labeled ‘FTD-upper’ and ‘FTD-lower’ indicate the number of upper-level and lower-level iterations in the FTD scheme, respectively. Note that the computational complexity in the FTD scheme is mainly affected by the number of lower-level iterations. The curve labeled ‘I3D’ indicates the number of iterations of the I3D scheme. Referring to Fig. 5 and Fig.6 at the same SNR, we find that the peak value of the iterations locates close to the turning region in the cliff region, for both the I3D and FTD schemes. We also find that the curve labeled ‘FTD-upper’ is only slightly smaller than the curve labeled ‘I3D’. This hints us that the decoding gain is mainly obtained by the upper-level iterations in the FTD scheme, and too

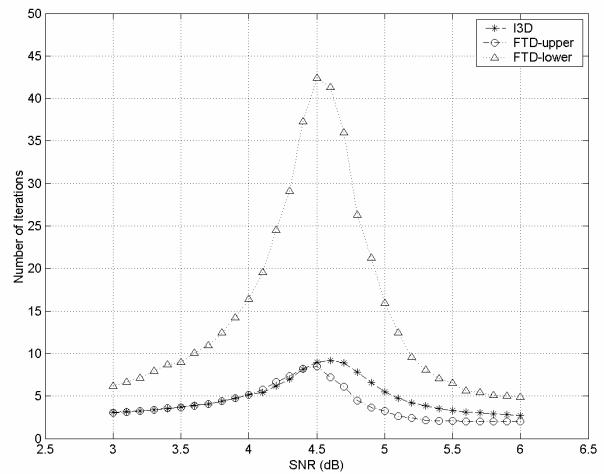


Figure 6. Average number of iterations versus SNR for the I3D schemes, the upper-level and lower-level loops of the FTD scheme.

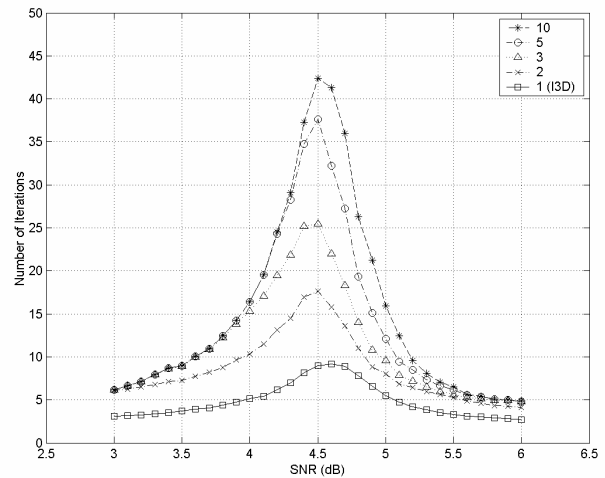


Figure 7. Average number of lower-level iteration required on the FTD scheme versus SNR with different limitations. The limitation is defined as the maximum number of lower-level iterations per each upper-level iteration.

much lower-level iterations merely waste the computational power to obtain only a few decoding gain. In fig. 6, we find that the large number of lower-level iterations in the FTD scheme wastes most of the computational power.

To get the idea why the I3D scheme provides the complexity reduction relative to the FTD scheme and suffers little coding gain, we present another view point in this paragraph. One way to reduce the number of total lower-level iterations in the FTD scheme is limiting the maximum number of lower-level iterations per upper-level iteration. Fig. 7 shows the effect of lowering the limit. The vertical axis is the average number of total lower-level iterations required at different SNRs, which can be an indicator of the computational complexity. Each curve is

labeled by a number, which indicates the maximum number limit of lower-level iterations per upper-level iteration. Note that the stop condition is also the same as previous, and the maximum number of upper-level iterations is also 10. In this figure, we find that the average number of total lower-level iterations is reduced if we lower the limit, and the computational complexity is also reduced. The extreme case is that there is only one lower-level iteration allowed per upper-level iteration, which results in exactly the same as the I3D scheme. This case results in the lowest computational complexity in Fig.7 and suffers little coding gain as shown in Fig.5.

IV. COMPARISON OF COMPUTATIONAL COMPLEXITY BETWEEN THE I3D SCHEME AND THE SCC SCHEME

Since the PCCC scheme with I3D and the SCC scheme both consist of only one single decoding loop, it is worth to compare the computational complexity between these two schemes. We have learned that the error floor of the PCCC scheme with I3D is lower than the SCC scheme in Section III. As for the computational complexity, the iterative loop of the I3D scheme consists of the PR-Dec, Conv-Dec1, and Conv-Dec2 blocks, but that of the SCC scheme consists of only two decoding blocks, the PR-Dec block and the decoding block matched to the convolutional encoder. We expect that the computational complexity of the I3D scheme is slightly higher than the SCC scheme.

The computational complexity of the BCJR algorithm [7] is proportional to the block size multiplied by the number of states in the corresponding coding system. Assume that the block size for packing the user bits is N , the coding rate is R , the convolutional encoder is described as $(g_1, g_2) = (31, 33)$, and the signal is transmitted under the PR4 channel. While decoding the convolutional code, the number of states is 16 and the block size is N , which results the computational complexity is $16N$. While decoding the PR4 channel with the precoder $1/(1 \oplus D^2)$, there are two delay elements, thus the number of states is 4, and the block size is N/R due to the additional parity bits, and it results the computational complexity is $4N/R$.

In the SCC scheme, the decoding computational complexity is $(4N/R + 16N)$ per iteration. In the I3D scheme, the complexity is $(4N/R + 16N + 16N)$ per iteration. Thus the computational complexity ratio of the SCC scheme to the I3D scheme per iteration is

$$\frac{\frac{4N}{R} + 16N}{\frac{4N}{R} + 16N + 16N} = \frac{1 + 4R}{1 + 8R}. \quad (2)$$

For example, at coding rate $R = 8/9$, the above ratio equals to $41/73 = 0.56$. Thus 1 iteration on the SCC scheme equals to 0.56 iteration on the PCCC scheme with the I3D scheme measured on the same computational complexity.

V. CONCLUSIONS

We have introduced a new decoding scheme, the I3D scheme, for decoding the PCCC structure in the PR channel, which is usually decoded by the FTD scheme in prior arts to get the best performance. Since the extrinsic information is not fed back in the lower-level iteration of the I3D scheme, the computational complexity of the I3D scheme is much less than that of the FTD scheme. Moreover, the error floor of the I3D scheme is as low as the FTD scheme.

The computational complexity of the I3D scheme is also compared with the simple SCC structure in the PR channel. The I3D scheme is slightly more complex than the SCC structure, but the SCC structure possesses a problem of higher error floor which the I3D scheme doesn't have. Thus, the I3D scheme for the PCCC structure provides a good alternative in the PR channel.

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] S. Benedetto and G. Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," *IEEE Trans. on Information Theory*, vol. 42, no. 2, Mar. 1996, pp. 409-428.
- [3] W. Ryan, "Performance of high rate turbo codes on a PR4-equalized magnetic recording channel," in *Proc., IEEE Int. Conf. on Communications*, Atlanta, Georgia, USA, Jun. 1998, pp. 947-951.
- [4] 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, USA, Mar. 1998, pp. 489-493.
- [5] 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. on Communications*, Vancouver, BC, Canada, Jun. 1999, pp. 1638-1642.
- [6] T. V. Souvignier, M. Öberg, P. H. Siegel, R. E. Swanson, and J. K. Wolf, "Turbo Decoding for Partial Response Channels," *IEEE Trans. on Communications*, vol. 48, no. 8, Aug. 2000, pp. 1297-1308.
- [7] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft decision outputs and its applications," in *Proc., IEEE Intl. Conf. Global Telecommunication*, Rio de Janeiro, Brazil, Dec. 1999, pp. 723-727.
- [8] L. Bahl, J. Cocker, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. 20, Mar. 1974, pp. 284-287.
- [9] S. Shamai, L. Ozarow, and A. Wyner, "Information rates for a discrete-time Gaussian channel with intersymbol interference and stationary inputs," *IEEE Trans. Information Theory*, Nov. 1991, pp. 1527-1539.