ITERATIVE DECODING FOR DIGITAL RECORDING SYSTEMS

Jan Bajcsy, James A. Hunziker and Hisashi Kobayashi
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
e-mail: bajcsy@ee.princeton.edu, hisashi@ee.princeton.edu
Fax: (609) 258-3745, Tel: (609) 258-1984

Abstract

In many existing optical and magnetic digital recording systems, data is usually encoded by a cascade of several encoders, followed by a channel with ISI (intersymbol interference). Conventional receivers follow a one-path approach, where an equalizer is followed by a cascade of appropriate decoders.

We propose a backward compatible iterative receiver which uses generalized erasures. An AZD (ambiguity zone detection) detector labels unreliable symbols as erasures. These are then resolved in an iterative process by the equalizer and the decoders.

The proposed decoding scheme can significantly improve the system performance. It is implementable with low decoding complexity, and introduces only a small decoding delay.

1 Introduction

Magnetic disk drives and tapes, digital audio tape and digital video tape are important examples of magnetic recording applications, and compact disc, DVD and rewritable DVD are among optical recording systems. The main technical goals in these digital recording products are to increase recording density with greater speed and reliability at lower cost.

Most existing digital recording systems can be viewed as a generalized concatenated system [1] in which several component encoders are concatenated in series. In the recording process the data is first encoded by one or more error correcting encoders, followed by a modulation code and a partial response channel [14], [11], [15]. The error correcting codes are usually Reed-Solomon codes or their variants, e.g., shortened codes or product codes. The recording code is usually a sophisticated version of a run-length limited sequence encoding which alleviates the ISI (intersymbol interference) introduced by the channel [3], [9], [11], [12]. The channel with ISI is usually followed by an equalizing filter that confines the span of ISI within a reasonable number of symbols, thus creating, in effect, some sort of a partial response (PR) channel.

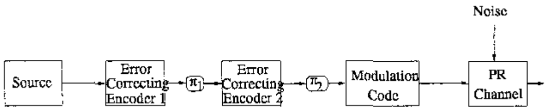

Figure 1: Structure of a generic recording system.

The conventional receiver for these systems uses a one-path structure, as depicted in Figure 3. The PR channel is followed by a maximum-likelihood decoder, as first proposed by Kobayashi [10]. The data is passed through decoders for the modulation and error correcting codes.

Motivated by the success of the iterative decoding of Turbo codes [4] and by soft decision decoding scheme discussed by Hagenauer et al. in [15] in decoding a concatenated code with Reed-Solomon code and convolutional code, we propose an iterative decoder for existing recording systems, e.g., CD, DVD or digital video tape. The basic idea is to create a joint teamwork of the decoders throughout the iterations. Joint operations among the decoders significantly reduces the BER (bit error rate) as compared to the conventional one-path receiver.

The structure of this paper is as follows: Section 2 gives a description of an assumed recording system model and its conventional receiver. Section 3 describes our proposed iterative receiver. Section 4 shows the performance results of the new receiver for different channels. Section 5 provides concluding remarks and directions of future work.

2 Recording System

We consider a recording system depicted in Figure 1 whose structure and parameters are similar to those of the CD, DVD and digital video tape systems.

The first error correcting code is a $(28, 24, 5)$ shortened Reed-Solomon (RS) code over $GF(256)$ with primitive polynomial $P(x) = 1 + x^2 + x^3 + x^4 + x^8$ and the generator polynomial of the code

$$g_1(x) = \prod_{i=1}^{4}(x - \alpha^i). \qquad (1)$$

0-7803-4984-9/98/$10.00 ©1998 IEEE.

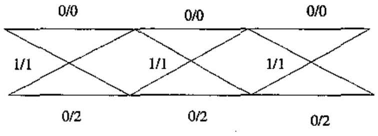

Figure 2: Trellis representation of the precoded duobinary channel. Edge labels denote the input bit and the corresponding output symbol.

Before entering the second encoder, the data passes through a block interleaver π_1 of size 28×28. The second code is a $(32, 28, 5)$ shortened RS code over the same Galois field, and with the same generator polynomial.

The second encoder is followed by another interleaver π_2, which is similar to the one used in the DVD system. It writes data bytes row-wise into a 28 x 32 array, then permutes the rows so that Row_i goes to Row_n, where

$$n = \begin{cases} i + \lfloor (i-1)/6 \rfloor & \text{for } i = 1, 2, ..., 24 \\ (i-24) * 7 & \text{for } i = 25, .., 28 \end{cases}, \qquad (2)$$

and $\lfloor \cdot \rfloor$ denotes the lower integer part. The data is then read out column-wise and then passed to the modulation encoder, which operates on the bit stream, taking in 8 bits and outputting 9. For a thorough exposition on the topic of modulation codes see either [9] or [12].

The output of the modulation code becomes the input to the partial response channel. In our case the PR channel includes a possible precoder, the ISI channel and the equalizing filter, and can be represented by a polynomial of finite degree

$$g(D) = g_0 + g_1 D + ... + g_n D^n, \qquad g_i \in R. \qquad (3)$$

The precoded message $m(D)$ that has passed through the PR channel is mapped into a sequence, which in polynomial form is given by

$$x(D) = m(D) g(D). \qquad (4)$$

The overall noiseless PR channel can then be represented by a trellis diagram. In our analysis, we model the channel as a duobinary channel (or a PR class 1 with $h(D) = 1 + D$) with a differential precoder. This is primarily for the sake of in presenting our idea, but the approach can be easily generalized to any form of PR. Its trellis representation is depicted in Figure 2. The input labels denote data bits and the output labels are hypothetical channel symbols in a noiseless case.

Finally, channel noise corrupts the channel symbols. We discuss two simple and idealized cases for the channel noise. The first case is where the effects of the noise are reflected in erased symbols at the quantized output. In the second case we will consider an additive white Gaussian noise model.

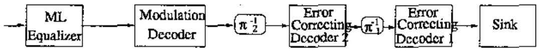

Figure 3: Conventional 1-path receiver in a generic recording system.

The conventional receiver, depicted in Figure 3, first performs maximum likelihood (ML) decoding on the noisy channel symbols using the Viterbi algorithm and the PR trellis. This technique is often denoted PRML and first explored by Kobayashi in [10]. The decoded bits are then passed to the modulation decoder, which decides whether the data has been received reliably or should be declared as "erasure" or ambiguous digit. The output bytes and erasures then proceed through the decoders for the RS codes, where the errors and erasures are corrected and resolved. Finally, the data is passed to the sink. Interleaving of the data is performed throughout the process, so that the data enter the decoders in an appropriate order, as specified at the recording side.

3 Proposed Iterative Decoding System

3.1 Principle of Operation

The proposed receiver, depicted in Figure 4, functions as follows. First, the PR channel output symbols

$$y_k = x_k + n_k, \qquad (5)$$

where x_k are the noiseless duobinary symbols and n_k's are the noise samples, are decoded using the Viterbi algorithm (VA). The decoded bits are passed to the decoder of the modulation code, which serves as a generalized AZD detector. Using bits from ML decoder, the modulation decoder tries to resolve transmitted bytes/symbols. If it cannot do this, given byte/symbol is labeled as erased. This data is then passed to the RS decoders.

A decoding technique with AZD was first discussed by Kobayashi and Tang [8] as a suboptimal but algebraic alternative to the maximum likelihood decoding (MLD) or the Viterbi decoder, which is a probabilistic decoder [9]. Here, AZD utilizes the fact that the modulation code has a certain amount of redundancy, i.e., only 256 of all 512 binary sequences of length 9 are allowed at the output of the modulation encoder. The modulation encoder can be thought as a map

$$f : \mathcal{M} \rightarrow \mathcal{C} \subseteq \{0, 1\}^9, \qquad (6)$$

where $\mathcal{M} = \{0, 1, .., 255\}$ is the set of bytes and $\mathcal{C}$ is a codebook of 256 modulation codeword. The AZD detector partitions the set $\{0; 1\}^9$ into two sets $\mathcal{C}$ and $\mathcal{E}$. If an illegal sequence $\hat{\mathbf{a}} \in \{0, 1\}^9$, $\hat{\mathbf{a}} \notin \mathcal{C}$, is detected at the input of the modulation decoder an erased byte is output, i.e.,

$$AZD_{out}(\hat{\mathbf{a}}) = \begin{cases} f^{-1}(\hat{\mathbf{a}}) & \text{for } \hat{\mathbf{a}} \in \mathcal{C} \\ \mathrm{E} & \text{for } \hat{\mathbf{a}} \in \mathcal{E} \end{cases} \tag{7}$$

The concatenated RS decoders form a loop. They are separated by a permutation (in the feed-back path) and its inverse-permutation (in the forward path) to preserve the order of the data. These two decoders are capable of performing decoding of sequence that contains erasures, e.g., using a Reed-Solomon decoding algorithm based on the Euclidean/continued fraction algorithm. If the codeword can be corrected, the corrected codeword (both data symbols and parity symbols) is put into the interleaver. If the codeword cannot be corrected due to an excessive number of errors or erasures, it is passed through without any change.

During the first iteration, the AZD output sequence is decoded by the inner decoder, then passed to the inverse permutation, the outer decoder and the permutation (Figure 4). At the end of the first iteration, the original AZD output sequence is modified by the "error/erasure corrector", which incorporates the corrections made in the first path. The second iteration applies to this modified AZD output, which revisits the receiver blocks in the same order as in the first iteration. This cyclical decoding procedure repeats itself.

In each iteration, some of the remaining errors and/or erasures will be resolved, and the "error/erasure corrector" modifies the AZD output sequence, by using a simple logic circuit (or logic table) which substitutes some digits of the AZD sequence by their corrected values. In the first iteration, the "error/erasure corrector" plays no role, since the feedback loop does not provide any information when the iterative decoding just begins.

The iterative procedure should end when all erasures are resolved and no errors are detected, or when no further resolution of error/erasure is achieved, or after a prescribed maximum number of iterations is reached.

The decision block at the end of the first path checks if the error correction has been completed. (It starts using an error detection code in the source data after certain number of iterations.) If the detecting test is passed or the maximum prespecified number of iterations is achieved, the data is passed to the sink. Otherwise, next decoding iteration will start.

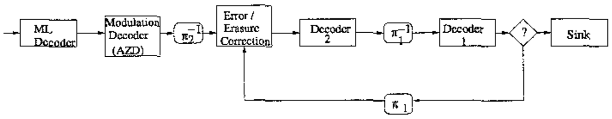

Figure 4: Proposed iterative decoder for the recording system.

3.2 An Ilustrative Example

The following simple example shows concretely the operation of the proposed iterative decoder from Figure 4. For the sake of this example's simplicity, we take Encoder 1 and Encoder 2 to be both (3,2) shortened RS codes over $GF(2^8)$ with $d_{min} = 2$ and generator matrices

$$\mathbf{G_1} = \mathbf{G_2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & \alpha \end{bmatrix}. \tag{8}$$

The encoders are separated by a 2x3 block interleaver π_1. For the simplicity of this example, we assume the other interleaver π_2 is an identity interleaver and an actual modulation code is represented by a (9,8) single parity check code. The PR channel is a precoded duobinary channel represented by the trellis in Figure 2.

1. Consider four information bytes given by a stream
$$\mathbf{I}_1 = (\mathbf{0000}). \tag{9}$$
Note that bold face numbers will represent bytes, whereas bits and duobinary symbols will be denoted by regular numbers.

2. Encoder 1 segments the data into blocks of 2 bytes and each block is then encoded using the (3, 2) RS code. Then encoder output is then formed by the following 6 bytes
$$\mathbf{I}_2 = (\mathbf{000}, \mathbf{000}), \tag{10}$$
where we put commas between consecutive coded blocks.

3. Permutation π_1, a 2 × 3 block interleaver, will store the above 6 bytes row-wise in the following array structure
$$\pi_1 \equiv \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{11}$$
and the output is obtained by reading out the above array contents column-wise. We again include commas just for clarity of presentation:
$$\mathbf{I}_3 = (\mathbf{00}, \mathbf{00}, \mathbf{00}). \tag{12}$$

4. Encoder 2 encodes the blocks in $\mathbf{I}_3$ using the (3,2) RS code obtaining

$$\mathbf{I}_4 = (\mathbf{000}, \mathbf{000}, \mathbf{000}). \tag{13}$$

5. The modulation encoder first converts each byte in $\mathbf{I}_4$ into 8 bits, which then get encoded into 9 bits. Hence the resulting stream contains 81 bits and dummy 0 could be added as the 82nd element to help the ML decoder find the ML path. The modulation encoder output is thus

$$\begin{aligned} \mathbf{I}_5 = \; & (000000000, 000000000, 000000000, \\ & 000000000, 000000000, 000000000, \\ & 000000000, 000000000, 000000000), \end{aligned} \tag{14}$$

   where the commas separate sequences of bits corresponding to different bytes.

6. The noiseless duobinary sequence on the channel is then given by

$$\begin{aligned} \mathbf{I}_6 = \; & (000000000, 000000000, 000000000, \\ & 000000000, 000000000, 000000000, \\ & 000000000, 000000000, 000000000), \end{aligned} \tag{15}$$

7. Suppose that after receiving the noisy version of I_6 from the channel, the ML decoder makes some errors and outputs the following bits

$$\begin{aligned} \mathbf{I}_7 = \; & (000000001, 111000000, 000000000, \\ & 000000001, 000000000, 000000100, \\ & 000000000, 000000000, 000000000). \end{aligned} \tag{16}$$

8. At the modulation decoder, the AZD detector finds out that sequences 1, 2, 4 and 6 in $\mathbf{I}_7$ do not belong to the codebook of the modulation encoder. The decoder consequently outputs bytes 1, 2, 4 and 6 as erased

$$\mathbf{I}_8 = (\mathbf{EE0}, \mathbf{E0E}, \mathbf{000}). \tag{17}$$

9. Decoder 2 takes $\mathbf{I}_8$ and at this stage cannot correct any of the erasures, hence it outputs

$$\mathbf{I}_9 = (\mathbf{EE}, \mathbf{E0}, \mathbf{00}). \tag{18}$$

10. By applying π_1^{-1} (i.e., 2×3 de-interlever) we obtain the following array, where the write-in is performed vertically, and read-out is done horizontally

$$\pi_1^{-1} \equiv \begin{bmatrix} \mathbf{E} & \mathbf{E} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{19}$$

   The output of the de-interleaver is therefore given by

$$\mathbf{I}_{10} = (\mathbf{EE0}, \mathbf{E00}). \tag{20}$$

11. By applying Decoder 1 to each block in $\mathbf{I}_{10}$ we can resolve one erasure obtaining

$$\mathbf{I}_{11} = (\mathbf{EE0}, \mathbf{000}). \tag{21}$$

12. Therefore, after the first iteration two of the four information bytes are still erased, so the second iteration will be started.

13. At the start of the second iteration, permutation π_1 re-interleaves the result of the first iteration $\mathbf{I}_{11}$ by writing it row-wise into the following array

$$\pi_1 \equiv \begin{bmatrix} \mathbf{E} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{22}$$

   and reading it out column-wise

$$\mathbf{I}_{12} = (\mathbf{E0}, \mathbf{E0}, \mathbf{00}). \tag{23}$$

14. The "Error/Erasure Correction" block compares the original output of the AZ detector $\mathbf{I}_8$ and the above $\mathbf{I}_{12}$, and an updated version of the AZ detector output is obtained as

$$\mathbf{I}_{13} = (\mathbf{E00}, \mathbf{E0E}, \mathbf{000}). \tag{24}$$

15. Decoder 2, with $\mathbf{I}_{13}$ as its new input, can correct one of the remaining erasures and outputs

$$\mathbf{I}_{14} = (\mathbf{00}, \mathbf{E0}, \mathbf{00}). \tag{25}$$

16. Inverse permutation π_1^{-1} is then applied to $\mathbf{I}_{14}$

$$\pi_1^{-1} \equiv \begin{bmatrix} \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{26}$$

   and the output of the de-interleaver is then given by

$$\mathbf{I}_{15} = (\mathbf{0E0}, \mathbf{000}). \tag{27}$$

17. Decoder 1 now corrects the last erasure in I_{15} obtaining

$$\mathbf{I}_{16} = (\mathbf{000}, \mathbf{000}). \tag{28}$$

18. Therefore, all the ambiguities among the information bytes have been resolved by the end of the second iterative step. Consequently, the iterative decoder outputs decoded information bytes

$$\hat{\mathbf{I}}_1 = (\mathbf{0000}). \tag{29}$$

   and stops.

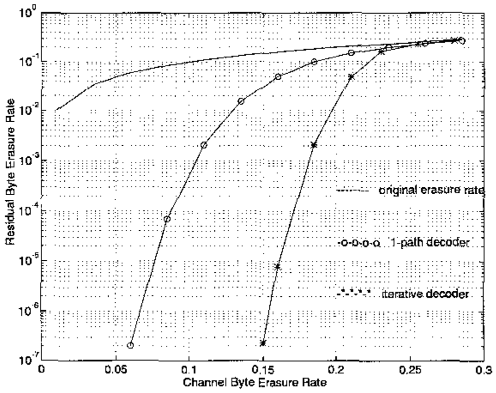

Figure 5: Performance curves for the conventional and iterative receiver on pure erasure channel.

4 Performance Results

We compare the performance of the one-path receiver and our proposed iterative receiver for the recording system specified in Section 2. First, we assume an idealized model of the recording channel without errors. We consider a channel with pure erasures, i.e., the modulation decoder outputs are either correct values or complete erasures. The byte erasures are assumed to be i.i.d. distributed at the output of the modulation decoder.

The simulation results are shown in Figure 5, which depicts the original erasure rate vs. residual bit erasure rate. For this channel the RS decoders will not introduce any errors in the decoding process, but they will be unable to decode if there are excessive erasures. For the desired erasure rate of 10^{-6}, the iterative receiver can resolve about two and a half times as many erasures as the one-path receiver. Another way to interpret these results is that at the symbol erasure rate of 15% our decoder outperforms the existing decoder by almost 5 orders of magnitude. Note that the decoder finished decoding after 2 to 4 iterations in most cases, thus making it practical from the decoding delay point of view.

Then we compared the performance of the conventional and new schemes for a continuous channel model. We assumed the noise at the duobinary channel is additive white Gaussian noise as in [10]. Consequently, the decoders have to deal with both errors and erasures.

The simulation results are shown in Figure 6, which depicts SNR vs. residual bit error rate (BER). The four curves depict the performance after one through four iterations. The first iteration result is equivalent to the result for the one-path receiver. For the desired BER rate of 10^{-6}, the decoding gain of about 1 dB is achieved due to the iterative decoding. Another way to view the results is for the channel SNR of 7.5 dB the iterative receiver lowers the BER rate by a factor of almost 1000 in 4 iterations. These results are consistent with the initial observation for the pure erasure channel.

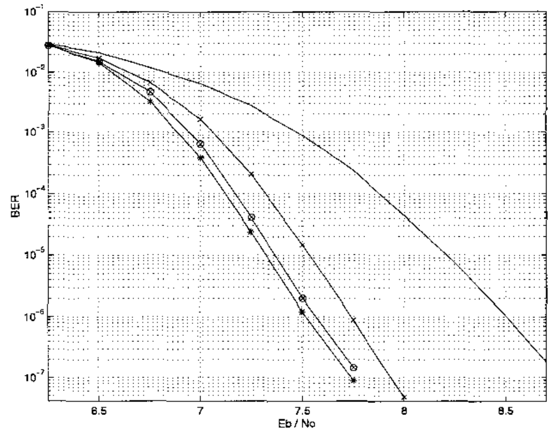

Figure 6: Performance curves for the conventional 1-path receiver (the top curve) and of the proposed iterative receiver iterations 1-4.

5 Conclusion and Future Work

Given the fact that most existing digital recording systems can be viewed as a generalized concatenated system, the proposed iterative decoding scheme should be applicable to many digital recording systems. Since the proposed solution is backward compatible in the sense that the recording system does not have to be changed at all, it can be used in many existing systems such as digital disk drives, compact disc players, DVD, etc.

As the simulation results show, our scheme offers a significant performance improvement over the conventional systems including PRML based products. It can be achieved with small added complexity in the decoder, since most improvements are achieved within the first four iterations.

Currently, we are considering two ways of improving the proposed iterative receiver. First, the ML decoder for the PR channel would provide soft decisions, using the SOVA algorithm of Hagenauer et. al. [7]. This will provide a better way of determining unreliable symbols at the AZD detection unit of the modulation decoder. The next goal is to extend the iterative decoding over all four decoders, as shown in Figure 7 and tailor the noise model to the specifics of the considered application. As our preliminary results indicate, this scheme enables us to gain further

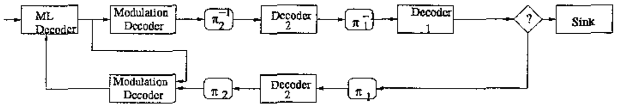

Figure 7: Iterative decoder for the overall system.

improvement.

Acknowledgements:

The present work has been supported, in part, by the National Science Foundation, the New Jersey Commission on Science and Technology, Asahi Chemical Co. Ltd and the Ogasawara Foundation for the Promotion of Science and Engineering.

References

[1] J. Bajcsy and H. Kobayashi, "Error Control of Generalized Concatenated Systems, " *PACRIM'97*, Victoria, B.C., Canada, *Conference Record*, pp. 749-752.

[2] J. Bajcsy and H. Kobayashi, Invention disclosures submitted to Princeton University, June 28 and Aug. 22, 1996; filed to the U.S. Patent Office, *" Error Control of Generalized Concatenated Systems."*

[3] J. W. M. Bergmans, *Digital Baseband Transmission and Recording*, Kluwer Academic Publishers, 1996.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes (I)," *Proc. ICC '93*, pp. 1064-1070, Geneva Switzerland, May 23-26, 1993.

[5] C. Berrou and A. Glavieux, "Turbo Codes: General Principles and Applications", *Sixth Tirrenia Workshop on Digital Communication,* pp. 215-226, 1993.

[6] J. D. Forney, *Concatenated Codes*, MIT Press, 1966.

[7] J. Hagenauer, P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *GLOBECOM 1989*, Dallas, Texas, *Conference Record*, pp. 1680-1686.

[8] H. Kobayashi and D. T. Tang, "On Decoding of Correlative Level Coding Systems with Ambiguity Zone", *IEEE Trans. Communications*, Vol. COM-19, Aug. 1971, pp. 467-477.

[9] H. Kobayashi, "A survey of coding schemes for transmission or recording of digital data", *IEEE Transactions on Communication Technology,* Dec. 1971, pp. 1087-1100.

[10] H. Kobayashi, "Application of probabilistic decoding to magnetic recording systems", *IBM J. of Res. Develop.* , vol. 15, Jan. 1971, pp. 64-74.

[11] K. A. S. Immink, "The Digital Versatile Disc (DVD): System Requirements and Channel Coding," *SMPTE Journal*, vol. 105, Aug. 1996, pp. 483-489.

[12] B. H. Marcus, P. H. Siegel, and J. K. Wolf, "Finite-State Modulation Codes for Data Storage", *IEEE Journal on Selected Areas in Comm.* , vol. 10, Jan. 1992, pp. 5-37.

[13] M. Mansuripur and G. Sincerbox, "Principles and Techniques of Optical Data Storage", *Proc. of the IEEE* , vol. 85, Nov. 1997, pp. 1780-1796.

[14] M. Umemoto, Y. Eto and T. Fukinuki, "Digital Video Recording", *Proceedings of the IEEE,* vol. 83, July 1995, pp. 1044-1054.

[15] S. B. Wicker and V. K. Bhargava (eds.), *Reed-Solomon Codes and their Applications*, IEEE Press, 1994.