

A RATE 8/9 SLIDING BLOCK TRELLIS CODE WITH STATIONARY DETECTOR FOR MAGNETIC RECORDING

Borivoje Nikolic^{1,2}, Michael Leung¹, Leo Fu¹

¹Texas Instruments Storage Products Group, 2460 North First St, Suite #220, San Jose, CA 95131

²Department of Electrical and Computer Engineering, University of California, Davis, CA 95616

Abstract— A new trellis code for partial-response magnetic recording channels, that eliminates most frequent errors while keeping the code trellis invariant in time, is proposed. Recently published codes either had lower code density or resulted in time varying trellises with a period of 9, thus requiring a higher complexity of detectors. The new code introduces dependency between codewords to achieve the same coding constraints as 8/9 code, with the same code density, resulting in a trellis that has a period of two. The new trellis code eliminates two states in every second step of the E²PR4 trellis, requiring a 14-state two-step sequence detector.

I. INTRODUCTION

Contemporary magnetic recording channels use high density of recording bits that involves partial-response detection. It was identified that at high levels of interference between recorded bits in partial response magnetic recording channels, the most common errors are produced by the failure to detect a sequence of three or more transitions [1]. The use of trellis coding increases the distance between codewords by introducing the redundancy such that the most common error events are avoided by coding constraints. This was first observed by Behrens and Armstrong [2] on the E²PR4 channel by the use of $d = 1$, rate 2/3 code with a detector that matches the resulting trellis. This code increased the minimum squared error distance in E²PR4 channel from 6 to 10, producing a 2.2dB coding gain. Karabed and Siegel [3] characterized the low-distance error events in the E²PR4 channel, and shown that higher rate codes can be implemented that achieve the same coding gain. Several codes have been proposed which eliminate dominant error events by coding constraints [4-7]. The major disadvantage of these codes is that their constraints either limit the recording density, have long burst error propagation, or require the implementation of the Viterbi detector that is variable in time. This emphasizes the need for higher rate code that results in stationary detector and has short error propagation.

The new trellis code is presented, that reduces the complexity of the detector while keeping its two-step implementation time invariant. It has the same properties as earlier proposed rate 8/9 codes, which required a time-varying detector, with burst error propagation limited to two bytes.

II. DOMINANT ERROR SEQUENCES IN E²PR4 CHANNELS

High-density magnetic recording channels are usually modeled by polynomials of the form

$$h(D) = (1 - D)(1 + D)^N, N \geq 1 \quad (1)$$

where D denotes unity sample delay.

In the case of $N = 1$, the channel is known as partial response class-4 (PR4), $N = 2$ corresponds to extended PR4 (EPR4), and $N = 3$ is usually denoted as E²PR4.

Classifications of error events have shown that some particular error sequences occur more often than the others in partial response channels. At high user densities most dominant error events correspond to false detection of sequences of three or more transitions. These dominant errors could be eliminated, if the modulation code does not allow certain sequences to appear in the channel.

User data words (bytes) in NRZI notation are input to the encoder. The encoder generates codewords, which are serialized and passed through the precoder of the type $1 \oplus D$. The precoder converts NRZI sequence to NRZ sequence. In NRZ notation, 0 and 1 represent two levels of write current, which result in two levels of magnetization. In NRZI notation 1 represents a transition and 0 represents no transition. The sequence of two consecutive transitions is usually denoted as a dibit, three consecutive transitions form a tribit, and four transitions form a quadbit.

The detected sequence in the read head is the convolution of recorded sequence with channel response. We can define error events (sequences) as the difference between recorded and detected data at different points in the channel:

$$\text{User data: } e_u(D) = u(D) - u'(D) \quad (2)$$

$$\text{Channel input: } e_x(D) = x(D) - x'(D) \quad (3)$$

$$\text{Channel output: } e_y(D) = y(D) - y'(D) \quad (4)$$

The sequence at the output of the channel is corrupted by noise, equalized and brought to Viterbi detector, which produces the maximum-likelihood estimate of the recorded sequence.

Error event distance is defined as a squared Euclidean norm

$$d^2(e_x) = \|e_y(D)\|^2. \quad (5)$$

Table 1: Closed error events in E²PR4.

Type	d^2	Event
1	6	+ - + 0 0 0 0
2	8	+ - + 0 0 + - + 0 0 0 0
3	8	+ - [+ -] 0 0 0 0
4	8	+ - + [- +] 0 0 0 0
5	10	+ 0 0 0 0
6	10	+ 0 0 + - + 0 0 0 0
7	10	+ - + 0 - + - 0 0 0 0
8	10	+ - + 0 0 + - + - [+ -] 0 0 0 0
9	10	+ - + 0 0 0 - + - 0 0 0 0
10	10	+ - + 0 0 + 0 0 + - + 0 0 0 0
11	10	+ - + 0 0 + - + 0 0 + - + 0 0 0 0

The minimum distance error event in E²PR4 channel has the Euclidean distance of 6, and there are three more error events with squared error distances of 8, while the matched-filter-bound for E²PR4 is 10.

Error sequences with lowest squared distances in E²PR4 channel are shown in Table 1 [7, 8]. In Table 1, the pattern within the brackets can be repeated one or more times in succession.

Elimination of the first four events from Table 1 by coding constraints increases the minimum squared error distance in E²PR4 channels from 6 to 10.

The errors of type 3 and 4 are caused by detector failure to recognize a quadbit. Therefore, elimination of quadbits in the code eliminates these types of errors.

The errors of type 1, 2, 6, 7, 8, 9, 10 and 11 are either due to existence of a quadbit or to the detector's inability to determine the position of a tribit, effectively shifting the start of the tribit by one. Therefore by limiting the occurrence of tribits only on odd or even positions in the codewords, these errors could be eliminated. After applying these constraints, single-bit error event (type 5) becomes dominant.

Recently, several codes have been proposed for partial response signaling, which eliminate the smallest squared distance error events by coding constraints. Maximum transition runlength (MTR) codes, proposed by Moon and Brickner [4], limited number of consecutive transitions to two, but resulted in reduced, $C = 0.8791$ recording capacity. The same authors shown the design of 4/5 rate code and have proven that the constraints can be set to allow 6/7 code rate. Keirn and McCarthy [5] demonstrated that tribits can be allowed on either odd or even positions in the codeword, but not both, which resulted in rate 8/9 code. Bliss at the same time presented the code that mapped 16 user bits to 18 bit codewords [6]. This code had long burst error propagation of 4 bytes. Moision, Siegel and Soljanin [7] introduced rate 8/9 code with time-varying MTR (TMTR) constraint, that limited error propagation to 2 bytes. The major disadvantage of this code is that its restriction on allowed tribits requires the implementation of the Viterbi detector that is variable in time, significantly increasing its complexity and reducing the speed.

III. THE NEW CODE DESIGN

The code proposed here, has the code rate of 8/9, a maximum zero runlength of 10 and the same distance properties as the TMTR code [7]. The resulting trellis has a period of 2, rather than 9. The application of this code keeps 2-step implementation of the detector time-invariant. An example is that when this code is applied to the E²PR4 trellis, only a 14-state one-step lookahead stationary Viterbi detector is required vs. a 16-state time-varying detector for TMTR code.

The reduced complexity of the detector should increase its speed and power performance. Another advantage is to ease the synchronization of the detector. With a TMTR code, the Viterbi detector is required to synchronize to 1 out of 9 cycles and therefore has to rely on the synchronization pattern in the read cycle. With added latency in the Viterbi detector, the resulting system usually has to pass the first few bytes of data in a regular, non-trellis mode. In many cases the first few data samples tend to have more distortion from timing recovery and AGC due to mode switching from acquisition to tracking, which involves the change of the loop filter time constant. With this new code, which only needs one out of two cycles to synchronize, the synchronization can be done with the 4T preamble pattern and detector could be aligned to receive first data samples or even the synchronization byte in trellis mode.

The application of the code is not limited to partial response channel, it can also be used in other signal processing architectures, e.g. fixed-delay-tree search (FDTS) to allow the stationary implementation of an interleaved two-step scheme to achieve high data rate.

The new code eliminates all the small distance error events similarly to 8/9 TMTR code, thus by not allowing quadbits in codewords and limiting the occurrence of tribits to certain bit positions inside the codeword. Allowed tribits in 8/9 code are shown in Figure 1, limiting the number of available codewords that satisfy the coding constraints to 267. Since the beginning of the tribit is allowed at positions 2, 4, 6 and 9, with tribit starting at position 9, wrapping up in the next word, this implementation results in a time-varying trellis. Two states, 0101 and 1010 are eliminated from E²PR4 trellis, but trellis changes every nine cycles to conform to code constraints.

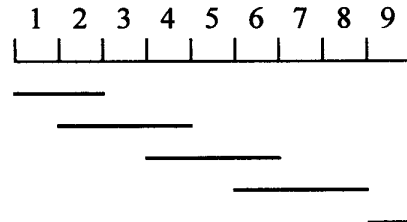


Figure 1: Allowed tribits in 8/9 code [7].

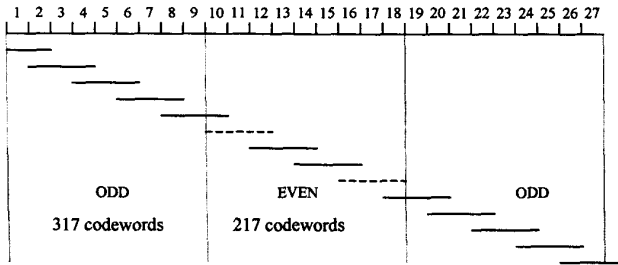


Figure 2: Extension of 8/9 constraints to 16/18 code.

Expanding the constraints to 18 bit codewords, as shown in Figure 2, does result in stationary trellis. But, 16 to 18 bit mapping would require large encoding/decoding logic and resulting code would have long byte error properties. Error propagation could be limited by separate encoding of bytes. If the bytes are encoded separately, the tribit starting at position 10 (beginning of the even code word) and at position 16 (end of even code word) would not be allowed, which is marked by dashed lines in Figure 2. Under these constraints, the number of available code words in the first 9-bit group is 317 and there are only 217 available code words in the second. This allows the mapping of 16 user bits to 18 channel bits, but does not allow separate mapping of two user bytes to two sequences of 9 channel bits.

To achieve the trellis stationary in time, the input data bits are separated into two 8-bit groups (bytes). The odd and even codewords are encoded separately, using sliding block technique.

The sequences of four consecutive ones are not allowed and the sequences of three consecutive ones are allowed on even bit positions in odd codewords and odd bit positions in even codewords as shown in Figure 2.

The new code increases the number of available codewords by introducing the correlation between odd and even bytes. The most common errors are eliminated by applying the following rules to odd and even codewords:

- (i) No sequence of four consecutive ones is allowed,
- (ii) Sequences of three consecutive transitions can begin only at even bit positions in odd codewords, i.e. 2nd, 4th, 6th, 8th, or at 3rd and 5th bit positions in even codewords. In addition, after applying the rule (iv), tribits at positions 1 and 7 in even codewords may occur.
- (iii) The even byte encoder is constructed in such a way that it allows codewords starting with 1100x, but eliminates codewords starting with 1101x, in addition to 1110x (rule (ii)), 1111x (rule (i)). It also allows codewords ending with x0011, but eliminates those ending with x1011 in addition to x0111 (rule (ii)), x1111 (rule (i)).
- (iv) When an even codeword starts with 11x and preceding odd codeword ends with x1, bit 9 of odd codeword is changed from 1 to 0 and bit at position 3 in even codeword is changed from 0 to 1. When an even codeword ends with x11 and succeeding odd codeword starts with 1x, the bit 1 of odd codeword is changed from 1 to 0 and bit at position 7 in even codeword is changed from 0 to 1.

Rule (iv) states that in cases when an odd codeword ends with x1, the even codeword cannot start with 11x. Or, in other words, if the even code word is encoded first, and if it starts with 11x, the previous odd code word cannot end with xx1 (last bit has to be 0). Symmetrically, if an even codeword ends with x11, the following odd codeword has to start with 0x.

Before applying the rule (v) bits 3 and 7 in even codewords will never already be a 1, because codewords starting with 111x or ending with x111 are not allowed (rules (ii) and (iii)). This change in even codewords will not cause the existence of quadbit, because of rule (iii).

After initial encoding, the codewords are searched for tribit position violations at the boundaries of odd and even codewords. If a tribit at an invalid position is detected, it is remapped to a valid position, applying rule (iv), as illustrated in Figure 3. In the decoding process, if tribits at positions 1 and 7 in even codewords are detected, they are changed to a dibit, and corresponding bits in adjacent odd codewords are changed from 0 to 1.

The code proposed here requires different encoders for odd and even codewords. The encoding of the even bytes is done independently, while the encoding of the odd bytes depends on the encoding of the previous and the following even bytes.

By using this encoding scheme, the incoming even bytes will be encoded first, as shown in Figure 5. According to the result of encoding of even bytes, odd byte will be encoded. If the previous even codeword ends with the pattern ...xxx11 the first bit of the odd codeword will be forced to 0, producing the codeword 0xxxxxxx. If the next codeword starts with 1xxxxxxx, the last bit of the odd codeword will be forced to 0 as described in rule (v). Among total of 317 possible odd codewords, patterns are distributed as shown in Table 2. The codeword distribution allows this mapping, because there is more than 64 codewords with 0xxxxxxx0, 0xxxxxxx1 and 1xxxxxxx0 patterns.

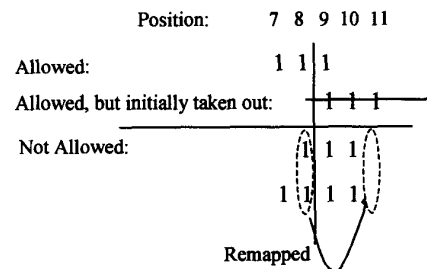


Figure 3: Resolving boundary tribits and quadbits.

Table 2: Pattern distribution in odd codewords.

Pattern	Count
0xxxxxxx0	100 words
0xxxxxxx1	78 words
1xxxxxxx0	78 words
1xxxxxxx1	61 words

III. ENCODER/DECODER DESIGN

The information about the change in the odd codeword will be stored in even codewords, by using the redundancy in the code. If the starting bit is forced to 0, the information will be stored in bit 7 in the previous even codeword, resulting in the user byte being re-mapped from codeword xxxxxx011 to xxxxxx111. Also the change in the last bit will be stored in the following even codeword by changing the bit 3 from 0 to 1. By allowing this redundancy, there exist 263 code words that satisfy constraints for the even bytes.

Since the odd and even bytes are, in general, encoded independently, the code will only allow two byte error bursts. The only exception is in the case of the even codeword of type 1100x0011, when an error event of length 5, hitting bits 3-7 would cause a 3-byte error. However, the probability of the error event of length 5, is low, since the dominant error events in trellis coded E²PR4 channels are $E_x = \pm 1$ and $E_x = (1, -1)$ have the error lengths less than 5.

The resulting stationary trellis for this code is shown in Figure 4. The trellis does not change in time and two states, 0101 and 1010, are permanently eliminated in every other step.

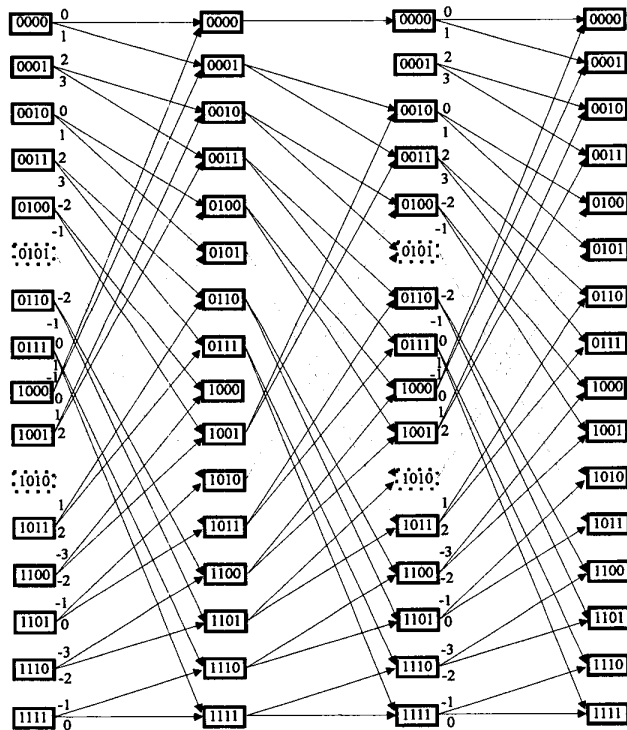


Figure 4: E²PR4 trellis for the proposed code.

The subset of desired codewords is selected from all the possible codewords by eliminating the quasi-catastrophic sequences and sequences that distract the variable-gain amplifier loop in even set. The maximum zero runlength even set is limited to 7 by eliminating the additional 3 codewords.

The zero runlength is limited to three in the odd set. This results in total maximum zero runlength for the code of 10.

The user data is mapped into codewords by using partitioning. The encoder block diagram is shown in Figure 5.

This encoding scheme detects tribits at not-allowed positions by using a simple 3-input AND gate, and upon detection relocates them by OR-ing bits 3 (or 7) in even codeword and inverting the last (or first) bit in odd codeword. The codewords are decoded in the opposite way.

IV. CONCLUSION

The new trellis code that eliminates the most common errors in E²PR4 channels, while keeping the two-step detector implementation stationary in time is proposed. The new code, proposed here has 8/9 code rate. It has maximum zero runlength of 10, with only two byte error propagation. The new trellis code eliminates two states in the every second step of the E²PR4 trellis, requiring 14-state two-step sequence detector.

REFERENCES

- [1] R. Karabed, N. Nazari, "Analysis of Error Sequences for PRML and EPRML Signaling Performed over Lorenzian Channel," *Proc. Globecom '96*, pp. 368-373, London, UK, Nov. 1996.
- [2] R. Behrens, A. Armstrong, "An Advanced Read/Write Channel for Magnetic Disk Storage," *Proc. 26th Asilomar Conference on Signals, Systems and Computers*, pp. 956-960, Pacific Grove, CA, Oct. 1992.
- [3] R. Karabed, P. Siegel, "Coding for higher order Partial Response Channels," *Proc. 1995 SPIE International Symposium on Voice, Video and Data Communications*, vol. 2605, pp. 115-126, Philadelphia, PA, Oct. 1995.
- [4] J. Moon, B. Brickner, "Maximum Transition Run Codes for Data Storage Systems," *IEEE Transactions on Magnetics*, vol. 32, no. 5, Sept. 1996.
- [5] Z.A. Keim, S.G. McCarthy, "Experimental Performance Comparison of FTDS/DFE Detectors: 8/9 (0,k) vs. 4/5 MTR Codes," *Digest of IEEE International Magnetics Conf. InterMag'97*, pp. BS-04, New Orleans, April 1-4, 1997.
- [6] W.G. Bliss, "An 8/9 Rate Time-Varying Trellis Code for High Density Magnetic Recording," *Digest of IEEE International Magnetics Conf. InterMag'97*, pp. BS-09, New Orleans, April, 1997.
- [7] B.E. Moision, P.H. Siegel, E. Soljanin, "Distance-Enhancing Codes for Digital Recording," *IEEE Transactions on Magnetics*, vol. 34, no. 1, pp. 69-74, Jan. 1998.
- [8] S. A. Altekari, M. Berggren, B.E. Moision, P.H. Siegel, J.K. Wolf, "Error-Event Characterization on Partial-Response Channels," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 241-247, Jan. 1999.

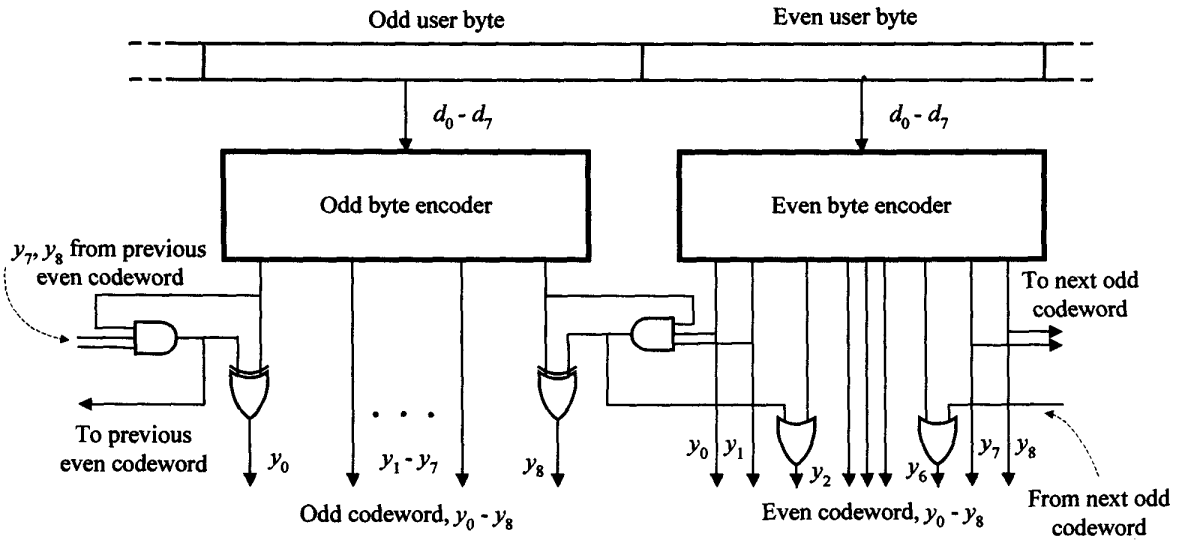


Figure 5: Encoder design for the new code.