Rate 8/9 Sliding Block Distance-Enhancing Code with Stationary Detector

Borivoje Nikolić, *Member, IEEE*, Michael Ming-Tak Leung, and Leo Ki-Chun Fu

***Abstract*—A new distance-enhancing code for partial-response magnetic recording channels eliminates most frequent errors, while keeping the two-step code trellis time invariant. Recently, published trellis codes either have lower code rates or result in time-varying trellises with a period of nine, thus requiring a higher complexity of detectors and code synchronization. The new code introduces dependency between code words in order to achieve the same coding constraints as the 8/9 time-varying maximum transition runlength (TMTR) code, with the same code rate, but resulting in a trellis that has a period of 2. This code has been applied to the E^{2}PR4 and a 32-state generalized partial response (GPR) ISI target. The resulting two-step trellises have 14 and 28 states, respectively. Coding gain is demonstrated for both targets in additive white Gaussian noise.**

***Index Terms*—Magnetic recording, partial response, trellis coding, Viterbi detection.**

I. INTRODUCTION

CONTEMPORARY magnetic recording channels use partial-response equalization with maximum likelihood (ML) detection to achieve a high density of recording bits. 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 coding constraints that avoid the most common error events. This was first observed by Behrens and Armstrong [2] on the E^{2}PR4 channel by using $d = 1$, rate 2/3 RLL code, with a detector that matched the resulting trellis. This code increased the minimum squared error distance in the E^{2}PR4 channel of six to ten, producing a 2.2 dB coding gain. Karabed and Siegel [4] have characterized low-distance error events in the E^{2}PR4 channel and have shown that higher rate codes that achieve the same coding gain can be implemented. Several codes that eliminate dominant error events by coding constraints have been proposed [4]–[11]. The major disadvantage of these codes is that they either have a low code rate and a long burst error propagation or require the implementation of a time-varying Viterbi detector. This emphasizes the need for a higher rate code that allows for a stationary detector and has short error propagation.

A new trellis code that reduces the complexity of the detector by keeping its two-step implementation time invariant is presented here. It has the same distance-enhancing properties as the earlier proposed rate 8/9 time-varying maximum transition runlength (TMTR) codes, but it does not require a time-varying detector, and it limits burst error propagation caused by the two most dominant error events to two bytes. Section II analyzes the distance properties of higher order partial response channels and reviews the design and properties of distance-enhancing codes for magnetic recording. Section III describes the design of a new distance-enhancing code, while its implementation issues are covered in Section IV. Section V discusses the application of the new code to the E^{2}PR4 trellis and to a modified 11-level, 32-state target. Section VI contains performance comparisons and Section VII concludes the paper.

II. DOMINANT ERROR EVENTS IN HIGHER ORDER PARTIAL RESPONSE CHANNELS

By examining and characterizing dominant errors in partial response channels, it is possible to construct codes that enhance the minimum error distance for the channel. The procedure of constructing such codes consists of [3], [11]

a) determining a set of input error sequences that correspond to most probable error events in a particular channel;
b) determining such a system of constrained sequences, that the difference between any two sequences in the set does not correspond to a minimum error distance set;
c) constructing an efficient and practical encoder and a decoder for this constraint;
d) designing a sequence detector that matches the channel and code constraints.

For a number of partial response channels, dominant error events have been characterized. Error events (sequences) can be defined as the difference between recorded and detected data

$$\text{User data:}\quad e_z(D) = z(D) - z'(D). \tag{1}$$

$$\text{Channel input:}\quad e_x(D) = x(D) - x'(D). \tag{2}$$

$$\text{Channel output:}\quad e_y(D) = y(D) - y'(D). \tag{3}$$

The channel output error sequence is equal to the input error sequence, convolved with the channel transfer function

$$e_y(D) = h(D)e_x(D). \tag{4}$$

The error event distance is defined as the Euclidean norm

$$d^2(e_x) = ||e_y(D)||^2. \tag{5}$$

Manuscript received February 18, 2000; revised December 7, 2000.

B. Nikolic was with the Texas Instruments Storage Products Group, San Jose, CA 95131 USA. He is now with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770 USA (e-mail: bora@eecs.berkeley.edu).

M. Leung and L. Fu were with the Texas Instruments Storage Products Group, San Jose, CA 95131 USA. They are now with Tripath Technology, Santa Clara, CA 95054 USA.

Publisher Item Identifier S 0018-9464(01)03496-3.

TABLE I
CLOSED ERROR EVENTS IN E^2PR4.
SEQUENCES IN BRACKETS CAN OCCUR ONE OR MORE TIMES

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

For the binary signaling alphabet, consisting of symbols 0 and 1, the error bits could be −1, 0, and +1, frequently shorthand noted as "−," "0," and "+." Using the results from [13], the list of short-distance closed error events for the E^2PR4 channel is shown in Table I. The sequences inside the brackets (such as [+, −]) can be repeated one or more times in the error sequence.

The closed errors of type 3 and 4 are caused by the detector's failure to recognize a sequence of four NRZI ones, which is frequently referred to as a quadbit. Therefore, elimination of quadbits in the code eliminates these types of errors. The errors of type 1 and 2 are due either to the existence of a quadbit or to the detector's inability to determine the position of a sequence of three NRZI ones (which is frequently referred to as a tribit), effectively shifting the start of the tribit by one. By limiting the start of a tribit to only odd or only even positions in the codewords, these errors can be eliminated. After applying these constraints, the single-bit error event (type 5) becomes dominant. Elimination of the dominant error events from Table I by coding constraints increases the minimum squared error distance in the E^2PR4 channels from six to ten.

Recently, several codes that eliminate the most common error events by coding constraints have been proposed for partial response signaling.

1) Maximum Transition Runlength Codes: Behrens and Armstrong first observed that application of a rate 2/3, $(d, k) =$ (1, 7) code on the E^2PR4 channel results in an increase of 2.2 dB in minimum squared error distance [2]. This coding gain could be obtained by using a detector matched to this code that conforms to the $d = 1$ constraint, thus achieving the matched filter bound (MFB) of the channel. The capacity of the $d = 1$ constraint is low, $C =$ 0.6942 [12], allowing a maximum code rate of $R =$ 2/3.

The rate 4/5 sliding block code [4] provided the same coding gain with a higher, but still small code rate. Maximum transition runlength (MTR) codes, proposed in [5], eliminate all possible sequences of three or more transitions, which results in a low capacity of the code, up to rate 8:10. Other codes such as [7] and [8] allow for sequences of three NRZ transitions (NRZI tribits) at specific locations within each codeword that start at either odd- or even-numbered bits but not at both.

The rate 8/9 code, proposed in [9], allowed for tribits at different positions within a codeword, which result in a mapping of 16 user bits to 18 bits of channel data. The code has a burst error propagation of four bytes, which is high for the three-way interleaving ECC configuration commonly employed in disk drives.

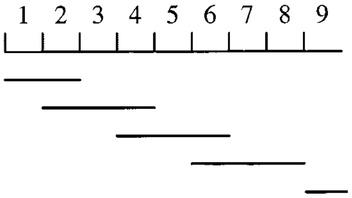


Fig. 1. Allowed tribits in TMTR code.

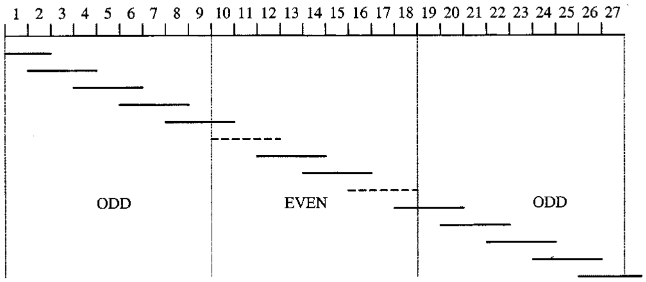


Fig. 2. Rate 16:18 TMTR block code.

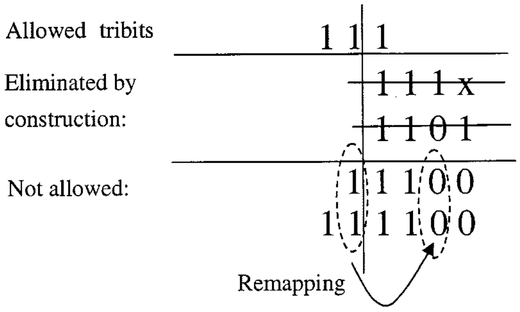


Fig. 3. Resolving boundary quadbits and tribits.

2) Time-Varying Maximum Transition Runlength (TMTR) Codes: Bliss [9], Karabed, Siegel, and Šoljanin [11] and Fitzpatrick and Modlin [15] introduced an 8/9 block code that independently maps eight input bits into nine code bits with time-varying MTR constraints. This code limits the practical propagation of error to two user bytes. The constraint on the position of the tribits is illustrated in Fig. 1. The tribits are shown as horizontal lines of three bits in length. They can start at positions 2, 4, and 6. The tribit that starts at position 9 ends at positions 1 and 2 of the next codeword.

In order to freely concatenate the codewords, a discontinuity is introduced in allowed tribits at the boundaries. Allowed tribits in the 8/9 TMTR code limit 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 the 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 the E^2PR4 trellis, but the trellis changes every nine cycles to conform to the code constraints. This requires that the detector be aligned to accommodate these changes with a period of nine. This implementation of a TMTR block code results in maximum zero runlengths of $k = 11$.

The major disadvantage of this code is that its restriction on allowed tribits requires an implementation of the Viterbi detector that is variable in time and therefore significantly increases the detector's complexity and reduces its speed.

Higher rate codes can be achieved [15] by relaxing the MTR constraints. The paper [15] reports a construction of a rate 9/10 code, but this code also requires a time-varying detector. [16] describes a Turbo-E^{2}PR4 channel which uses a 16/17 rate quasi-MTR (QMTR) code. The code eliminates all the quadbits but leaves all the tribits, which results in a higher code rate. The set of dominant errors is limited and these errors are removed by a postprocessor, which follows the detector.

III. NEW RATE 8/9 TRELLIS CODE

The new code eliminates all small distance error events similarly to the 8/9 TMTR code. By not allowing quadbits in the codewords and by limiting the occurrence of tribits to certain bit positions inside the codeword.

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

A. Sliding-Block Code Construction

To achieve a trellis stationary in time, short error propagation, and simple encoding/decoding logic, the input data bits are separated into two 8-bit groups (bytes). Initially, the odd and even codewords are encoded separately, using the sliding block technique.

The sequences of four consecutive ones are not allowed, and the sequences of three consecutive ones are allowed at even bit positions in odd codewords and in odd bit positions in even codewords, as illustrated in Fig. 2.

The new code increases the number of available codewords by introducing a correlation between odd and even bytes. The code-construction procedure consists of initial encoding in rules 1–3, and remapping of the invalid tribits in rule 4.

1) No sequence of four consecutive NRZI ones (quadbits) is allowed.
2) Sequences of three consecutive NRZ transitions (NRZI ones) can begin only at second, fourth, sixth, and eighth bit positions in odd codewords or at third and fifth bit positions in even codewords.
3) Even codewords starting with 1101 and ending with 1011 are deleted. They are reserved for later remapping of invalid tribits.
4) When an initially encoded even codeword starts with 11 and the preceding odd codeword ends with 1, bit 9 of the odd codeword is changed from 1 to 0 and the bit at position 3 in the even codeword is changed from 0 to 1. When an even codeword ends with 11 and the succeeding odd codeword starts with 1, the first bit of the odd codeword is changed from 1 to 0 and the bit at position 7 in the even codeword is changed from 0 to 1.

TABLE II
ERROR EVENT STATISTICS FOR AWGN/LORENTZIAN CHANNEL AT A USER DENSITY OF 3.0

Error Event Length	Event Count	Error Event %
2 bit	585	62.97%
3 bit	279	30.03%
5 bit	57	6.14%
6 bit	1	0.11%
7 bit	2	0.22%
8 bit	1	0.32%
9 bit	0	0.00%
10 bit	0	0.00%
> 11 bit	2	0.22%

TABLE III
THREE-BYTE BURST ERROR POSSIBILITY FOR ERROR EVENT LENGTH OF FIVE, SIX, AND EIGHT

Error Event Length	Error Event %	Possible Position within Even Codewords	Average Number of Affected Codewords	3 Byte Error Percentage
5 bit	6.14%	1	2.0	0.0026%
6 bit	0.11%	2	9.5	0.0004%
7 bit	0.22%	3	30.3	0.0043%
8 bit	0.32%	4	55.0	0.0154%
Total				0.023%

After the initial encoding, using rules 1–3, some tribits may appear at invalid positions, and they then have to be remapped to valid positions. Before rule 4 is applied, bits 3 and 7 in the even codewords will not be equal to 1, because codewords starting or ending with a tribit, quadbit, or 1101 (1011) are not allowed by rules 1, 2, and 3. This prevents the remapping in even codewords from creating a quadbit.

Codewords are searched for tribit position violations at the boundaries between odd and even codewords after the initial encoding. If a tribit is detected at an invalid position, it is remapped to a valid position by applying rule 4, as illustrated in Fig. 3. In the decoding process, if tribits are detected at positions 1 and 7 in even codewords, they are changed to a dibit (a sequence of two consecutive NRZI ones), and the corresponding bits in adjacent odd codewords are changed from 0 to 1.

The code presented in this paper has a rate of 8/9, a maximum zero runlength of ten, and the same distance properties as the rate 8/9 TMTR code. The resulting trellis has a period

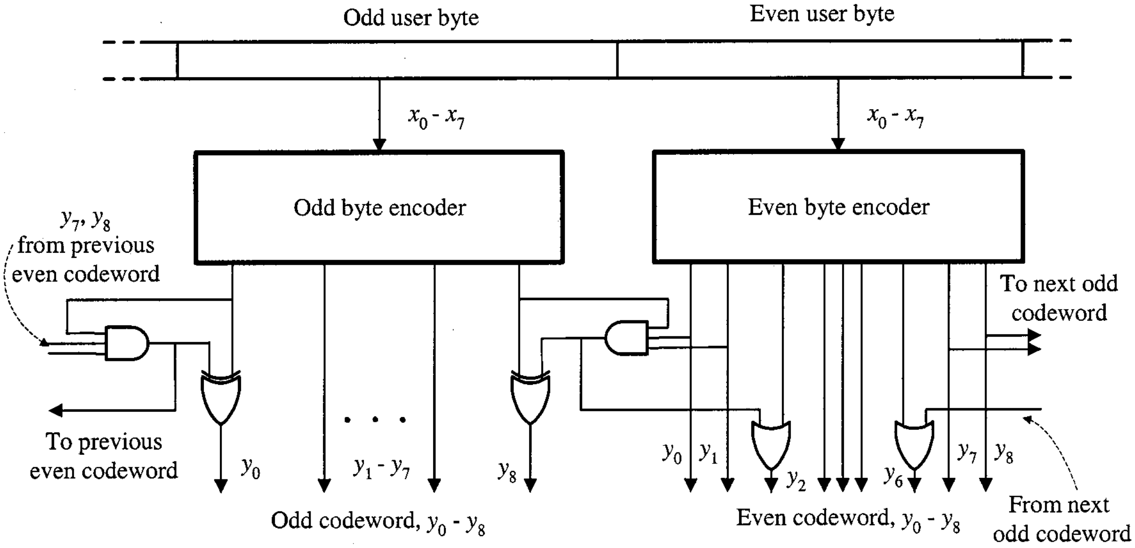


Fig. 4. Encoder design.

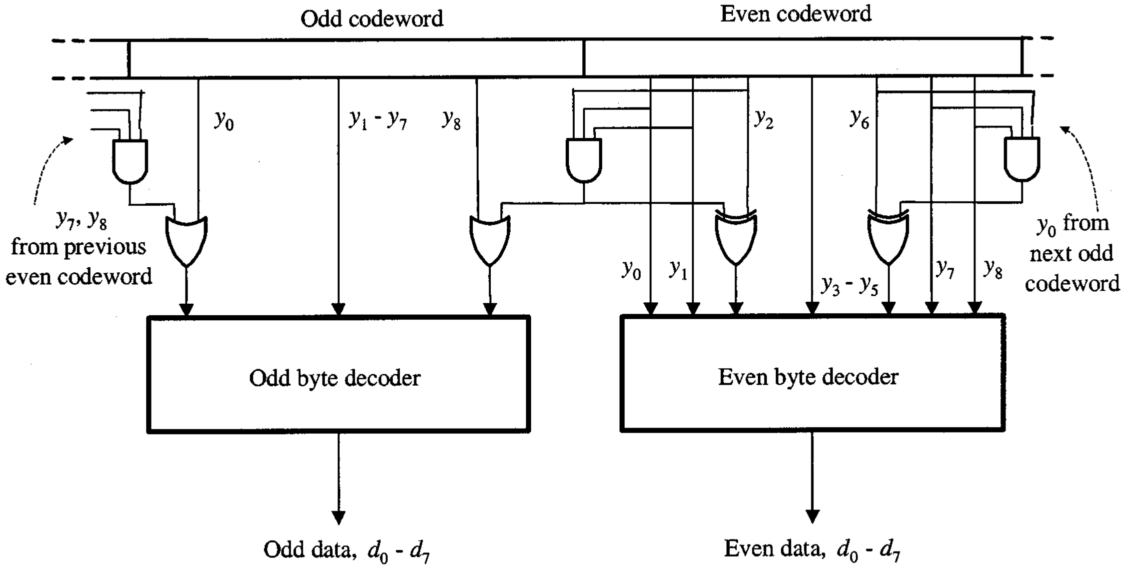


Fig. 5. Decoder design.

of two rather than nine. The application of this code keeps a two-step implementation of the detector time-invariant. For example, when this code is applied to the E^2PR4 trellis, only a 14-state two-step stationary Viterbi detector is required, versus a 16-state time-varying detector for the TMTR code. A time-invariant trellis with some permanently eliminated states would require a lower complexity sequence detector to match the code constraints.

The reduced complexity of the detector should increase its speed and power performance. Moreover, the detector synchronization is eased. With a TMTR code, the Viterbi detector is required to synchronize in one out of nine cycles and therefore must rely on the synchronization pattern in the read cycle. With added latency in the Viterbi detector, the resulting system usually must pass the first few bytes of data in a regular, nontrellis mode. In many cases, the first few data samples tend to have more distortion from timing recovery and automatic gain control (AGC) due to the 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 preamble pattern, and the detector could be aligned to receive the first data samples or even the synchronization byte in trellis mode. This is an important advantage over the TMTR code, since no particular data sequence must be used to align the detector.

The application of this code is not limited to ML sequence detectors. It can also be used in other signal processing architectures [e.g., in a fixed-delay-tree search (FDTS)] to allow for stationary implementation of an interleaved two-step scheme in order to achieve a high data rate).

IV. IMPLEMENTATION ISSUES

A. Error Propagation Properties

Since the odd and even bytes are, in general, encoded independently, the code will only allow for a two-byte error propa-

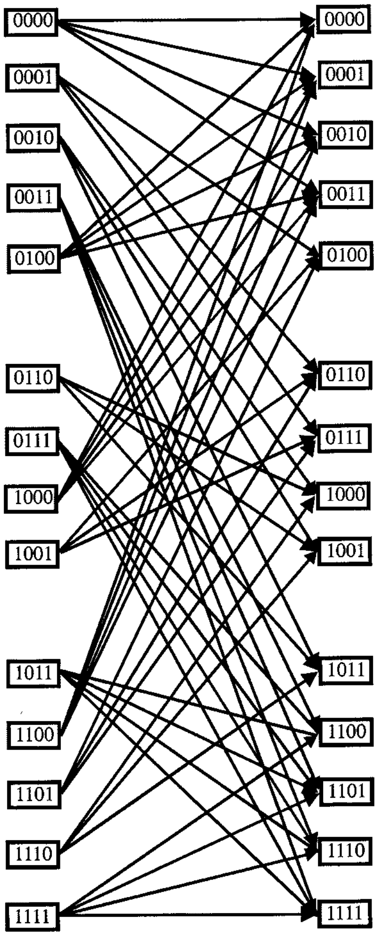
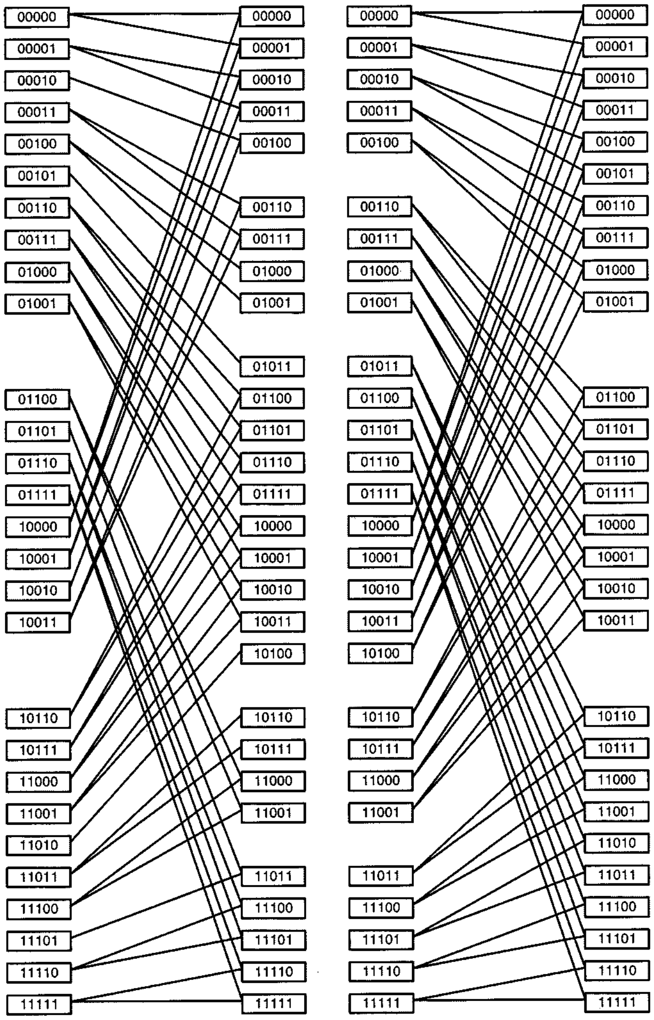


Fig. 6. E^2PR4 trellis for the new code.



a) b)

Fig. 7. Two possible 32-state trellises for the new code.

gation. The exception occurs when error events of length five or larger hit the even codeword, causing adjacent bytes to be corrupted due to the replacement rules. Table II shows the error event statistics for a Lorentzian channel with white noise at user density of 3.0 using the proposed trellis code. One billion channel bits are run at the channel BER of around 10^{-6}. Error event bit lengths are listed in the NRZ domain. With the normal $(1+D)$ precoder for trellis codes, NRZI dominant error events of [1], $[1\ -1]$ and $[1\ 0\ 0\ 1]$, will generate NRZ error events of length two, three, and five, respectively. As shown in Table II, the two most dominant error events are two and three bits long, and the probability of a 5-bit error event is around 6.1%. No 3-byte error propagation is observed. To evaluate the impact of error propagation on the proposed trellis code, a simple analysis is carried out. For error events with length larger than 11, these patterns can cause 3-byte errors in any code. The additional 3-byte burst error caused by the replacement comes from error events of length five to ten. For an error event of length five, a 3-byte error will only occur for codewords 1100x0011 being hit at bits three through seven, as the only valid 11xxxxx11 codeword patterns are 1100x0011. The probability of occurrence of a 3-byte burst error caused by a 5-bit error event is

$$\begin{aligned}\text{3-byte error\%} =\ &(\text{probability of a 5-bit error event})\\ &\times (\text{probability of even codewords})\\ &\times (\text{number of 1100x0011 patterns}\\ &\quad \text{out of 256 valid codewords})\\ &\times (\text{bit position 3 out of 9 bit positions})\\ =\ &6.1\% \times \tfrac{1}{2} \times 2/256 \times 1/9 = 0.0027\%.\end{aligned}$$

Similar considerations are carried out for length six, seven, and eight error events, and Table III shows both the number of possible positions within the even codeword and the average number of codewords (out of 256) that will generate 3-byte errors when being hit by error events of various lengths. The overall additional 3-byte burst error percentage is 0.023%. Thus, the code practically limits error propagation to two bytes.

B. Encoder/Decoder Design

A subset of desired codewords is selected from the set of all possible codewords by eliminating the quasi-catastrophic sequences (codewords 001 100 110 and 011 001 100) and the sequences that could distract the variable-gain amplifier loop (codeword 011 011 011) in the even set. The maximum zero runlength in even codewords is limited to seven through elimination

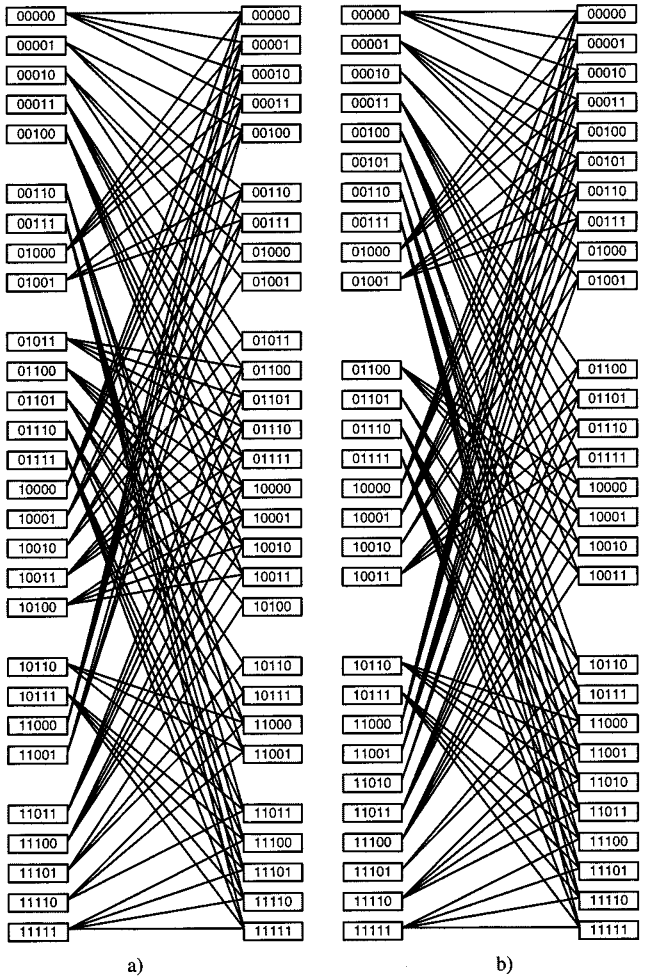


Fig. 8. Two possible two-step 32-state trellises for the new code.

of the additional three codewords (000000000, 000000001, and 100 000 000).

The zero runlength in the odd set is limited to three, which results in the total maximum zero runlength for the code of ten.

The user data is mapped into codewords by partitioning [18]. The encoder block diagram is shown in Fig. 4 [17].

This encoding scheme detects tribits at dis-allowed positions by using a simple three-input AND gate. Upon detection, it relocates them by OR-ing bits three (or seven) in the even codeword and by inverting the last (or the first) bit in the odd codeword. The codewords are decoded in the opposite way, as shown in Fig. 5.

V. E^{2}PR4 AND 32-STATE TRELLISES

A. E^{2}PR4 Trellis

The resulting two-step E^{2}PR4 trellis for this code is shown in Fig. 6. The two-step trellis does not change in time and two states, 0101 and 1010, are permanently eliminated in every other step. The detector matched to this code is equivalent to the detector required for the rate 16:18 block code [9].

The two-step (one step lookahead) detector [19] matched to the code requires only 14 add-compare-select units for its implementation. Out of the 14 units, eight are four-way and six are three-way as shown in Fig. 6. This results in over 30% savings in complexity over the full detector implementation. Reduced number of timing-critical four-way units and shorter interconnect length can be used for improvements in the speed performance of the detector.

B. Modified Target Trellis

At higher user densities, channel bit-error rate (BER) can be improved by employing a 32-state Viterbi detector that corresponds to targets better matched to the channel response. The partial response of $(1-D)\cdot(1+D)^2\cdot(2+D+D^2)$ is selected to demonstrate the application of the code to a 32-state target. The selected response matches the Lorentzian channel at the user density of 3.0. The resulting target has 11 discrete signal levels [−5, −4, −3, −2, −1, 0, 1, 2, 3, 4, 5] and requires a 32-state Viterbi detector. The dominant error events for this channel are $[+\,-\,+]$ with squared distance of ten, $[+\,-]$ with squared distance of 20, and a single bit error event with squared distance of 22. Thus, the coding gain of using the trellis code with the 32-state response is 3.4 dB.

Since the new code does not allow for four consecutive ones, two states, 01010 and 10101, are removed in every step. Additionally, since three consecutive transitions can start only at even or at odd bit positions but not at both, either one of the pairs of states 00 101 and 11 010 or 01 011 and 10 100 will not exist. The two possible trellis configurations of the resulting 28-state trellis are shown in Fig. 7. Applying the one-step lookahead to this trellis results in a 28-state two-step trellis from Fig. 8. It is not time varying, and it requires 16 four-way ACS units and 12 three-way ACS units.

VI. COMPARISON RESULTS

To verify the performance, several systems are compared at user densities of 3.0 for a Lorentzian channel. Additive white Gaussian noise is used in the simulation, and SNR is defined as the zero-to-peak amplitude of the input signal over the noise power within two times the Nyquist bandwidth. Different code rate systems have a slightly different Nyquist bandwidth. For example, a lower code rate system implies a higher noise bandwidth- the code rate difference is therefore automatically adjusted in the SNR in order to have a fair comparison between different code rate systems. BER curves are generated with 100 errors for each data point in the graph (i.e., 10^8 bits are exercised for the BER at 10^{-6}). The simulated systems are: 1) E^{2}PR4 with rate16/17 RLL code; 2) E^{2}PR4 with new 8/9 code; 3) 32-state target with 16/17 RLL code; and 4) 32-state target with 8/9 code.

Comparison results are shown in Fig. 9. The new rate 8/9 code has a 1.75 dB advantage over the RLL-coded E^{2}PR4 system with code-rate loss accounted for. It also improves the performance of the 32-state target by 1 dB when compared to the RLL coded system.

VII. CONCLUSION

Proposed is a new trellis code that eliminates the most common errors in higher order partial response recording channels while keeping the two-step detector implementation

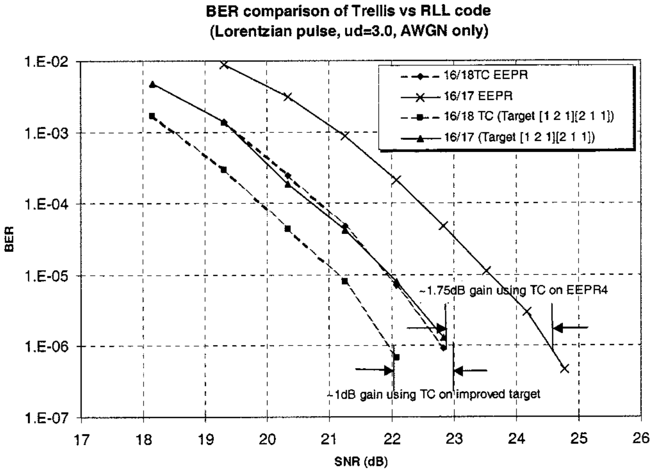


Fig. 9. Comparison of BER versus SNR for trellis and RLL coded E^2PR4 and [1 2 1] [2 1 1] channels.

stationary in time. The new code has a rate 8/9. It has a maximum zero runlength of ten, with only two-byte practical error propagation. The new trellis code was applied to two advanced recording channels. It eliminates two states in every other step of the E^2PR4 trellis, requiring a 14-state two-step sequence detector. When applied to an eleven-level, 32-state target, it eliminates four states in the two-step trellis. Coding gains were demonstrated in both cases with the Lorentzian channel under additive white Gaussian noise. Application of this code is not limited to ML detectors. It also applies to DFE/FDTS systems.

REFERENCES

[1] R. Karabed and N. Nazari, "Analysis of error sequences for PRML and EPRML signaling performed over Lorenzian channel," in *Proc. Globecom '96*, London, UK, Nov. 1996, pp. 368–373.

[2] R. Behrens and A. Armstrong, "An advanced read/write channel for magnetic disk storage," in *Proc. 26th Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Oct. 1992, pp. 956–960.

[3] K. A. Schouhamer Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2260–2299, Oct. 1998.

[4] R. Karabed and P. Siegel, "Coding for higher order partial response channels," in *Proc. 1995 SPIE Int. Symp. Voice, Video and Data Communications*, vol. 2605, Philadelphia, PA, Oct. 1995, pp. 115–126.

[5] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Trans. Magn.*, vol. 32, pp. 3992–3994, Sept. 1996.

[6] B. Brickner and J. Moon, "Design of a rate 6/7 maximum transition run code," *IEEE Trans. Magn.*, vol. 33, pp. 2749–2751, Sept. 1997.

[7] Z. A. Keirn and S. G. McCarthy, "Experimental performance comparison of FTDS/DFE detectors: 8/9 (0, k) vs. 4/5 MTR codes," *IEEE Trans. Magn.*, vol. 33, pp. 2731–2733, Sept. 1997.

[8] S. G. McCarthy and Z. A. Keirn, "Distance enhancing codes for E^2PRML: Performance comparison using spinstand data," *IEEE Trans. Magn.*, vol. 33, pp. 2734–2736, Sept. 1997.

[9] W. G. Bliss, "An 8/9 rate time-varying trellis code for high density magnetic recording," *IEEE Trans. Magn.*, vol. 33, pp. 2746–2748, Sept. 1997.

[10] B. E. Moision, P. H. Siegel, and E. Soljanin, "Distance-enhancing codes for digital recording," *IEEE Trans. Magn.*, vol. 34, pp. 69–74, Jan. 1998.

[11] R. Karabed, P. H. Siegel, and E. Soljanin, "Constrained coding for binary channels with high intersymbol interference," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1777–1797, Sept. 1999.

[12] B. E. Moision, A. Orlitsky, and P. H. Siegel, "Bounds on the rate of codes which forbid specified difference sequences," in *Proc. 1999 Global Telecommunications Conf. (GLOBECOMM'99)*, Rio de Janeiro, Brazil, Dec. 5–9, 1999, pp. 878–882.

[13] S. A. Altekar, M. Berggren, B. E. Moision, P. H. Siegel, and J. K. Wolf, "Error-event characterization on partial-response channels," *IEEE Trans. Inform. Theory*, vol. 45, pp. 241–247, Jan. 1999.

[14] A. D. Weathers, S. A. Altekar, and J. K. Wolf, "Distance spectra for PRML channels," *IEEE Trans. Magn.*, vol. 33, pp. 2809–2811, Sept. 1997.

[15] K. Knudson-Fitzpatrick and C. S. Modlin, "Time-varying MTR codes for high density magnetic recording," in *1997 Global Telecomm. Conf. (GLOBECOM'97), Conf. Record*, Phoenix, AZ, Nov. 3–8, 1997, pp. 1250–1253.

[16] T. Nishiya, K. Tsukano, T. Hirai, S. Mita, and T. Nara, "Rate 16/17 maximum transition run (3; 11) code on an EEPRML channel with an error-correcting postprocessor," *IEEE Trans. Magn.*, vol. 35, pp. 4378–4386, Sept. 1999.

[17] B. Nikolić, M. Leung, and L. Fu, "A rate 8/9 sliding block code with stationary detector for magnetic recording," in *Proc. 1999 Int. Conf. Communication, ICC'99*, vol. 3, Vancouver, BC, Canada, June 6–10, 1999, pp. 1653–1657.

[18] B. Nikolić and M. Leung, "Sliding block (rate 8/9) trellis code for magnetic recording," U.S. patent 6 081 210, June 27, 2000.

[19] P. Black and T. Meng, "A 140 MB/s 32-state radix-4 Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1877–1885, Dec. 1992.

Borivoje Nikolić (S'93–M'99) received the Dipl.Ing. and M.Sc. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, and the Ph.D. degree from University of California, Davis, in 1992, 1994, and 1999. respectively.

He was on the Faculty of the University of Belgrade from 1992. to 1996. During his graduate studies, he spent two years with Silicon Systems, Inc, Texas Instruments Storage Products Group, San Jose, CA, working on disk drive signal processing electronics. In 1999, he joined the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, where he is now an Assistant Professor. His research interests include high-speed and low-power digital integrated circuits and VLSI implementation of communications and signal processing algorithms.

Dr. Nikolić received the College of Engineering best doctoral dissertation award and the Anil K. Jain prize for the best doctoral dissertation in Electrical and Computer Engineering from the University of California, Davis, in 1999, as well as the City of Belgrade award for the best Diploma thesis in 1992.

Michael Ming-Tak Leung received the B.Eng. degree in electronic and communication engineering from the University of Birmingham, Birmingham, U.K., in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1988 and 1993, respectively.

He was the Manager of the Advanced Channel Aarchitecture Group, Texas Instruments Storage Products Group, San Jose, CA, where he worked on the definition of signal processing architectures and implementation of advanced HDD storage channel ICs. He is currently the manager of the DSP Architecture Group, Tripath Technology, Santa Clara, CA, and is working on architecture of ADSL and optical transceiver IC products. His technical interests include communication channel theory, coding, DSP architecture and implementation, and high-speed CMOS digital design.

Leo Ki-Chun Fu received the B.S degree in electrical engineering and computer science from the University of California, Berkeley (UCB), in 1993.

From 1992 to 1993, he was a Research Assistant with the Center of X-ray Optics, Lawrence Berkeley Laboratory, UCB, where he conducted research on diffraction properties of X-ray through varies metal. In 1993, he joined the advanced channel architecture group in Silicon Systems, Inc., San Jose, CA, and later joined Storage Products Group, Texas Instruments, where he worked as a System Engineer on signal processing architecture and modeling DSP algorithms on advanced HDD storage channel ICs. His technical interests include coding, and DSP architecture and implementation. In 2000, he joined the DSP Architecture Group, Tripath Technology, Santa Clara, CA, and is currently working on architecture of ADSL and optical transceiver IC products.