Reduced Complexity Sequence Detection for High-Order Partial Response Channels

Michael Leung, Borivoje Nikolić, *Member, IEEE*, Leo Ki-Chun Fu, and Taehyun Jeon

***Abstract*—Detector hardware complexity of high-order partial response magnetic read channels is a major obstacle to high data rate operation and reduced area and power consumption. The method presented here reduces the complexity of single-step and two-step implementations of the Viterbi detector by applying a distance-enhancing code that eliminates some states from the code trellis. The complexity of the detector is further reduced by eliminating less-probable branches from the trellis. This is accomplished by a simple control mechanism that uses the signs of the consecutive input samples. The reduced set of add-compare-select (ACS) units is dynamically assigned to the detector states, decreasing the complexity of the Viterbi detector by roughly 50%. This method is demonstrated on high-order partial response systems with the E^{2}PR4 target and an 11-level/32-state target. The simulation results show negligible bit error rate (BER) degradation for signal-to-noise ratios in the range of operation of contemporary disk drive read channels.**

***Index Terms*—Magnetic recording, partial response signalling, recording codes, Viterbi detection.**

I. Introduction

The LEVEL OF intersymbol interference between neighboring recorded bits in magnetic recording channels increases with recording density. For such channels, maximum-likelihood (ML) sequence detection is proven to be the optimum decoding method [1]. The asymptotically optimal method of implementing the ML detection technique uses the Viterbi algorithm [2], [3]. A practical technique to employ ML detection is to equalize the channel to partial response (PR) targets [4], [5] of the form

$$h(D) = (1-D)(a_0 + a_1 D + a_2 D^2 + \cdots + a_N D^N), \quad (1)$$

where D denotes a unit sample delay.

A frequently used set of targets is represented by a class of polynomials [6]

$$h(D) = (1-D)(1+D)^N, \qquad N \geq 1. \quad (2)$$

When $N = 1$, the channel is known as partial response class-4 (PR4); $N = 2$ corresponds to extended PR4 (EPR4); $N = 3$ is usually denoted as E^{2}PR4.

The Viterbi detector for a partial response channel of degree N requires a 2^{N+1}-state implementation. It is generally realized as an array of 2^{N+1} single-step add-compare-select (ACS) units. As the recording density increases, the number of terms in the suitable PR target will grow to avoid excessive noise enhancement. Hardware complexity will, thus, also increase because the complexity of the Viterbi detector will double for every additional allowed bit of interference.

The method described in this paper significantly reduces the complexity of the detector by eliminating the less-probable paths in the trellis. It is general and can be applied to any type of coding, to a variety of PR channels, as well as to both single-step and two-step detectors. However, the resulting reduction in complexity depends on the code used. The method that we developed is here applied to two high-order PR channels, which results in roughly 50% reduced complexity.

This paper is organized into eight sections, and this introduction serves as Section I. Section II describes recent coding and detection techniques employed in practical magnetic recording systems. In Section III, the new trellis code is introduced and the application of the new code to reduce the detector complexity is discussed. Section IV describes the new technique of trellis reduction based on pruning the unlikely branches. Section V shows the application of the trellis reduction to the E^{2}PR4 channel, and Section VI deals with the reduced complexity detection in other high-order PR channels. Section VII presents a complexity estimation, and Section VIII summarizes the paper.

II. Techniques in Magnetic Channels

A. Two-Step Viterbi Detector Implementation

The performance of the Viterbi detector is crucial to the performance of the entire magnetic read channel. Design of Viterbi detectors for high data rate operation is a well-known challenge because a tight feedback loop in the ACS unit makes pipelining difficult. Applying a one-step look-ahead scheme to the ACS unit roughly doubles the throughput of the system [7], resulting in a two-step (radix-4) implementation. In single-step implementations, the ACS units are two-way, and in the case of two-step implementations, they are four-way. Each single-step ACS unit consists of two adders and one subtractor, whereas a four-way ACS unit consists of four adders and six parallel subtractors. As shown in Table I, the overall complexity of a four-way ACS is about three times that of a two-way ACS, given that an adder is 1.5 times larger than a subtractor. The increase in complexity makes it difficult for a two-step Viterbi detector to meet the desired two times speed improvement, whereas the power and area of the implementation are increased. Reducing

Manuscript received March 1, 2000; revised August 15, 2000.

M. Leung and L. K.-C. Fu were with Texas Instruments, San Jose, CA USA. They are now with Tripath Technology, Santa Clara, CA 95054 USA (e-mail: mleung@tripath.com; lfu@tripath.com).

B. Nikolić is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770 USA (e-mail: bora@eecs.berkeley.edu).

T. Jeon is with Texas Instruments, Inc., San Jose, CA 95134 USA (e-mail: taehyun.jeon@ti.com).

Publisher Item Identifier S 0733-8716(01)01757-7.

TABLE I
SIZES OF DIFFERENT ACS UNITS

	Number		Size		Equivalent
ACS	Adder	Comparator	Adder	Comparator	Size
1-way	1	0	3	0	3
2-way	2	1	6	2	8
3-way	3	3	9	6	15
4-way	4	6	12	12	24

TABLE II
CLOSED ERROR EVENTS IN E^{2}PR4. 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

the complexity of the two-step detector allows for trading area and power for speed.

B. Time-Varying Maximum Transition Runlength (TMTR) Codes and Quasi-MTR (QMTR) Code

Recently, several trellis codes that eliminate the most common error events by using coding constraints have been proposed for PR signaling [8]–[18]. Most common errors in higher order PR channels are caused by sequences of three or more consecutive recorded transitions. Therefore, restricting the maximum transition runlength (MTR) in the recording code can eliminate dominant error events. The list of short distance error events for the E^{2}PR4 channel is summarized in Table II. Error events (sequences) can be defined as differences between recorded and detected data

$$\text{User data:} \quad ez(D) = z(D) - z'(D) \tag{3}$$

$$\text{Channel input:} \quad ex(D) = x(D) - x'(D) \tag{4}$$

$$\text{Channel output:} \quad ey(D) = y(D) - y'(D). \tag{5}$$

Bliss [12]–[14], Karabed *et al.* [11] and Knudsen–Fitzpatrick *et al.* [17] introduced at the same time a 8/9 block code that maps eight input bits into nine code bits with time-varying MTR (TMTR) constraints. 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, which significantly increases the detector's complexity and reduces the speed.

The TMTR block code is constructed by deleting all codewords that contain quadbits and by restricting tribits to certain positions [11]. Because the beginning of the tribit is allowed at positions 2, 4, 6, and 9, with the tribit starting at position 9 being wrapped up in the next word, this implementation results in a time-varying trellis. When the code is applied to an E^{2}PR4 trellis, the resulting detector still has 16 states. Two states, 0101 and 1010, have to be eliminated from the trellis in cycles 1, 3, 5, 7, and 8, and the trellis changes every nine cycles to conform to code constraints. The realignment of the detector is, thus, required every nine cycles to accommodate these changes. The TMTR block code results in maximum zero runlength of $k = 11$ [14]. Higher rate codes are possible by relaxing the MTR constraints. In [17], a construction of a rate 9/10 code, which also requires a time-varying detector, is reported.

To avoid code-rate loss in the TMTR code, a rate 16/17 quasi-MTR (QMTR) code was reported in [19]. The code eliminates quadbits but leaves all tribits, which results in a higher code rate. The application of the QMTR code in general does not increase the minimum error distance, but it reduces the number of possible error events. In the E^{2}PR4 channel, the QMTR eliminates the dominant error events of types 2, 3, and 4 from Table II, leaving only two remaining error patterns to be corrected by a postprocessor.

C. Turbo-Postprocessing and Parity Bits

An alternative way of coping with dominant error events uses a postprocessor to correct them. As originally proposed by Wood [20], Turbo-PRML enhances the performance of the PR4 channel. A Turbo-PRML detector consists of a conventional PR detector followed by a postprocessor controlled by an error filter block. The filter correlates the ideal PRML samples with actual equalized PRML samples. If the correlation exceeds a specified threshold value, the possible error is corrected in the postprocessor.

To enhance the performance of error postprocessing, parity bits can be appended to channel data blocks consisting of several codewords [21]. Usually, one or more parity bits can be used to identify occurrence of an error in a block. Error filters, matched to a selected set of dominant error events, are used to pinpoint the error location and to correct the error if the detected parity is incorrect.

D. Reduced Complexity Sequence Detector

It has been demonstrated that the sequence could be estimated using reduced-state estimation with a small loss in the error rate [22]. The performance loss is larger than 1 dB for the EPR4 channel and is unacceptable for the magnetic recording applications. Near-optimal performance was achieved in EPR4-equalized magnetic read channels, with significantly reduced computational requirements [23]. A method proposed in [24], based on the elimination of paths with higher error distances, resulted in a capability to share hardware between states of the detector.

The complexity of the Viterbi detector can be reduced by dynamically eliminating improbable branches from the trellis [23]. Shafiee and Moon proposed a method of reducing the detection complexity of the EPR4 equalized sequence by dividing the range of data into six "ambiguity zones." Ambiguity zone detection was first proposed by Kobayashi in 1971 [25]. Equalization targets in EPR4 are $+2$, $+1$, 0, -1, and -2, and they represent the boundaries between six ambiguity zones: $(+\infty, +2)$, $(+2, +1)$, $(+1, 0)$, $(0, -1)$, $(-1, -2)$, and $(-2, -\infty)$. This elimination of trellis branches is dependent on the values of channel samples, and a maximum of only two values are allowed as the output x_k of the PR channel. Two values are allowed for zones $(+2, +1)$, $(+1, 0)$, $(0, -1)$, $(-1, -2)$, and only one value is allowed for zones $(+\infty, +2)$ and $(-2, -\infty)$. The probability of noise added—such that it brings the sample outside of the zone—is small, assuming a high signal-to-noise ratio (SNR) operating range. This leads to a simplification of the EPR4 trellis by elimination of restricted branches. The size of the detector is, thus, significantly reduced, but its implementation is complicated by adding time variance.

Other approaches to reducing the complexity of the Viterbi detector have been proposed in [26] and [27]. Instead of eliminating improbable branches from the trellis, these approaches focus on merging trellis states. Using similar concepts as in set partitioning [28], symmetric trellis states are paired and merged to share ACS resources by creating parallel branches. Incoming sample values [27] or preliminary decisions in the survival registers [26] are used to discriminate the paths within the parallel branches. In [27], a compare-select-add (CSA) structure is used [29], which requires the comparison, selection, and addition operations to be carried out sequentially to reduce the hardware. The trellis simplification method in [27] is not systematic and produces nonoptimal CSA sharing. Multiplexers are required at the input and output of ACS units and in branch metrics additions. Using preliminary decisions [26] reduces the number of trellis states by half, but it introduces an additional tight feedback loop. Decisions from the ACS units of the current computation cycle are needed to generate branch metrics for the next cycle. A typical implementation requires parallel branch metrics hardware, and the ACS decisions would be used to multiplex in the correct branch metric values at the end of the computation cycle.

III. Trellis Reduction by Coding Constraints and Rate 8/9 TMTR Code with Stationary Detector

Application of coding can affect the trellis and the complexity of the detector by defining constraints that restrict the occurrence of certain patterns in the code. One of the commonly used codes that reduces the complexity of the matching detector in the E^{2}PR4 channel is the rate 2/3 (1, 7) runlength-limited (RLL) code [30]. The resulting trellis has 10 states, compared with 16 states for a full trellis. The code has distance-enhancing properties similar to the TMTR code, but the code rate is low. However, a rate 8/9 TMTR code requires a time-varying detector. A time-invariant trellis with some states permanently eliminated allows for the use of a lower complexity sequence detector to match the code constraints.

To explore the TMTR code structure for possible trellis complexity reduction, a new rate 16 : 18 TMTR code is derived. The code has a maximum zero runlength of $k = 10$ and the same distance properties as the rate 8/9 TMTR code. The resulting trellis has a period of 2, rather than of 9. The application of this code keeps the two-step implementation of the detector stationary.

The new code eliminates the same distance error events as the 8/9 TMTR code, by not allowing quadbits in codewords and by limiting the occurrence of tribits to certain bit positions inside the codeword. Expansion of the constraints to 18-bit codewords [12] and creation of a 16 : 18 block code result in a stationary trellis. However, direct 16- to 18-bit mapping would require large encoding/decoding logic and the resulting code would have long byte error properties.

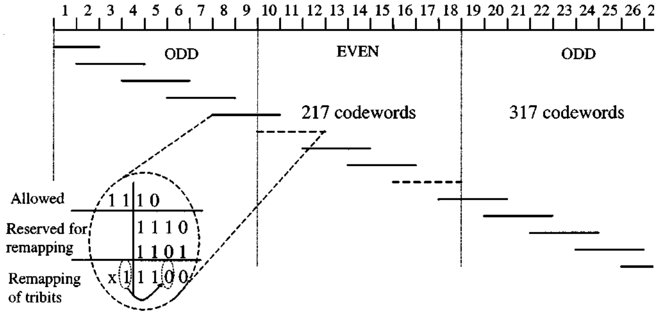


Fig. 1. Rate 16 : 18 TMTR code with an illustration of resolving boundary quadbits and tribits.

The new code is constructed by an initial encoding of odd and even user data bytes separately. The odd codewords allow tribits to start at even bit positions, whereas even codewords allow tribits to start at odd bit positions, thus, resulting in possible concatenation. If after the initial encoding either a quadbit or a tribit at a position that violates the MTR constraint is detected at the boundary between the odd and even codewords, the even codeword is remapped to a set of reserved codewords. Information about the remapping is encoded in the adjacent odd codeword, as illustrated in Fig. 1 [15]. Practical error propagation is limited to two bytes. Three-byte error propagation is possible only when an error event of 5 bits in length hits both remapped bits in the odd codeword.

The encoder and decoder are designed using "gated" partitions [16], which results in about 2.5 more complexity than a TMTR encoder/decoder. However, this increased complexity is still a very small portion of the read channel chip.

This code, when applied to the E^{2}PR4 trellis, requires only a 14-state, two-step, stationary Viterbi detector instead of a 16-state detector. These 14 states require eight four-way and six three-way ACS units for implementation, as shown in Fig. 2.

IV. Improved Method of Trellis Reduction

The complexity of the Viterbi detector can be reduced by dynamically eliminating improbable branches from the trellis, based on the values of channel samples [23]. The scheduling of dynamically assigned ACS units is complicated, especially when it is applied to high-order PR targets. When postprocessing techniques are used to correct dominant errors in the Viterbi detector, the number of pruned branches has to be carefully determined so that the reduced trellis operation will not affect the overall bit error rate (BER). An example of an E^{2}PR4 trellis is used as a demonstration of the improved trellis reduction scheme.

With the application of a trellis code to the E^{2}PR4 channel, the minimum squared distance for an error event is 10. This corresponds to a single-bit channel-input error event, $ex = [+]$, or equivalently, $ey = [1 \;\; 2 \;\; 0 \;\; -2 \;\; -1]$ at the output of the

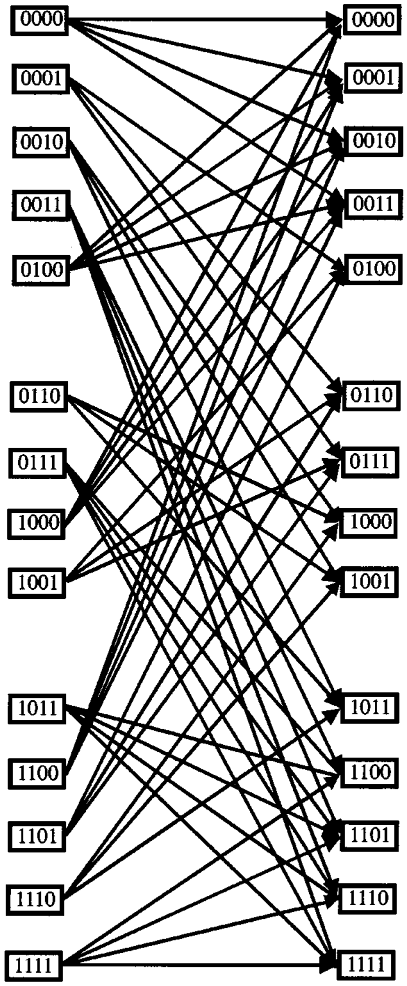


Fig. 2. Two-step E^2PR4 trellis with states 1010 and 0101 eliminated by coding constraints.

channel. For sample-by-sample threshold detection, the equivalent sample error event for a noiseless sample of value s being segmented into the regions $[s+2,\ s+3]$ or $[s-2,\ s-3]$ is $ey = [4]$. Therefore, for an observed sample of value s, pruning off all trellis branches except those that correspond to $[s_c+1,\ s_c,\ s_f,\ s_f-1]$, where s_c and s_f are the ceiling and floor of s, will have little effect on the overall system BER. The effective error distance of pruning out the true branch is $10{*}\log_{10} 1.6 = 2.04$ dB larger than that of the dominant error event of type $ex = [+]$. Moreover, the error distance of common trellis-coded E^2PR4 error events, e.g., $ex = [+]$, $[+\ -]$, $[+\ 0\ 0\ +]$, will further be reduced by correlation between the noise samples. The probability of misdetecting the true signal sequence S_k as another signal sequence $S'_k = S_k + ey_k$ (assuming S'_k is also a valid sequence) is

$$P(ey_k) = Q(d_k/2\sigma_k), \tag{6}$$

where $d_k^2 = \Sigma ey_k^2$ and σ_k^2 is the variance of the noise projected into the error event subspace along the direction of the error event ey_k

$$\sigma_k^2 = (\underline{ey}'_k \mathbf{R} \underline{ey}_k)/(\underline{ey}'_k \underline{ey}_k). \tag{7}$$

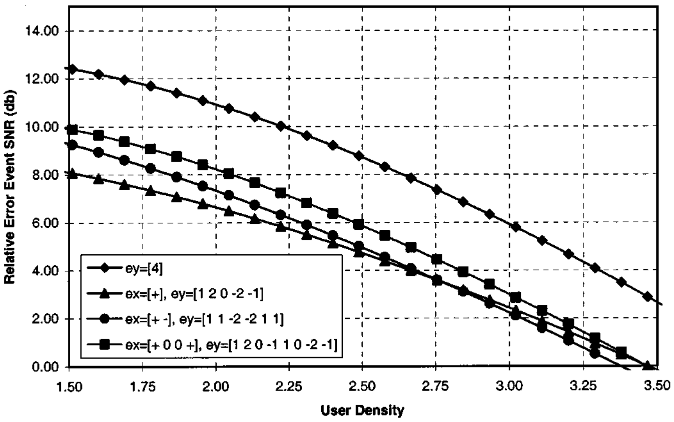


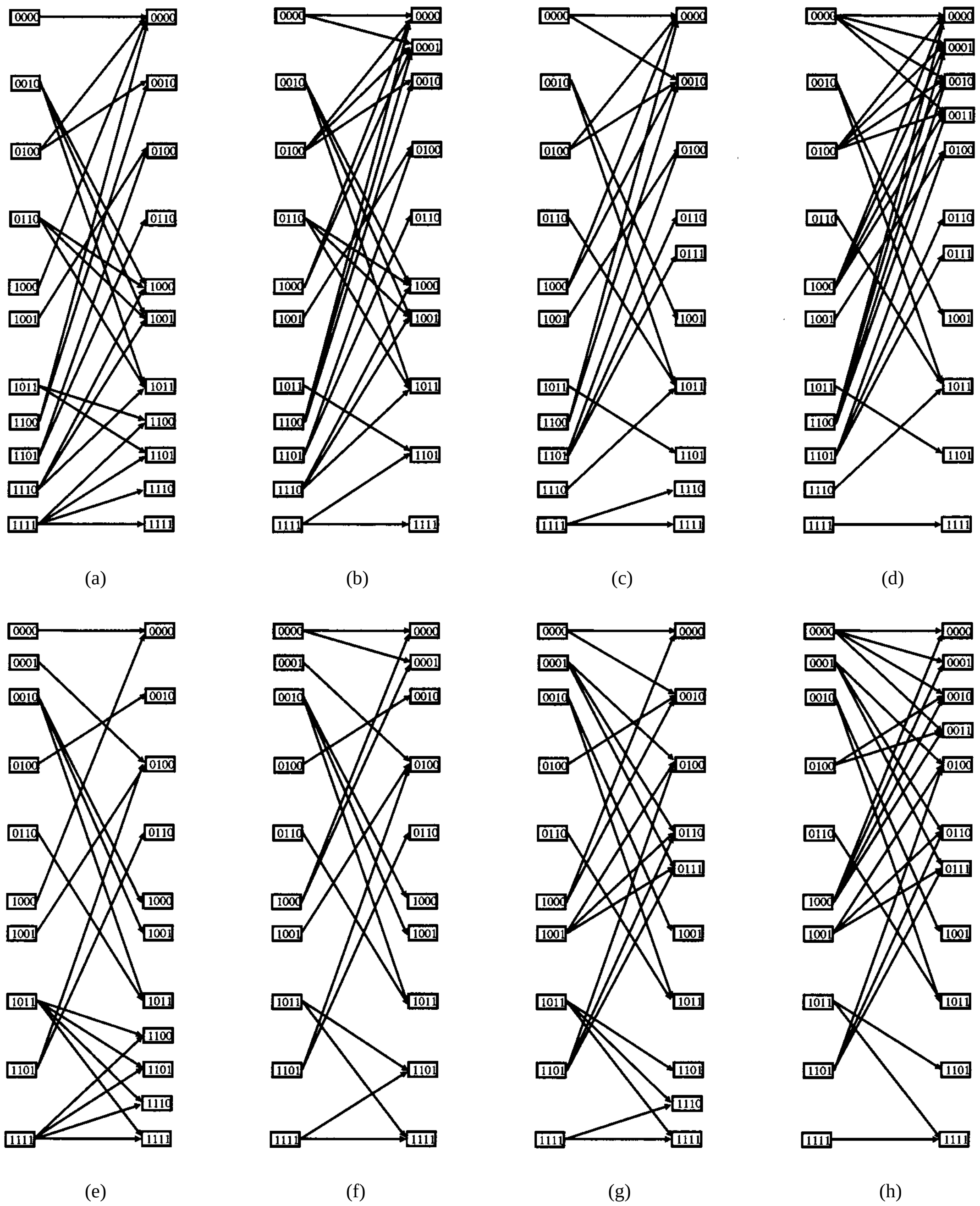
Fig. 3. Relative error event SNR for pruning error and three dominant regular error events. SNR is normalized such that the relative SNR for $Ex = [+]$ is zero at user density of 3.5.

$\underline{ey}_k$ is the vector representation of ey_k, and $\mathbf{R}$ is the autocorrelation matrix of the noise at the output of the equalizer. For the error event $ex = [+]$, $[+\ -]$, $[+\ 0\ 0\ +]$, the corresponding ey's are $[1\ 2\ 0\ -2\ -1]$, $[1\ 1\ -2\ -2\ 1\ 1]$, and $[1\ 2,\ 0\ -1\ 1\ 0\ -2\ -1]$. To compare the probability $P(ey_k)$ of pruning the true branch (ey = [4]) to various dominant error events in trellis-coded E^2PR4 systems, we denote $(d_k/\sigma_k)^2$ as the error event SNR for ey_k.

Fig. 3 shows the relative error event SNR for a Lorentzian pulse with white Gaussian noise. Four error event SNR curves are charted, including the trellis pruning error event, $ey = [4]$, and the three most dominant error events in a trellis-coded E^2PR4 system. As shown in Fig. 3, the SNR for the pruning error event is at least 3 dB larger than for the other dominant error events. It is around 4 dB better than $ex = [+]$ (rather than only 2.04 dB), due to the effect of noise sample correlation. Furthermore, the SNR penalty for pruning this error event is significantly larger than for the next two regular dominant error events. Therefore, the effect of branch pruning on system BER should be insignificant even if postprocessing is used to further eliminate major dominant error events. This effect will be further demonstrated by simulation results for a practical system.

The separation of all possible input cases that correspond to $[s_c+1,\ s_c,\ s_f,\ s_f-1]$ for each sample value would result in complicated control and selection logic. In this paper, a simple methodology of trellis reduction is traced. A single threshold at zero is used. The method uses the sign of a short sequence of data samples to configure the detector:

1) If the input signal is greater than 0, most of the negative signal levels are disallowed. When the input signal is negative, most of the positive signal levels are disallowed.
2) Further trellis reduction is possible by observing the signs of preceding and following input samples. Because certain states would be disallowed by the polarity of previous samples and because other states will be disallowed by the polarity of the following samples, additional branches can be pruned from the trellis in the current state.
3) Additional trellis branches are added back to the pruned trellis to simplify the ACS unit assignment.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Reduced trellises for the incoming signal samples, previous one, current two, and next one, marked as positive $(+)$ or negative $(-)$: (a) $- - - -$, (b) $- - - +$, (c) $- - + -$, (d) $- - + +$, (e) $- + - -$, (f) $- + - +$, (g) $- + + -$, and (h) $- + + +$.

The effect of these trellis-pruning steps can be more easily understood by examining their application to the E^{2}PR4 trellis in the next section.

V. Reduced Complexity Detector Implementation in the E^{2}PR4 Channel

With the application of the new TMTR code, the states 0101 and 1010 are permanently eliminated and the resulting two-step E^{2}PR4 trellis has only 14 states, as shown in Fig. 2. The method outlined in Section IV is applied to the resulting trellis.

1) If the input signal level is greater than 0, the allowed input levels are $+3$, $+2$, $+1$, 0, -1, and when the input signal level is less than 0, the allowed input levels are $+1$, 0, -1, -2, -3.
2) A sequence of four incoming samples is analyzed, and the resulting two-step trellis is formed. The selection is based on the preceding sample, two current samples, and the next sample.

Based on the above criteria, the reduced trellises in Fig. 4 are formed. Only eight out of 16 reduced trellises are shown in Fig. 4, because the remaining eight are symmetrical to the ones shown.

For the implementation of this type of sequence detector, a smaller number of ACS units is needed. A minimal implementation of the 14-state E^{2}PR4 trellis contains the following:

- four four-way ACS units;

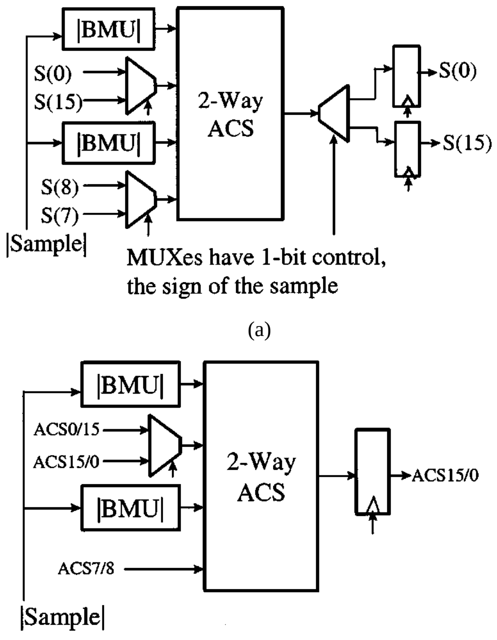


Fig. 5. (a) Example of sharing two-way ACS between states 0 and 15. (b) Example of sharing two-way ACS by using only input multiplexer.

- one three-way ACS unit;
- four two-way ACS units;
- two adders (one-way ACS).

The 14 states will be dynamically assigned to 11 ACS units, which require additional multiplexers in the critical path and control logic outside of the critical path.

This approach reduces the hardware complexity of the Viterbi detector by approximately 50%. For example, the total number of adders in ACS units is reduced to 29, compared with 64 in the full trellis implementation.

For different sequences of inputs, different detector structures are needed, as seen in Fig. 4. The requirements for the number and size of ACS units for each state vary with different sample sequences. The direct assignment plan for the states that share the reduced set of ACS units would lead to a complicated control, resulting in significant overhead in multiplexer logic. The multiplexers of different sizes, from two-way to five-way, would be needed.

A. Simplified ACS Assignment

A simple solution is to add more resources besides the initial four four-way, one three-way, four two-way ACS units, and two adders. The straightforward assignment plan can be derived if the symmetrical states (0 and 15, 1 and 14, 2 and 13, 3 and 12, 4 and 11, 6 and 9, 7 and 8) share the same resources. In this case, a few branches are added back to the pruned trellis and some of the ACS units are extended to accept more inputs. The minimum solution requires four four-way, three three-way, three two-way ACS units, and one adder.

In the proposed implementation, the branch metrics (BM) and ACS units would be assigned together. This implementation eliminates the BM multiplexers in front of the ACS units.

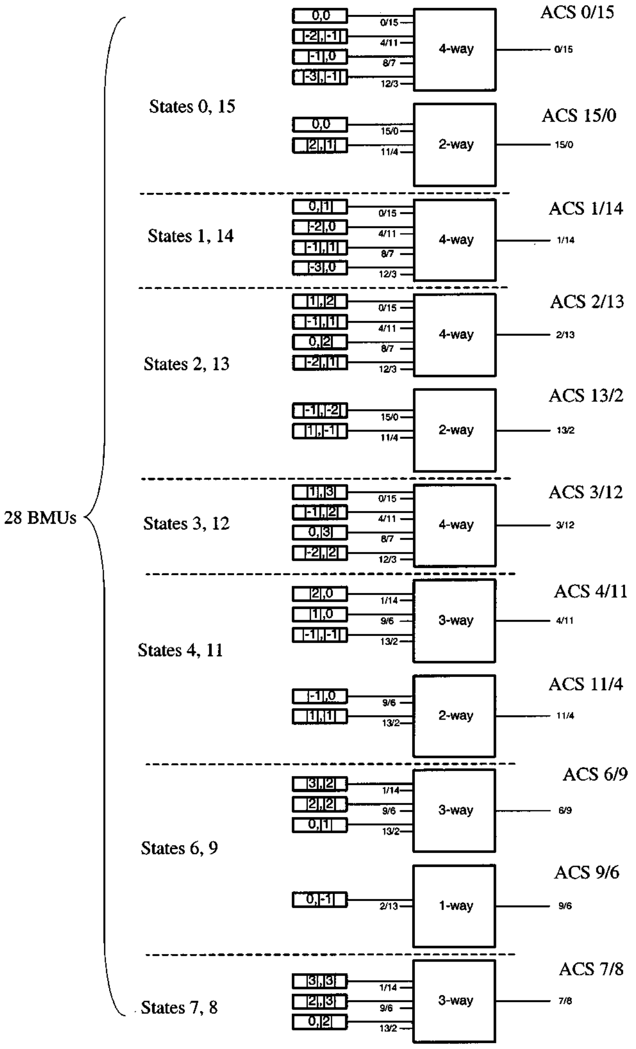


Fig. 6. Reduced complexity detector structure.

Sharing of ACS units can be done by adding the multiplexers both at the inputs and at the outputs of the ACS units. Because the assignment is simple, only two control signals are needed at the input and at the output, which will control 11 two-way multiplexers at the input and 11 two-way demultiplexers at the output. Because the states {0, 15}, {1, 14}, {2, 13}, {3, 12}, {4, 11}, {6, 9}, and {7, 8} share the same resources, the control logic is based only on a single sample sign [Fig. 5(a)].

Moreover, the demultiplexers after ACS units can be eliminated if the implementation is ACS-based, or assignment driven, as opposed to being based on conventional states [Fig. 5(b)] [31]. In this case, we are basing our detection on ACS units, not on actual states. This is the reason why ACS units are named: 0/15, 15/0, 1/14, 2/13, 13/2, 3/12, 4/11, 11/4, 6/9, 9/6, 7/8. This results in a slightly more complicated control of 11 two-way multiplexers at the input. The resulting ACS assignment plan is shown in Table III, and the detector architecture is shown in Fig. 6.

A problem that arises by adding extra branches in the trellis is that some of the states that generate them do not exist. Such states should not be considered in the calculation of new states. This can be done by adding a single validity tag bit to every state.

TABLE III
ACS USAGE DEPENDING ON THE INPUT SAMPLE SIGNS WITH ADDED RESOURCES AND SIMPLIFIED ASSIGNMENT. NUMBERS INDICATE THE SIZE OF THE ACS UNIT

Seq. State	----	---+	--+-	--++	-+--	-+-+	-++-	-+++	+---	+--+	+-+-	+-++	++--	++-+	+++-	+++
0	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2
1	x	4	x	4	x	4	x	4	x	4	x	4	x	4	x	4
2	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2
3	x	4	x	4	x	4	x	4	x	4	x	4	x	4	x	4
4	2	2	2	2	3	3	3	3	2	2	2	2	3	3	3	3
5	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
6	1	1	3	3	1	1	3	3	1	1	3	3	1	1	3	3
7	x	x	3	3	x	x	3	3	x	x	3	3	x	x	3	3
8	3	3	x	x	3	3	x	x	3	3	x	x	3	3	x	x
9	3	3	1	1	3	3	1	1	3	3	1	1	3	3	1	1
10	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
11	3	3	3	3	2	2	2	2	3	3	3	3	2	2	2	2
12	4	x	4	x	4	x	4	x	4	x	4	x	4	x	4	x
13	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4
14	4	x	4	x	4	x	4	x	4	x	4	x	4	x	4	x
15	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4

Invalid states could exist in cases in which two states share only one ACS resource (states 1 and 14, 3 and 12, 7 and 8).

The reduced complexity detector has only four timing critical four-way ACS units, which results in a much shorter interconnect length. The speed penalty associated with adding the gate that evaluates the tag bit and multiplexers is well compensated for by reduced interconnect length in the ACS array.

The detector structure resulting from the assignment plan using symmetric states is similar to the structure reported in [27]. As described in Section II-D, the method in [27] produces nonoptimal ACS sharing, which requires 12 trellis states when applied to the TMTR-coded trellis. The addition of extra branches back to the pruned trellis results in only one multiplexing operation at the input of ACS, whereas for the detector in [27], multiplexers are required at the input and at the output of ACS units as well as in BM additions. Moreover, the multiplexing operations at the input of ACS are relatively complex.

TABLE IV
REQUIRED SIZES AND OPERATIONS IN ACS UNITS

ACS	Type	States	Branch metrics
0/15	4-way	0,15	(0,0), (-2,-1), (-1,0), (-3,-1)
15/0	2-way	0,15	(0,0), (2,1)
1/14	4-way	1,14	(0,1), (-2,0), (-1,1), (-3,0)
2/13	4-way	2,13	(1,2), (-1,1), (0,2), (-2,1)
13/2	2-way	2,13	(-1,-2), (1,-1)
3/12	4-way	3,12	(1,3), (-1,2), (0,3), (-2,2)
4/11	3-way	4,11	(2,0), (1,0), (-1,-1)
11/4	2-way	4,11	(-1,0), (1,1)
6/9	3-way	6,9	(3,2), (2,2), (0,1)
9/6	1-way	6,9	(0,-1)
7/8	3-way	7,8	(3,3), (2,3), (0,2)

The reduced complexity system was simulated to verify the proper operation and to evaluate the possible loss in BER compared with full implementation. The resulting comparison of trellis-coded E^2PR4 channels, at a user density of 3.0, implementing an 8/9 code and a full trellis from [11] with stationary TMTR code with reduced trellis, is shown in Fig. 7. A baseline performance is shown for the RLL-coded, rate 16/17 EPR4 detector. Simulations confirm that, indeed, there is a small loss—below 0.3 dB—at lower SNRs, and virtually no loss at higher SNR with BER lower than 10^{-6}. Additive white

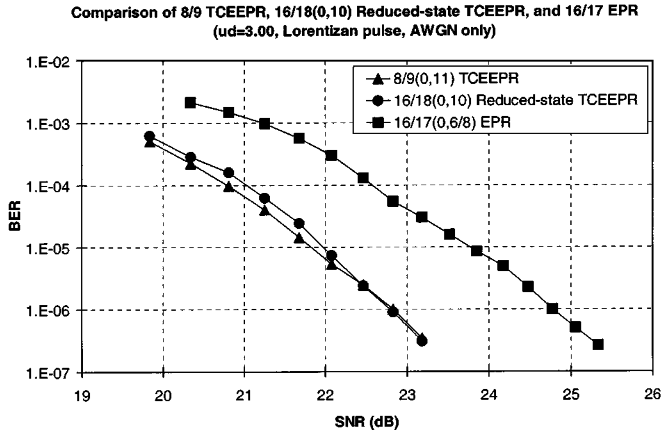


Fig. 7. Comparison of BER versus SNR for trellis-coded E^2PR4 channels (SNR levels are shown before equalization).

Gaussian noise (AWGN) is added to the input of the analog front end of the channel. SNR is defined as the zero-to-peak amplitude of the input signal over the noise power within two times the Nyquist bandwidth. BER curves are generated with 100 errors for each data point in the graph (i.e., 10^8 bits are exercised for BER at 10^{-6}). Fig. 8 demonstrates the performance of the reduced complexity detector when used with an error postprocessor. Four performance curves are shown; they correspond to the following detectors:

1) E^2PR4 with stationary TMTR code, full trellis;
2) E^2PR4 with stationary TMTR code, reduced trellis;
3) E^2PR4 with stationary TMTR code, full trellis with ideal postprocessing;
4) E^2PR4 with stationary TMTR code, reduced trellis with ideal postprocessing.

All graphs are shown with rate 8/9 (16 : 18) codes; i.e., there was no code-rate adjustment for eventual parity bits used to enhance the postprocessing. Postprocessing systems are denoted EEPR+ in Fig. 8.

The ideal postprocessing is implemented in a simulator by not counting errors of type $[+]$ and $[+\ -]$, thus, assuming ideal postprocessor performance. The purpose of this setup is to verify the performance of the reduced complexity detector in the presence of the error postprocessor. As seen from Fig. 8, there is no performance degradation between full and reduced trellis implementations with or without a postprocessor.

VI. Application of the Method to Other Partial Response Channels

To demonstrate the wide application range of the proposed method for reduction of detector complexity, it was applied to several higher order PR channels suitable for application at higher user densities. At higher user densities, channel BER can be improved by employing a 32-state Viterbi detector that corresponds to targets that better match the channel response. The PR of type

$$h(D) = (1-D) \cdot (1+D)^2 \cdot (2+D+D^2) \tag{8}$$

matches the Lorentzian channel at a 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 a squared distance of 10, $[+\ -]$ with squared distance of 20, and a single bit error event with a squared distance of 22. Thus, the coding gain obtained by using the TMTR code that eliminates all short-distance error events is 3.4 dB. With code-rate loss accounted for, the new target demonstrates about 1.7 dB improvement over the E^2PR4 channel with a 16/17 RLL code, or about 1 dB improvement with the use of the TMTR code, as shown in Fig. 9.

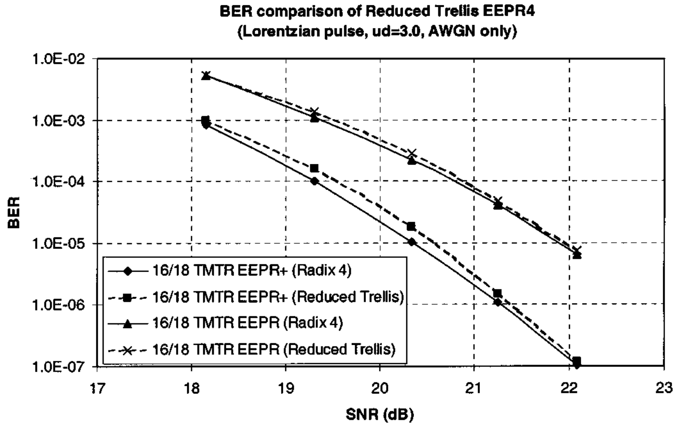


Fig. 8. Comparison of BER versus SNR for full and reduced implementations of a trellis-coded E^2PR4 channel with and without postprocessing.

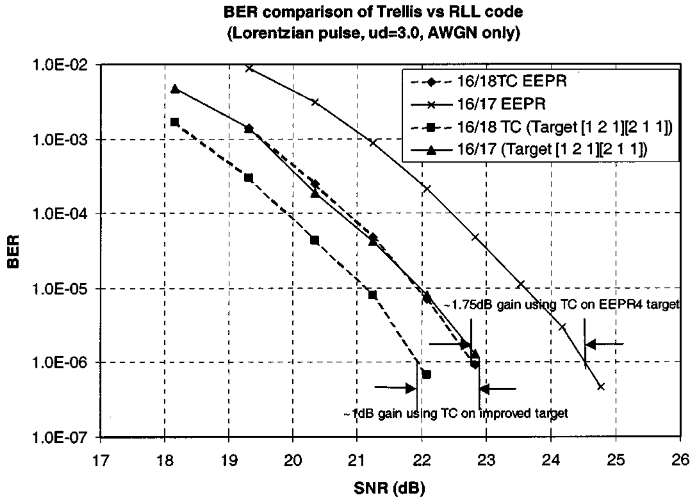
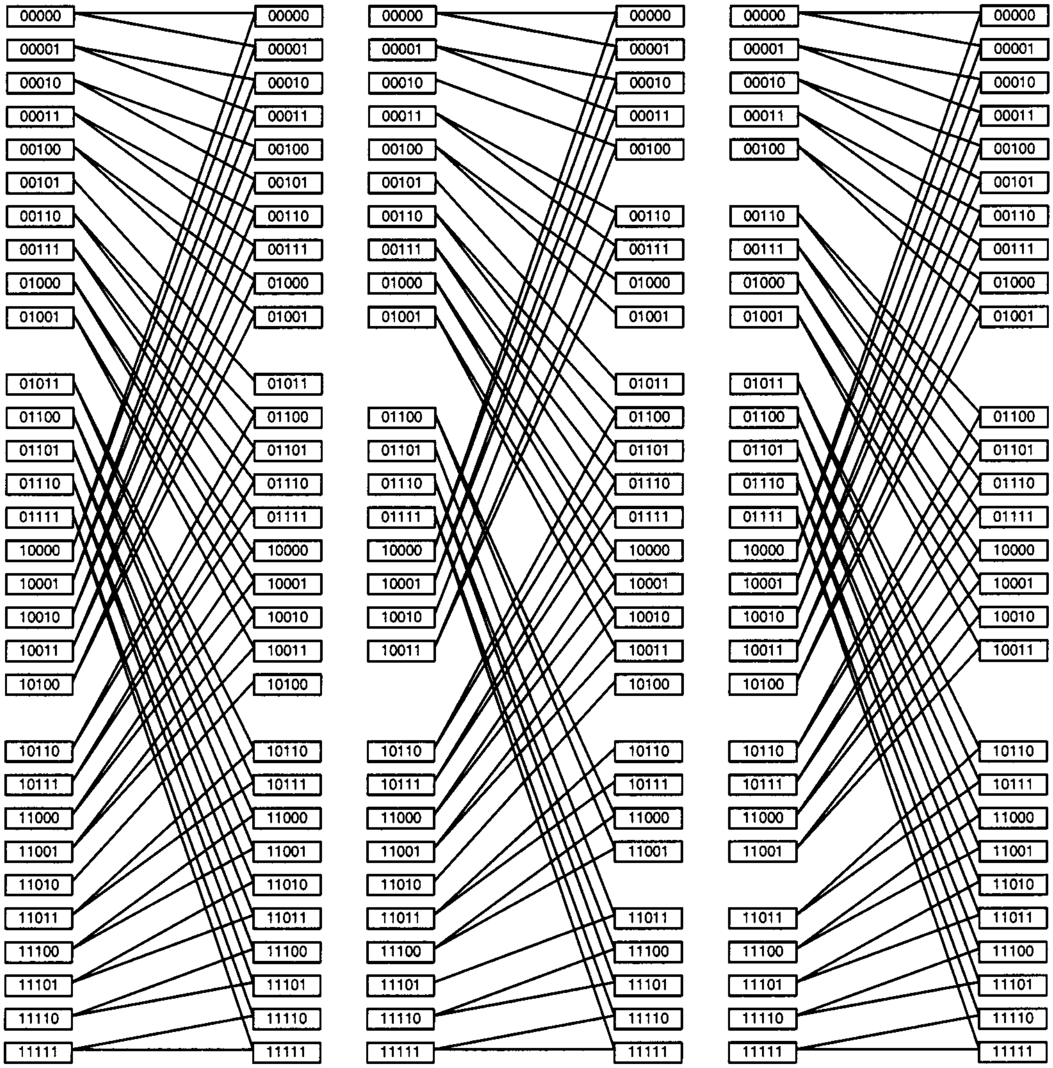


Fig. 9. Performance comparison of the new [1 2 1] [2 1 1] target versus E^2PR4, with trellis and RLL codes.

Application of the QMTR code to the 32-state trellis permanently eliminates two states (01010 and 10101) in every step, as shown in Fig. 10(a). Application of the TMTR code to the 32-state trellis eliminates the states 01010 and 10101 as well as two more states, but results in a step-by-step time varying structure, as shown in Fig. 10(b) and (c).

Using a two-step detector structure for both QMTR and TMTR codes results in stationary trellises, as shown in Fig. 11(a). Two states are removed in the case of the QMTR code, and four states are removed when using a TMTR code resulting in two possible trellis structures, as shown in Fig. 11(b) and (c).



(a) (b) (c)

Fig. 10. (a) Single-step trellis diagram with QMTR code, (b) single-step trellis diagram with TMTR code (version A), and (c) single-step trellis diagram with TMTR code (version B).

The reduction method described in Section IV was applied to the new target with a 16/18 TMTR code and a 16/17 QMTR code. When the incoming sample is positive, any BM that corresponds to a target value of -2 or below is removed from the trellis. When the sample is negative, BMs that correspond to targets $+2$ or above are eliminated. To demonstrate the effect of trellis reduction in BER, the following test cases were selected:

1) new target with QMTR code (rate 16/17);
2) new target with TMTR code (rate 16/18);
3) new target with QMTR code with ideal postprocessing (rate 8/9, assumes adding 3 bits for 6 bytes for required parity for postprocessing);
4) new target with TMTR code with ideal postprocessing (rate 48/58, assumes adding 4 bits for 6 bytes for required parity for postprocessing).

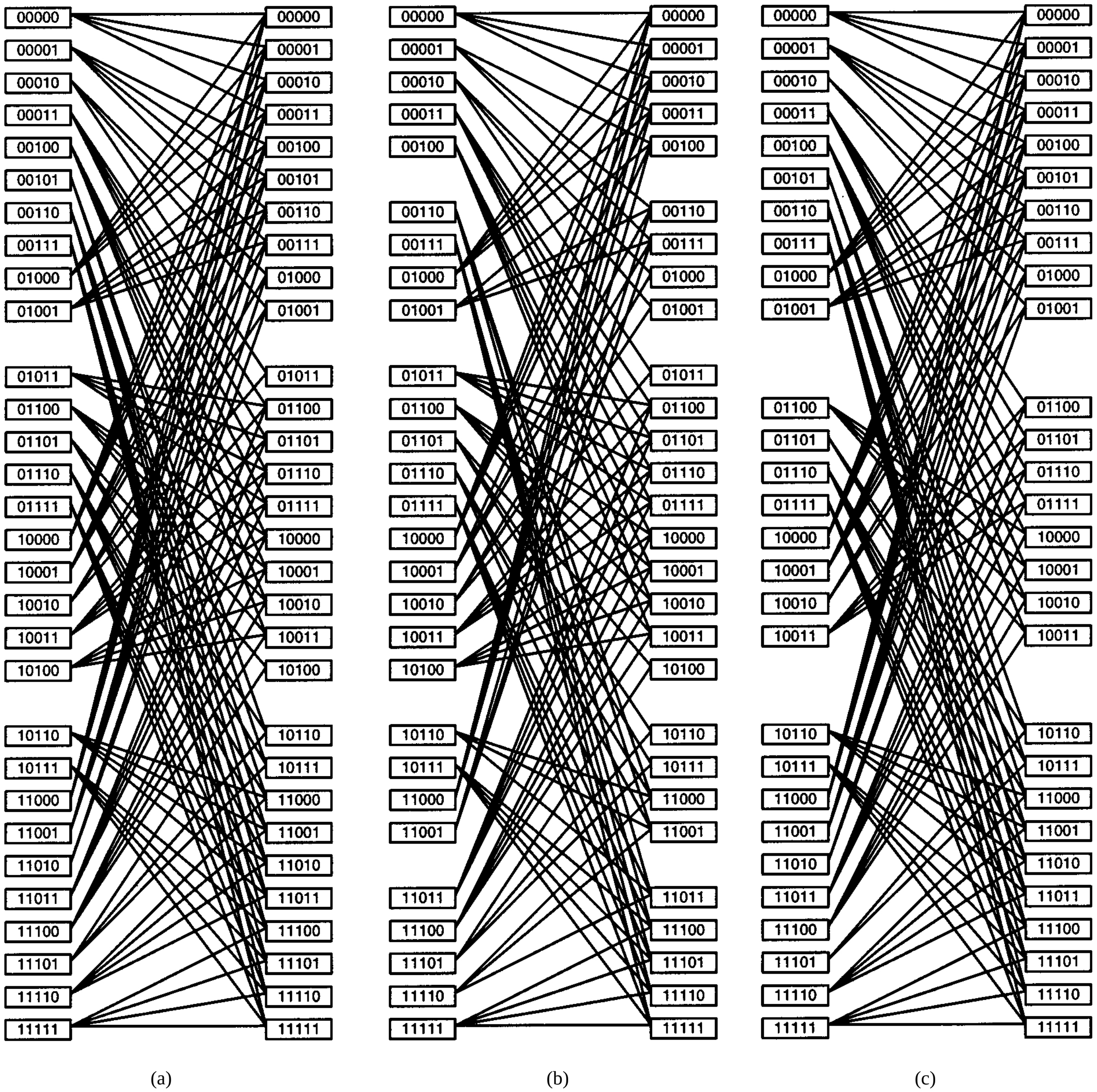
All four of these cases were simulated twice with full and with reduced complexity detectors. The results summarized in Fig. 12 show no performance degradation in any of the cases. The postprocessing operations are assumed ideal in the simulations. Three error event types had simply been omitted in error counting. The removed error patterns were as follows:

- in QMTR: $[+]$, $[+\ -]$, $[+\ -\ +]$;
- in TMTR: $[+]$, $[+\ -]$, $[+\ 0\ 0\ +]$.

To each data block of 6 user bytes, even and odd parity bits are added with additional bits needed to avoid code constraint violation.

VII. Complexity Estimation

The implementation of a single-step E^2PR4 Viterbi detector requires seven branch metrics units (BMU), 16 ACS units, and 16 survival registers. Each single-step ACS unit consists of two adders and one comparator, usually implemented as a subtractor. As the subtractors require only carry chain for implementation of the comparison, they are less complex than adders. For full two-step implementation, BMs are added together to form 39 different two-step BMs. A four-way ACS unit consists of four adders and six parallel subtractors.



(a) (b) (c)

Fig. 11. (a) Two-step trellis diagram with QMTR code, (b) two-step trellis diagram with TMTR code (version A), and (c) two-step trellis diagram with TMTR code (version B).

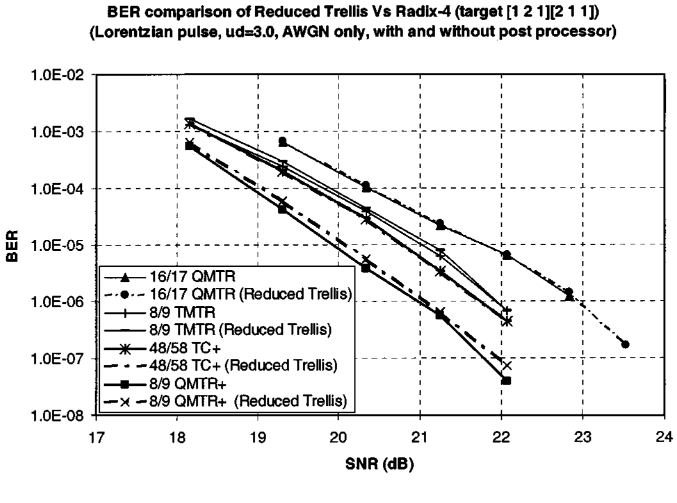


Fig. 12. Performance comparison of full versus reduced complexity implementation for TMTR and QMTR codes with and without postprocessing.

Table I shows the number of adders and comparators required and the equivalent sizes of one-, two-, three-, and four-way ACS units, assuming the size of an adder is 1.5 times larger than that of a comparator.

Table V shows that reduced complexity detector implements the Viterbi algorithm at less than a 30% of hardware increase from single-step while achieving two-step throughput. Full two-step implementation is approximately 2.7 times bigger than is single-step implementation.

To estimate the savings in hardware requirements when the reduction method is applied to the PR (8), the following test cases are selected:

1) stationary TMTR code with two-step detector;
2) QMTR code with two-step detector;
3) QMTR code with single-step detector.

Table VI shows the number of ACS units used with QMTR and TMTR codes for single-step Viterbi implementation, where

TABLE V
COMPLEXITY COMPARISON BETWEEN DIFFERENT IMPLEMENTATIONS OF E^2PR4 DETECTORS

Implementation / Unit	Full 2-step		Full 1-step		14-state, 2-step		Reduced, 2-st.	
	Count	%	Count	%	Count	%	Count	%
ACS Adders	64	100	32	50	50	78	32	50
ACS Subtract.	96	100	16	17	66	69	36	37
ACS Size	384	100	128	33	282	73	168	43
1-step BMUs	14	100	7	50	14	100	14	100
BM adders	39	100	-	-	39	100	25	64
Survival Regs.	16	100	16	100	14	87	11	69

TABLE VI
COMPLEXITY COMPARISON OF SINGLE-STEP DETECTORS FOR THE NEW TARGET WITH DIFFERENT CODES

Type / ACS	Single-step	Single-step	Single-step	Single-step	Reduced single-step
	RLL	TMTR_A	TMTR_B	QMTR	QMTR
1-way	-	6	-	2	5
2-way	32	22	28	28	14
3-way	-	-	-	-	-
4-way	-	-	-	-	-
Total	32	28	28	30	19
Size	256	194	224	230	127

TABLE VII
COMPLEXITY COMPARISON OF TWO-STEP DETECTORS FOR THE NEW TARGET WITH DIFFERENT CODES

Type / ACS	Two-step	Two-step	Two-step	Two-step	Reduced two-step	Reduced two-step	Reduced two-step
	RLL	QMTR	TMTR_A	TMTR_B	QMTR	TMTR_A	TMTR_B
1-way	0	0	0	0	2	3	2
2-way	0	2	6	0	3	4	2
3-way	0	4	0	12	2	0	6
4-way	32	24	22	16	12	11	8
Total:	32	30	28	28	19	18	18
Size	768	652	576	564	348	305	304

TMTR_A denotes the trellis used in the odd cycle and TMTR_B denotes the trellis used in the even cycle. It also compares the hardware complexity of the regular, single-step 32-state Viterbi using a QMTR code with the reduced single-step structure and shows that the reduced trellis method results in a 45% reduction in size.

The number of ACS units required and the estimated size of different two-step detectors are shown in Table VII. The table summarizes the complexities of the full two-step implementations and the savings that correspond to different codes. The reduced-complexity detector shows about a 45% reduction in size for both QMTR and TMTR codes, compared with the full implementation.

VIII. Conclusion

The proposed method significantly reduces the implementation complexity of the Viterbi detector for high-order PR channels. This is achieved by eliminating less likely taken branches from the trellis, in conjunction with applied trellis coding. The elimination uses a simple method based on the sign of the input sample. The smaller area and power of the detector, achieved by using a simple assignment of ACS units, could be traded for speed enhancement. System level simulations have shown that there is no significant loss in BER when compared with full trellis implementation. The method is demonstrated on a trellis-coded E^2PR4 channel resulting in a 50% complexity reduction. Similar complexity and performance results are achieved for a 32-state detector that matches a different equalization target, with different recording codes and detector implementations.

Acknowledgment

This work was initiated by the late R. Yamasaki, for whose guidance the authors are deeply grateful.

References

[1] G. Forney, Jr., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363–378, May 1972.

[2] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[3] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

[4] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, pp. 921–934, Sept. 1975.

[5] H. Kobayashi and D. T. Tang, "Application of partial-response channel coding to magnetic recording systems," *IBM J. Res. Devel.*, vol. 14, pp. 368–375, July 1970.

[6] H. Thapar and A. Patel, "A class of partial response systems for increasing storage density in magnetic recording," *IEEE Trans. Magn.*, vol. MAG-23, pp. 3666–3668, Sept. 1987.

[7] 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.

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

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

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

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

[12] W. G. Bliss, "An 8/9 rate time-varying trellis code for high density magnetic recording," in *Proc. Dig. IEEE Int. Magn. Conf., InterMag'97*, New Orleans, p. BS-09.

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

[14] ——, "Trellis coding system for disc storage systems," U.S. Patent 6.032.284, Feb. 29, 2000.

[15] B. Nikolić, M. Leung, and L. Fu, "A rate 8/9 sliding block code with stationary detector for magnetic recording," IEEE Trans. Magn., vol. 37, pp. 1653–1657, May 2001, to be published.

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

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

[18] 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.

[19] 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.

[20] R. Wood, "Turbo-PRML: A compromise EPRML detector," *IEEE Trans. Magn.*, vol. 29, pp. 4018–4020, Nov. 1993.

[21] T. Conway, "A new target response with parity coding for high density magnetic recording channels," *IEEE Trans. Magn.*, vol. 34, pp. 2382–2386, July 1998.

[22] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 989–995, Aug. 1989.

[23] H. Shafiee and J. Moon, "A reduced-complexity trellis search decoding algorithm for extended class IV partial response systems," in *SUPERCOMM/ICC '92*, vol. 1, Chicago, IL, pp. 120–124.

[24] R. Behrens *et al.*, "Method and apparatus for reduced-complexity Viterbi-type sequence detectors," U.S. Patent 5.291.499, Mar. 1994.

[25] H. Kobayashi and D. T. Tang, "On decoding of correlative level coding systems with ambiguity zone detection," *IEEE Trans. Commun.*, vol. COM-19, pp. 467–477, Aug. 1971.

[26] R. He, J. R. Cruz, and H. Song, "A reduced-state sequence estimation technique and its applications," *IEEE Trans. Magn.*, vol. 35, pp. 2331–2333, Sept. 1999.

[27] K. Tsukano, T. Nishiya, T. Hirai, and T. Nara, "Simplified EEPR Viterbi detector based on a transformed radix-4 trellis for a disk drive," *IEEE Trans. Magn.*, vol. 35, pp. 4387–4401, Sept. 1999.

[28] J. Bergmans, *Digital Baseband Transmission and Recording*. Dordrecht: Kluwer Academic, 1996.

[29] G. Fettweis, R. Karabed, P. H. Siegel, and H. K. Thapar, "Reduced-complexity Viterbi detector architectures for partial response signaling," in *Proc. 1995 Global Telecommun. Conf. (Globecom'95),* Singapore, pp. 559–563.

[30] R. T. Behrens and A. J. Armstrong, "An advanced read/write channel for magnetic disk storage," in *Proc. Conf. Record 26th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Oct. 26–28, 1992, pp. 956–960.

[31] B. Nikolić, M. Leung, L. Fu, V. G. Oklobdžija, and R. Yamasaki, "Reduced complexity sequence detection for E^2PR4 magnetic recording channel," in *Proc. 1999 IEEE Global Conf. Commun., GLOBECOM'99, Conf. Record*, vol. 1, part B, Rio de Janeiro, Brazil, pp. 960–964.

Michael 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 Architecture Group in Texas Instrument's Storage Products Group, San Jose, CA, where he had 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 with 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.

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

He is an Assistant Professor in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley. 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. His research interests include high-speed and low-power digital integrated circuits and VLSI implementation of communications and signal processing algorithms.

He 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 at the University of California at Davis in 1999, as well as the City of Belgrade award for the best Diploma thesis in 1992.

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

From 1992 to 1993, he worked as a Research Assistant in the Center of X-ray Optics at Lawrence Berkeley Laboratory, 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., later acquired by Texas Instruments Incorporated (TI) as TIs Storage Products Group, San Jose, CA, where he worked as a System Engineer on signal processing architecture and modeling DSP algorithm on advanced HDD storage channel ICs. His technical interests include coding, DSP architecture, and implementation. In 2000, he joined the DSP Architecture Group at Tripath Technology, Santa Clara, CA, and is currently working on the architecture of ADSL and optical transceiver IC products.

Taehyun Jeon received the B.S. degree from Yonsei University, Seoul, Korea, in 1989, and the M.S. and Ph.D. degrees from the University of Minnesota, Minneapolis, in 1993 and 1997, respectively, all in electrical engineering.

From 1994 to 1997, he was a Research Assistant with the University of Minnesota, Minneapolis. Upon completion of his graduate studies, he joined Motorola, Inc., where he worked on the advanced read channel architecture development for high-density data storage systems. Since 1998, he has been with Texas Instruments, Inc., San Jose, CA, where he is currently a Member of the Technical Staff. His current research interests include communication theory and signal processing for digital data transmission systems with emphasis on channel characterization and modeling, modulation coding, equalization, and detection.