Advanced Signal Processing Technique for Storage Systems

Jun Lee and Kees A. Schouhamer Immink, *Fellow*, IEEE

Abstract — *A post-Viterbi processor has found wide acceptance in recording systems since it can correct dominant error events at the channel detector output using only a few parity bits, and thereby significantly reduce the correction capacity loss of the error correction code. This paper presents two novel techniques for minimizing the mis-correction of a post-Viterbi processor based on an error detection code. One is a method for achieving a low probability of mis-selection in actual error-type. The other is a method for achieving a low probability of mis-positioning in error-location of an occurred error event. Simulation results show that an application of these techniques to conventional post-Viterbi processor considerably reduces the probability of mis-correction and the performance approaches the corresponding bit error rate and symbol error rate bound.*

Index Terms — Error detection code, dominant error event, matched-filtering type post-Viterbi processor, **cyclic redundancy check code.**

I. INTRODUCTION

The demand for high-density digital data storage systems has been growing steadily. Although technological innovations in the design of recording media and head are key to achieving high density recording systems, the role of sophisticated coding and signal processing techniques for data recovery is increasingly becoming crucial in supporting and augmenting these advancements.

To further improve the performance under aggressive recording conditions, currently, there is a great deal of activity exploring turbo coding and soft iterative decoding for future storage products [1]-[3]. Turbo codes and low-density parity-check (LDPC) codes are promising soft error correction codes (ECC) with potential to approach the channel capacity. However, due to the high implementation complexity and long latency, these schemes have so far not found their way to commercial recording systems. In recent years, there has been also a lot of interest in error detection codes that have error correction properties [4]-[10]. The reason is that unlike the situation in conventional read channels, where ECC is expected to correct all the errors at the output of the constrained decoder, dominant error events can be corrected by the application of a low redundancy error detection code. This error detection code is an inner ECC, which could correct dominant error events at the channel detector output, using only a few parity bits. In this way, the correction capacity loss of the outer ECC is significantly reduced and the error propagation of the modulation decoder is also minimized. Compared with the iteratively decodable codes, the error detection codes based approach, which is referred to post-Viterbi processor (or maximum likelihood (ML) post-processor), has found wide acceptance since the performance-complexity trade-off offered by these codes is very attractive and affordable. This approach has been widely studied for magnetic recording channels [4][6]-[10], and for DVD optical recording systems [5].

There has been unignorable mis-correction of a post-Viterbi processor due to channel impairments [5][6]. This paper proposes two new techniques for reducing mis-correction of a post-Viterbi processor based on an error detection code. One is a technique for minimizing the probability of mis-selection that the actual error event is judged as the incorrect one. The other is a technique for minimizing the probability of mis-positioning in error-location of an occurred error event. The performance has been evaluated under perpendicular magnetic recording channel. The techniques can be also applied to optical storage systems for performance improvement.

The paper is organized as the following. We firstly start with some preliminaries in Section II. Section III introduces new techniques and their examples for reducing the miss-correction. In Section IV, simulation results are given. Finally, conclusion is given in Section V.

II. PRELIMINARIES

In storage systems, data is encoded by an error detection encoder [10]-[12] to form codeword a=$[a_1, a_2, a_3, ..., a_N]$ with length N before it is transmitted. An equalizer reshapes the codeword transmitted through a readback channel into a sequence that is matched to a channel target response, and then ML detector recovers the codeword from the output of equalizer q=$[q_1, q_2, q_3, ..., q_N]$ based on Viterbi algorithm. However, the recovered codeword $\hat{a}$=$[\hat{a}_1, \hat{a}_2, \hat{a}_3, ..., \hat{a}_N]$ from the ML detector may contain errors caused by noise on the readback channel.

The performance of a partial response maximum likelihood (PRML) system can be improved by employing a post-Viterbi processor [4]-[10] that corrects a dominant error event at the output of a detector. An error detection decoder computes a syndrome to check for the presence of errors in the recovered codeword. When the syndrome is non-zero, a post-Viterbi processor is activated. The post-Viterbi processor first calculates the error signal $e_{q,k} = q_k - \hat{q}_k$ between an output signal of an equalizer q_k and a signal $\hat{q}_k = \sum_{l=0}^{L_g-1} g_l \hat{a}_{k-l}$ generated by convolution of an output of ML decoder with a channel target response $g(D) = g_0 + g_1 D + ... + g_{L_g-1} D^{L_g-1}$, where D is one-symbol delay. The error signal is then passed to a bank of matched filters

Jun Lee is with Data and Storage R &D Laboratory, LG Electronics in Korea (e-mail: leejun28@naver.com).

Kees A. Schouhamer Immink is with Institute for Experimental Mathematics, Ellernstrasse 29-31, Essen, Germany (e-mail: immink@turing-machines.com).

Contributed Paper
Manuscript received 10/14/10
Current version published 12/23/10
Electronic version published 12/30/10.

corresponding to the dominant error events. Here, the matched-filter for a given error event is the time-reversed version of the convolution between the error event and the channel target response. The outputs (namely, likelihood values) of the matched filters are normalized by subtracting a set of offset values (η_j s) associated with the corresponding error events. The maximum normalized output is used to determine the error-location of the corresponding error event, for each matched-filter. The resulting normalized maximum outputs of all the filters are compared and the largest one is selected. The error event ê chosen from conventional post-Viterbi processor is given by

$$\hat{e}=\arg\max_{e^j_{i\in\{1,2,\ldots,N\}}}\left\{\sum_{k=1}^{N_t} e_{q,k}\left(\sum_{l=0}^{L_g-1} g_l e^j_{i,k-l}\right)-\eta_j\right\}, \qquad (1)$$

where $\eta_j=\frac{1}{2}\sum_{k=1}^{N_t}\left(\sum_{l=0}^{L_g-1} g_l e^j_{i,k-l}\right)^2$, $N_t=N+L_e+L_g-2$ and $e^j_{i,k}=e_{k-i+1}$ for $k=i,i+1,\ldots,i+L_e-1$ is the k-th element of error event e^j_i with length L_e of type j, starting at position $i\in\{1,2,\ldots,N\}$. Finally, based on an error-type j and error starting position i of an error event ê corresponding to the largest likelihood value, the error event is corrected by the post-Viterbi processor. However, it is observed that there has been unignorable mis-correction of a post-Viterbi processor due to correlated noise and residual inter-symbol interference [5][6]. The mis-correction can be classified into mis-selection in actual error-type and mis-positioning in error-location of an occurred error event.

An error detection code [10]-[12] with a primitive (or a non-primitive) generator polynomial of degree m produces the same syndrome sequence of period (2^m-1) (or period < (2^m-1)) with different order of syndrome values for any dominant/non-dominant error events. Accordingly, in addition to the detection of an occurrence of an error event, the error detection code can generate a set of likely error starting positions for each detectable error event based on the syndrome corresponding to the recovered codeword, which is the output sequence of PRML system. Moreover, the size of the set can be decreased by checking whether bit pattern between each starting position and ending position can generate the corresponding error event or not. Consequently, the set of more probable error positions can be found. The procedure is referred to "*optimal position search*" in this paper. On the other hand, in conventional post-Viterbi processor, without the consideration of useful property of error detection code (namely, a periodic property of syndrome sequence) and characteristic of the error event, a set of error starting positions associated with any detectable error event is the same as {1, 2,…,*N*}, which are all positions in recovered codeword, and the set includes the positions where the corresponding error event cannot occur. Thus, the error correction of a post-Viterbi processor based on the set can result in mis-positioning in error-location of an occurred error event.

Conventional post-Viterbi processor performs the error correction of an occurred error event based on the error-type and its likely error starting position corresponding to the largest likelihood value. The correction based on the largest one is the possibility of mis-selection that the actual error event is judged as the incorrect one in case that the largest one is similar to the other ones. In order to reduce the possibility, the paper introduces an error correction technique of an occurred error event that considers the other ones as well as the largest likelihood value among the outputs of the matched filters. The technique is referred to "*list-correction*" in the paper.

III. ADVANCED POST-VITERBI PROCOESSOR

In conventional post-Viterbi processor, all positions in recovered codeword are considered as error starting positions, and each matched filter computes the likelihood values of the corresponding error event over these all positions and then it outputs a position corresponding to the maximum one. Finally, based on error-type and error starting position of error event corresponding to the largest likelihood value among the outputs of the matched filters, conventional post-Viterbi processor performs the error correction. However, due to channel noise, there is the possibility of mis-positioning that the selected error starting position is incorrect. In addition to mis-positioning, there is also the possibility of mis-selection that the selected error-type is incorrect.

Firstly, in order to lessen mis-positioning in error-location, the Section introduces a method for producing a set of more probable error starting positions of each detectable error event using a periodic property of syndrome sequence and characteristic of error event instead of all positions in recovered codeword. Secondly, in order to decrease mis-selection of actual error-type, the Section introduces an error correction technique of a new post-Viterbi processor based on the second and third largest ones, etc. as well as the largest likelihood value. One common technique used as parity-check code for error detection is known as a cyclic redundancy check (CRC) code [10]-[12]. The (*n*, *k*) CRC code is specified by its generator polynomial and the syndrome is the remainder when a recovered codeword polynomial is divided by the generator polynomial, where *n* is the length of codeword bits and *k* is the length of information bits. A primitive (or a non-primitive) generator polynomial of degree *m* for CRC code produces the same syndrome sequence of period (2^m-1) (or period < (2^m-1)) with different order of syndrome values for any dominant/non-dominant error events. When the CRC code

TABLE 1
CRC-BASED SYNDROME SEQUENCE WITH A PRIMITIVE GENERATOR POLYNOMIAL $G(X)=1+X^2+X^3$

Dominant error event	Syndrome sequence in decimal number
±[2,-2]	6 3 4 2 1 5 7
±[2,-2, 2]	7 6 3 4 2 1 5
±[2,-2, 2,-2]	2 1 5 7 6 3 4
±[2,-2, 2,-2, 2]	5 7 6 3 4 2 1
±[2,-2, 0, 2,-2]	4 2 1 5 7 6 3
±[2,-2, 2,-2, 2,-2]	3 4 2 1 5 7 6

detects the occurrence of any error event whose syndrome is non-zero, the corresponding syndrome can provide a set of likely error starting positions of an occurred any detectable error event in recovered codeword. However, in conventional post-Viterbi processor, the CRC code is not utilized to generate a set of more likely error starting positions corresponding to each dominant error event, but it is used to only check for the presence of errors in recovered codeword. With a CRC code with a primitive generator polynomial $G(x) = 1+x^2+x^3$, Table 1 shows the syndrome sequence in decimal number associated with each dominant error event that arises in typical perpendicular magnetic recording systems [10][11]. Here, the [+2, -2] error event, for example, arises when the correct bit pattern of [1, 1, 1, -1, 1, 1] is erroneously decoded as [1, 1, -1, 1, 1, 1], and it is denoted by difference between [1, 1, 1, -1, 1, 1] and [1, 1, -1, 1, 1, 1]. +2 error and -2 error in [+2, -2] error event are not distinguished; we simply use 1 to represent an erroneous bit in [+2, -2] and 0 a correct bit. Accordingly, [+2, -2] can be expressed as [1, 1]. In Table 1, the first syndrome value in the second column means the syndrome value when a corresponding dominant error event starts to occur in the first position in a recovered codeword. For example, syndrome value 6 for a dominant error event ±[2,-2] indicates that the error event starts to occur in the 1st position in a recovered codeword. Likewise, syndrome value 6 for a dominant error event ±[2,-2, 2,-2] is the syndrome value when the error event starts to occur in the 5th position in a recovered codeword. Since the syndrome sequence for each dominant error event is a repeated version of a period-7 ($=2^3$-1) syndrome sequence, a set of likely error starting positions for each dominant error event in codeword block can be generated. For example, assuming that syndrome value computed from a recovered codeword is 6, a set of possible error starting positions for a dominant error event ±[2,-2] is {1, 8,…,7*k*-6}, where k=1, 2,…,N/7. Moreover, from the set, it is possible to generate a set of more probable error starting positions by eliminating the positions in codeword block without possibility that the dominant error event occurs. As a simple example for *optimal position search*, (36, 33) CRC code based on $G(x) = 1+x^2+x^3$ in Figure 1 is considered. Suppose that an error event [-2, 2](=[1, 1]) occurs in the 3rd and 4th positions. The syndrome is computed as 4 in decimal numbers, which assumes that an error event [-2, 2] is occurred within a codeword. Based on the syndrome 4, the error event must start either in position 3, 11, 18, 25 or 32, which is a set of possible starting positions of the error event, $\boldsymbol{t}$ can be expected. After checking two bits between each starting position and ending position, i.e., [3, 4], [11, 12], [18, 19], [25, 26] and [32, 33], it is found that bits in positions 18 and 19 in $\boldsymbol{r}$ are all 1's, i.e., $\boldsymbol{r}_{18} = \boldsymbol{r}_{19} = 1$, and bits in positions 32 and 33 in $\boldsymbol{r}$ are all 0's, i.e., $\boldsymbol{r}_{32} = \boldsymbol{r}_{33} = 0$. Thus, from the set $\boldsymbol{t}$, these positions are eliminated because there is no probability that the error event [-2, 2] occurs in these positions. Consequently, a set of possible starting positions for the error event $\boldsymbol{s}$ is achieved. An error event ê selected from a new post-Viterbi processor based on *optimal position search* is given by

$$\hat{\mathrm{e}} = \arg\max_{\mathrm{e}_{i\in S_j}^{j}} \left\{ \sum_{k=1}^{N_t} e_{q,k} \left(\sum_{l=0}^{L_g-1} g_l e_{i,k-l}^{j} \right) - \eta_j \right\}, \qquad (2)$$

where S_j means a set of error starting positions corresponding to error-type j achieved by *optimal position search*. Main difference between equations (1) and (2) is a set of assumed error starting positions. The set in equation (1) is all positions in recovered codeword irrespective of error events, while the set in equation (2) depends on characteristic of the corresponding error event and syndrome of recovered codeword. It is no doubtable that the set in equation (2) is more probable.

Conventional post-Viterbi processor performs the correction of an occurred error event based on error-type and its likely error positions associated with the largest likelihood value among the outputs of all matched filters. The correction based on the largest one can cause mis-selection of an occurred error event in case that distinction between the largest one and the other ones is ambiguous. Therefore, for error correction, the consideration of the error types and their error starting positions corresponding to the other ones as well as the largest one can reduce mis-selection of an occurred error event. We call the process *list-correction*. The *list-correction* is performed by following steps.

Position	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
c	0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0
e	0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
r	0 0 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0
t	3 4 11 12 18 19 25 26 32 33
s	3 4 11 12 25 26

- H : parity check matrix
- c : transmitted codeword ($cH^T = [0\ 0\ 0]$)
- e : error event ([1 1])
- r : ML codeword ($S = rH^T = 4 = [1\ 0\ 0]$), where S is syndrome
- t : set of possible starting positions of the error event based on syndrome 4
- s : set of more probable starting positions achieved by *optimal position search*

Fig. 1. A way to find the possible positions for an occurred error event [-2 +2](=[1, 1]) based on its syndrome value 4.

TABLE 2
AN EXAMPLE OF A POST-VITERBI CORRECTION BASED ON *LIST-CORRECTION* WITH $L=2$

Likelihood value	Dominant error event	Error position
$4.8694e^{-001}$ (the 1st largest)	±[2, -2, 2, -2]	[148, 149, 150, 151]
$4.8214e^{-001}$ (the 2nd largest)	±[2, -2, -2]	[201, 202, 203]
$3.9771e^{-001}$	±[2,-2, 2,-2, 2, -2]	[80, 81, 82, 83, 84, 85]
$3.9625e^{-001}$	±[2,-2, 2,-2, 2]	[27, 28, 29, 30, 31]
$3.6407e^{-002}$	±[2, -2]	[137, 138]
$3.5399e^{-002}$	±[2,-2, 0, 2,-2]	[51, 52, 54, 55]

Step 1. Perform the error correction in recovered codeword based on error type j_k and its likely error starting position $i_k \in \{1,2,3,...,N\}$ corresponding to the *k-th* largest one, where k is initially 1, and generate the re-estimated codeword $\tilde{a}_{i_k}^{j_k} = \hat{a} + \mathrm{e}_{i_k}^{j_k}$.

Step 2. Compute syndrome of the re-estimated codeword $S = \tilde{a}_{i_k}^{j_k} H^T$, where $\boldsymbol{H}$ is a parity check matrix.

Step 3. $k = k+1$ if the syndrome is non-zero. Otherwise, stop the process and output $\tilde{a}_{i_k}^{j_k}$.

Step 4. Repeat 1~3 steps until $k > L$, where L is the maximum number of error correction through *list-correction*.

If syndrome after *L-th* correction is non-zero, it means that the correction fails, and then the processor outputs the recovered codeword with no correction. Conventional post-Viterbi processor just performs **Step 1** with L=1.

With (203, 200) CRC code based on $g(x) = 1+x^2+x^3$, Table 2 shows an example of the error correction of a new post-Viterbi processor based on *list-correction*. In Table 2, the second column shows the considered dominant error events, and the corresponding likelihood value and error position are given in the first and third columns, respectively. The result is obtained through simulation when actual occurred error type and its error starting position are [2 -2 2] and 201, respectively. In this example, the likelihood values corresponding to [2 -2 2 -2] and [2 -2 2] are the first and second largest ones among the maximum outputs of all matched filters, respectively. A conventional post-Viterbi processor accomplishes the error correction based on j_1 = [2 -2 2 -2] and i_1 = 148 associated with the largest one, and then it outputs re-estimated codeword $\tilde{a}_{i_k}^{j_k}$. As a result, the output includes more errors than before correction and the result stems from mis-selection of an occurred error event [2 -2 2] to [2 -2 2 -2]. On the other hand, a post-Viterbi processor based on *list-correction* with L = 2 performs error correction using j_2 = [2 -2 2] and i_2 = 201 corresponding to the second largest one, again, and then it outputs the re-estimated codeword with zero-syndrome (in other words, no error). As shown to an example of Table 2, we can identify that the efficiency of the *list-correction* is increased in case it is ambiguous that we judge which one between maximum outputs (for example, the first and second largest ones) of each matched filter is more reliable. Table 3 shows an example of a post-Viterbi correction using both *optimal position search* and *list-correction* with L = 2 based on (203, 200) CRC code with $G(x) = 1+x^2+x^3$. The result is obtained through simulation when the syndrome of the recovered codeword is 2, and an occurred error event and the corresponding error starting position are ±[2 -2] and 88, respectively. In Table 3(a), for example, *optimal position search* for an error event ±[2 -2 2 -2] is performed as the following. The error event starts to occur in the 1^{st} position in a recovered codeword because syndrome value computed from recovered codeword is 2, and thus a set of likely error starting positions for ±[2 -2 2 -2] becomes {1, 8,…,7k-6}, where k=1, 2, 3,…, N/7, since the syndrome sequence for each dominant error event is a repeated version of a period-7 syndrome sequence. Moreover, from the set, by eliminating positions in codeword without possibility that ±[2 -2 2 -2] occurs, set of optimal error starting positions of the error event S_3=[29, 106, 148] is obtained. After *optimal position search* for the considered error events in Table 3(a), each matched filter finds and outputs a position with the maximum likelihood value among the corresponding possible error starting positions, which are all elements of S_j. As example, a matched filter associated with ±[2 -2 2 -2] outputs the likelihood value $5.2407\times e^{-2}$ corresponding to position 106. Finally, in Table 3(b), based on the error type ± [2 -2] and its error positions [88 89] associated with the largest one among outputs of all matched filters, *list-correction* is performed and then re-estimated codeword with zero-syndrome is generated. Accordingly, *list-correction* is completed without processing until a given L=2. The reason is because distinction between the first and second largest ones is noticeable. If there is ambiguity, the error correction may be performed twice (L=2). From Table 3(a), we can observe that the size of S_j is smaller when length of an error event is longer. The fact implies that probability of mis-positioning of error event with longer length is much less than that of one with smaller length.

Figure 2 shows the block diagram of a new post-Viterbi processor. Here, K and L are the number of the considered dominant error events and the maximum number of error correction by *list-correction*, respectively, and $K \geq L$. In Figure 2, bolded block and the routine for *list-correction* are added to reduce mis-correction of conventional post-Viterbi processor. Bolded block plays a significant role in decreasing the probability of mis-positioning based on a set of more probable error starting positions associated with each detectable error event generated through *optimal position search*. An error detection code belongs to two valuable properties. The first one is error detection and the second one is the generation of likely error positions of a detectable error event using periodic property of syndrome sequence. O*ptimal position search* can be realized through the utilization of the second property. However, in conventional post-Viterbi processor, error detection code is not used to generate a set of likely error starting positions, but it is utilized to only check

TABLE 3
AN EXAMPLE OF ERROR CORRECTION OF A NEW POST-VITERBI PROCESSOR

(a) *OPTIMAL POSITION SEARCH* (S_j)

Dominant error event	A set of error starting positions (S_j)
±[2,-2]	[4, 11, 18, 25, 32, 39, 46, 60, 74, 81, 88, 95, 102, 109]
±[2,-2, 2]	[26, 82, 89, 131, 159, 173, 180, 201]
±[2,-2, 2,-2]	[29, 106, 148]
±[2,-2, 2,-2, 2]	[27]
±[2,-2, 0, 2,-2]	[23, 37, 177]
±[2,-2, 2,-2, 2,-2]	[80]

(b) *LIST-CORRECTION* WITH L=2

Likelihood value	Dominant error event	Error position
$4.5966e^{-001}$(the 1^{st} largest)	±[2,-2]	[88, 89]
$2.2748e^{-001}$	±[2,-2, 0, 2,-2]	[37, 38, 40, 41]
$2.0321e^{-001}$	±[2,-2, 2]	[131, 132, 133]
$1.3347e^{-001}$	±[2,-2, 2,-2, 2]	[27, 28, 29, 30, 31]
$7.8609e^{-002}$	±[2,-2, 2,-2, 2, -2]	[80, 81,82,83, 84, 85]
$5.2407e^{-002}$	±[2,-2, 2,-2]	[106, 107, 108, 109]

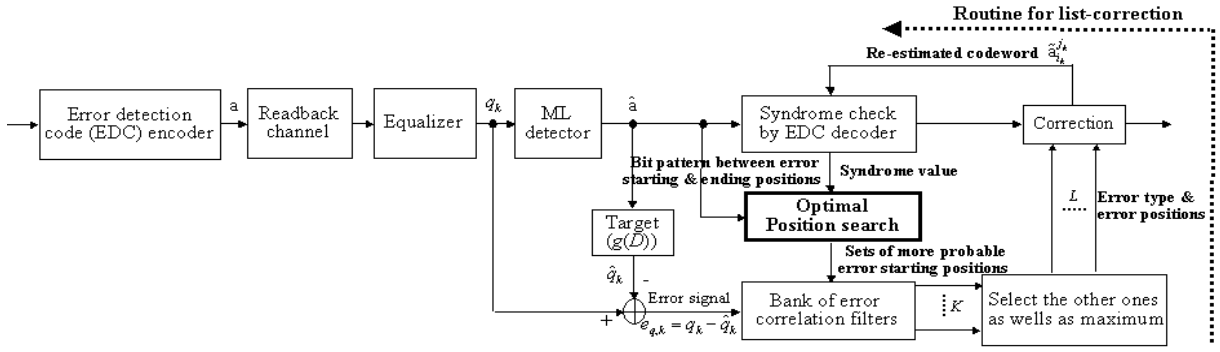

Fig. 2. Block diagram of a new post-Viterbi processor based on both *optimal position search* and *list-correction* with a given *L*.

for the presence of errors in recovered codeword. The utilization of the second property can improve performance of conventional post-Viterbi processor. Routine of *list-correction* plays a significant role in lessening mis-selection of an occurred error event through the consideration of the other ones as well as the largest likelihood value. Mis-selection occurs in case that a likelihood value corresponding to actual error-type does not become the largest one due to colored noise. *List-correction* tries to minimize the impact of the colored noise through at most *L*-time correction. In case that *L* is 1 and *optimal position search* block is removed, the scheme is the same as the conventional post-Viterbi processor. Figure 2 operates as follows. Data is encoded by an error detection code encoder to produce the codeword (a) and then the codeword is transmitted through the noisy channel. An equalizer reshapes the transmitted codeword into equalizer target response. The output of an equalizer (q_k) is fed to the ML detector and the ML detector recovers the transmitted codeword based on the output of the equalizer. An error detection code decoder computes the syndrome of the recovered codeword (â). When the syndrome is non-zero, the scheme generates the error signal ($\hat{q}_k$) and performs *optimal position search*. Based on the error signal and possible error starting positions for each dominant error event, each matched filter finds a position yielding the maximum likelihood value and then it outputs likelihood value associated with the position. Finally, based on the error types and their error positions corresponding to the first and second largest ones, and so on among these likelihood values, the scheme performs *list-correction* with a given *L* and then it generates re-estimated codeword ($\tilde{a}_{i_k}^{j_k}$). The validation of the correction depends on syndrome of the codeword. If the syndrome is zero, the process is halted. Otherwise, the process is repeated until at most *L*. It means that the error correction fails in case that a codeword with zero-syndrome is not found until *L-th* error correction.

IV. SIMULATION RESULTS

A (203, 200) CRC code generated by $G(x) = 1+x^2+x^3$ is used as an error-detection code and the code can detect the dominant error events for perpendicular recording [10][11]. The bit error rates (BERs) of conventional and new post-Viterbi processors are simulated and compared at user density D_u=1.4 with the equalizer target response of $g(D) = 1 + 6D + 7D^2 + 2D^3$ over perpendicular recording. User density D_u is given by $D_u = R_{CRC} \times D_c$, where R_{CRC} is code rate of the CRC code and D_c is channel density. The dominant error events (K=6) at user density 1.4 are the same as the first column in Table 1. The maximum number of error correction through *list-correction L* is 3. As a reference, BER of PRML (1, 6, 7, 2) system and the ideal BER (bound), which is obtained under assumption that the correction of any single dominant/non-dominant error event occurred in recovered codeword is perfect, are also shown at the same user density. The signal-to-noise ratio (SNR) has been defined in [11] as the energy of the first derivative of the transition response E_{dt} to the noise spectral density. The noise parameter N_{50} in the SNR definition signifies a mixture of 50% additive white Gaussian noise and 50% jitter noise. Figure 3 shows BERs of conventional and new post-Viterbi processors. It is seen that both post-Viterbi processors produce considerable performance gains compared to PRML (1, 6, 7, 2) system. In particular, new schemes provide more performance gain than the conventional one. The fact implies that new schemes

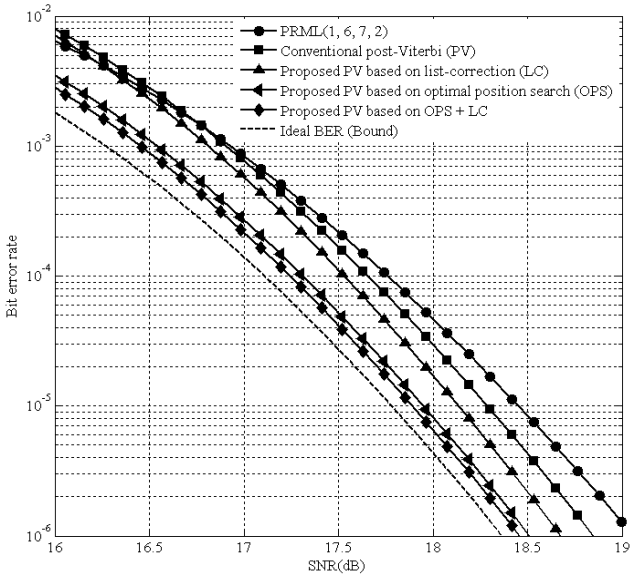

Fig. 3. Comparison of BERs at user density 1.4.

yield rare mis-correction through *optimal position search* and *list-correction* with L=3, while conventional scheme suffers from unignorable mis-correction for the set of dominant error events. The performance loss of conventional scheme stems from generation of either an error event of the same length as the occurred dominant error event due to mis-positioning, or an error event even longer than an occurred error event due to mis-selection. However, the frequency is significantly reduced through application of both *optimal position search* and *list-correction* to conventional scheme. From performance comparison of the schemes using *optimal position search* (*OPS*), *list-correction* (*LC*) or both techniques, we estimate that most of mis-correction is generated by mis-positioning because scheme using *OPS* outperforms scheme using *LC*. Meanwhile, the performance of the scheme using both techniques is close to that of the scheme using *optimal position search* and their performance difference is not noticeable. The result implies that *optimal position search* itself considerably reduces the probability of mis-selection as well as mis-positioning.

An (n, k, t) Reed–Solomon (RS) code can correct up to t symbol errors in an n-symbol code block that includes k information symbols. Here, we attempt to compute the symbol error rate (SER), defined as the ratio of the number of uncorrectable sectors to the total number of received sectors. One popular way of doing this calculation is based on the multinomial distribution for the probabilities of the symbol error events with different lengths [4][6]-[9][13].

For each length-i symbol error (i.e., i consecutive erroneous RS symbols), where $i = 1, \ldots, k$, let x_i and p_i respectively be the number and the probability of occurrence. The probability density function based on the multinomial distribution is then described by [6][10]:

$$f_{X_1,\cdots,X_k}(x_1,\cdots,x_k) = \frac{n!}{x_1!x_2!\cdots x_{k+1}!}\cdot p_1^{x_1}p_2^{x_2}\cdots p_{k+1}^{x_{k+1}} \quad (3).$$

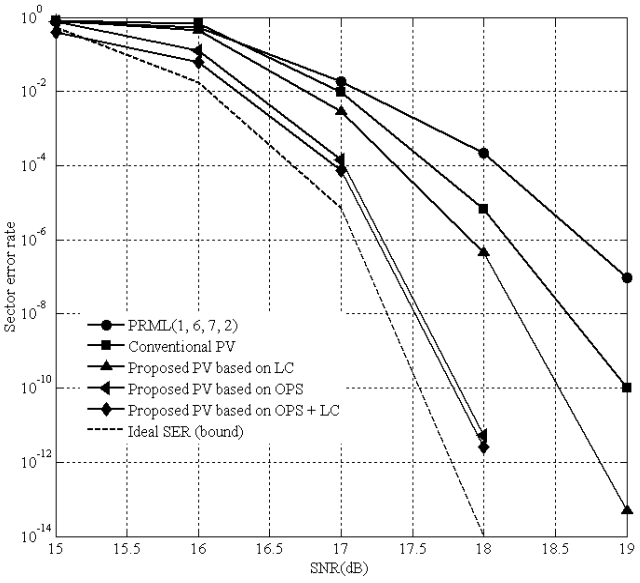

Fig. 4. Comparison of SERs at user density 1.4.

Here, x_{k+1} and p_{k+1} denote the number and the probability of no erroneous symbols in a sector, and accordingly, $\sum_{i=1}^{k+1} p_i = 1$ and $\sum_{i=1}^{k+1} x_i = n$. The probabilities p_i's are estimated from the captured symbol error statistics.

To compute the SER for 512-information-byte sectors, a shortened RS code based on *Galois field GF* (2^{10}) is used [9]. The overall user density D_u' taking into account the outer RS code is defined as $D_u' = D_u \times R_{RS}$, where R_{RS} is the code rate of the outer RS code. Since an (n, k, t) shortened RS code can correct up to t symbol errors in a sector and does not require an interleaving, the probability of an uncorrectable sector, which is the SER, is simply given by

$$P_{\text{sector}} = 1 - \sum_{x_1}\cdots\sum_{x_k} f_{X_1,\cdots,X_k}(x_1,\cdots,x_k), \quad (4)$$

where the sum is over all combinations of $x_1,\cdots,x_k$ such that $\sum_{i=1}^{k} i\cdot x_i \le t$.

Figure 4 compares the SERs of PRML (1, 6, 7, 2) system and conventional and new post-Viterbi processors using a (420, 410, 5) outer RS code. The overall user density D_u' is approximately 1.33. The channel density D_c for post-Viterbi processor is around 1.35. The proposed schemes attain the SER improvement and specially, the performance of schemes using *optimal position search* or both techniques is noticeable and is close to that of ideal SER. The SNR gain is seen to be around 1.2 dB at SER = 10^{-10} compared to conventional scheme.

V. CONCLUSION

The paper has proposed two methods for reducing mis-correction, which is main source of bit errors in conventional post-Viterbi processor based on error detection code. One is *optimal position search* for reducing mis-positioning in error-location, and the other is *list-correction* for reducing mis-selection in error type. The SNR gains of new system using these techniques relative to conventional scheme have been evaluated at both the same BERs as well as at the same SERs at the output of the RS decoder. The gains are significant. Thus, we conclude that new techniques can be a candidate as desirable solution for future high-capacity storage systems.

REFERENCES

[1] L. L. McPheters and S. W. McLaughlin, "Turbo-coded optical recoding channels with DVD minimum mark size," *IEEE Trans. Magn*, vol.38, no.1, pp.298-302, Jan. 2002.

[2] J. Li, K.R. Narayanan, E.M. Kurtas, and C.N. Georghiades, "On the performance of turbo product codes and LDPC codes over PR channels," *IEEE Trans. Commun.*, vol.50, no.5, pp.723-734, May 2002.

[3] H. Song, B. V. K. V. Kumar, E. M. Kurtas, Y. Yuan, L. L. McPheters, and S.W. McLaughlin, "Iterative decoding for partial response (PR), equalized, magneto-optical (MO) data storage channels," *IEEE J. Selected Areas Commun.*, vol.19, no.4, pp.774-782, April 2001.

[4] T. Conway, "A new target response with parity coding for high density magnetic recording channels," *IEEE Trans. Magn.*, vol. 34, no. 4, pp. 2382–2386, July 1998.
[5] K. Cai and K.A.S. Immink, "A General construction of constrained parity-check codes for optical recording," *IEEE Trans. on Commun*., vol. 55, no. 7, pp. 1070-1079, July 2008.
[6] R. D. Cideciyan, J. D. Coker, E. Eleftheriou, and R. L. Galbraith, "Noise predictive maximum likelihood detection combined with parity-based post-processing," IEEE Trans. Magn., vol. 37, no. 2, pp. 714–720, Mar. 2001.
[7] W. Feng, A. Vityaev, G. Burd, and N. Nazari, "On the performance of parity codes in magnetic recording systems," in *Proc. IEEE GLOBECOM 2000*, pp. 1877–1881.
[8] K. Saeki and Z. Keirn, "Optimal combination of detection and error correction coding for magnetic recording," *IEEE Trans. Magn*., vol. 37, no. 2, pp. 708–713, Mar. 2001.
[9] Z. A. Keirn, V. Y. Krachkovsky, E. F. Haratsch, and H. Burger, "Use of redundant bits for magnetic recording: Single-parity codes and Reed Solomon error-correcting code," *IEEE Trans. Magn.*, vol. 40, no. 1, pp. 225–230, Jan. 2004.
[10] J. Moon, J. Park and J. Lee, "CRC-based high-rate error detection code for perpendicular recording," *IEEE Trans. on Magn*., vol. 42, no. 5, pp 1626-1628, May 2006.
[11] J. Moon and J. Park, "Detection of prescribed error events: Application to perpendicular recording," in *Proc. IEEE ICC 2005*, vol. 3, pp. 2057–2062, May 2005.
[12] S. Lin and D. J. Costello, Error control coding : Fundamentals and application (second edition), Prentice Hall, 2004.
[13] W. G. Bliss, R. Karabed, K. Zhang, C. C.Varanasi, and J. Ashley, "Sector error rate estimation of concatenated codes in magnetic recording," in *Proc. IEEE ICC 2001*, pp. 2726–2730.

BIOGRAPHIES

Jun Lee received his B.S. and M.S. degree from Dongguk University, Seoul, Korea in 1998 and 2000, respectively. Since March 2000, he has been a Ph.D. student in Dept. of Electronic Engineering at Dongguk University. In 2003, In 2003, he received Ph. D. degree and he joined the faculty of Samsung Advanced Institute of Technology (SAIT), Suwon, Korea, and he is currently working with LG Electronics. His research interests are signal processing and coding for storage systems and communication theory.

Kees Schouhamer Immink received his PhD degree from the Eindhoven University of Technology. He was with Philips Research Labs in Eindhoven from 1968 till 1998. He founded and became president of Turing Machines Inc. in 1998. He is, since 1994, an adjunct professor at the Institute for Experimental Mathematics, Essen University, Germany. Immink designed coding techniques of virtually all consumer-type digital audio and video recording products, such as Compact Disc, CD-ROM, CD-Video, Digital Audio Tape recorder, Digital Compact Cassette system, DCC, Digital Versatile Disc, DVD, Video Disc Recorder, and Blu-ray Disc. He received widespread recognition for his many contributions to the technologies of video, audio, and data recording. He received a Knighthood in 2000, a personal 'Emmy' award in 2004, the 1996 IEEE Masaru Ibuka Consumer Electronics Award, the 1998 IEEE Edison Medal, 1999 AES Gold Medal, and the 2004 SMPTE Progress Medal. He was named a fellow of the IEEE, AES, and SMPTE, and was inducted into the Consumer Electronics Hall of Fame, and elected into the Royal Netherlands Academy of Sciences and the US National Academy of Engineering. He served the profession as President of the Audio Engineering Society inc., New York, in 2003.