Error Event Corrections Using List-NPML Decoding and Error Detection Codes

Suayb S. Arslan, Jaewook Lee, and Turguy Goker

Quantum Corporation Irvine, CA 92617 USA

A List-Noise Predictive Maximum Likelihood (List-NPML) decoding algorithm based on a periodic error detection mechanism is proposed for magnetic recording systems to minimize the number of error events. The proposed detector keeps a list of candidate paths (N candidates per state of a trellis) based on the observation that most of the error events can be recovered by finding a set of most likely paths. A periodic decision making process is utilized for every $\mathcal{P}$ bits based on error detection codes. With this approach, a tradeoff between performance and complexity is studied with various combinations of N and $\mathcal{P}$. The proposed structure is robust to miscorrections and time-varying error events, eliminating the need for knowing the error event distributions prior to its operation. We also introduced a novel design of parity bits that meets the run length constraints of the channel and a trellis update architecture for improved performance. Simulation results show that the proposed List-NPMLD gives us significant BER performance and post-ECC gains at the expense of some increase in complexity.

***Index Terms*—Data detection, error correction coding (ECC), error detection, magnetic recording, Partial Response 4 (PR4).**

I. Introduction

A TYPICAL magnetic recording read channel system is constituted of a concatenation of channel decoder, Run Length Limited (RLL) code decoder and Error Correction Code (ECC) decoder. The Partial Response Maximum Likelihood (PRML) detector for Partial-Response (PR) Class-4 channels achieves the near optimal performance at low normalized linear densities $D_c = PW50/T$, where $PW50$ is the pulse width measured at half the peak amplitude of the channel's step response and T is the bit period [1]. However, at higher linear recording densities such as $D_c > 2.5$, it is well known that PR4 equalizer leads to a dramatic noise enhancement. In the past, a subclass of generalized partial response polynomials (EPRML^4 or E^{2}PRML^4) as well as generalized partial response polynomials with non-integer coefficients are shown to match to the recording channel frequency characteristics better and therefore used to suppress noise enhancement at the expense of increased complexity [2]. In particular, a Noise Predictive Maximum Likelihood Detection (NPMLD) is proposed in [3]. This study uses a polynomial of the form $G(D) = [1 - D^2)(1 + P(D)]$ where $P(D) = p_1 D + p_2 D^2 + \cdots + p_L D^L$ is the polynomial of the noise whitening filter. Such a Finite Impulse Response (FIR) filter is introduced in order to approximately whiten the noise at the input of the Maximum Likelihood (ML) detector at the expense of larger number of trellis states and increased decoding complexity. A feedback loop in the trellis is used to reduce the complexity of the detection algorithm at the expense of some loss in performance.

The NPMLD is recently used in conjunction with post-Viterbi processing methods for increased performance, particularly to correct some of the dominant error events [4]. Such schemes combined with various detection codes are shown to be helpful when the frequency of error-event occurrences at the output of the NPMLD is uneven and known to the post processor, as discussed in [5] and [6]. One of the advantage of the post processing is the low complexity implementation. However, post processors are suboptimum solutions and usually not robust to miscorrection of error events. In addition, it is reported that post-ECC gains are not as much as the pre-ECC gains [5]. We note that the frequency of occurrences of such error events are functions of system parameters such as the recording density and the physical conditions of the read/write heads and media. As such parameters might change in time, the frequency of occurrences of error events at the output of the NPMLD changes [2], [7], requiring an adaptive system to be able see reported performance gains.

In this study, a List-NPMLD based on Error Detection Codes (EDCs) is introduced as an alternative detection algorithm that can either be used by its own or incorporated with one of the popular post processing methods for improved performance. Unlike other post processors, the proposed scheme is shown to be robust to miscorrections and time-varying error events due to a verification stage using EDCs.

II. System Model

We consider the serially concatenated block diagram shown in Fig. 1(a). User data initially goes through a Reed-Solomon (RS) encoding. We use a combined modulation/error-detection code to satisfy run length requirements of ones and zeros. A coded bit stream REF A goes through an EDC insertion in which it is splitted into $\mathcal{P}$-bit chunks (period) and equal amount of EDC parity bits are inserted at the end of each $\mathcal{P}$-bit chunks. This insertion is made so that the violation to the modulation constraints is minimum. Later, the data is mapped into the symbol sequence $a_n \in \{+1, -1\}$ and written on a storage medium for readback. Read signal goes through a Low Pass Filter (LPF) and an Analog-Digital Converter (ADC) before the PR4 equalization. Let the samples at the output of the PR4 equalizer be y_n at time nT, where n is the running index. Thus, y_n constitutes the PR4 signal plus the distortion on which the List-NPMLD will perform the sequence estimation, where $y_n = a_n - a_{n-2} + w_n$ and w_n is the total distortion due to white and/or colored noise as well as the residual error after quantization and equalization. After the whitening filter, the output symbol stream z_n is sent as an input to the proposed List-NPMLD, where $z_n = y_n - \sum_{i=1}^{L} y_{n-i} p_i$.

Manuscript received November 02, 2012; revised February 12, 2013; accepted February 22, 2013. Date of current version July 15, 2013. This work was presented at the 12th Joint MMM/Intermag conference, Session No: HW-12, Chicago, USA, Jan. 2013. Corresponding author: S. S. Arslan (e-mail: Suayb.Arslan@Quantum.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2013.2249658

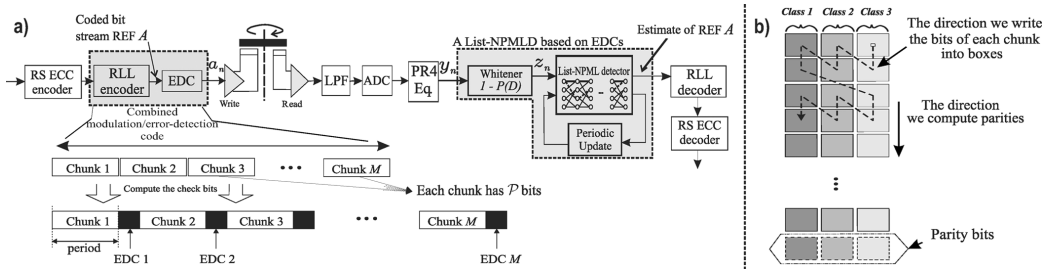

Fig. 1. a) System model used to record and readback the user data using the proposed List-NPMLD based on error detection codes. EDC: Error Detection Code. RLL: Run Length Limited. PR4 Eq.: Partial Response 4 Equalization. RS ECC: Reed Solomon Error Correction Coding. b) Computation of the three parity bits based on the current chunk of $\mathcal{P}$-bits. We assume that $\mathcal{P}$ is divisible by 3 in this example. EDC codewords need not be systematic as shown.

The List-NPMLD starts decoding the corresponding incoming symbol sequence of the first chunk using the feedback bits from different path memories and passes the possible set of candidate paths to the update module. The update module periodically makes a decision on the correct path based on the result of EDC decoding and updates the accumulated metrics which are then passed on to the List-NPMLD for further processing. Decoding of each chunk follows the same steps. Since a decision is made after each update step, the algorithm continually outputs the decoded bits.

III. Background

Noise Predictive Maximum Likelihood Detection: NPMLD is a sequence detection algorithm with an extra imbedded prediction stage at the output of an equalizer [3]. An NPMLD takes the PR4 equalized output y_n and finds the maximum likely path in the trellis. At time n, the cost due to the state transition $s_k \rightarrow s_j$ is given by [3]

$$c_n(s_k, s_j) = \left[z_n + \sum_{i=K+1}^{L+2} \hat{a}_{n-i}(s_k) g_i + \sum_{i=1}^{K} a_{n-i}(s_k) g_i - a_n \right]^2 \quad (1)$$

where $\hat{a}_{n-i}(s_k)$ are the past decisions in the path history associated with the state s_k and the symbols $\{a_{n-i}(s_k)\}_{i=1}^{K}$ in the second summation represent the state information. The coefficients $\{g_i\}_{i=1}^{L+2}$ are given by the polynomial multiplication $G(D) = (1-D^2)[1-P(D)] = 1 - g_1 D - \cdots - g_{N+2} D^{L+2}$.

As can been seen, the effective intersymbol interference length of the NPMLD is $L+2$. A large K increases the number of states of the NPMLD and decreases the length of the imbedded feedback whereas a lower K exhibits a reverse trend. Due to unreliable past decisions, a long feedback will usually degrade the performance of the system. This way, a complexity/performance trade-off can be achieved.

Design of Error Detection Codes: EDCs are suitable functions that add a fixed-length redundancy (a tag) to a message for error detection. After the channel, the tag is recomputed and compared with the original tag in order to decide whether there has been any change in the original message during transmission. We either add an m-bit CRC or m-bit parity to each $\mathcal{P}$-bit chunk to produce a $(\mathcal{P}+m)$-bit EDC codeword (Fig. 1(a)). Note that CRC code design is independent of the error event distribution of the current system. For parity code, we compute a parity bit value per each row (a bit that is added to the data bits in each row to make the modulo 2 sum of the total bits including the parities zero) and therefore add three parity bits to each $\mathcal{P}$-bit chunk (see Fig. 1(b)). Method of addition of bits minimizes the violation to run length constraints, by placing the parities to unconstrained bit positions of the modulation code. Parities are generated to maximize the detection probability of dominant error events at the output of the Viterbi detection. Assuming that only a single error event happens within a chunk, we can show that this 3-bit parity code can detect various error events. In general, if at least one of the classes of bits have odd number of bit errors, the designed parity code will perform successfully.

The main difference of a parity code is that each bit is designed to detect errors for the corresponding class only, whereas the CRC code considers all three classes together.

IV. List-NPMLD Based on EDCs

Notation: Let $\phi_n(j,l), 1 \leq l \leq N$ denote the l-th lowest accumulated metric to reach state j (s_j) at time n from some starting state at time $\hat{n} < n$. At time n, let $\beta_n(j,l)$ denote the state covered by the l-th best path at time $n-1$, which passes through state j at time n. Similarly, $r_n(j,l)$ characterizes the ranking of the l-th best path at time $n-1$, when this path passes through state j at time n. Since path memories are used in the branch metric computation of reduced-state noise predictive detector architectures, branch metrics corresponding to a state transition $s_j \rightarrow s_k$ using distinct paths arriving s_j are not necessarily the same. For example, the best path and the second best path arriving s_j have the same trellis feedback (tentative decisions) for the current branch metric computation. Thus, we denote the incremental branch metric that corresponds to a state transition $s_j \rightarrow s_k$ using the l-th best path of s_j at time n as $c_n^{(l)}(s_j, s_k)$.

A List-NPMLD Algorithm: The proposed algorithm is a combination of periodic updates and a parallel implementation of the List-NPMLD algorithm which simultaneously produces a rank ordered list of the N globally best candidates for each state in a trellis while maintaining the imbedded noise prediction. For the List-NPMLD at time n, the cost due to the state transition $s_k \rightarrow s_j$ (assuming that s_j and s_k are adjacent states), using the l-th best path of s_j at time n is given by

$$c_n^{(l)}(s_k, s_j) = \left[z_n + \sum_{i=K+1}^{L+2} \hat{a}_{n-i}^{(l)}(s_k) g_i + \sum_{i=1}^{K} a_{n-i}(s_k) g_i - a_n \right]^2 \quad (2)$$

where $\hat{a}_{n-i}^{(l)}(s_k)$ is the past decision in the l-th best path history associated with s_k and $a_{n-i}(s_k)$ is due to the hypothetical state transition $s_k \rightarrow s_j$. If $n-i<0$, then $\hat{a}_{n-i}^{(l)}(s_k)$ is assumed to be

"-1". Also, if s_k and s_j are not adjacent, then $c_n^{(l)}(s_k, s_j) = \infty$. The operation of a parallel List-NPMLD algorithm implementation for decoding the ω-th chunk, $\omega = 1, 2, \ldots, M$ is summarized in Algorithm 1.

Algorithm 1 A List-NPMLD algorithm for ω-th chunk

Initialization: $n = (\omega - 1)\mathcal{P} + 1$,

for $1 \le j \le 2^K$ **do**

for $1 \le l \le N$ **do**

if $\omega = 1$ **then**

$\phi_n(j, l) = c_n^{(1)}(s_1, s_j)$ and $\beta_n(j, l) = 1 \triangleq j^*$.

else

$\phi_n(j, l) = \phi_{n-1}(j^*, m^*) + c_n^{(l)}(s_{j^*}, s_j)$,
$\beta_n(j, l) = j^*$ where j^* is the state at which the previous path decision ends and m^* is the ranking of that path at state j^*.

end if

end for

end for

Recursion:

for $(\omega - 1)\mathcal{P} + 1 < n \le \omega\mathcal{P}$ **do**

for $1 \le j \le 2^K$ **do**

for $1 \le l \le N$ **do**

$$\phi_n(j, l) = \min_{\substack{1 \le k \le 2^K \\ 1 \le t \le N}}{}^{(l)} \left\{ \phi_{n-1}(k, t) + c_n^{(t)}(s_k, s_j) \right\}$$

$$(\beta_n(j, l), r_n(j, l)) = \arg\min_{\substack{1 \le k \le 2^K \\ 1 \le t \le N}}{}^{(l)} \left\{ \phi_{n-1}(k, t) + c_n^{(t)}(s_k, s_j) \right\}$$

where $\arg\min^{(l)}$ outputs the pair (k^*, t^*) that gives the l-th smallest value.

end for

end for

end for

Decision: Find l-th best path at time $n = \omega\mathcal{P}$

$$(j_{\omega\mathcal{P}}^{(l)}, m_{\omega\mathcal{P}}^{(l)}) = \arg\min_{\substack{1 \le k \le 2^K \\ 1 \le t \le N}}{}^{(l)} \phi_{\omega\mathcal{P}}(k, t)$$

Then, l-th ($l \in \{1, \ldots, N\}$) most likely state sequence is given by $(j_{(\omega-1)\mathcal{P}+1}^{(l)}, j_{(\omega-1)\mathcal{P}+2}^{(l)}, \ldots, j_{\omega\mathcal{P}-1}^{(l)}, j_{\omega\mathcal{P}}^{(l)})$ where $j_n^{(l)} = \beta_{n+1}(j_{n+1}^{(l)}, m_{n+1}^{(l)})$ and $m_n^{(l)} = r_{n+1}(j_{n+1}^{(l)}, m_{n+1}^{(l)})$ for $(\omega - 1)\mathcal{P} + 1 \le n \le \omega\mathcal{P} - 1$. We also have $j_{(\omega-1)\mathcal{P}+1}^{(l)} \triangleq j^*$ and $m_{(\omega-1)\mathcal{P}+1}^{(l)} \triangleq m^*$.

Update Step and Decision Making: As the trellis evolves in time, the algorithm eliminates half of the possible paths at each time step for each state (smallest N out of $2N$). At the end of each $\mathcal{P}$-bits period, a decision is made and accumulated metrics are updated. The reason of employing a periodic update is to increase the probability of correction of error events. This is achieved by saving the algorithm from eliminating the candidate paths. For example, for large $\mathcal{P}$, the algorithm eliminates more candidates than it does for small choices of $\mathcal{P}$. It is more likely for large $\mathcal{P}$ that the algorithm discards the actual path (the path without error). This suggests for a fixed N, if $\mathcal{P}$ is large, we expect less error correction. On the contrary, we can choose $\mathcal{P}$ small to increase the probability of correction. However, having small $\mathcal{P}$ implies more frequent updates and more EDC bits unless some other mechanism is used for detection. Therefore, the proposed algorithm offers a tradeoff that an increased performance is possible at the expense of a decreased user density.

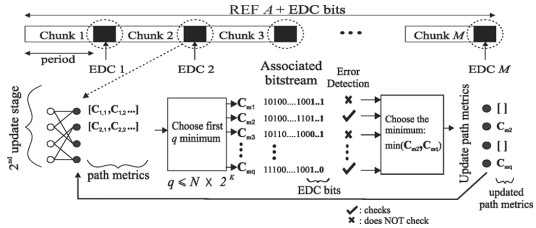

Fig. 2. Update stage of the proposed algorithm. EDC: Error detection code.

The details are best explained by an example shown in Fig. 2. Let $N = 2$ and $K = 2$. The update step chooses the first q smallest metrics out of $2^K N$ possible paths where $q \le 2^K N$. We put a constraint on the set of possible paths to help the detection capability of actual EDCs, since candidate paths might exhibit different characteristics of error and cause EDC to fail. By choosing an appropriate q value, we can prevent the EDC from checking other paths that potentially may be causing more error events than is the maximum likely path. Error detection is performed based on the smallest q accumulated metrics (i.e., the ordered set $\mathcal{C}_q \triangleq \{\mathbf{C}_{m1}, \mathbf{C}_{m2}, \ldots, \mathbf{C}_{mq}\}$). In Fig. 2, a "✓" denotes that the EDC does not detect any error and flags a 1, a "×" denotes that the EDC does not check and flags a 0. In this example, second and q-th elements of $\mathcal{C}_q$ flag 1s. Using practical EDCs, more than one path may flag 1 as shown in this example. The algorithm chooses the path whose EDC flags a 1 and has the smallest accumulated metric. Finally, accumulated metrics are updated; the accumulated metrics of all the paths except the second and q-th paths are discarded i.e., set to ∞. If EDCs indicate $\times$ for all the candidates, then the algorithm chooses the path with $\mathbf{C}_{m1}$. We do not update the accumulated metrics if there is no path with a flag 1.

V. Numerical Results

We consider PR4 signalling and set $K = 2$ (4-state trellis) and $L = 3$ (3 bits in feedback). We start with a Lorentzian channel model [8]. Electronics and stationary transition noise samples are added to the signal waveform at the input of the LPF as modeled in [7] i.e., w_n is modeled as a mixture of electronics and transition noises. Signal to Noise Ratio (SNR) is computed at the input of the LPF and given by $2/(N_0 + N_m)$ where $N_0/2$ and $N_m/2$ are the two sided spectral densities of a white Gaussian noise (i.e., electronic noise) and a colored noise (i.e., transition noise) sources. We approximate the ratio of the transition noise power to the total noise power by $\beta = N_m/(N_0 + N_m)$. LPF and PR4 equalization are done as in [7]. Whitening filter coefficients $\{p_i\}_{i=1}^L$ are selected using linear prediction in Least Mean Squares (LMS) sense based on the current noise samples w_n.

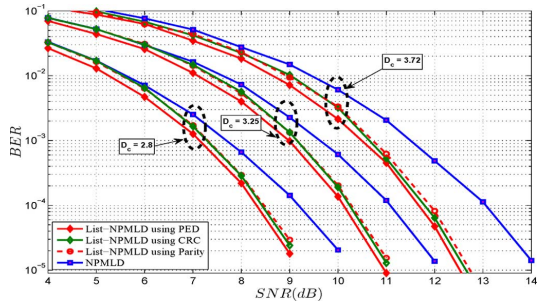

Fig. 3. Simulation result using Lorentzian channel model and various D_c values. We set $\beta = 0.5$.

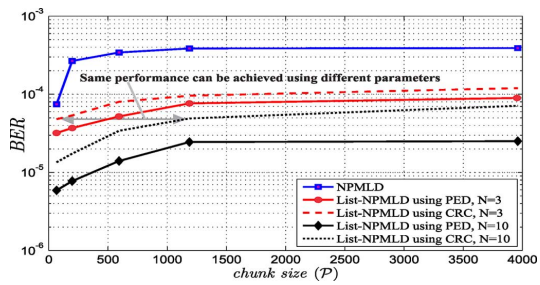

Fig. 4. Performance as a function of period size ($\mathcal{P}$).

We show the performance of the proposed system using a CRC code with a generator polynomial $X^4 + 1$ and three parity bits as described. We set $\mathcal{P} = 198$ bits, $\beta = 0.5$ and used $q = 6$ for $N = 3$. Note that with this parameter selections, we use 1 extra bit per 66-bit for error detection. In order for a fair comparison, we assume that a bit period T increases to $T_{NPMLD} = (67 \times T)/66$ when we simulate the NPMLD performance. We tested linear density values $D_c = 2.8, 3.25$ and 3.72 in Fig. 3 for the List-NPMLD system. This corresponds to simulating the NPMLD performance using $PW50/T_{NPMLD} \approx 2.76, 3.2$ and 3.66, respectively. The figure shows the performance of the detection codes relative to Perfect Error Detection (PED) case. The performance degradation due to using actual detection codes is shown to be minor and only noticeable at around 1e-2 to 1e-3 BER range. The gain increases slightly with growing D_c. This is because the proposed scheme treats the error events equally. We finally note that EDCs based on CRC and parity bits show similar results for the selected simulation parameters and gains increase with growing N. For example, we observed a gain of almost 2 dB over the NPMLD with $N = 50$ at $D_c = 3.25$ which is not shown here.

Fig. 4 shows BER performance at SNR 10.5 dB with varying $\mathcal{P}$. List-NPMLD uses $D_c = 3.25$ and corresponding densities for NPMLD are $\approx 3.11, 3.2, 3.23, 3.24, 3.25$ for chunk sizes 66,198,594, 1188 and 3960 bits, respectively. When $N = 10$, we set $q = 12 \leq 80$ to help EDC perform satisfactorily. The detection codes are observed to be challenged by the choice of q and the performance gap between the PED case and actual EDC increase as the List-NPMLD uses larger N. As expected, increasing the chunk size decreases the gain and the number of redundant bits. However, we can compensate the performance degradation by using larger N. The same figure, for example, shows that using CRC bits, we can get almost the same BER performance with $N = 3$ using $\mathcal{P} = 66$ bits and $N = 10$ using $\mathcal{P} = 1188$ bits.

TABLE I
PERFORMANCE RESULTS FOR A TAPE CHANNEL. NPMLD: NOISE PREDICTIVE MAXIMUM LIKELIHOOD DETECTOR, LNPMLD (N): LIST-NPMLD USING N CANDIDATE PATHS AND $\mathcal{P} = 198$ BITS. EE: ERROR EVENT

Error Event	NPMLD	LNPMLD(3)	LNPMLD(50)
"1"-bit EE	13365 (53.96%)	5854 (56.36%)	1843 (62.6%)
"2"-bit EE	4744 (19.15%)	2117 (20.4%)	550 (18.7%)
"3"-bit EE	3456 (13.9%)	1146 (11.03%)	194 (6.6%)
"4"-bit EE	1004 (4.05%)	338 (3.25%)	56 (1.9%)
"5"-bit EE	374 (1.51%)	102 (1%)	23 (0.8%)
TOTAL	24769 (100%)	10387 (100%)	2942 (100%)

Last, we consider a real Linear Tape Open (LTO) tape channel. Tested tape drive system has $D_c \approx 2.3$. We present the results in Table I, for chunk size $\mathcal{P} = 198$ bits and $N = 3$ and 50. SNR is computed at the output of the PR4 equalizer and is the equalized signal power over the total noise power. As a benchmark study, we assume PED and consider only the tape waveforms with average SNR satisfying 14.5 dB $\geq$ SNR ≥ 10 dB. We observe that the performance of the List-NPMLD improves with increasing N, at the expense of an increase in complexity. Moreover, we can correct approximately 58% and 88% of the error events using $N = 3$ and $N = 50$, respectively. The frequency of these dominant error events are shown to be roughly the same, suggesting the idea of using the proposed scheme with a post processing method targeting a specific error event distribution.

VI. CONCLUSION

We presented a List-NPMLD algorithm based on a periodic EDC updates for effective error event detection and correction. We used parity/CRC codes for detection of error events although any code with detection capability can be used. We show that the proposed scheme in conjunction with EDC codes might be an alternative detection algorithm that can either be used by its own or be incorporated with a traditional post processor for improved performance. We also note that the proposed scheme using CRC codes does not make any assumptions about the error event distributions. Simulation and test results show that the List-NPMLD detection with periodic updates based on error detection codes might be a viable solution for next generation magnetic recording systems.

REFERENCES

[1] R. D. Cideciyan, "A PRML system for digital magnetic recording," *IEEE J. Select. Areas Commun.*, vol. 10, no. 23, pp. 38–56, Jan. 1992.
[2] B. Vasic and E. M. Kurtas, *Coding and Signal Processing for Magnetic Recording Systems*. Boca Raton: CRC Press, 2004.
[3] E. Evangelos and W. Hirt, "Improving performance of PRML/EPRML through noise prediction," *IEEE Trans. Magn.*, vol. 32, no. 5, pp. 3968–3970, Sept. 1996.
[4] J. L. Sonntag and B. Vasic, "Implementation and bench characterization of a read channel with parity check postprocessor," in *Proc. TMRC*, Santa Clara, CA, Aug. 2000.
[5] R. D. Cideciyan, J. D. Coker, E. Evangelos, 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.
[6] J. Moon, J. Park, and J. Lee, "Cyclic redundancy check code based high-rate error-detection code for perpendicular recording," *IEEE Trans. on Magn.*, vol. 42, no. 5, pp. 1626–1628, May 2006.
[7] R. D. Cideciyan, E. Evangelos, and T. Mittelholzer, "Perpendicular and longitudinal recording: A signal-processing and coding perspective," *IEEE Trans. Magn.*, vol. 38, no. 4, pp. 1698–1704, July 2002.
[8] J. Moon, "Discrete-time modeling of transition noise dominant channels and study of detection performance," *IEEE Trans. Magn.*, vol. 27, no. 6, pp. 4573–4578, Nov. 1991.