A General Construction of Constrained Parity-Check Codes for Optical Recording

Kui Cai, *Member, IEEE,* and Kees A. Schouhamer Immink, *Fellow, IEEE*

***Abstract*—This paper proposes a general and systematic code design method to efficiently combine constrained codes with parity-check (PC) codes for optical recording. The proposed constrained PC code includes two component codes: the normal constrained (NC) code and the parity-related constrained (PRC) code. They are designed based on the same finite state machine (FSM). The rates of the designed codes are only a few tenths below the theoretical maximum. The PC constraint is defined by the generator matrix (or generator polynomial) of a linear binary PC code, which can detect any type of dominant error events or error event combinations of the system. Error propagation due to parity bits is avoided, since both component codes are protected by PCs. Two approaches are proposed to design the code in the non-return-to-zero-inverse (NRZI) format and the non-return-to-zero (NRZ) format, respectively. Designing the codes in NRZ format may reduce the number of parity bits required for error detection and simplify post-processing for error correction. Examples of several newly designed codes are illustrated. Simulation results with the blu-ray disc (BD) systems show that the new $d = 1$ constrained 4-bit PC code significantly outperforms the rate 2/3 code without parity, at both nominal density and high density.**

***Index Terms*—Constrained codes, finite state machine (FSM), parity-check codes, cyclic redundancy check (CRC) codes, post-processor.**

I. Introduction

MODULATION codes [1], [2], also known as constrained codes, are used in recording systems to translate an arbitrary sequence of user data into a sequence with special properties required by the systems. They are usually characterized by the so-called (d,k) constraints, or runlength constraints [2]. Binary sequences satisfying the (d,k) constraints have at least d and at most k, $k > d$, '0's between successive '1's. These constraints mitigate the problems of intersymbol interference and inaccurate timing. For optical recording, modulation codes also need to have the dc-free property [3], [4], *i.e.* they should have almost no content at very low frequencies. The dc-free constraint avoids interference between data and servo signals, and also permits filtering of low-frequency disc noise.

Development of 'efficient and powerful channel codes' is key to ensuring good reception performance under aggressive recording conditions. For optical recording, the d constraint has been reduced from $d = 2$ [3], [4] used in the compact disc (CD) and digital versatile disc (DVD) to $d = 1$ [5], [6] in the blu-ray disc (BD) or high-definition digital versatile disc (HD-DVD). In [7], Immink *et al.* have introduced a new family of finite-state modulation codes with $d = 1$ or $d = 2$ constraints, whose rates are higher than those of the standard codes, and are very close to the Shannon capacity.

To further improve the system performance at high recording densities, in recent years, the combination of Reed-Solomon (RS) outer codes and parity-check (PC) inner codes in conjunction with post-processing [8], [9], [10] has found wide acceptance in magnetic recording systems since the performance-complexity trade-off offered by these codes is very attractive and affordable. This approach also shows high potential for optical recording systems [11], [12]. The PC code is an inner error correction code (ECC), which can detect the specific dominant error events (*i.e.* the most likely error events that can occur) at the output of the channel detector, using only a few parity bits. For error correction, the matched-filtering type post-processor that combines syndrome and soft-decision decoding [10], [12] is widely used due to its simplicity. Cyclic redundancy check (CRC) codes [13] are simple and efficient error detecting PC codes, and have been used in [9], [10], [12]. In [14], specific event error detection code is proposed, which can detect any error event from an arbitrary list of error events.

When the PC constraints are imposed on the data, the modulation constraints should be satisfied simultaneously. This will result in additional code rate loss. The minimum overhead is one user bit per parity bit. Equivalently, $\frac{1}{R_c}$ channel bits are needed per parity bit, where R_c is the rate of the constrained code [11]. For example, for the rate $2/3$ $d = 1$ codes [5], [6] used in BD and HD-DVD, the minimum feasible overhead is 1.5 channel bits per parity bit. Let there be p parity bits per codeword of length n. Then, the capacity of constrained PC codes is given by

$$C_{pc} = \frac{(n - \frac{p}{R_c})R_c}{n} = R_c - \frac{p}{n}. \quad (1)$$

There have been several attempts in recent years to efficiently combine constrained codes with PC codes. For example, in the scheme described by Perry *et al.* [15], a constrained data sequence is parsed into shorter blocks of equal length, and a parity data block is inserted between each pair of these blocks. The data and parity blocks are connected such that the modulation constraints are not violated. The major disadvantage of this scheme is that it can only correct specific mixed-type errors.

Gopalaswamy and Bergmans [16] proposed concatenated coding to construct modulation codes with error detection

Paper approved by T.-K. Truong, the Editor for Coding Theory and Techniques of the IEEE Communications Society. Manuscript received September 6, 2006; revised July 23, 2007.

K. Cai is with Data Storage Institute, Singapore 117608 (e-mail: Cai_Kui@dsi.a-star.edu.sg).

K. A. Schouhamer Immink is with Turing Machines Inc., The Netherlands.

Digital Object Identifier 10.1109/TCOMM.2008.060502.

capability. In this scheme, the PC information is first calculated for each constrained data block. This information is then encoded by a standard constrained encoder and appended to the end of the corresponding data block. In this way, the proposed scheme achieves high coding efficiency. For the rate 2/3 $d = 1$ code, a parity bit requires 1.5 channel bits. However, in this scheme, the channel bit-stream corresponding to the parity bits is not protected by PCs. Therefore, errors occurring in this portion may cause further errors during decoding. This results in error propagation.

A combi-code scheme proposed by Coene *et al.* [11] achieves high efficiency similar to [16], and avoids the parity-bit related error propagation. In this scheme, the constrained PC code consists of two sliding block codes, which are designed to detect single-bit transition shift errors (*i.e.* a '1' in the (d, k) constrained channel bit-stream is shifted a single place to the left or the right). Because the two constituent codes are based on the same FSM, no additional channel bits are needed for stitching the two codes together. By using this scheme, efficient PC codes with $d = 2$ constraint, which achieve 2 channel bits per parity bit, have been designed. However, efficient combi-codes with $d = 1$ constraint are not available. Furthermore, this scheme is not general enough to detect any arbitrary error event or error event combinations. The efficiency of this approach can be further improved by using the method proposed in [7].

In the write path of a data storage system, a precoder, *i.e.* a modulo-2 integration operation, converts the binary outputs of the constrained encoder into a corresponding modulated signal, which is then stored on the storage medium. The constrained encoded bits before and after the precoder are referred to as a non-return-to-zero-inverse (NRZI) sequence, and a non-return-to-zero (NRZ) sequence, respectively. Most of the prior art schemes design codes in the NRZI format [15], [16], [11], [17].

In this paper, a novel code design technique is proposed that overcomes all the drawbacks of the prior art schemes. The designed constrained PC codes can detect any type of dominant error events or error event combinations in optical recording systems, without introducing the parity-bit related error propagation. The rates of the designed codes are only a few tenths of a percent below the capacity. Two approaches are proposed to design constrained PC codes either in NRZI format or in NRZ format. Designing the codes in NRZ format is found to be more preferable for the PC code and post-processing based detection approach. In addition, although this paper focuses on designing codes for optical recording, the proposed code design technique is general, and can encompass other recording channels, such as the magnetic recording channels. This technique can also be generalized to combine constrained codes with other types of ECCs (*e.g.* the RS codes). However, it is out of the scope of this paper.

This paper is organized as follows. Section II presents the general principle of the new code design approach. Detailed methods for designing codes in NRZI and NRZ format are presented in Section III and Section IV, respectively. In Section V, examples of several newly designed efficient codes are illustrated. Their performances are presented in Section VI. The paper is concluded in Section VII.

II. General Principle of the New Code Design Approach

The general principle of the new code design is as follows. A segment of user data, which is typically the binary output of a RS-ECC encoder, is partitioned into several data words. All the data words except the last one are encoded by any suitable finite state constrained encoder, such as that proposed in [7]. The resulting codewords are referred to as "normal constrained (NC) codewords". The last data word is encoded by a parity-related constrained encoder, and the resulting codeword is referred to as the "parity-related constrained (PRC) codeword". In particular, the PRC encoder maps the last data word into a specific codeword chosen from a candidate codeword set, so that a certain PC constraint is realized over the combined codeword, which is a concatenation of the sequence of NC codewords and the PRC codeword. This PC constraint corresponds to a predetermined generator matrix, which can be defined to detect any type of error events in the system. The corresponding details can be found in [13], [14]. For ease in imposing the modulation constraints, the generator matrix needs to be designed to generate a systematic PC code. This code design principle is based on the following proposition.

Proposition 1: Consider the encoder for a $[l, n]$ systematic linear binary PC code C, which transforms an n-bit information word into an l-bit codeword, with $p = l - n$ being the number of parity bits. Let $\mathbf{u_1}$ and $\mathbf{u_2}$, respectively, denote row vectors with n_1 bits and $n_2 = n - n_1$ bits consisting of a sequence of NC codewords and a PRC codeword, with $0 < n_1 < n$. If the parity bits of $[\mathbf{u_1} \mid \underbrace{0, \cdots, 0}_{n_2}]$ and $[\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{u_2}]$ are equal, then the combined constrained codeword $[\mathbf{u_1} \mid \mathbf{u_2}]$, with p bits of zeros appended, generates a codeword of C.

Proof: Let $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$ be a generator matrix that describes the encoder of C, where $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathbf{P}$ is a $n \times p$ matrix. The parity bits of $[\mathbf{u_1} \mid \underbrace{0, \cdots, 0}_{n_2}]$ and $[\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{u_2}]$ are computed as

$$\mathbf{p_1} = [\mathbf{u_1} \mid \underbrace{0, \cdots, 0}_{n_2}]\mathbf{P} \quad \text{and} \quad \mathbf{p_2} = [\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{u_2}]\mathbf{P}.$$

If $\mathbf{p_1} = \mathbf{p_2}$, we get

$$[\mathbf{u_1} \mid \mathbf{u_2}]\,\mathbf{P} = [\underbrace{0, \cdots, 0}_{p}]. \tag{2}$$

Thus, $[\mathbf{u_1} \mid \mathbf{u_2} \mid \underbrace{0, \cdots, 0}_{p}]$ is a codeword of C. □

The structure of our constrained PC code, thus, includes two component codes: the NC code and the PRC code. Both codes serve as information words of the PC code C. The NC codewords are first constructed and connected. The parity bits of the sequence of NC codewords (with n_2 trailing zeros appended) are then computed. After that, a specific PRC codeword, which produces the same parity bits when n_1

leading zeros are appended, is selected from a candidate codeword set and concatenated directly with the NC codewords, thus forming the combined constrained PC codeword. The combined codeword is transmitted over the channel without appending its parity bits $[\ \underbrace{0, \cdots, 0}_{p}\ ]$, since the latter is fixed and known by the receiver. At the detector output, by checking the parity bits reconstructed from the received constrained PC codeword according to (2), which are equal to the syndrome of the received codeword (with p bits of zeros appended), we can detect errors in the received codeword that are within the error detection capability of the corresponding PC code C. Note that, in principle, we could always choose $\mathbf{p_1}+\mathbf{p_2}=\mathbf{a}$, where $\mathbf{a}$ is an arbitrary p bits row vector, and generate a codeword of C in terms of $[\mathbf{u_1} \mid \mathbf{u_2} \mid \mathbf{a}]$. At the detector output, errors can be detected by checking the syndrome of the received codeword with the parity bits $\mathbf{a}$ appended. Choosing $\mathbf{a}=\mathbf{0}$ makes the syndrome equal to the parity bits reconstructed from the received codeword, and simplifies decoding.

Both the NC code and the PRC code are finite-state constrained codes designed based on the same FSM. This enables the two component codes to be connected in any order without violating the modulation constraints, and also facilitates simpler hardware implementation of the encoder/decoder. In principle, any efficient FSM can be used in conjunction with the proposed code design approach. Here, we choose to use the FSM proposed in [7], since capacity approaching codes can thereby be obtained. Furthermore, since the PRC code is also protected by PCs, error propagation due to the PRC code is avoided. In addition, by applying the Guided Scrambling (GS) scheme [2], [7] to the NC code, whose codewords occupy the major portion of the constrained PC code, as shown in [7], satisfactory dc-free performance can be achieved.

The rate of our constrained PC code is given by

$$\begin{aligned} R &= \frac{m}{n} = \frac{n_1 R_1 + n_2 R_2}{n} \\ &= R_1 - \frac{n_2}{n}(R_1 - R_2), \end{aligned} \tag{3}$$

where m and n are the lengths of the segment of user data and the combined constrained PC codeword, respectively, and R_1 and R_2 are the rates of the NC code and the PRC code, respectively. The choice of n depends on the specific recording system and is a compromise between the code rate loss due to PC and the error correction capability of the post-processor [12]. The optimum codeword length has been found to be around 100 channel bits per parity bit, for $d=1$ coded optical recording channels. The corresponding details are given in Section VI.

III. Code Design in NRZI Format

A. Encoder Description

In this section, the codes are designed in NRZI format. Fig. 1 is a block schematic for encoding a constrained PC code in NRZI format. As illustrated, an m-bit segment of user data is partitioned into $K+1$ data words. The K leading data words are individually encoded into the first component codewords by the NC encoder[1]. Therefore, we have $K=\frac{n_1 R_1}{m_{NC}}$, with m_{NC} being the length of each input data word of the NC encoder. The $(K+1)^{\text{th}}$ data word is encoded into the second component codeword by the PRC encoder. During encoding, the next-state information (obtained from a code table) is passed from each codeword to the next. It indicates the next state from which to select a codeword for encoding the next data word. The encoder also includes a PC unit, which calculates the parity bits of the sequence of the leading K NC codewords (appended with n_2 trailing bits of zeros). The parity bits are then passed to the PRC encoder, and used to guide the encoding of the $(K+1)^{\text{th}}$ data word into the PRC codeword. Concatenating the NC codewords and the PRC codeword together, results in the combined constrained PC codeword in NRZI format. The NRZI codewords are then converted into NRZ format by a precoder, which is not shown in the figure, before they are transmitted over the channel.

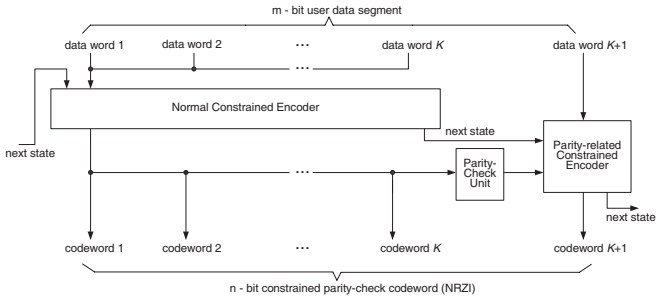

Fig. 1. Block schematic for encoding a constrained PC code in NRZI format.

B. Design of the Component Codes

To design the NC code, we use the FSM proposed in [7], since rates of the resulting constrained codes are very close to the capacity. To achieve high encoding efficiency, one data word is mapped into one codeword only in each state of the FSM.

To design the PRC code, we propose a novel approach to design sets of codewords with distinct parity bits, based on the same FSM of the NC code. These parity bits correspond to a predetermined generator matrix. We first propose a set of criteria that guides the design of the PRC code. With a given PC constraint, these criteria indicate how to choose the number of encoder states and assign valid codewords to these states to maximize the code rate.

To design a PRC code with m_2 user data bits and p parity bits, the number of codewords leaving a state set should be at least 2^{m_2+p} times the number of states within the state set. Based on the FSM of $d=1$ codes proposed in [7], we can obtain the following criteria:

$$r|X_{00}| + r_1|X_{01}| \geq r_1 2^{m_2+p}, \tag{4}$$

$$r\left(|X_{00}| + |X_{10}|\right) + \ r_1\left(|X_{01}| + |X_{11}|\right) \geq r 2^{m_2+p}, \tag{5}$$

where X_{ab} denotes the set of codewords starting with a 'a' and ending with a 'b', and $|X_{ab}|$ denotes the size of X_{ab}, where a, $b \in \{0,1\}$. The encoder has r states, which are divided into

[1]Guided scrambling needs to be applied to the NC codewords to impose the dc-free constraint. This is not shown in the figure for the sake of simplicity.

two state subsets of a first and second type. The encoder has r_1 states of the first type and $r_2 = r - r_1$ states of the second type. All codewords in states of the first type must start with a '0', while codewords in states of the second type start with either a '0' or a '1'.

Furthermore, for each set of codewords with the same parity bits, the number of codewords leaving a state set should be at least 2^{m_2} times the number of states within the state set. The criteria that guide the design of each set of codewords are therefore given by

$$r|\tilde{X}_{00}| + r_1|\tilde{X}_{01}| \geq r_1 2^{m_2}, \tag{6}$$

$$r\left(|\tilde{X}_{00}| + |\tilde{X}_{10}|\right) + r_1\left(|\tilde{X}_{01}| + |\tilde{X}_{11}|\right) \geq r2^{m_2}, \tag{7}$$

where $\tilde{X}_{ab}$ denotes the set of codewords with the same parity bits that start with a 'a' and end with a 'b'. In each set of the codewords with the same parity bits, each codeword has an assigned next state. A codeword that ends with '0' (*i.e.* codewords in $\tilde{X}_{00}$ and $\tilde{X}_{10}$) can be assigned up to r different next states in both the first and second state sets, and therefore can be used to map to r different user data words. A codeword that ends with '1' (*i.e.* codewords in $\tilde{X}_{01}$ and $\tilde{X}_{11}$) can only be assigned up to r_1 next states in the first state set, and therefore can be used to map to r_1 different user data words. The particular mapping of the codeword to the data word is a matter of design choice, and is not critical to the operation of the system. However, to ensure unique decodability, the sets of codewords that belong to a given state must be disjoint.

Similar criteria for designing PRC codes with $d = 2$ constraint can be derived. They are given by

$$r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| \geq r_1 2^{m_2+p}, \tag{8}$$

$$\begin{aligned} r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| \\ + r|X_{0100}| + (r_1 + r_2)|X_{0110}| + r_1|X_{0101}| \\ \geq (r_1 + r_2)2^{m_2+p}, \end{aligned} \tag{9}$$

$$\begin{aligned} r|X_{0000}| + (r_1 + r_2)|X_{0010}| + r_1|X_{0001}| \\ + r|X_{0100}| + (r_1 + r_2)|X_{0110}| + r_1|X_{0101}| \\ + r|X_{1000}| + (r_1 + r_2)|X_{1010}| + r_1|X_{1001}| \\ \geq r2^{m_2+p}, \end{aligned} \tag{10}$$

where X_{abcd} denotes the set of codewords that start with 'ab' and end with 'cd', where a, b, c, $d \in \{0, 1\}$. For $d = 2$ codes, the encoder has r states, which are further classified into three sets of states. The first set has r_1 states and it includes codewords that start with '00'. The second set has r_2 states and it includes codewords that start with either '01' or '00'. The third set has $r_3 = r - r_1 - r_2$ states and it includes codewords that start with '10', '01' or '00'.

The criteria that guide the design of each set of codewords that has the same parity bits are expressed as

$$r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \geq r_1 2^{m_2}, \tag{11}$$

$$\begin{aligned} r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \\ + r|\tilde{X}_{0100}| + (r_1 + r_2)|\tilde{X}_{0110}| + r_1|\tilde{X}_{0101}| \\ \geq (r_1 + r_2)2^{m_2}, \end{aligned} \tag{12}$$

$$\begin{aligned} r|\tilde{X}_{0000}| + (r_1 + r_2)|\tilde{X}_{0010}| + r_1|\tilde{X}_{0001}| \\ + r|\tilde{X}_{0100}| + (r_1 + r_2)|\tilde{X}_{0110}| + r_1|\tilde{X}_{0101}| \\ + r|\tilde{X}_{1000}| + (r_1 + r_2)|\tilde{X}_{1010}| + r_1|\tilde{X}_{1001}| \\ \geq r2^{m_2}, \end{aligned} \tag{13}$$

where $\tilde{X}_{abcd}$ denotes the set of codewords with the same parity bits. For each set of codewords with the same parity bits, a codeword that ends with '00' (*i.e.* codewords in $\tilde{X}_{0000}$, $\tilde{X}_{1000}$ and $\tilde{X}_{0100}$) can be assigned up to r different following states, and therefore can be used to map to r different user data words. A codeword that ends with '10' (*i.e.* codewords in $\tilde{X}_{0010}$, $\tilde{X}_{1010}$ and $\tilde{X}_{0110}$) can only be assigned up to r_1 following states in the first state set and r_2 states in the second state set, and therefore can be used to map to $r_1 + r_2$ different user data words. A codeword that ends with '01' (*i.e.* codewords in $\tilde{X}_{0001}$, $\tilde{X}_{1001}$ and $\tilde{X}_{0101}$) can be only assigned up to r_1 different following states in the first state set, and therefore can be used to map to r_1 different user data words. In addition, different states cannot contain the same codeword.

Note that the above inequalities are equivalent to the *approximate eigenvector equation* [1], and they are necessary conditions for code construction. Following these criteria, and by using either computer search or analytical approaches proposed in [18], we can determine the optimum number of encoder states to maximize the rate of the PRC code. The corresponding code rate is given by

$$R_2 = m_2/n_2. \tag{14}$$

The main steps for the design of the PRC code are as follows.

(1) For a PRC code with m_2 user data bits and p parity bits, use the criteria described above (*i.e.* (4) to (7) for $d = 1$ codes, and (8) to (13) for $d = 2$ codes) to determine the codeword length n_2 and the optimum number of encoder states. Note that at this step, the maximum runlength constraint k is temporarily relaxed (*e.g.* larger than $k = 7$ for $d = 1$ codes, and larger than $k = 10$ for $d = 2$ codes).

(2) Enumerate all the valid d constrained codewords of length n_2. Based on the given generator matrix, compute the parity bits of each codeword (with n_1 leading zeros appended) and distribute them into a group of codeword sets. A total of 2^p codeword sets are obtained.

(3) For each set of codewords with the same parity bits, allocate the codewords to various encoder states by following the FSM of the NC code [7]. This results in a set of 2^p sub-tables.

In each of the sub-tables, the principles for distributing the codewords to the encoder states are as follows. For $d = 1$ codes, the encoder states include two types of state subsets. The codewords in states of the first type must start with a '0', while codewords in states of the second type start with either a '0' or a '1'. In the sub-tables, every codeword has an assigned

next-state, which specifies the state from which to select the codeword for encoding the next data word. A codeword that ends with a '0' can be assigned to any of the encoder states, while a codeword that ends with a '1' can only be assigned to the states of the first type. This prohibits that a codeword ending with '1' entering states of the second type. Similarly, for $d = 2$ codes, the encoder states are divided into three sets of states. The first set includes codewords that start with '00', the second set includes codewords that start with either '01' or '00', and the third set includes codewords that start with '10', '01' or '00'. For each set of codewords with the same parity bits, a codeword that ends with '00' can be directed to any of the encoder states. A codeword that ends with '10' can only be directed to states in the first and second state sets. A codeword that ends with '01' can be directed to states in the first state set only. Furthermore, for both $d = 1$ and $d = 2$ codes, to ensure unique decodability, different states cannot have codewords in common.

(4) Concatenate the 2^p sub-tables, and form a code table for encoding/decoding the PRC code. Compared with the code table of the NC code, the PRC code table is enlarged by a factor of 2^p. In each state of the FSM, there is a set of 2^p codewords potentially mapped to one user data word. During encoding, as illustrated in Fig. 1, the parity bits associated with the sequence of NC codewords are first calculated. The PRC codeword having the same parity bits is selected from the codeword set and assigned to the user data word.

(5) Tighten the k constraint of the designed PRC code by optimizing the code table obtained from Step (4) by deleting codewords that start or end with long runs of '0's, or by increasing the number of states of the FSM.

Example 1: An example might be helpful to understand the whole process to design the PRC code. Assume the design of a PRC component code with $m_2 = 4$, for a $d = 1$ constrained single-bit even PC code (*i.e.* $p = 1$). The NC component code is assumed to be the rate 9/13 (1,18) code with 5 encoder states (*i.e.* $r = 5$, $r_1 = 3$, $r_2 = 2$) as proposed in [7]. First, by using the criteria of (4) to (7), we determine a minimum codeword length of $n_2 = 8$ for the PRC code. Next, generate all the valid $d = 1$ codewords of length $n_2 = 8$ and allocate them to various encoder states according to the principles described above. An example of the code table for the resulting rate $4/8$ PRC code is illustrated by Table I. As can be seen, the code table includes two sub-tables, which contain codewords with even and odd parity, respectively. In each sub-table, the first column shows the input data word. The second to the sixth columns show the codewords mapped to the data word and its associated next-state, for encoder states 1 to 5, respectively. Note that in the sub-tables, each codeword can be mapped to multiple data words, with the corresponding next-states being different. Note also that different encoder states do not have the same codeword.

Furthermore, in each of the encoder states, there is a set of two codewords mapped to one data word. This ensures that during encoding, a suitable PRC codeword from the codeword set can always be chosen, to be concatenated with the sequence of NC codewords and to realize an even PC constraint over the combined codeword. Such a fact can be seen from the following example. Let us assume that in Fig. 1, the PRC encoder's current input data word (*i.e.* the $(K+1)^{\text{th}}$ data word) is '1111'. Based on Table I, all possible cases that might arise during encoding are listed in Table II. Therefore, the even PC constraint can be achieved on the combined codeword, and the NC codewords and the PRC codeword can be connected without violating the $d = 1$ constraint.

TABLE I
CODE TABLE OF A RATE 4/8 PRC CODE DESIGNED IN NRZI FORMAT.

Parity even										
Data word	State 1		State 2		State 3		State 4		State 5	
	Codeword	State	Codeword	State	Codeword	State	Codeword	State	Codeword	State
0000	00001010	1	00100010	1	01000010	1	10000010	1	10010000	1
0001	00001010	2	00100010	2	01000010	2	10000010	2	10010000	2
0010	00001010	3	00100010	3	01000010	3	10000010	3	10010000	3
0011	00001010	4	00100010	4	01000010	4	10000010	4	10010000	4
0100	00001010	5	00100010	5	01000010	5	10000010	5	10010000	5
0101	00010010	1	00100100	1	01000100	1	10000100	1	10101010	1
0110	00010010	2	00100100	2	01000100	2	10000100	2	10101010	2
0111	00010010	3	00100100	3	01000100	3	10000100	3	10101010	3
1000	00010010	4	00100100	4	01000100	4	10000100	4	10101010	4
1001	00010010	5	00100100	5	01000100	5	10000100	5	10101010	5
1010	00010100	1	00101000	1	01001000	1	10001000	1	10100101	1
1011	00010100	2	00101000	2	01001000	2	10001000	2	10100101	2
1100	00010100	3	00101000	3	01001000	3	10001000	3	10100101	3
1101	00010100	4	00101000	4	01001000	4	10001000	4	10101001	1
1110	00010100	5	00101000	5	01001000	5	10001000	5	10101001	2
1111	00100001	1	01000001	2	01010101	3	10010101	4	10101001	5
Parity odd										
Data word	State 1		State 2		State 3		State 4		State 5	
	Codeword	State	Codeword	State	Codeword	State	Codeword	State	Codeword	State
0000	00001000	1	01001010	1	01001001	1	10001010	1	10100010	1
0001	00001000	2	01001010	2	01001001	2	10001010	2	10100010	2
0010	00001000	3	01001010	3	01001001	3	10001010	3	10100010	3
0011	00001000	4	01001010	4	01000101	1	10001010	4	10100010	4
0100	00001000	5	01001010	5	01000101	2	10001010	5	10100010	5
0101	00010000	1	01010010	1	01000101	3	10010010	1	10100100	1
0110	00010000	2	01010010	2	00101001	1	10010010	2	10100100	2
0111	00010000	3	01010010	3	00101001	2	10010010	3	10100100	3
1000	00010000	4	01010010	4	00101001	3	10010010	4	10100100	4
1001	00010000	5	01010010	5	01010010	1	10010010	5	10100100	5
1010	00101010	1	01010100	1	01010010	2	10010100	1	10101000	1
1011	00101010	2	01010100	2	01010010	3	10010100	2	10101000	2
1100	00101010	3	01010100	3	01010010	4	10010100	3	10101000	3
1101	00101010	4	01010100	4	01010010	5	10010100	4	10101000	4
1110	00101010	5	01010100	5	00100000	1	10010100	5	10101000	5
1111	00010101	1	00100101	3	00100000	2	10000101	4	10100001	5

TABLE II
EXAMPLE OF THE ENCODING PROCESS OF A RATE 4/8 PRC CODE DESIGNED IN NRZI FORMAT.

Parity bit of the K NC codewords	Next-state associated with the K^{th} NC codeword	PRC codeword	Next-state associated with the PRC codeword
0	1	00100001	1
	2	01000001	2
	3	01010101	3
	4	10010101	4
	5	10101001	5
1	1	00010101	1
	2	00100101	3
	3	00100000	2
	4	10000101	4
	5	10100001	5

C. Decoder Description

Based on the same FSM, the operation of the PRC decoder is generally the same as that of the NC decoder [7], but with the code tables being different. Both decoders are sliding-block decoders with a look ahead of one codeword. However,

unlike the NC decoders which are based on only one code table, the PRC decoder is based on two code tables. One is the NC code table, which is used to determine the state of the next codeword, while the other is the PRC code table, which is used to decode the current codeword using the obtained state information of the next codeword.

Example 2: We use the rate 4/8 code of Example 1, to illustrate the decoding process of the PRC code. Assume a received PRC codeword is '00001010'. According to Table I, the codeword of '00001010' is assigned to the data words '0000', '0001', '0010','0011', and '0100', together with the next-states 1 to 5, respectively. Therefore, we have to look at the next received codeword (*i.e.* the first NC codeword of the next constrained PC code) to obtain the encoder state that the PRC codeword is directed to. Assume the next NC codeword is '0001010001010', and it is found to belong to State 2, according to the code table of the NC code [7]. This means that the next-state associated with the PRC codeword '00001010' is State 2. Therefore, it is determined that the PRC codeword represents the data word '0001'. In the same manner, other NC and PRC codewords can be decoded sequentially.

IV. Code Design in NRZ Format

In this section, we present an approach to design constrained PC codes in NRZ format. For PC codes and post-processing based detection approaches, it is preferable to encode the data in NRZ format due to the following reasons. In the NRZI case, error detection and post-processing have to be done at the output of the 'NRZ to NRZI inverse precoder'. The process of inverse precoding will cause error propagation and thus increase the length of error events. For example, a single bit error in NRZ format will be converted into a transition shift error of 2 bits in NRZI format. As a result, the number of parity bits required for detecting errors may increase. Furthermore, carrying out post-processing at the detector output is more straightforward than doing it at the inverse precoder output.

The conventional approach for detection and correction of errors in NRZ format is to use a concatenation of a modulation encoder with a precoder, followed by a PC encoder [8], [9], [10]. However, this approach will considerably weaken the modulation constraint of the encoded channel data stream. In [17], Cideciyan *et al.* proposed the cascade of a modulation encoder with a PC encoder followed by a precoder. In this approach, before precoding, the user data is first encoded into a constrained PC code in NRZI format, which can detect and correct NRZ errors. This is done by translating the PC matrix at the output of the precoder into that at the input of the precoder, under the condition that the PC code at the output of the precoder must contain the all-one codeword.

We now present a new approach to design the constrained PC code in NRZ format, without PC matrix transformation and without the specific requirement on the PC code. In our approach, the code table of the NC code remains the same as that of the NC code in NRZI format. However, the code table for the PRC code is designed in a different way. The details are as follows.

(1) For a PRC code with m_2 user data bits and p parity bits, determine the codeword length n_2 and the optimum number of

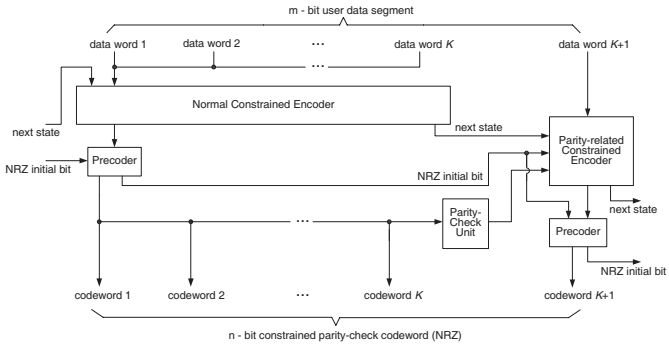

Fig. 2. Block schematic for encoding a constrained PC code in NRZ format.

encoder states. The criteria that guide the design are similar to those in the NRZI case. The only difference is that the parity bits of each codeword are computed in the NRZ format, rather than in the NRZI format, based on an assumed initial NRZ bit. To do this, '0' and '1' are used to denote NRZ bits '-1' and '$+1$', respectively.

(2) Enumerate all the valid d constrained codewords of length n_2 in NRZI format. Compute the parity bits of the codewords in NRZ format with an assumed initial NRZ bit.

(3) Distribute each set of NRZI codewords with the same NRZ parity bits obtained from Step (2) into different encoder states, and form a set of 2^p sub-tables. The principles for distributing the codewords in each sub-table are the same with those in the NRZI case.

(4) Concatenate the 2^p sub-tables together to form the code table for encoding/decoding of the PRC code in NRZ format. For the two different initial NRZ bits (*i.e.* '$+1$' and '-1'), we use the same code table to simplify encoding/decoding. However, the order of codeword sets with the same parity bits may need to be adjusted according to the initial NRZ bit.

To do encoding, as shown in Fig. 2, the NC codewords are first constructed and connected as in the NRZI case. The resulting codewords are then converted into NRZ format by a precoder, and the associated parity bits are computed. Based on these parity bits as well as the last bit of the NRZ sequence, the PRC codeword in NRZI format that has the same NRZ parity bits is selected from the codeword set. The PRC codeword needs to be converted into NRZ format before concatenating with the NRZ format NC codewords. During decoding, the detected NRZ data sequence is first converted into NRZI format through an inverse precoder, and the resulting NRZI sequence is then decoded based on the code tables of the NC code and the PRC code, along the same lines as those described in Section III-C.

Example 3: As an example, we show by Table III the code table of a rate 4/8 PRC code that is designed in NRZ format. Similar to Example 1, the code serves as the PRC component code of a single-bit event PC code with $d = 1$ constraint. The NC component code is also the same with that in Example 1. Similar to Table I, Table III also includes two sub-tables, which contain sets of codewords in NRZI format. However, the parity bit of each codeword is computed in the NRZ format, instead of in the NRZI format as in Table I. It can be verified that with an assumed intial NRZ bit of '-1', all the codewords in the first sub-table have an even parity,

TABLE III
CODE TABLE OF A RATE 4/8 PRC CODE DESIGNED IN NRZ FORMAT.

Parity even for initial NRZ bit '-1' or '+1'										
Data word	State 1		State 2		State 3		State 4		State 5	
	Codeword	State	Codeword	State	Codeword	State	Codeword	State	Codeword	State
0000	00001000	1	00100010	1	01000100	1	10000010	1	10010100	1
0001	00001000	2	00100010	2	01000100	2	10000010	2	10010100	2
0010	00001000	3	00100010	3	01000100	3	10000010	3	10010100	3
0011	00001000	4	00100010	4	01000100	4	10000010	4	10010100	4
0100	00001000	5	00100010	5	01000100	5	10000010	5	10010100	5
0101	00001010	1	00101000	1	01010000	1	10001000	1	10100010	1
0110	00001010	2	00101000	2	01010000	2	10001000	2	10100010	2
0111	00001010	3	00101000	3	01010000	3	10001000	3	10100010	3
1000	00001010	4	00101000	4	01010000	4	10001000	4	10100010	4
1001	00001010	5	00101000	5	01010000	5	10001000	5	10100010	5
1010	00010100	1	00101010	1	01010010	1	10001010	1	10101000	1
1011	00010100	2	00101010	2	01010010	2	10001010	2	10101000	2
1100	00010100	3	00101010	3	01010010	3	10001010	3	10101000	3
1101	00010100	4	00101010	4	01010010	4	10001010	4	10101000	4
1110	00010100	5	00101010	5	01010010	5	10001010	5	10101000	5
1111	00010001	1	00100101	2	01010101	3	10000101	4	10100101	5
Parity odd for initial NRZ bit '-1' or '+1'										
Data word	State 1		State 2		State 3		State 4		State 5	
	Codeword	State	Codeword	State	Codeword	State	Codeword	State	Codeword	State
0000	00010010	1	01000010	1	01010100	1	10000100	1	10100100	1
0001	00010010	2	01000010	2	01010100	2	10000100	2	10100100	2
0010	00010010	3	01000010	3	01010100	3	10000100	3	10100100	3
0011	00010010	4	01000010	4	01010100	4	10000100	4	10100100	4
0100	00010010	5	01000010	5	01010100	5	10000100	5	10100100	5
0101	00010000	1	01001000	1	00100001	1	10010000	1	10010101	1
0110	00010000	2	01001000	2	00100001	2	10010000	2	10010101	2
0111	00010000	3	01001000	3	00100001	3	10010000	3	10010101	3
1000	00010000	4	01001000	4	00001001	1	10010000	4	10100001	1
1001	00010000	5	01001000	5	00001001	2	10010000	5	10100001	2
1010	00100100	1	01001010	1	00001001	3	10010010	1	10100001	3
1011	00100100	2	01001010	2	01000101	1	10010010	2	10101001	1
1100	00100100	3	01001010	3	01000101	2	10010010	3	10101001	2
1101	00100100	4	01001010	4	01000101	3	10010010	4	10101001	3
1110	00100100	5	01001010	5	01010001	1	10010010	5	10000001	1
1111	00010101	1	00101001	3	01010001	2	10001001	4	10000001	5

while all the codewords in the second sub-table have an odd parity. It can be further verified that with an intial NRZ bit of '+1', the code table remains the same[2]. Therefore, based on the NRZ parity bits of the sequence of NC codewords and the last NRZ bit of the NRZ sequence, a suitable PRC codeword in the NRZI format with the same NRZ parity bit can always be selected from the code table. By converting the PRC codeword into NRZ format and concatenating it with the sequence of NRZ format NC codewords, an even PC constraint can be realized over the combined codeword in NRZ format. Finally, we remark that the rate 4/8 PRC codes described in Examples 1 to 3 are only for illustration purpose. Several more efficient newly designed codes are shown in the next section.

V. EXAMPLES OF NEWLY DESIGNED EFFICIENT CODES

In this section, we present several efficient constrained PC codes, designed in NRZ format, using the above code design method.

First of all, a new (1,18) constrained single-bit even PC code is designed. The rate 9/13 (1,18) code with 5 states (*i.e.* $r = 5$, $r_1 = 3$, $r_2 = 2$) FSM proposed in [7] is used as the NC code, since its rate is 3.85% higher than that of the rate 2/3 $d = 1$ codes used in BD and HD-DVD systems. A new rate 12/19 (1,18) code with 5 states is designed as the PRC code, which requires only 1 channel bit per parity bit with respect to the rate 2/3 $d = 1$ codes.

[2]Note that for other PRC codes designed in the NRZ format, the order of the sub-tables may differ with different initial NRZ bits, depending on the PC constraint and the codeword lengths of both the PRC code and the combined constrained PC code.

TABLE IV
DISTRIBUTION OF CODEWORDS IN THE VARIOUS ENCODER STATES FOR A RATE 12/19 (1,18) PRC CODE.

(i) Parity even for initial NRZ bit '-1' (Parity odd for '+1')

	$\widetilde{X}_{00}$	$\widetilde{X}_{01}$	$\widetilde{X}_{10}$	$\widetilde{X}_{11}$
Size	2135	1275	1275	805
State 1	605	358	0	0
State 2	599	369	0	0
State 3	603	362	0	0
State 4	0	0	589	384
State 5	0	0	600	368

(ii) Parity odd for initial NRZ bit '-1' (Parity even for '+1')

	$\widetilde{X}_{00}$	$\widetilde{X}_{01}$	$\widetilde{X}_{10}$	$\widetilde{X}_{11}$
Size	2046	1309	1309	792
State 1	589	384	0	0
State 2	599	368	0	0
State 3	594	378	0	0
State 4	0	0	605	357
State 5	0	0	598	370

Table IV shows the distribution of codewords in $r = 5$ encoder states, for the rate 12/19 PRC code. Through enumeration, we find that among the total 10946 valid $d = 1$ codewords of length 19, there are 5490 codewords having even parity with an assumed initial NRZ bit of '-1' (or odd parity with an initial NRZ bit of '+1'). Among these codewords, we further find $|\tilde{X}_{00}| = 2135$, $|\tilde{X}_{01}| = |\tilde{X}_{10}| = 1275$ and $|\tilde{X}_{11}| = 805$. We also find that there are 5456 codewords having odd parity with an assumed initial NRZ bit of '-1' (or even parity with an initial NRZ bit of '+1'), among which we have $|\tilde{X}_{00}| = 2046$, $|\tilde{X}_{01}| = |\tilde{X}_{10}| = 1309$ and $|\tilde{X}_{11}| = 792$. Each sub-table in Table IV illustrates the distribution of codewords with the same parity bit among the $r = 5$ states.

We take Table IV (i) as an example, which contains all the codewords having even parity with an assumed initial NRZ bit of '-1'. Observe that the set $\tilde{X}_{00}$ has 605 codewords allocated in State 1, 599 codewords in State 2, and 603 codewords in State 3. The total number of assigned codewords is $605 + 599 + 603 = 1807$, which is smaller than the set size 2135. Similarly, for each of the other codeword set, the total number of assigned codewords is smaller than the size of the set. On the other hand, in each state, the codewords are distributed according to the restrictions that a codeword ending with a '0' can be assigned to up to $r = 5$ different user data words, while a codeword that ends with a '1' can only be assigned to up to $r_1 = 3$ different user data words. Therefore, for State 1, the total number of assigned codewords is $605 \times 5 + 358 \times 3 = 4099$, which is sufficient to map $2^{12} = 4096$ user data words. Similarly, it can be verified that from any of the $r = 5$ encoder states, there are at least 4096 codewords that can be assigned to the user data words. This means that 12-bit user data words can be encoded. In the same manner, codewords having odd parity with an assumed initial NRZ bit of '-1' are distributed as shown in Table IV (ii), which also shows that 12-bit user data words can be supported. Hence, following Table IV, a rate 12/19 (1,18) PRC code can be constructed. We remark that the distribution of codewords given above may not be unique.

As a second example, using the same rate 9/13 code as the NC code, we design new constrained 2-bit and 4-bit PC codes.

TABLE V
SUMMARY OF NEWLY DESIGNED EFFICIENT CONSTRAINED PC CODES.

Code type	Generator polynomial	(d,k)	Parity overhead (channel bits / parity bit)	R	$\eta = R/C_{pc}$
9/13 code (NC) & 12/19 code (PRC)	$g(x) = 1 + x$	(1,18)	1	66/97 = 0.6804	0.9945
9/13 code (NC) & 9/16 code (PRC)	$g(x) = 1 + x + x^2$	(1,18)	1.5	135/198 = 0.6818	0.9965
9/13 code (NC) & 9/19 code (PRC)	$g(x) = 1 + x + x^4$	(1,18)	1. 5	279/409 = 0.6822	0.9970
6/11 code (NC) & 10/20 code (PRC)	$g(x) = 1 + x$	(2,15)	1.25	52/97 = 0.5361	0.9900
6/11 code (NC) & 5/17 code (PRC)	$g(x) = 1 + x + x^4$	(2,15)	1.75	215/402 = 0.5348	0.9877

They are defined by generator polynomials $g(x) = 1+x+x^2$ and $g(x) = 1+x+x^4$, respectively. The corresponding PRC codes are a rate 9/16 (1,18) code and a rate 9/19 (1,18) code, respectively. With respect to the rate 2/3 $d = 1$ codes, these PRC codes achieve 1.5 channel bits per parity bit.

As a third example, we consider $d = 2$ codes. With the rate 6/11 (2,15), 9-state (*i.e.* $r = 9$, $r_1 = 4$, $r_2 = 2$, $r_3 = 3$) code proposed in [7] as the NC code, whose rate is 2.27% higher than that of the rate 8/15 (2,10) EFM-like codes [4], [2] used in DVD systems, we have designed a new constrained single-bit even PC code and a new constrained 4-bit PC code defined by $g(x) = 1+x+x^4$. The PRC codes are a 9-state rate 10/20 (2,15) code and a 9-state rate 8/22 (2,15) code, respectively. With respect to the EFM-like codes, these PRC codes achieve 1.25 and 1.75 channel bits per parity bit, respectively.

The above examples of newly designed efficient codes are summarized in Table V. The codeword length, n, is chosen such that the number of channel bits per parity bit is around 100 (see Section VI for details). As can be seen, the new codes achieve minimum parity overhead, and the efficiency of most of the new codes is only a few tenths of a percent below capacity[3].

It should be noted that for the above new codes, the sizes of input symbols of all the component codes are not 8 bits. As a result, error propagation due to the mismatch of symbol sizes between the constrained code and the conventional byte-oriented RS-ECC [5] may arise. However, this error propagation can be avoided by using the 'modified concatenation' scheme [19]. Alternatively, a non-byte-oriented RS-ECC can be used to eliminate this error propagation. For example, when the rate 9/13 code without parity, the rate 135/198 2-bit PC code and the rate 279/409 4-bit PC code in Table V are used in conjunction with a 9-bit/symbol RS-ECC, error propagation is avoided since the size of the input symbols of all the codes (or component codes) is 9 bits. Finally, we remark here that it is possible to impose stricter k constraint and the repeated minimum transition runlength (RMTR) constraint [5], [6] on the designed codes, by increasing the number of states of the FSM, and/or by applying the GS scheme.

[3]Following (1), for constrained PC codes with the $(1,\infty)$ constraint, the capacity is $C_{pc(1,\infty)} = R_{(1,\infty)} - 1/100 = 0.6842$, with 100 channel bits per parity bit. Similarly, the capacity of constrained PC codes with the $(2,\infty)$ constraint is $C_{pc(2,\infty)} = R_{(2,\infty)} - 1/100 = 0.5415$.

VI. PERFORMANCE EVALUATION

In this section, the performance of the newly designed efficient constrained PC codes with $d = 1$ constraint is evaluated using BD systems. In particular, the performance of various constrained codes is compared using the symbol error rate (SER) at the output of the constrained decoder as the performance criterion. In the simulations, we assume a RS-ECC with 9 bits/symbol, since in such cases there is no error propagation due to the mismatch of symbol sizes between ECC and the constrained codes whose input symbol size is 9 bits (*i.e.* the rate 9/13 code, the rate 135/198 2-bit PC code and the rate 279/409 4-bit PC code).

In our study, it is assumed that the optical read-out is linear and a generalized Braat-Hopkins model [20] is used to describe the channel. The Fourier transform of the channel symbol response is given by

$$H(\Omega) = \begin{cases} \frac{2RT_u \sin(\pi\Omega)}{\pi^2\Omega}\left[\arccos(|\frac{\Omega}{R\Omega_u}|) - |\frac{\Omega}{R\Omega_u}|\sqrt{1-(\frac{\Omega}{R\Omega_u})^2}\right] & \text{for } |\frac{\Omega}{R\Omega_u}| < 1, \\ 0 & \text{for } |\frac{\Omega}{R\Omega_u}| \geq 1, \end{cases} \tag{15}$$

where Ω is the frequency normalized by the channel bit rate, and R is the rate of the $d = 1$ constrained PC code. The quantity $\Omega_u = f_c T_u$, which is the optical cut-off frequency f_c normalized by user bit rate $1/T_u$, is a measure of the recording density. For an optical recording system using a laser diode with wavelength λ and a lens with numerical aperture NA, the normalized cut-off frequency is given by $\Omega_u = \frac{2\mathrm{NA}}{\lambda} L_u$, where L_u is the spatial length of one user bit. For the BD systems using the rate 2/3 17PP [5], with λ=405 nm, NA=0.85 and L_u=112.5 nm, we get $\Omega_u \approx 0.5$. In this paper, cut-off frequencies $\Omega_u = 0.5$ and $\Omega_u = 0.375$ are considered. These choices represent recording systems with nominal density and high density, respectively, according to current standards [5]. The variance σ^2 of additive white Gaussian channel noise, is determined by the user signal-to-noise-ratio (SNR) defined as $\mathrm{SNR_u(dB)} = 10\log_{10}\left(\frac{\sum h_{ku}^2}{\sigma_u^2}\right)$ and $\sigma^2 = \frac{1}{R}\sigma_u^2$, with σ_u^2 being the noise power in the user bandwidth, and h_{ku} is the channel symbol response for $R = 1$ [20]. When studying the performance over different user densities, the reference signal power in the user SNR needs to be independent of density. For this, h_{ku} is evaluated for a particular user density, *e.g.* $\Omega_u = 0.33$, which is independent of the densities at which the channel and receiver are tested. The above definitions of SNR and channel response help to fairly reflect the impact of code rate in the system performance evaluation.

In this study, a Viterbi detector that is matched to a 7-tap optimized partial response (PR) target is used as the detector [12]. The dominant error events at the Viterbi detector output turn out to be $\pm\{2,0,-2\}$, $\pm\{2\}$, $\pm\{2,0,-2,0,2\}$, $\pm\{2,0,-2,0,2,0,-2\}$, and $\pm\{2,0,0,-2\}$. At the output of the Viterbi detector, a matched-filtering type post-processor is used, which can correct both single and double error events that occur within each detected codeword [21]. The post-processor is essentially a soft-decision decoder of the PC code, and it is widely accepted in practice due to its simplicity [10], [12]. In particular, the post-processor is designed to correct a specific number of the dominant error events at the output

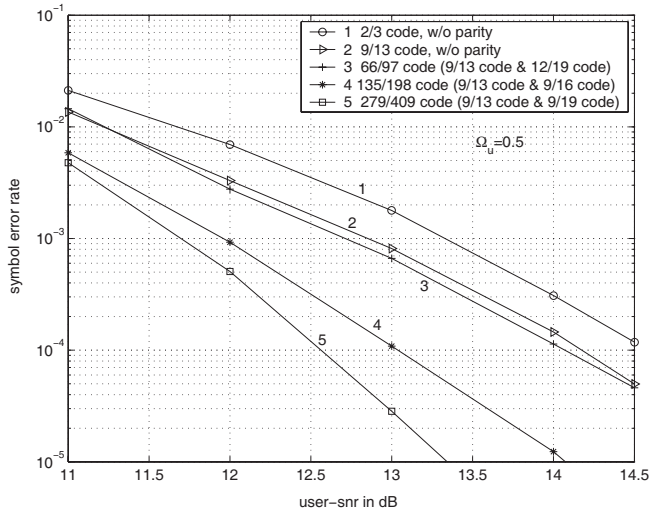

Fig. 3. SER performance of the BD system with various codes, $\Omega_u = 0.5$.

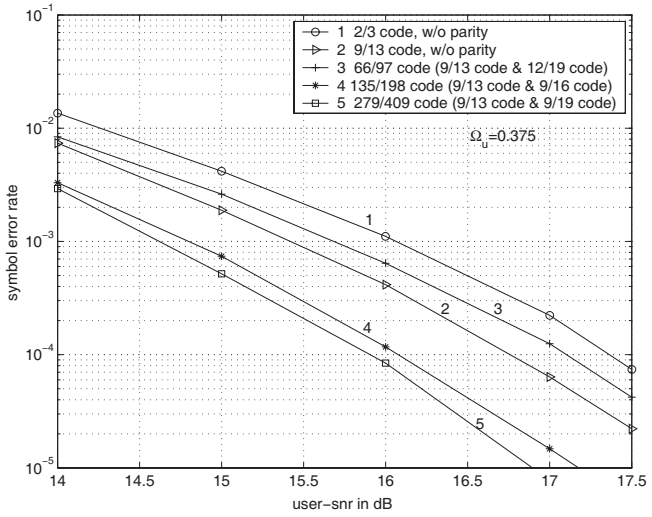

Fig. 4. SER performance of the BD system with various codes, $\Omega_u = 0.375$.

of the channel detector, by exploiting the syndromes of the received constrained PC codewords, and by computing the Euclidean distance of the candidate error events [10], [12]. We remark that the code design technique proposed above is general, and it is suitable to other types of decoders of the PC code as well.

The suitability and efficiency of different PC codes for given channel and detector are determined by the error event distribution at the detector output. The new single-bit even PC code can only detect error events with odd number of errors, and therefore cannot detect the error events $\pm\{2, 0, -2\}$, $\pm\{2, 0, -2, 0, 2, 0, -2\}$ and $\pm\{2, 0, 0, -2\}$. The new 2-bit PC codes, defined by the generator polynomial $g(x) = 1 + x + x^2$, can detect most of the dominant error events, except $\pm\{2, 0, -2, 0, 2\}$ and $\pm\{2, 0, 0, -2\}$. The new 4-bit PC codes, with $g(x) = 1+x+x^4$, however, can detect all the dominant error events. For real-life channels, the dominant error events may differ from those illustrated above. However, following the code design method described in the above sections, we can easily define a generator matrix that detects all the required error events [14], and design the constrained PC code accordingly.

For a given PC code, the choice of its codeword length (n) is a trade-off between the code rate loss and error correction power. In our study, for each code, SERs are compared with different codeword lengths and SNRs. Simulation results show that the minimum SER is obtained with around 100 channel bits per parity bit, over a wide range of SNRs. Therefore, we choose to use a codeword length of around 100 bits per parity bit, for the designed constrained PC codes.

Figs. 3 and 4 illustrate the SER performance of the system with the rate $2/3$ code, rate $9/13$ code, and the new constrained PC codes, at nominal density and high density, respectively. The data storage systems typically require an error rate of 10^{-12} or less after ECC. For BD, an ECC failure rate of 10^{-16} corresponds to a SER of around 4×10^{-3} [5], [22]. Therefore, the performance of various codes is compared at SER $= 10^{-4}$, to keep an additional margin for the allowable SER of various codes and SNRs. From Fig. 3, we observe that compared to the performance of the system with the rate $2/3$ code and without parity (Curve 1), the rate 9/13 code without parity (Curve 2) gives a gain of 0.4 dB at SER $= 10^{-4}$, due to its higher code rate. Compared to the rate 9/13 code without parity, the new single-bit PC code (Curve 3) gives no significant gain, since it cannot detect the error event $\pm\{2, 0, -2\}$. The new 2-bit PC code (Curve 4), however, achieves a gain of 1.2 dB over the rate 9/13 code, since it can detect the error event $\pm\{2, 0, -2\}$. Using the new 4-bit PC code (Curve 5), around 0.5 dB gain is obtained over the 2-bit PC code. The reason is that it can detect all the dominant error events. Overall, the new constrained 4-bit PC code gains 2 dB over the system with the rate $2/3$ code and without parity. At $\Omega_u = 0.375$, observe from Fig. 4 that compared with the results at nominal density, the performance gains of PC codes are more modest. According to Fig. 4, the new 4-bit PC code achieves an overall performance gain of 1.5 dB at high density. This is due to the reason that at high density there are many non-dominant error events, which are long events with small probabilities. To detect these types of error events, more parity bits are needed. They are also difficult to correct, since mis-correction of these long events will cause many more errors. Using appropriate coding techniques (*e.g.* RMTR codes) may eliminate the underlying data patterns that support these events and improve the performance of PC codes and post-processing. This is beyond the scope of the paper.

VII. Conclusions

In this paper, a general and systematic code design technique has been proposed for constructing capacity-approaching constrained PC codes, which can detect any type of dominant error events or error event combinations in optical recording systems. The PC constraint corresponds to linear systematic binary PC codes. The modulation constraint can be any practical d constraint (*i.e.* $d = 1$ and $d = 2$). Furthermore, error propagation due to parity bits is avoided, since errors are corrected equally well over the entire constrained PC codeword. Approaches have been proposed to design the code in NRZI format and NRZ format. Designing the codes in

NRZ format is found to be preferable. Using the proposed method, various new codes for different optical recording systems can be designed. Application of this technique to other recording systems is straightforward. Examples of several newly designed efficient codes have been illustrated, and their SER performances have been evaluated with the BD systems. Simulation results show that the new $d = 1$ constrained 4-bit PC code can detect all the dominant error events. Compared to the rate 2/3 code without parity, it achieves a performance gain of 2 dB at nominal density, and 1.5 dB at high density, at SER $= 10^{-4}$.

ACKNOWLEDGMENT

Authors would like to thank Prof. Jan W. M. Bergmans and Dr. George Mathew for their valuable insights and help in preparation of this paper.

REFERENCES

[1] B. H. Marcus, P. H. Siegel, and J. K. Wolf, "Finite-state modulation codes for data storage," *IEEE Trans. Commun.*, vol. 10, no. 1, pp. 5–37, Jan. 1992.

[2] K. A. S. Immink, *Codes for Mass Data Storage Systems.* Shannon Foundation Publishers, 1999, chaps. 4, 5, and 13.

[3] K. A. S. Immink and H. Ogawa, "Method for encoding binary data," US patent 4,501,000, Feb. 1985.

[4] K. A. S. Immink, "EFMPlus: the coding format of the multimedia compact disc," *IEEE Trans. Consumer Electron.*, vol. 41, no. 3, pp. 491–497, Aug. 1995.

[5] T. Narahara, S. Kobayashi, Y. Shimpuku, G. van den Enden, J. Kahlman, M. van Dijk, and R. van Woudenberg, "Optical disc system for digital video recording," *Jpn. J. Appl. Phys.*, pt. 1, vol. 39, no. 2B, pp. 912–919, 2000.

[6] K. Kayanuma, C. Nota, and T. Iwanaga, "Eight to twelve modulation code for high density optical disk," in *Proc. Tech. Dig. Intl. Symp. Optical Memory* (ISOM), Nara, Japan, Nov. 2003, pp. 160–161.

[7] K. A. S. Immink, J. Y. Kim, S. W. Suh, and S. K. Ahn, "Efficient DC-free RLL codes for optical recording," *IEEE Trans. Commun.*, vol. 51, pp. 326–331, Mar. 2003.

[8] 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.

[9] W. Feng, A. Vityaev, G. Burd, and N. Nazari, "On the performance of parity-check codes in magnetic recording systems," in *Proc. IEEE Intl. Conf. Global Telecommun.* (GLOBECOM), San Francisco, USA, Nov. 2000, pp. 1877–1881.

[10] 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.

[11] W. M. J. Coene, H. Pozidis, and J. W. M. Bergmans, "Run-length limited parity-check coding for transition-shift errors in optical recording," in *Proc. IEEE Intl. Conf. Global Telecommun.* (GLOBECOM), San Antonio, USA, Nov. 2001, pp. 2982–2986.

[12] K. Cai, V. Y. Krachkovsky and J. W. M. Bergmans, "Performance bounds for parity coded optical recording channels with d=1 constraint", in *Proc. IEEE Intl. Conf. Commun.* (ICC), Alaska, USA, May 2003, pp. 2914–2918.

[13] S. Lin and D. J. Costello, *Error Control Coding Fundamentals and Applications.* Prentice-Hall Inc., 1983, Chap. 4.

[14] A. Patapoutian, B. Shen, and P. A. McEwen, "Event error control codes and their applications," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2595–2603, Sept. 2001.

[15] P. Perry, M. C. Lin, and Z. Zhang, "Runlength-limited codes for single error-detection and single error-correction with mixed type errors," *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1588–1592, July 1998.

[16] S. Gopalaswamy and J. W. M. Bergmans, "Modified target and concatenated coding for constrained magnetic recording channels," in *Proc. IEEE Intl. Conf. Commun.* (ICC), New Orleans, USA, Jun. 2000, pp. 89–93.

[17] R. D. Cideciyan and E. Eleftheriou, "Codes satisfying maximum transition run and parity-check constraints," in *Proc. IEEE Intl. Conf. Commun.* (ICC), Paris, France, June 2004, pp. 635–639.

[18] K. Cai and K. A. S. Immink, "On the number of encoder states a type of RLL codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 7, pp. 3313–3319, July 2006.

[19] W. G. Bliss, "Circuitry for performing error correction calculations on baseband encoded data to eliminate error propagation," *IBM. Tech. Discl. Bul.*, vol. 23, pp. 4633–4634, 1981.

[20] K. Cai, G. Mathew, J. W. M. Bergmans, and Z. Qin, "A generalized description of Braat-Hopkins model for optical recording channels," in *Proc. IEEE Intl. Conf. Consumer Electronics* (ICCE), Los Angeles, USA, June 2003, pp. 324–325.

[21] K. Cai, K. A. S. Immink, J. W. M. Bergmans, and L. P. Shi, "Constrained parity-check code and post-processor for advanced blue laser disc systems," *Jpn. J. Appl. Phys.*, vol. 45, no. 2B, pp. 1071–1078, Feb. 2006.

[22] A. Padiy, B. Yin, C. Verschuren, R. Vlutters, T. Jansen, and J. Lee, "Signal processing for 35GB on a single-layer Blu-ray disc," in *Proc. Tech. Dig. Int. Symp. Optical Data Storage* (ODS), Monterey, USA, Apr. 2004, pp. 34–36.

Kui Cai received B.E. degree in information and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 1992, M.Eng degree in electrical engineering from National University of Singapore, Singapore, in 2000, and joint Ph.D. degree in electrical engineering from Technical University of Eindhoven, The Netherlands, and National University of Singapore, Singapore, in 2007. Since 1999, she has been with Data Storage Institute, Singapore. Her research interests include coding theory, communication theory, and signal processing.

Kees A. Schouhamer Immink received Masters and Ph.D. degrees from the Eindhoven University of Technology in 1975 and 1984, respectively. Since 1998, he is founder and president of Turing Machines Inc.. He is an adjunct professor at the Institute for Experimental Mathematics, Essen-Duisburg University, Germany, and affiliated with the Data Storage Institute and National University of Singapore as a Distinguished Visiting Professor.

He has made many contributions to digital audio, video, and data recording devices. He designed the channel coding techniques of virtually all consumer-type digital audio and video recording products such as CD, DCC, DVD, Blu-Ray, etc. His research resulted in four books, more than 100 journal articles, and around 1000 international patents. He was named a Knight by Beatrix, Queen of the Netherlands, and he received an 'Emmy' award from the National Academy of Television Arts and Sciences (NATAS), the IEEE Edison Medal, AES Gold and Silver Medals, and the SMPTE Progress Medal. He was named a fellow of the IEEE, AES, SMPTE, IEE, member of Royal Netherlands Academy of Sciences, and foreign associate of the National Academy of Engineering, and was inducted into the Consumer Electronics Hall of Fame. In 1998, the IEEE Information Theory Society has recognized his pioneering and influential achievements to Information Theory by awarding him with the Golden Jubilee Award for Innovation. He served the profession as President of the Audio Engineering Society in 2003-2004.