The Art of Combining Distance-Enhancing Constrained Codes with Parity-Check Codes for Data Storage Channels

Kui Cai[1], Kees A. Schouhamer Immink[2], Yuan Xing Lee[3], and Zhiliang Qin[1]
[1]Data Storage Institute, Singapore, 117608.
[2]Turing Machines Inc., The Netherlands.
[3]LSI Corporation, USA.
Tel: (65)68748666, Fax: (65)67772053
Email: Cai_Kui@dsi.a-star.edu.sg, immink@turing-machines.com

Abstract

A general and systematic code design methodology is proposed to efficiently combine constrained codes with PC codes for data storage channels. The proposed constrained PC code includes two component codes: the normal constrained (NC) code and the parity-related constrained (PRC) code. The NC code can be any distance-enhancing constrained code, such as the maximum transition run (MTR) code or repeated minimum transition runlength (RMTR) code. The PRC code can be any linear binary PC code. The constrained PC codes can be designed either in non-return-to-zero-inverse (NRZI) format or non-return-to-zero (NRZ) format. The rates of the designed codes are only a few tenths of a percent below the theoretical maximum. The proposed code design method enables soft information to be available to the PC decoder and facilitates soft decoding of PC codes. Furthermore, since errors are corrected equally well over the entire constrained PC codeword, error propagation due to parity bits is avoided. Efficient finite-state encoding methods are proposed to design capacity-approaching constrained codes and constrained PC codes with RMTR or MTR constraint. The generality and efficiency of the proposed code design methodology are shown by various code design examples for both magnetic and optical recording channels.

Index Terms

Distance-enhancing constrained codes, error correction codes (ECCs), parity-check codes, finite-state encoding method, post-processor, soft decoding.

I. INTRODUCTION

There are two types of channel codes in data storage channels, namely, constrained codes [1], [2] and error correction codes (ECCs) [3]. The constrained codes, also known as modulation codes, are used to match the data to the recording channel characteristics, to improve the detector performance and to help in the operation of control loops (*e.g.* timing/gain loops) at the receiver. Runlength limited (RLL) codes [2] are examples of widely used constrained codes, which are characterized by the minimum runlength constraint d and the maximum runlength constraint k. In recent years, a maximum transition run (MTR) constraint j [4] has been further introduced to magnetic recording channels. The MTR codes prohibit input data patterns that support some of the dominant error events (*i.e.* the most likely error events that can occur) at the output of the channel detector, and therefore increase the minimum distance between those that remain [5]. Therefore, they are known as distance-enhancing codes [6]. In optical recording systems, $d = 1$ constrained codes, examples of early distance-enhancing codes, are used for the latest blue laser disc systems [7], [8]. The $d = 1$ codes are MTR codes with a $j = 1$ constraint. Furthermore, there is a second distance-enhancing constraint t [7], [8], which stipulates in the $d = 1$ constrained channel bit stream the maximum number of consecutive minimum distance transitions, *i.e.* '1010', '1010 10', '10101010', '101010101' and so on patterns. The corresponding codes are referred to as repeated minimum transition runlength (RMTR) codes. The improvement of coding gain of the above distance-enhancing codes in data storage channels has been reported in [4], [6], [7], [8]. Moreover, it has also been found that the distance-enhancing codes can help to improve the error-rate floor performance of low-density parity-check (LDPC) code [9], through eliminating the long error events of the channel [10]. For given code constraint(s), the achievable maximum code rate was derived by Shannon and is called the Shannon Capacity [11]. The efficiency of a code, referred to as *coding efficiency*, is defined as the ratio of the code rate to the Shannon Capacity.

ECC codes, on the other hand, are used to detect and correct errors that may occur when data is transmitted over data storage channels by adding additional parity bits/symbols into the data. In data storage channels, the error control system typically consists of a Reed-Solomn (RS) code, which is capable of correcting combinations of random and burst errors. In recent years, the soft-decodable inner ECC codes have been further adopted, to make use of the soft information from the channel and effectively correct short random errors that are not covered by the outer RS-ECC. The low redundancy parity-check (PC) polynomial codes are one type of inner ECC codes which are widely used for data storage channels. The PC code can detect the specific dominant error events of the system using only a few parity bits. For error correction, the matched-filtering type post-processor that combines syndrome and soft-decision decoding [12], [13] is widely used due to its simplicity. Alternatively, the PC codes can also be decoded iteratively by using a soft-input soft-output (SISO) channel detector (*e.g.* the BCJR detector [14]) and the sum-product algorithm (SPA) decoder [15]. The LDPC codes are another type of inner ECC codes which have the potential to approach the Shannon-limit. Currently, LDPC codes are being explored to replace the outer RS-ECC for data storage channels. One major obstacle is the error floor phenomenon of LDPC codes [16], since the data storage systems require extremely low error rates.

Many attempts have been made to efficiently combine constrained codes with PC codes. For example, in the scheme described by Perry *et al.* [17], 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. Due to the insertion of additional stitching bits between the data and parity blocks, the coding efficiency of this scheme is limited.

Cideciyan *et al.* [18] designed a rate $96/104$ code which satisfies the $j = 3$ MTR constraint and a specific 2-bit PC constraint. In this scheme, between every block of six the rate $16/17$ MTR codes with $j = 3$ constraint[1], two parity bits are inserted. A rate $36/36$ parity-check-preserving block encoder is further used to replace the codewords that violate the $j = 3$ constraint and ensure the PC constraint is still satisfied over the combined codeword. Although this scheme achieves a high coding efficiency, it is not general enough to design codes that satisfy an arbitrary MTR constraint with any given PC constraint.

Vasic *et al.* [19] proposed a method to impose the k constraint in the LDPC codewords through deliberate bit-flippings with the expectation that the LDPC code is able to correct both the deliberate errors as well as channel errors that occur during data detection. However, for data storage channels with strong distance-enhancing constraints, such as the MTR or RMTR

[1]At the border of two codewords, the MTR constraint is relaxed to $j = 4$.

constraint, the number of flipped bits increases significantly compared with that in k-constrained channels. This will result in substantial performance losses and a high error floor of the LDPC codes.

Another efficient scheme proposed by Wijngaarden and Immink [20] to combine the k constrained codes with ECCs is based on the concept of “constrained codes with unconstrained positions”. In [20], various methods are proposed to construct high-rate k constrained codewords with unconstrained bit positions preserved for the ECC parity bits. Like the method proposed in [19], this scheme also works well only for systems with loose modulation constraints (*e.g.* the k constraint).

In this paper, we propose a general and systematic code design methodology, which can efficiently combine constrained codes with PC codes for data storage channels. The constrained codes can be any distance-enhancing constrained codes, such as the various MTR codes and RMTR codes. The PC codes can be any linear binary PC codes, which are defined to detect and correct any type of dominant error events and error event combinations in any data storage channels. The rates of the designed codes are only a few tenths of a percent below the capacity. The proposed method enables soft information to be available to the PC decoder and soft decoding of PC codes. Furthermore, since errors are corrected equally well over the entire constrained PC codeword, error propagation due to parity bits is avoided. Approaches have been proposed to impose the PC constraint into the channel bit streams, either before or after the precoder. In addition, although this paper focuses on combining distance-enhancing constrained codes with PC codes, the proposed code design technique is general, and can encompass any practical modulation constraint for data storage systems, such as the k constraint.

This paper is organized as follows. In Section II, we present in details the code construction methodology to efficiently combine distance-enhancing constrained codes with PC codes. To show the generality of the proposed methodology, code design examples for both magnetic and optical recording channels are further illustrated. In particular, in Section III, examples of several newly designed constrained PC codes for optical recording channels with $d = 1$ (*i.e.* $j = 1$) and RMTR constraints are illustrated. In Section IV, constrained PC codes satisfying various MTR constraints, such as the $j = 2$ and $j = 3$ constraints, are designed for magnetic recording channels. The paper is concluded in Section V.

II. Code Construction Methodology

A. Capacity of Constrained PC Codes

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, $\frac{1}{C_{NC}}$ channel bits are needed per parity bit, where C_{NC} is the capacity of the constrained code. Let there be p parity bits per codeword of length n. Then, the capacity of constrained PC codes is given by

$$C_{PC} = \frac{(n - \frac{p}{C_{NC}})C_{NC}}{n} = C_{NC} - \frac{p}{n}. \tag{1}$$

B. General Principle

The general principle of the new code design is as follows. A segment of user data is partitioned into several data words. All the data words except the last one are encoded by any efficient constrained encoder that can achieve a high coding efficiency. The resulting codewords are referred to as "normal constrained (NC) codewords". The last data word, however, is encoded by a specific 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(of n bits), which is a concatenation of the sequence of NC codewords (of n_1 bits) and the PRC codeword (of $n_2 = n - n_1$ bits). This PC constraint corresponds to a predetermined generator matrix (or generator polynomial) of a $[l, n]$ linear binary PC code [3], which is defined to detect any type of error events in the system. For ease in imposing the modulation constraints, the generator matrix needs to be designed to generate a systematic code.

To support the above described code design principle, we now show that, *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 $\mathbf{v_1}$ and $\mathbf{v_2}$, 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 $[\mathbf{v_1} \mid \underbrace{0, \cdots, 0}_{n_2}]$ and $[\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{v_2}]$ are equal, then the combined constrained codeword $[\mathbf{v_1} \mid \mathbf{v_2}]$, with p bits of zeros appended, generates a codeword of C.*

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 $[\mathbf{v_1} \mid \underbrace{0, \cdots, 0}_{n_2}]$ and $[\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{v_2}]$ are computed as

$$\mathbf{p_1} = [\mathbf{v_1} \mid \underbrace{0, \cdots, 0}_{n_2}]\mathbf{P} \quad \text{and} \quad \mathbf{p_2} = [\underbrace{0, \cdots, 0}_{n_1} \mid \mathbf{v_2}]\mathbf{P}.$$

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

$$[\mathbf{v_1} \mid \mathbf{v_2}]\,\mathbf{P} = [\underbrace{0, \cdots, 0}_{p}]. \tag{2}$$

Thus, $[\mathbf{v_1} \mid \mathbf{v_2} \mid \underbrace{0, \cdots, 0}_{p}]$ is a codeword of C.

Therefore, as shown in Fig. 1, the structure of our constrained PC code, includes two component codes: the NC code and the PRC code. Both codes serve as information words of the PC code C. During encoding, a m-bit segment of user data is partitioned into $I+1$ data words. The I leading data words are first encoded into NC codewords. 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. 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 $[\mathbf{v_1} \mid \mathbf{v_2} \mid \mathbf{a}]$. Choosing $\mathbf{a} = \mathbf{0}$ makes the syndrome equal to the parity bits reconstructed from the received codeword, and simplifies decoding.

Furthermore, the NC code and PRC code should be constructed based on the same code design method. This enables the two component codes to be connected in any order without the addition of addition stitching bits, and also facilitates simpler hardware implementation of the encoder/decoder. The details for designing the NC code and the PRC code are presented in the next section. The proposed scheme enables soft information to be available to the PC decoder, and therefore facilitates soft decoding of PC codes. In addition, since the PRC code is also protected by PCs, error propagation due to parity bits is avoided.

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.

C. *Design of the Component Codes*

In the write path of a data storage system, a precoder of the form $1/(1 \oplus D)$ 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 [17], [20]. In this section, we start with the design of the constrained PC codes in NRZI format.

1) Design of the NC Codes: In the proposed scheme, any efficient code design method can be used to design the NC code. Among various code design techniques, the finite-state encoding method [1] can achieve high coding efficiency, limited decoder error propagation, and satisfy strong modulation constraints, such as the MTR or RMTR constraint. Therefore, in this paper, we use this method to design the NC code. Furthermore, unlike the *state-splitting method* [21] starting with the *labeled graph*, we propose simple and efficient finite-state encoding methods, which directly specify the encoding/decoding principles for various distance-enhancing codes. The rates of the designed codes are only a few tenths of a percent below the capacity.

2) Design of the PRC Codes: To design the PRC code, we use the same code design method of the NC code. This enables the two component codes to be connected seamlessly without violating the modulation constraints. However, for the NC code, to achieve a high coding efficiency, there is only one codeword mapped to each user data word, in each of the encoder state. For the PRC code, on the other hand, there is a set of 2^p candidate codewords potentially mapped to each user data word, in each state of the encoder state, where p is the number of parity bits. Therefore, compared with the NC code with the same input user data word length, the codeword length of the PRC code needs to be longer to achieve the additional PC constraint, and hence its code rate is smaller. Therefore, to achieve a high coding efficiency, as shown in Fig. 1, each combined constrained PC code consists multiple NC codewords. The PRC codeword is only used as the last codeword in the combined codeword.

We propose the following criteria that guides the design of the PRC code.

- *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. 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 rate of the PRC code is then given by

$$R_p = m_2/n_2. \tag{4}$$

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 to determine the codeword length n_2 and a suitable number of encoder states.

(2) Enumerate all the valid constrained codewords of length n_2. Based on the given generator matrix, compute the parity bits of each codeword (appended with n_1 leading bits of zeros) 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 encoding method of the NC code. Thus results in a set of 2^p sub-tables.

(4) Concatenate the 2^p sub-tables together, and form a code table for encoding/decoding of 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 of the encoder state, there is a set of 2^p codewords potentially mapped to one user word.

Based on the same code design method, the operation of the PRC decoder is generally the same as that of the NC decoder, but with the code tables being different. For the finite-state constrained PC codes proposed in this paper, both the NC and PRC decoders are sliding-block decoders [1] with the least decoding window of length 2.

D. Code Design in NRZ Format

In the previous section, we present the code design method in NRZI format. In this section, we propose 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 reason that 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. As a result, the number of parity bits required for detecting errors may increase.

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 [12], [22]. However, this approach will considerably weaken the modulation constraint of the encoded channel data stream. In [18], 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. First, determine the codeword length n_2 and the suitable number of encoder states for the PRC code follow criteria 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. Secondly, enumerate all the valid 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. Thirdly, distribute each set of NRZI codewords with the same NRZ parity bits into different encoder states, and form a set of 2^p sub-tables. Fourthly, 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 parity bits of each codeword set may differ.

To do encoding, 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.

III. Examples of Code Design for Optical Recording Channels

In this section, we present several efficient distance-enhancing constrained PC codes designed for optical recording channels, using the above code design method. The codes are designed in both NRZI and NRZ formats.

A. $d = 1$ *(i.e.* $j = 1$*) Constrained PC Codes*

1) Finite-State Encoding Method to Design $d = 1$ *Constrained Code:* The $d = 1$ codes are MTR codes with a $j = 1$ constraint. The standard codes adopted by blue laser disc systems are rate 2/3 $d = 1$ codes [7], [8]. In [23], Immink *et al.* have proposed an efficient finite-state encoding method, using which even more efficient $d = 1$ codes have been designed. The code design method can be summarized as follows. A codeword is a binary string of length n that satisfies the $d = 1$ constraint. The set of codewords, X, is divided into four subsets X_{00}, X_{01}, X_{10} and X_{11}, which are characterized as follows. Codewords in X_{00} start and end with a '0', codewords in X_{01} start with a '0' and end with a '1', *etc.* The encoder has s states, which are divided into two state subsets of a first and second type. The encoder has s_1 states of the first type and $s_2 = s - s_1$ 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'. Codewords that end with a '0', *i.e.*, codewords in subsets X_{00} and X_{10}, may enter any of the s encoder states. Codewords that end with a '1' may enter the s_1 states of the first state set only. Furthermore, the sets of codewords that belong to a given state must be disjoint. During decoding, by observing both the current and the next codewords, the decoder can uniquely decide which information word was actually transmitted. Based on the above code design method, both the NC code and the PRC code can be designed for $d = 1$ constrained PC codes.

2) $d = 1$ Constrained PC Codes: 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.* $s = 5$, $s_1 = 3$, $s_2 = 2$) proposed in [23] is used as the NC code, since its rate is 3.85% higher than that of the standard rate 2/3 codes. A new rate 7/12 (1,18) code with 5 states is designed as the PRC code, which requires only 1.5 channel bit per parity bit with respect to the rate 2/3 codes.

We show the details of the code design in NRZ format. Table I shows the distribution of codewords in $s = 5$ encoder states, for the rate 7/12 PRC code. Through enumeration, we find that among the total 377 valid $d = 1$ codewords of length 12, there are 195 codewords having even parity no matter the initial NRZ bit is '-1' or '+1'. Among these codewords, we further find $|X_{00}| = 76$, $|X_{01}| = 43$, $|X_{10}| = 51$ and $|X_{11}| = 25$. We also find that there are 182 codewords having odd parity irrespective of the initial NRZ bit, among which we have $|X_{00}| = 68$, $|X_{01}| = 46$, $|X_{10}| = 38$, and $|X_{11}| = 30$. Each sub-table in Table I illustrates the distribution of codewords with the same parity bit among the $s = 5$ states.

We take Table I (i) as an example, which contains all the codewords having even parity. Observe that the set X_{00} has 20 codewords allocated in each of State 1, State 2, and State 3. The total number of assigned codewords is $20 \times 3 = 60$, which is smaller than the set size 76. 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 $s = 5$ different user data words, while a codeword that ends with a '1' can only be assigned to up to $s_1 = 3$ different user data words. Therefore, for State 1, the total number of assigned codewords is $20 \times 5 + 10 \times 3 = 130$, which is sufficient to map $2^7 = 128$ user data words. Similarly, it can be verified that from any of the $s = 5$ encoder states, there are at least 129 codewords that can be assigned to the user data words. This means that 7-bit user data words can be encoded. In the same manner, codewords having odd parity are distributed as shown in Table I (ii), which also shows that 7-bit user data words can be supported. Hence, following Table I, a rate 7/12 (1,18) PRC code can be constructed. We remark that the distribution of codewords given above may not be unique.

Concatenating the rate 9/13 NC codes with the rate 7/12 PRC code , we obtain a new rate 70/103 (1,18) constrained single-bit even PC code, whose coding efficiency is 99.33% compared to the capacity[2]. Note that for optical recording channels, the codeword length, n, is chosen such that the number of channel bits per parity bit is around 100. According to [13], [24], such choice of n achieves a compromise between code rate loss due to PC and error correction power of the post-processor.

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. 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. The rates of the combined 2-bit and 4-bit constrained PC codes are 135/198, and 279/409, respectively, with coding efficiencies of 99.65% and 99.70% respectively. The coding gain of $d = 1$ constrained PC codes is reported in [13], [24].

Note that for either the 2-bit or 4-bit PC code proposed above, the sizes of input symbols of both the NC and PRC component codes are the same. Therefore, if these codes are used in conjunction with an outer RS-ECC with the same symbol size, error propagation due to the mismatch of symbol sizes between the constrained code and RS-ECC is avoided. All the constrained PC codes proposed in the following sections are designed in a similar fashion.

B. RMTR Constrained Codes and RMTR Constrained PC Codes

1) A New Finite-State Encoding Method to Design RMTR Code with $d = 1$ and $t = 3$ constraints: The code design starts from computing the capacity of the RMTR codes. A finite-state transition diagram (FSTD) which provides a graphical representation of a sequence with $d = 1$ and $t = 1$ RMTR constraints is illustrated in Fig. 2. The labels on the arrows correspond to NRZI code bits, and any sequence that can be formed by following the arrows through the states and reading off the transition labels is a valid constrained sequence. An adjacent matrix for the FSTD is constructed in which the element in the i^{th} row and the j^{th} column represents the number of transitions from state i to state j. In Fig. 2, the adjacent matrix is given by $A_{(d=1,t=1)} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. The capacity of the RMTR code is then computed as [11] $C_{(d=1,t=1)} = \log_2 \xi_{\max}[A_{(d=1,t=1)}] = 0.5515$, where $\xi_{\max}[A_{(d=1,t=1)}]$ is the largest real eigenvalue

[2]For optical recording channels, following (1), the capacity of the constrained PC codes is $C_{PC} = C_{NC} - 1/100$, with 100 channel bits per parity bit.

of $A_{(d=1,t=1)}$. Following a similar procedure, the capacities of various RMTR codes can be computed. They are illustrated in Table II.

From Table II, we observe that $t = 3$ is the minimum achievable RMTR constraint for $d = 1$ codes whose code rates are comparable to that of the standard rate 2/3 codes. Imposing RMTR constraints stronger than $t = 3$ will introduce additional code rate loss. Therefore, in the following, we present the design a new rate 8/12 code with the $d = 1$ and $t = 3$ constraints.

We propose a finite state encoding method to design the new code. First, we define a codeword to be a binary string of length v that satisfies the $d = 1$ and $t = 3$ constraints. As shown in Fig. 3, the encoder has 5 states, which are characterized as follows.

- Codewords in State 1 state start with '000'.
- Codewords in State 2 start with '0100', '00100', or '000'.
- Codewords in State 3 start with '010100', '001010', or '0100'.
- Codewords in State 4 state start with '100'.
- Codewords in State 5 start with '10100', '101010', '010101', '010100', or '100'.

In addition, to ensure unique decodability, there is no common codeword between the above five encoder states.

To facilitate reuse of codewords, *i.e.* mapping the same codeword to more than one user word to achieve a high coding efficiency, each codeword may enter more than one encoder state. The rules for assigning various codewords to the encoder states are as follows.

- Codewords that end with '00' can be assigned to any of the five states.
- Codewords that end with '0010' or '001010' cannot be assigned to State 5.
- Codewords that end with '001' or '00101010' can be assigned to States 1 to 3.
- Codewords that end with '00101' can be assigned to States 1 and 2.
- Codewords that end with '0010101' can be assigned to State 1 only.

The above state-transition rules ensure that the $d = 1$ and $t = 3$ constraints are always satisfied during the concatenation of codewords from various encoder states.

During decoding, by observing both the current and the next codewords, the decoder can determine the user data word that was actually transmitted.

Following the above described finite-state encoding method, a new rate 8/12 RMTR code with 5 encoder states ($s = 5$, $s_1 = 3$, and $s_2 = 2$) is designed, which satisfies the $d = 1$, $t = 3$, and $k = 16$ constraints. The rate of the code is only 1.86% below the capacity. Compared with the standard rate 2/3 codes, it imposes the minimum achievable RMTR constraint on the channel bit stream with the least decoding window length, without introducing additional code rate loss.

2) RMTR Constrained PC Codes: Using the above rate 8/12 code as the NC code, various RMTR constrained PC codes that satisfy the $d = 1$ and $t = 3$ constraints can be designed. We exemplify our code design with a new constrained 2-bit PC code and a new constrained 4-bit PC codes. They are defined by generator polynomials $g(x) = 1+x+x^2$ and $g(x) = 1+x+x^4$, respectively. A rate 8/15 ($d = 1$, $t = 3$, and $k = 16$) code and a rate 8/18 code ($d = 1$, $t = 3$, and $k = 16$) are designed as the PRC codes for the 2-bit and 4-bit PC code, respectively. These PRC codes achieve 1.5 channel bits per parity bit with respect to the rate 2/3 codes. The rates of the combined constrained PC codes are 136/207, and 272/414, respectively. Both codes achieve a coding efficiency of only 1.84% below the capacity. The performance improvement achieved by these codes over the standard rate 2/3 code is shown in [25].

IV. Examples of Code Design for Magnetic Recording Channels

In this section, we show efficient MTR constrained PC codes designed for magnetic recording channels. We use the $j = 2$ and $j = 3$ MTR codes as examples, although the code design method can be generalized to other MTR codes as well.

A. $j = 2$ Constrained MTR Code and $j = 2$ Constrained PC Codes

1) A New Finite-State Encoding Method to Design $j = 2$ Constrained MTR Code: The technique of look-ahead coding in conjunction with violation detection and substitution is widely use for constructing MTR codes [5]. In this section, we propose an efficient finite-state encoder which transforms u-bit user data words into v-bit codewords that satisfy the $j = 2$ constraint. As shown in Fig. 4, the encoder has s states, which are further classified into three sets of states. The first state set has s_1 states and it includes codewords that start with ‘0’. The second state set has s_2 states and it includes codewords that start with either ‘0’ or ‘10’. The third state set has $s_3 = s - s_1 - s_2$ states and it includes codewords that start with ‘0’, ‘10’ or ‘110’. Codewords that end with a ‘0’ may be assigned to any of the s encoder states. Codewords that end with a ‘01’ may be assigned to the s_1 states of the first set, or the s_2 states of the second set. Codewords that end with a ‘011’ may be assigned to the s_1 states of the first state set only. The above state-transition rules prohibit that a codeword ending with ‘01’ entering states of the third type, or a codeword ending with '011' entering states of the second and third types. Due to the reuse of codewords in encoding, to ensure unique decodability, each set of codewords that belongs to a given state must be disjoint. This attribute implies that any codeword can be unambiguously related to the state from which it emerged. Therefore, the decoder is a sliding-block decoder with one codeword look-ahead.

With the above described finite-state encoding method, we obtain the following conditions:

$$s|X_{0\cdots0}| + (s_1+s_2)|X_{0\cdots01}| + s_1|X_{0\cdots011}| \geq s_1 2^u, \tag{5}$$

$$\begin{aligned} &s\left(|X_{0\cdots0}| + |X_{10\cdots0}|\right) + (s_1+s_2)(|X_{0\cdots01}| + |X_{10\cdots01}|) \\ &+s_1\left(|X_{0\cdots011}| + |X_{10\cdots011}|\right) \geq (s_1+s_2)2^u, \end{aligned} \tag{6}$$

$$\begin{aligned} &s\left(|X_{0\cdots0}| + |X_{10\cdots0}| + |X_{110\cdots0}|\right) + (s_1+s_2)(|X_{0\cdots01}| + |X_{10\cdots01}| + |X_{110\cdots01}|) \\ &+s_1\left(|X_{0\cdots011}| + |X_{10\cdots011}| + |X_{110\cdots011}|\right) \geq s2^u, \end{aligned} \tag{7}$$

where $X_{a\cdots b}$ denotes the set of codewords starting with a binary string of 'a' and ending with a binary string of 'b', and $|X_{a\cdots b}|$ denotes the cardinality of the codeword set $X_{a\cdots b}$. Note that the above inequalities are equivalent to the *approximate eigenvector equation* [], and they are necessary conditions for the code construction. By using computer search, we can determine suitable integers s, s_1, s_2 and s_3 which satisfy conditions of (5) $\sim$ (7) and maximize the code rate u/v, and design the code accordingly.

By using the above described code design method, a new rate 7/8 MTR code with 6 encoder states ($s = 6$, $s_1 = 3$, $s_2 = 2$, and $s_3 = 1$) is designed, which satisfies the $j = 2$, and $k = 12$ constraints. As the capacity of $j = 2$ MTR codes is 0.8791 [5], the efficiency of this code is 99.53%. Furthermore, it can be verified that six is the smallest number of encoder states possible for a rate 7/8 $j = 2$ MTR code [26].

2) $j = 2$ Constrained PC Codes: Various $j = 2$ MTR constrained PC codes can be designed, based on the above described finite-state encoding method. First, using the rate 7/8 code as the NC code, we design a new constrained 2-bit MTR constrained PC code, which is defined by the generator polynomial $g(x) = 1 + x + x^2$. A new rate 7/11 ($j = 2$, $k = 12$) code is designed as the PRC code, which achieves 1.5 channel bits per parity bit with respect to the rate 7/8 code. As a second example, with the same rate 7/8 code as the NC code, we design a new constrained

4-bit constrained PC code which corresponds to the generator polynomial $g(x) = 1 + x + x^4$. The corresponding PRC code is a rate 7/13 (($j = 2$, $k = 12$) code. With respect to the rate 7/8 code, it achieves 1.25 channel bits per parity bit. The rates of the combined 2-bit and 4-bit constrained PC codes are 119/139 and 238/277, respectively[3]. Therefore, The corresponding coding efficiencies are 99% and 99.35%, respectively.

B. $j = 3$ Constrained MTR Code and $j = 3$ Constrained PC Codes

1) A New Finite-State Encoding Method to Design $j = 3$ Constrained MTR Code: In a similar fashion as in the case with $j = 2$ MTR code, we now propose an efficient finite-state encoding method to design the $j = 3$ MTR codes. As shown in Fig. 5, the encoder has s states, which are divided into four state sets of a first, second, third, and fourth type. The encoder has s_1 states of the first type, s_2 states of the second type, s_3 states of the third type, and $s_4 = s - s_1 - s_2 - s_3$ states of the fourth type. All codewords in states of the first type must start with a '0'. Codewords in states of the second type start with either a '0' or a '10'. Codewords in states of the third type start with '0', '10', or '110', while codewords in states of the fourth type start with '0', '10', '110', or '1110'.

The state-transition rules are as follows. Codewords that end with a '0', may enter any of the s encoder states. Codewords that end with a '01' may not enter the s_4 states of the fourth state set. Codewords that end with a '011' may enter the s_1 states of the first state set, or the s_2 states of the second state set. Codewords that end with a '0111' may enter the s_1 states of the first state set only. Furthermore, the set of codewords that belongs to a given state must have no codewords in common.

Following the above described method, we obtain the following conditions:

$$s|X_{0\cdots0}| + (s_1 + s_2 + s_3)|X_{0\cdots01}| + (s_1 + s_2)|X_{0\cdots011}| + s_1|X_{0\cdots0111}| \geq s_1 2^u, \tag{8}$$

$$\begin{aligned} &s\left(|X_{0\cdots0}| + |X_{10\cdots0}|\right) + (s_1 + s_2 + s_3)(|X_{0\cdots01}| + |X_{10\cdots01}|) \\ &+(s_1 + s_2)(|X_{0\cdots011}| + |X_{10\cdots011}|) + s_1\left(|X_{0\cdots0111}| + |X_{10\cdots0111}|\right) \geq (s_1 + s_2)2^u, \end{aligned} \tag{9}$$

[3]For magnetic recording channels, the codeword length n of the constrained PC code is chosen such that the number of channel bits per parity bits is around 70 [12]. The capacity of the constrained PC codes is $C_{PC} = C_{NC} - 1/70$.

$$s\left(|X_{0\cdots0}|+|X_{10\cdots0}|+|X_{110\cdots0}|\right)+(s_1+s_2+s_3)(|X_{0\cdots01}|+|X_{10\cdots01}|+|X_{110\cdots01}|)$$
$$+(s_1+s_2)(|X_{0\cdots011}|+|X_{10\cdots011}|+|X_{110\cdots011}|)$$
$$+s_1\left(|X_{0\cdots0111}|+|X_{10\cdots0111}|+|X_{110\cdots0111}|\right)\geq(s_1+s_2+s_3)2^u, \tag{10}$$

$$s\left(|X_{0\cdots0}|+|X_{10\cdots0}|+|X_{110\cdots0}|+|X_{1110\cdots0}|\right)+(s_1+s_2+s_3)(|X_{0\cdots01}|+|X_{10\cdots01}|$$
$$+|X_{110\cdots01}|+|X_{1110\cdots01}|)+(s_1+s_2)(|X_{0\cdots011}|+|X_{10\cdots011}|+|X_{110\cdots011}|+|X_{1110\cdots011}|)$$
$$+s_1\left(|X_{0\cdots0111}|+|X_{10\cdots0111}|+|X_{110\cdots0111}|+|X_{1110\cdots0111}|\right)\geq s2^u, \tag{11}$$

Following an approach similar to case with the $j=2$ code, we can obtain suitable encoder states s, s_1, s_2, s_3, and s_4 for a rate u/v $j=3$ constrained codes.

By using the above described code design method, a new rate 17/18 MTR code with 8 encoder states ($s=8$, $s_1=4$, $s_2=2$, $s_3=1$, and $s_4=1$) is designed, which satisfies the $j=3$ and $k=13$ constraints. The code achieves an efficiency of 99.75%, with respect to the the capacity of $j=3$ codes of 0.9468 [5].

2) $j=3$ Constrained PC Codes: Using the rate 17/18 code as the NC code, various $j=3$ MTR constrained PC codes can be designed. A new 2-bit and 4-bit MTR constrained PC codes are two examples. 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 17/21 ($j=3$, $k=13$) code and a rate 17/23 (($j=3$, $k=13$) code, respectively. With respect to the rate 17/18 code, these PRC codes achieve 1 channel bits and 1.5 channel bits per parity bit, respectively. The rates of the combined 2-bit and 4-bit constrained PC codes are 136/147, and 255/275, respectively. The corresponding coding efficiencies are 99.21% and 99.44% respectively. In particular, the coding efficiency of the rate 136/147 2-bit constrained PC code is higher than that of the rate 96/104 code designed in [18].

V. Conclusions

In this paper, we have proposed a general and systematic code design methodology to efficiently combine constrained codes with PC codes for data storage channels. The modulation constraint can be any distance-enhancing constraint, such as the MTR or RMTR constraint. The PC constraint corresponds to any linear binary PC code. Approaches have been proposed to design the constrained PC codes either in NRZI or NRZ format. The rates of the designed codes are only a few tenths of a percent below the capacity. The proposed code design method enables soft information to be available to the PC decoder, and therefore facilitates soft decoding of PC codes. Furthermore, error propagation due to parity bits is avoided, since errors are corrected equally well over the entire constrained PC codeword.

A variety of code design examples have been provided for both magnetic and optical recording channels. The distance-enhancing constraints include the $d = 1$ (*i.e.* $j = 1$) constraint and RMTR constraint for optical recording channels, as well as $j = 2$ and $j = 3$ constraints for magnetic recording channels. In particular, efficient finite-state encoding methods have been proposed to design capacity-approaching constrained codes and constrained PC codes with various RMTR and MTR constraints. These examples demonstrate the generality and efficiency of the proposed code design methodology.

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, 13.

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

[4] J. Moon and B. Brickner, “Maximum transition run codes for data storage systems,” *IEEE Trans. Magn.*, vol. 32, pp. 3992-3994, Sep. 1996.

[5] R.D. Cideciyan, E. Eleftheriou, B.H. Marcus, and D. S. Modha, “Maximum transition run codes for generalized partial response channels,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 619-634, Apr. 2001.

[6] E. Soljianin and A.J. van Wijngaarden, “Application of distance enhancing codes,” *IEEE Trans. Magn.*, vol. 37, no. 2, pp. 762-767, Mar. 2001.

[7] 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.

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

[9] R.G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[10] T. Kanaoka and T. Morita, “Structured LDPC codes with reversed MTR/ECC for magentic recording channels,” *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2561-2563, Oct. 2006.

[11] C.E. Shannon,“A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, 1948.

[12] 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.

[13] 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.

[14] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, Mar. 1974.

[15] X-Y. Hu, E. Eleftherious, D-M. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Intl. Conf. Global Telecommun.* (GLOBECOM), San Antonio, USA, Nov. 2001, pp. 1036-1036E.

[16] T. Rchardson, "Error floors of LDPC codes," in *Proc. of the 41st Allerton Conf.* Oct. 2003.

[17] 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, Jul. 1998.

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

[19] B. Vasic and K. Pedagani, "Run-length-limited low-density parity-check codes based on deliberate error insertion," *IEEE Trans. Magn.*, vol. 40, no. 3, pp. 1738-1743, May 2004.

[20] A.J. van Wijngaarden and K.A.S. Immink, "Maximum runlength limited codes with error control capabilities," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 602-611, Apr. 2001.

[21] R.L. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes: An application of symbolic dynamics to information theory," *IEEE Trans. Inform. Theory*, vol. 29, pp. 5-22, Jan. 1983.

[22] 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.

[23] 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.

[24] K. Cai, K.A.S. Immink, J.W.M. Bergmans, and L.P. Shi, "Novel constrained parity-check code and post-processor for advanced blue laser disk," in *Tech. Dig. Int. Symp. Optical Media and Optcal Data Storage* (ISOM/ODS), Hawaii, USA, Jul. 2005, pp. TuB5.

[25] K. Cai, K.A.S. Immink, S. Zhang, Z. Qin, and X. Zou, "RMTR constrained parity-check codes for high-density blue laser disk systems ," in *Tech. Dig. Int. Symp. Optical Media and Optcal Data Storage* (ISOM/ODS), Hawaii, USA, Jul. 2008, pp. MoA3.

[26] B.H. Marcus and R.M. Roth, "Bounds on the number of states in encoder graphs for input-constrained channels," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 742-758, May 1991.

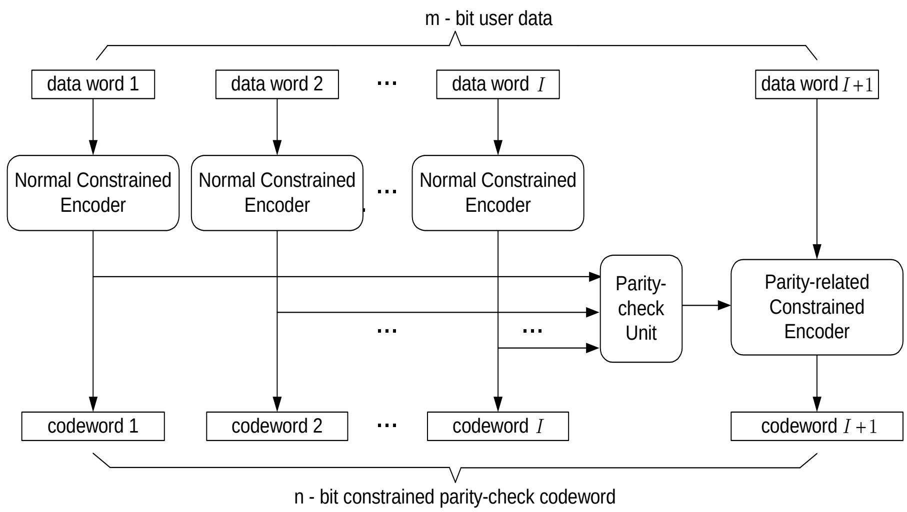

Fig. 1. Block diagram for encoding a constrained PC code.

TABLE I

DISTRIBUTION OF CODEWORDS IN THE VARIOUS ENCODER STATES FOR A RATE 7/12 (1,18) PRC CODE.

(i) Parity Even

	Size	State 1	State 2	State 3	State 4	State 5
X_{00}	76	20	20	20	0	0
X_{01}	43	10	10	10	0	0
X_{10}	51	0	0	0	20	20
X_{11}	25	0	0	0	10	10

(ii) Parity Odd

	Size	State 1	State 2	State 3	State 4	State 5
X_{00}	68	20	20	20	0	0
X_{01}	46	10	10	10	0	0
X_{10}	38	0	0	0	20	18
X_{11}	30	0	0	0	10	13

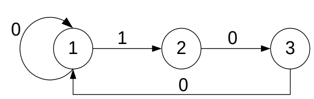

Fig. 2. FSTD for a $(d = 1,\ t = 1)$ sequence.

TABLE II

CAPACITY $C_{(d=1,t)}$ AS A FUNCTION OF t.

t	$C_{(d=1,\ t)}$
1	0.5515
2	0.6509
3	**0.6793**
4	0.6888
5	0.6922
6	0.6935
∞	0.6942

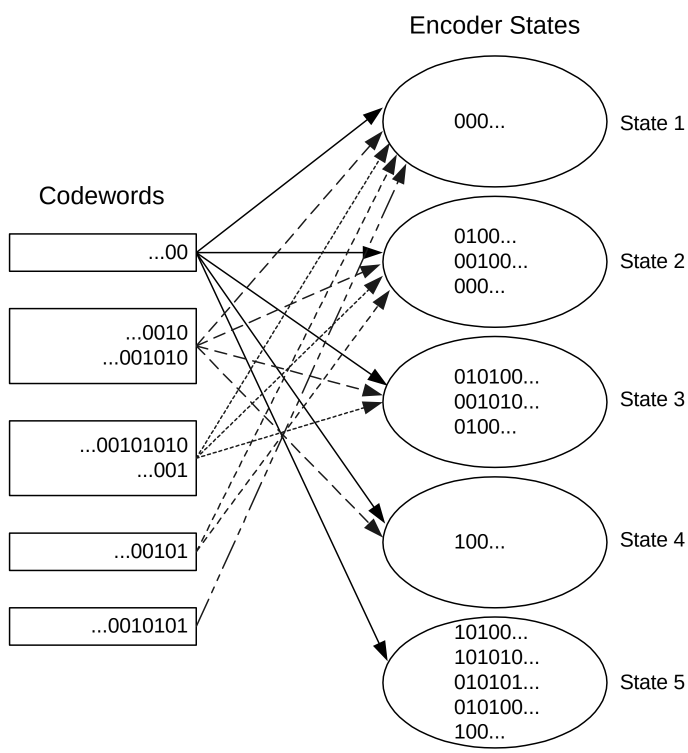

Fig. 3. Finite-state encoding method to design the rate 8/12 RMTR code with $d = 1$ and $t = 3$ constraint.

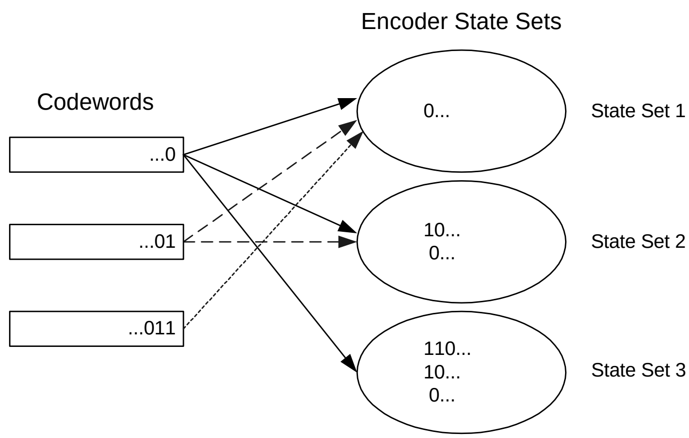

Fig. 4. Finite-state encoding method to design MTR codes with $j = 2$ constraint.

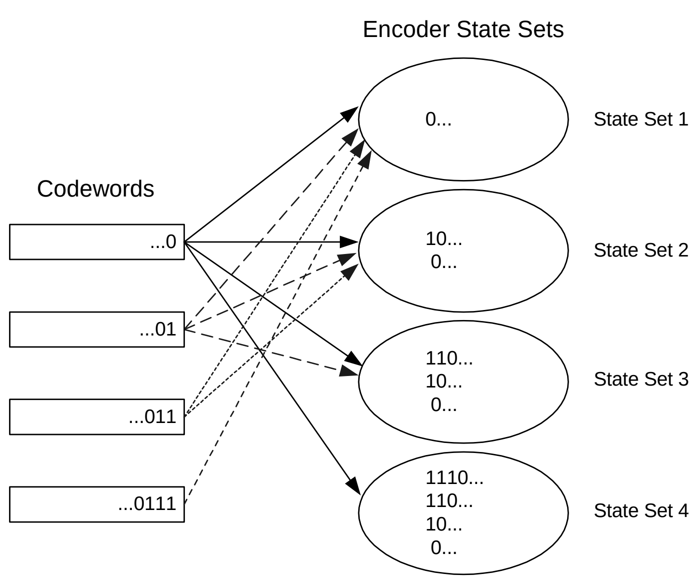

Fig. 5. Finite-state encoding method to design MTR codes with $j = 3$ constraint.