Design of Close-to-Capacity Constrained Codes for Multi-Level Optical Recording

Ashwin Kumar and Kees A. Schouhamer Immink, *Fellow, IEEE*

Abstract

We report a new method for designing (M, d, k) constrained codes for use in multi-level optical recording channels. The method allow us to design practical codes, which have simple encoder tables and decoders having fixed window length. The codes presented here for the $d = 1$ and $d = 2$ cases, achieve higher storage densities than previously reported codes, and come within $0.3 - 0.7\%$ of capacity.

Index Terms

Multi-level optical recording, runlength-limited sequences, channel capacity, (M, d, k) sequences.

I. INTRODUCTION

An effective approach to achieve high recording densities and data rates in optical data storage systems is multi-level (ML) recording technology. Pit depth modulation (PDM) [1] and electron trapping optical memories (ETOM) [2] are two ML systems that use media allowing multi-level marks to be written on the disc. The user data is stored on the optical disc as a series of ML symbols, thus, increasing the recording density of the disc. Modulation codes are used to transform user data bits into a ML symbol sequence conforming to certain channel constraints. These constraints may require the symbol sequence to have certain spectral properties (e.g. dc-free), prohibit the recording of certain sequences, or assist in timing recovery. The choice of the modulation code influences factors such as achievable storage density, data rate, and detection signal-to-noise ratio (SNR).

The ML runlength limited (ML-RLL) codes are a class of modulation codes used for the ML or M-ary ($M \geq 3$) recording channel. The codes produce (M, d, k) constrained sequences that have at least *d* and at most *k* zeros between consecutive non-zero M-ary symbols [3]. The efficiency of the code is $\eta = R/C$, where R is the rate of the code and C is Shannon capacity [12] of the (M, d, k) constrained channel. Previously reported ML-RLL codes [4], [5], [8] have been constructed using the *state-splitting algorithm* [7] and have efficiencies of upto 98.5%. In [9], two *ad-hoc* construction methods were presented, which give optimal (M, d, k) block codes with low complexity. These methods achieve 90-95% efficiency for the $d = 1$ case [4]. In [11], Immink *et al.* proposed a code construction approach for designing efficient RLL codes for 2-level optical recording. In this paper, we extend the methods reported in [11] to construct M-ary, $d = 1$ and $d = 2$ codes which have rates close to capacity and achieve higher sttorage densities. These codes have simple encoder tables and fixed decoder window lengths.

This paper is organized as follows. In Section II, we detail some properties of (M, d, k) constrained sequences. In Section III, we present the new ML-RLL code design techniques along with illustrative examples on the construction of close to capacity codes.

II. PROPERTIES OF (M, d, k) CONSTRAINED SEQUENCES

A. Capacity of (M, d, k) constrained sequences

A ML-RLL modulation code maps m user bits onto n ML symbols satisfying the (M, d, k) constraints, where m and n are integers. For a (M, d, k) constrained channel, the Shannon capacity C is found using the characteristic equation [3]. For the case (M, d, ∞), the characteristic equation is given by

$$z^{d+1} - z^{d} - (M-1) = 0. \tag{1}$$

Similarly, for the case of $d < k < \infty$, the characteristic equation is given by

$$z^{k+2} - z^{k+1} - (M-1)z^{k-d+1} + M - 1 = 0. \tag{2}$$

The capacity C for either case is obtained as

$$C = \log_2 \lambda, \tag{3}$$

where λ is the largest real root of the respective characteristic equation. Table I shows the capacity C for values of M ranging from 3 to 10, for $d = 1$, and k varying from $d + 1$ to ∞. In [13], it has been shown that for values of M, d and $k = \infty$ that give rational capacity, it is possible to construct 100% efficient codes. But, for $k < \infty$, the capacity C is an irrational number and hence, a code with rational rate R will only be able to approach the capacity.

A. Kumar is with the Electrical and Computer Engineering Department, National University of Singapore, Singapore, and Data Storage Institute, A*Star, Singapore (email: ashwin.kumar@nus.edu.sg).

K. A. S. Immink is with the Institute for Experimental Mathematics, Ellernsstrasse 29, 45326 Essen, Germany (email: immink@exp-math.uni-esen.de).

TABLE I

CAPACITIES FOR (M, d, k) CODES WITH $d = 1$

M / k	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=∞
M=3	0.82317	0.92548	0.96606	0.98390	0.99219	0.99616	0.99810	0.99905	1.00000
M=4	1.07300	1.15423	1.18347	1.19503	1.19982	1.20184	1.20271	1.20309	1.20337
M=5	1.25276	1.32099	1.34371	1.35196	1.35507	1.35626	1.35672	1.35690	1.35702
M=6	1.39362	1.45284	1.47132	1.47755	1.47973	1.48050	1.48078	1.48088	1.48093
M=7	1.50961	1.56216	1.57765	1.58256	1.58417	1.58470	1.58487	1.58493	1.58496
M=8	1.60830	1.65567	1.66895	1.67294	1.67417	1.67455	1.67467	1.67471	1.67472
M=9	1.69424	1.73746	1.74904	1.75235	1.75332	1.75360	1.75369	1.75371	1.75372
M=10	1.77040	1.81019	1.82043	1.82323	1.82401	1.82423	1.82429	1.82431	1.82431

TABLE II

INTEGERS m AND n SUCH THAT $1 < R < C(4, 1, \infty)$

m	n	Code rate R	$(1 - \eta)\%$
6	5	1.20000	0.28
7	6	1.16667	2.7775
8	7	1.14286	5.028
9	8	1.12500	6.5125
13	11	1.18182	1.79
19	16	1.18750	1.318
25	21	1.19048	1.071

B. *Counting (M, d, k) constrained sequences*

Given that the channel symbols are taken from the set $A = \{0, 1, \dots, M-1\}$, where M is the number of possible mark levels, it is of interest to know the number of sequences of length n that satisfy the (M, d, k) constraint. Setting $k = \infty$ and using enumeration, we obtain the number of distinct sequences of length n as

$$N_d(n; M) = 0, \quad \text{for} \quad n < 0 \tag{4a}$$

$$N_d(0; M) = 1. \tag{4b}$$

$$N_d(n; M) = n(M-1) + 1, \quad 1 \le n \le d+1 \tag{4c}$$

$$\begin{aligned} N_d(n; M) &= N_d(n-1; M) \\ &\quad + (M-1) N_d(n-d-1; M), \\ &\qquad \text{for} \quad n > d+1. \end{aligned} \tag{4d}$$

We use the above properties to first choose a rate $R = m/n$ which is as close to capacity as possible and then construct a practical encoder that realizes this rate. In the next section, we describe in detail the method to obtain high rate ML-RLL codes with $d = 1, 2$, and give practical codes constructed using this procedure.

III. EFFICIENT CODING SCHEME

Choose m and n to give rates m/n as close to capacity C as possible. A computer search will give possible values which can be used in practical code design. For $M = 4, d = 1$ case, Table II lists values of m and n that give a code rate R within 7% of capacity $C(4, 1, \infty)$. In the next subsection, we give necessary conditions for the existence of encoders that produce sequences with $d = 1$ constraint and have rate m/n.

A. *Construction of $d = 1$ ML-RLL codes*

We describe the conditions necessary to obtain a finite state encoder that generates M-ary $d = 1$ constrained sequences. Having chosen the values of m and n, we define the set E as the set of possible codewords of length n satisfying the $d = 1$ constraint. The codeword space is partitioned into subsets as shown in Fig. 1. The subsets E_{00}, E_{0i}, E_{i0} and E_{ij}, where $i, j \in \{1, 2, \dots, M-1\}$, are defined as follows. The subset E_{00} consists of all possible codewords beginning and ending with a zero, E_{0i} consists of all possible codewords beginning with a zero and ending with a non-zero symbol i, and so on. The directed arrows signify the *concatenation* rules of codewords to maintain the $d = 1$ constraint. From Fig. 1, it can be noted that codewords in the subset E_{ij} must be followed by codewords in subsets E_{0i} or E_{00}. If there are no directed arrows originating from a subset, then it is assumed that codewords in that subset can be *concatenated* with codewords from any of the other subsets.

Let the total number of encoder states to be r. We divide the set of states into two sets R_1 and R_2 consisting of r_1 and r_2 states, respectively. The set R_1 consists of codewords which start with a '0', while the codewords in R_2 may start with either

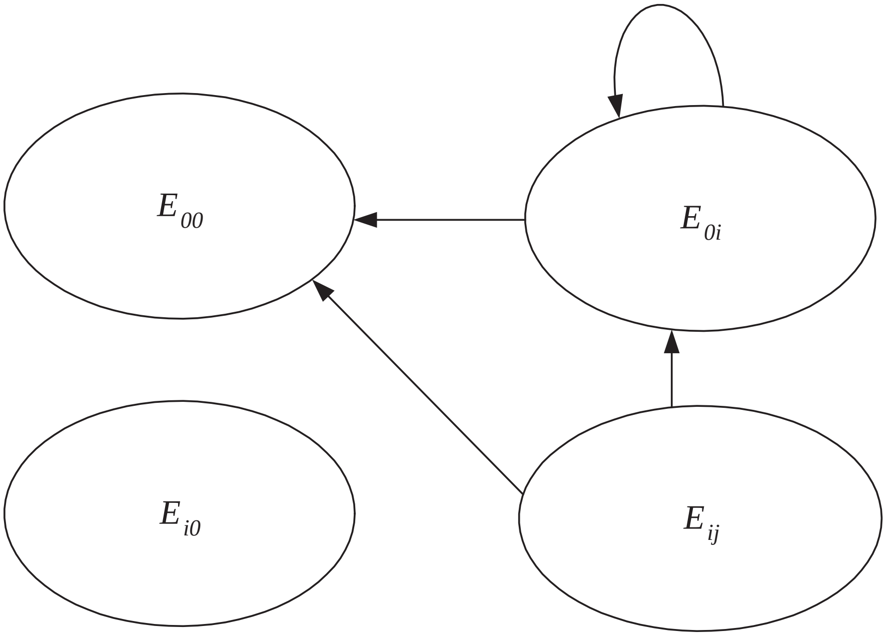

Fig. 1. Subsets of M-ary codewords of length n satisfying the $d = 1$ constraint

a '0' or a non-zero symbol 'i'. We now ignore the k constraint by setting $k = \infty$, and assign the subsets of codewords to encoder states. Codewords in E_{00} and E_{0i} can belong to any of the r states, while those in E_{i0} and E_{ij} can only belong to the states in R_2 states. Applying the rules for *concatenation* of the codewords, we obtain the encoder state transition rules as follows. The codewords ending with a zero may enter any of the r encoder states, but the rest of the codewords may only enter the states in the set R_1. An essential condition is that the sets of codewords belonging to each state should be disjoint. This implies that the codewords can be decoded as having come from a particular state, which makes it possible to assign the same codeword to more than one information word [11]. A sliding block decoder, which observes the current and next codeword, can be used to determine the transmitted information word. Hence, the decoder window length is fixed to 2 n-blocks.

From the encoder transition rules, the 'next state' associated with the codewords in subsets E_{00} and E_{i0} can be any of the r states. Thus, they can be assigned r times to different information words. On the other hand, the 'next state' associated with codewords in subsets E_{0i} and E_{ij} can only be states in set R_1. Thus, the codewords in the subsets E_{0i} and E_{ij} can be assigned r_1 times to information words. Now, let $|E_{xy}|$ denote the cardinality of the set E_{xy}, where $x, y \in \{0, 1, 2, \ldots, M-1\}$. Then, for the r_1 states there are $r|E_{00}| + r_1 \sum_{i=1}^{M-1} |E_{0i}|$ codewords leaving the states. For a rate m/n code, the total number of codewords leaving the r_1 states should be greater than or equal to the number of information words given by $r_1 2^m$. Thus, we get the first condition for the existence of the encoder as

$$r|E_{00}| + r_1 \sum_{i=1}^{M-1} |E_{0i}| \geq r_1 2^m. \tag{5}$$

Similarly, for the r states of the encoder, the total number of codewords leaving them should be greater than or equal to the number of information words $r2^m$. Thus, we have

$$\begin{aligned} & r\left(|E_{00}| + \sum_{i=1}^{M-1} |E_{i0}|\right) \\ & +r_1\left(\sum_{i=1}^{M-1} |E_{0i}| + \sum_{i=1}^{M-1}\sum_{j=1}^{M-1} |E_{ij}|\right) \geq r2^m. \end{aligned} \tag{6}$$

The above equations govern the size of the encoder and result in a specific encoder structure for constructing $d = 1$ constrained codes of rate m/n. A simple computer search will help us to obtain values of r_1 and r that satisfy the above two conditions. Knowing the encoder structure, we can assign the codewords to the corresponding states, to get a practical encoding scheme. In the next subsections, we present two $d = 1$ constrained codes constructed using the above design technique.

TABLE III

VALUES OF r_1 AND r THAT SATISFY THE CONDITIONS GIVEN BY (5) AND (6) FOR $M = 4$, $d = 1$ AND RATE $R = 6/5$

r_1	r_2	r
3	4	7
6	8	14
7	9	16
8	11	19
9	12	21
10	13	23
11	14	25

TABLE IV

DISTRIBUTION OF CODEWORD SUBSETS TO ENCODER STATES FOR $M = 4$, $d = 1$ AND RATE $R = 6/5$ CODE

	Number of codewords for each state						
	Set R_1			Set R_2			
Codeword subsets / State Number	1	2	3	4	5	6	7
E_{00}	5	7	7	0	0	0	0
E_{01}	3	2	2	0	0	0	0
E_{02}	3	2	2	0	0	0	0
E_{03}	4	1	1	1	0	0	0
E_{10}	0	0	0	3	1	2	1
E_{20}	0	0	0	3	2	1	1
E_{30}	0	0	0	3	1	1	2
E_{11}	0	0	0	0	1	1	2
E_{12}	0	0	0	0	1	1	2
E_{13}	0	0	0	0	1	1	2
E_{21}	0	0	0	0	2	1	1
E_{22}	0	0	0	0	1	1	2
E_{23}	0	0	0	0	2	1	1
E_{31}	0	0	0	0	1	1	2
E_{32}	0	0	0	0	1	3	0
E_{33}	0	0	0	0	2	2	0

B. *Rate* $6/5$ $(4, 1, 11)$ *code*

From Table I, we see that the maximum achievable capacity for the ($M = 4$, $d = 1$) code is 1.20337. In [4], a $(4, 1, 2)$ code has been designed with $m = 1$, $n = 1$ and $R = 1$ having efficiency 93.4% and storage density 2 bits/minimum-mark-length. We shall now describe the design of a ($M = 4, d = 1$) code with much higher efficiency and a practical value of k. The first step is to do a computer search to obtain the values of m and n for which higher code rates are achievable, and these are tabulated in Table II. The value of m is limited to 25, in order to reduce the complexity of the encoder. Choosing $m = 6$ and $n = 5$, we now attempt to construct a code having rate $R = m/n = 1.2$, which is 0.28% below capacity and achieves storage density of 2.4 bits/minimum-mark-length. For $M = 4$, $d = 1$ and $n = 5$, using Eqs. 4, the cardinalities of the codeword sets are computed as $|E_{00}| = 19$, $|E_{0i}| = |E_{i0}| = 21$, and $|E_{ij}| = 36$. Substituting these values in Eqs. (5) and (6), the possible values for r and r_1 are given in Table III.

Choosing the least number of states, i.e. $r = 7$, the assignment of the number of codewords from each subset to the encoder states is done as shown in Table IV. Thus, we get the resulting 7-state encoder. Note that the assignment of the codeword subsets to states is not unique and can result in different encoder structures.

From the assignment shown in Table IV, we see that the sum of each row is equal to the total number of codewords in each subset. We now calculate the number of possible codewords for State 1 as $7 \times 5 + 3 \times 10 = 65 > 2^6 = 64$. Thus, we have one codeword that need not be assigned to any transition. Similarly, for other states the total number of codewords is greater than or equal to 2^6. The excess codewords in each state may be used to set the k constraint or to control the dc-content of the symbol sequence. In the above case, we choose the codeword '00000' in State 1 not to be assigned for a transition back to the same state, limiting the k constraint to a maximum of 11. Thus, we have constructed a 7-state encoder with rate $6/5$ to obtain a $(4, 1, 11)$ constrained code. The decoder consists of two lookup tables, one using the next codeword to decide the next state, and the other using its output and the current word to decide on the information word. Thus, a single symbol error will at the most affect two m-bit binary information words.

Using the above design method, we also constructed a rate $7/6$ code with a 2-state encoder to obtain a $(4, 1, 5)$ constrained code. It was observed that the assignment of codewords to the 2-state encoder can be done following a simple rule. This rule states that all the codewords starting with a '0' be assigned to one state, and remainder of the codewords starting with a non-zero symbol be assigned to the other state. This results in a simplied decoder structure, where the state to which a codeword belongs can be decided by observing the first symbol alone.

TABLE V

INTEGERS m AND n SUCH THAT $1 < R < C(8, 1, \infty)$.

m	n	Code rate R	$(1-\eta)\%$
5	3	1.66667	0.481
10	6	1.66667	0.481
8	5	1.6	4.461
11	7	1.57143	6.1675
14	9	1.55556	7.115

TABLE VI

VALUES OF r_1 AND r THAT SATISFY CONDITIONS GIVEN BY EQS. (5) AND (6) FOR $M = 8$, $d = 1$ AND RATE $R = 10/6$

r_1	r_2	r
1	2	3
2	4	6
2	5	7
3	7	10
4	8	12

C. *Rate* $10/6$ $(8, 1, 11)$ *code*

We observe from Table I that the maximum achievable capacity for ($M = 8, d = 1$) is 1.67472. As outlined above, we first find the values of m and n that permit codes with rates close to capacity. Table V gives these values for code rates with efficiency within 8% of the capacity $C(8, 1, \infty)$. Choosing $m = 5$ and $n = 3$, we obtain an encoder structure which does not provide additional sequences. Hence, we choose to construct a ($M = 8, d = 1$) code with $m = 10$ and $n = 6$, which is also only 0.481% below capacity and gives a storage density of 3.33 bits/minimum-mark-length. For $M = 8$, $d = 1$ and $n = 6$, using Eqs. 4, the cardinalities of the codeword sets are computed as $|E_{00}| = 176$, $|E_{0i}| = |E_{i0}| = 71$, and $|E_{ij}| = 15$. Substituting these values in Eqs. (5) and (6), we get the possible values for r and r_1 shown in Table VI. In order to keep the encoder structure as simple as possible, we choose the number of states as $r = 3$. The assignment of the number of codewords from each subset to the encoder states is done as shown in Table VII. Thus, we get the resulting 3-state encoder. The assignments are shown considering the total number of sequences E_{0i}, E_{ij} and E_{i0}, over all nonzero symbols i, j. The actual details can be easily worked out. It can be verified that the row sum adds up to the total number of codewords in each subset. We see that for State 1, the number of codewords is $3 \times 176 + 1 \times 7 \times 71 = 1025 > 2^{10} = 1024$, giving one surplus codeword. As for the other two states, we have an excess of $1113 - 1024 = 89$ codewords, which can be used in imposing additional constraints on the ML symbol sequence. As before, eliminating the occurence of the longest zero sequence, we can restrict k to 11. Thus, we arrive at a rate $10/6$ $(8, 1, 11)$ constrained code for ML optical recording. The decoder structure is the same as before. In the next subsection, we outline the design technique to obtain $d = 2$ constrained M-ary codes.

D. *Constructing* $d = 2$ *codes*

As in the case of $d = 1$, we can now outline a similar method to construct M-ary codes with $d = 2$ constraint. We initially find integers m and n that satisfy $R = m/n < C(M, 2, \infty)$. Next, we divide the codewords into the sets E_{0000}, E_{0i0j}, E_{00i0}, E_{0ij0}, etc, where $i, j \in \{1, 2, \ldots, M-1\}$. The first two symbols denote the two starting symbols of the codeword and the latter two denote the last two symbols, i.e. E_{0ij0} is the subset of codewords starting with '$0i$' and ending with '$j0$'. We define the number of encoder states as r, and divide them into sets R_1, R_2 and R_3 containing r_1, r_2 and r_3 states, respectively. The codewords in set R_1 start with '00', the codewords in set R_2 start with either '$0i$' or '00' and those in the set R_3 start with '$i0$', '$0i$' or '00'. Following the *concatenation* rule to preserve the $d = 2$ constraint, the encoder state transition rules are described as follows. Codewords ending with '00' can enter any of the r states, codewords ending with '$i0$' can enter the states in R_1 or R_2 but not R_3, and codewords ending with '$0i$' can only enter states in R_1. Given that the sets of codewords assigned to the various states are disjoint, decoding can be done unambiguously and each codeword in a given state can be

TABLE VII

DISTRIBUTION OF CODEWORD SUBSETS TO ENCODER STATES FOR $M = 8$, $d = 1$ AND RATE $R = 10/6$ CODE

	Number of codewords for each state		
	Set R_1	Set R_2	
Codeword subsets / State Number	1	2	3
E_{00}	176	0	0
E_{0i}	497	0	0
E_{i0}	0	371	126
E_{ij}	0	0	735

TABLE VIII

LIST OF NEW ML-RLL CODES.

r	m	n	(M,d,k)	$R=m/n$	D	$1-\eta\%$
9	6	5	$(4,1,11)$	1.2	2.4	0.28
2	7	6	$(4,1,5)$	1.167	2.33	3.05
5	4	3	$(5,1,7)$	1.333	2.66	1.75
3	11	7	$(7,1,6)$	1.571	3.142	0.854
3	12	10	$(8,1,11)$	1.667	3.33	0.481
9	14	13	$(6,2,12)$	1.076	3.23	0.429
11	12	10	$(8,2,13)$	1.2	3.6	0.69

assigned multiple times to different input words. Following the encoder transition rules, we see that codewords ending with '00' (i.e. those in the subsets E_{0000}, E_{0i00} and E_{i000}) can have any of the r states as the 'next state',and thus be assigned r times to different input information words. Those that end with '$i0$' can only have $r_1 + r_2$ states as the 'next state', and can be assigned that many times to different information words. Finally, the codewords ending with a non-zero symbol can be assigned r_1 times to different information words. We can now define three conditions required to construct the encoder with rate m/n. With

$$\begin{aligned} A_1 =& r|E_{0000}| + (r_1+r_2)\sum_{i=1}^{M-1}|E_{00i0}| \\ &+ r_1\sum_{i=1}^{M-1}|E_{000i}|, \end{aligned} \tag{7}$$

$$\begin{aligned} A_2 =& r\sum_{i=1}^{M-1}|E_{0i00}| \\ &+(r_1+r_2)\sum_{i=1}^{M-1}\sum_{j=1}^{M-1}|E_{0ij0}| \\ &+r_1\sum_{i=1}^{M-1}\sum_{j=1}^{M-1}|E_{0i0j}|, \end{aligned} \tag{8}$$

$$\begin{aligned} A_3 =& r\sum_{i=1}^{M-1}|E_{i000}| \\ &+(r_1+r_2)\sum_{i=1}^{M-1}\sum_{j=1}^{M-1}|E_{i0j0}| \\ &+r_1\sum_{i=1}^{M-1}\sum_{j=1}^{M-1}|E_{i00j}|, \end{aligned} \tag{9}$$

the conditions are

$$A_1 \geq r_1 2^m, \tag{10}$$

$$A_1 + A_2 \geq (r_1+r_2)2^m, \tag{11}$$

$$A_1 + A_2 + A_3 \geq r2^m. \tag{12}$$

Using Eqs. 4, we can calculate the cardinalities of the codeword subsets for the $d=2$ constrained sequences. Then, we solve the above equations to get values for r_1, r_2 and r_3 which define the encoder structure. As examples, we can verify the existence of an 11-state encoder for $(8,2,13)$ constrained code with rate $12/10$, which is only 0.69% below capacity and achieves a storage density of 3.6 bits/minimum-mark-length. Also, a 9 state encoder for $(6,2,12)$ constrained code with rate $14/13$ can be constructed, which is only 0.429% below capacity and achieves a storage density of 3.23 bits/minimum-mark-length. The $M=6$ code is 5% more efficient than all existing codes, and the $M=8$ code is a new code designed for the ML recording channel. It is possible to find similar highly efficient codes for other values of M.

IV. CONCLUSIONS

We have shown a new construction method for obtaining highly efficient codes for ML optical recording, with $d = 1$ and $d = 2$ constraints. The choice of integers m and n during the design procedure, are seen to affect the complexity of the resulting encoder. The assignment of codeword subsets to states and subsequent assignment of codewords to encoder transistions can be automated using computer programs. Table VIII gives a list codes we have developed, along with number of encoder states r, the achievable storage density D in bits/minimum-mark-length and its proximity to capacity given by $(1 - \eta)\%$.

By designing $6/5$ $(4, 1, 11)$ and $14/13$ $(6, 2, 12)$ constrained codes, we have shown that it is possible to obtain codes that achieve higher rate than existing codes, providing a higher storage density. Also, we have constructed two new codes, a rate $5/3$, $(8, 1, 11)$ code and rate $12/10$, $(8, 2, 13)$ code, both within 0.7% of capacity.

ACKNOWLEDGMENT

The authors would like to thank Dr. G. Mathew, Department of Electrical and Computer Engineering, National University of Singapore, and Dr. K. S. Chan, Data Storage Institute, A*Star, Singapore, for their support and valuable advice related to the work.

REFERENCES

[1] S. Spielman et al., "Using pit-depth modulation to increase capacity and data transfer rates in optical discs," in *Proc. Optical Data Storage Topical Meeting*, May 1996, pp. 17-18.

[2] A. R. Calderbank, R. Laroia and S. W. McLaughlin, "Coded modulation and precoding for electron trapping optical memories" *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1011-1019, Aug. 1998.

[3] C. A. French and J. K. Wolf, "Results involvig (d, k) constrained M-ary codes," *IEEE Trans. Magnetics.*, vol. 23. no. 5, pp. 3678-3680, Sept. 1987.

[4] S. W. McLaughlin, "Five runlength-limited codes for M-ary recording channels," *IEEE Trans. Magnetics.*, vol. 33, no. 3, pp. 2442-2450, Mar. 1997.

[5] B. Vasic, S.Z. Denic, and M.C. Stefanovic, "M-ary RLL dc-free codes for optical recording," *Electron. Lett.*, vol. 36, no. 14, pp. 1214-1216, Jul. 2000.

[6] J. Lee, "4-ary $(1, 2)$ runlength limited code for optical storage channels," *Electron. Lett.*, vol. 35, no. 7, pp. 580-581, Apr. 1999.

[7] R. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes," *IEEE Trans. Inform. Theory.*, vol. IT-29, no. 1, pp. 5-37, Jan. 1992.

[8] S. W. McLaughlin, "Improved distance M-ary (d, k) codes for high density recording," *IEEE Trans. Magnetics.*, vol. 31, no. 2, pp. 1155-1160, Mar. 1995.

[9] S. Datta and S. W. McLaughlin, "Optimal block codes for M-ary runlength constrained channels," *IEEE Trans. Inform. Theory.*, vol. 47, no. 5, pp. 2069-2078, Jul. 2001.

[10] S. W. McLaughlin, L. Yung-Cheng, C. Pepin, and D. Warland, "Multilevel DVD: Coding beyond 3 bits/data-cell," in *Proc. International Symp. on Optical Memory and Optical Data Storage (ISOM/ODS)*, July 2002.

[11] K. A. S. Immink, K. Jin-Yong, S. Sang-Woon, and K. A. Seong, "Efficient dc-Free RLL codes for optical recording," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 326-331, Mar. 2003.

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

[13] S. W. McLaughlin, Jian Luo, and Qun Xie, "On the capacity of M-ary Runlength Limited Codes," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1508-1511, Sept. 1995.