EXTREMELY EFFICIENT DC-FREE RLL CODES FOR OPTICAL RECORDING

Kees A. Schouhamer Immink, Jin-Yong Kim, Sang-Woon Suh and Seong Keun Ahn

ABSTRACT

We will report on new dc-free runlength-limited codes (DCRLL) intended for the next generation of DVD. The efficiency of the newly developed DCRLL schemes is extremely close to the theoretical maximum, and as a result, significant density gains can be obtained with respect to prior art coding schemes.

Keywords: optical recording, capacity, constrained code, runlength-limited, RLL sequence, (d, k) sequence, dc-free code

I. INTRODUCTION

Optical recording, developed in the late 60s and early 70s, is the enabling technology of a series of very successful products for digital consumer electronics systems such as Compact Disc (CD), CD-ROM, CD-R, DVD, and many other products that are still in the offing. Notably spectral shaping (dc-free) and runlength-limited (RLL) codes have found widespread usage in consumer-type mass storage systems such as Compact Disc, DAT, DVD, and so on [1]. The design of codes for optical recording is essentially the design of combined *dc-free* and *runlength limited* (DCRLL) codes. Eight to Fourteen Modulation (EFM) developed by Immink & Ogawa in the early eighties [2] was adopted as the recording code for the Compact Disc (CD). EFMPlus [3], used in the DVD, is a code with the same basic parameters as EFM and a useful six percent higher efficiency. Table I gives a survey of recording codes, which are part of consumer-type optical recording products.

TABLE I
Survey of recording codes and their application area

Device	*Code*	*Type*	*d,k*	*Ref.*
CD	EFM	DCRLL	1,10	[2]
MiniDisc	EFM	DCRLL	1,10	
DVD	EFMPlus	DCRLL	1,10	[3]
DVR	(1,7)PP	DCRLL	1,7	[4]

Kees A. Schouhamer Immink is with Turing Machines Inc, 15 W. Alexanderlaan, 5664 AN Geldrop, The Netherlands. E-mail: immink@turing-machines.com. Jin-Yong Kim, Sang-Woon Suh, Seong Keun Ahn are with DCT Team, Multi-Media Labs, LG Electronics Inc., 16 Woomyeon-Dong, Seocho-Gu, Seoul 137-724, Korea

II. RUNLENGTH-LIMITED CODES

Binary sequences generated by a (d, k) RLL encoder have at least d and at most k 'zero's between successive 'one's. Let the integers m and n denote the information word length and codeword length, respectively. The *code rate*, $R = m/n$, is a measure of the code's efficiency. The maximum rate of an RLL code, given values of d and k, is called the *Shannon capacity*, and it is denoted by $C(d, k)$. As an example, Table II tabulates $C(d = 1, k)$ and $C(d = 2, k)$ for relevant values of k. We may observe, for example, that for $d = 1$ and $k = 7$ the Shannon capacity, $C(1, 7)$, has a value of 0.6793. Thus, an encoder that translates arbitrary sequences into sequences that have at least $d = 1$ and at most $k = 7$ 0's between successive 1's, cannot have a rate larger than 0.6793.

TABLE II
Capacity $C(1, k)$ and $C(2, k)$ as a function of k.

k	$C(1,k)$	$C(2,k)$
7	0.6793	0.5174
8	0.6853	0.5293
9	0.6888	0.5369
10	0.6909	0.5418
11	0.6922	0.5450
∞	0.6942	0.5515

Information recording has a constant need for enhancing the information density on the record carrier, and a possible solution to this end is an increase of the rate of the code.

III. VERY EFFICIENT CODING SCHEMES

For ease of presentation we will first focus on the design of RLL codes with $d = 1$. Later we will extend the ideas to the design of codes with $d = 2$.

Rate 2/3, (1,7) codes are known in the art for more than a quarter of a century, see for example [5, 6]. The code rate, 2/3, of the (1,7) code is slightly less than the Shannon capacity, 0.6793, and the code is therefore a highly efficient one. The efficiency of an RLL code is usually measured by a quantity called *code efficiency*, η, defined by

$$\eta = R/C(d, k). \qquad (1)$$

There are only two approaches for constructing a $(1, k)$ RLL code, whose rate is larger than two-thirds. Firstly, we may relax the maximum runlength k to a value larger than 7. Note that a (1,7) code was first put to practical use in

Original manuscript received June 25, 2001
Revised manuscript received September 26, 2001

the early seventies, and that since the advent of hard-disk drives (HDDs), significant improvements in signal processing for timing recovery circuits have made it possible to employ codes with a much larger maximum runlength k. Secondly, on top of that we may endeavor to design a more efficient code.

The efficiency of the rate 2/3, (1,7) code is 0.6667/0.6793 = 0.981, which reveals that we can at most gain 1.9% in rate by an alternative, more efficient, code redesign. If we fully relax the k constraint, i.e. set $k = \infty$, we can at most gain 3.97% in code rate. In other words, a viable improvement in code rate of a ($d = 1$) encoder ranges from 1.9 to 3.97%. To the best of the author's knowledge, extremely efficient ($d = 1$) codes having a rate exceeding two-thirds have not been reported in the literature. In the sequel of the paper, we will systematically design such extremely efficient codes. We start, in the next subsection, with a simple problem, namely finding integers m and n that improve the rate, 2/3, of the industry standard code.

A. Suitable integers m and n for $d = 1$

We will start with a simple, but very illuminating exercise, namely a search for pairs of integers m and n that are suitable candidates for a coding rate exceeding 2/3. All pairs of integers $2/3 < m/n < C(1,\infty)$, $n < 50$, are shown in Table III. Surprisingly there are just six m and n pairs whose quotient is larger than 2/3. We omitted trivial pairs, such as 18 and 26 etc., that are multiples of given smaller pairs. Perusal of the table reveals that the code rate $m/n = 9/13$ is highly attractive as it is just 0.28% below the Shannon capacity $C(1,\infty)$. The fact that the quotient 9/13 is less than capacity does not mean that a code with that rate can be *practically* constructed.

TABLE III

Integers m and n such that $2/3 < R = m/n < C(1,\infty)$. The quantity $\eta = R/C(1,\infty)$ expresses the code efficiency.

m	n	$1-\eta$ %
34	49	0.0525
9	13	0.2786
11	16	0.9711
13	19	1.4449
15	22	1.7895
17	25	2.0514

B. Encoder description

We start with a few ubiquitous definitions. The encoder has r states, which are divided into two state subsets of a first and second type. The state subsets are of size r_1 and $r_2 (= r - r_1)$, respectively. A codeword is a binary string of length n that satisfies the $d = 1$ constraint. The encoder state-transition rules are easily described. Codewords that end with a '0', i.e., codewords in subsets E_{00} and E_{10} may enter any of the r encoder states. Codewords that end with a '1' may be followed by codewords in the r_1 states of the first type only. With the above model we were able to construct many new codes including a rate 9/13, (1,14) code. Clearly this new code improves the rate of the traditional rate 2/3, (1,7) code by a factor of 27/26 (≈ 1.038) without seriously compromising the timing regeneration.

IV. Efficient $d = 2$ codes

Up till now we have concentrated on the design of efficient $d = 1$ codes, and as both code parameters, $d = 1$ and $d = 2$, are of great practical interest for optical recording, we will now repeat the exercise for the case $d = 2$.

A. Suitable integers m and n for $d = 2$

RLL codes with minimum runlength parameter $d = 2$ have been widely published. The highest reported rate of such a ($d = 2$) code is 8/15[1]. Table II tabulates $C(2,k)$ as a function of k, and from this table the reader can easily discern the head room available for the design of a code of rate $R = m/n > 8/15$. The rate 8/15 is, see Table II, 3.3% below channel capacity $C(2,\infty)$. Table IV shows values of m and n, where $8/15 \le m/n < C(2,\infty)$ and $n < 50$. The pairs of integers are ordered according to their efficiency $R/C(2,\infty)$. Clearly, the quotients 11/20, 6/11, and 7/13

TABLE IV

Integers m and n such that $8/15 < R = m/n < C(2,\infty)$. The quantity $\eta = R/C(2,\infty)$ expresses the code efficiency.

m	n	$1-\eta$ %
11	20	0.2720
17	31	0.5644
6	11	1.0962
19	35	1.5672
13	24	1.7830
20	37	1.9872
7	13	2.3642
15	28	2.8623
8	15	3.2940

are suitable candidate rates for the creation of small ($d = 2$) codes.

B. Encoder description

In this section we will describe a finite-state encoder that generates sequences that satisfy the $d = 2$ constraint (note that the k constraint will be ignored for a while). The encoder is assumed to have r states, which are divided into three state subsets of states of a first, second, and third type. The state subsets are of size r_1, r_2, and $r_3 (= r - r_1 - r_2)$, respectively. Codewords that end with the string '00' may enter any of the r encoder states. Codewords that end with a '10' may not be followed by codewords in a state of the third type. Similarly, codewords that end with a '1' may only be followed by codewords belonging to states of the first type. Table V summarizes the new

[1] At press time, the author became aware that Kim [7] has been granted a U.S. Patent on an example of a rate 7/13, (2,25) code.

RLL codes, $d = 1$ and $d = 2$, we have found. As we can see, the efficiency of the majority of the new codes is just a few tenths of a percent below capacity. At this junction, we have completed the description of the new RLL codes, and we are in the position to describe how we can turn the newly developed RLL codes into DCRLL codes.

TABLE V
Survey of newly developed codes.

m	n	d	k	states	$\eta = R/C(d,k)$
11	20	2	23	9	0.9975
7	13	2	11	9	0.9880
6	11	2	15	9	0.9915
9	13	1	14	13	0.9979
9	13	1	18	5	0.9973
11	16	1	10	13	0.9951

V. Guided Scrambling

There are various methods to transform an RLL code into a DCRLL code [1] by adding redundant dc-control bits, which are chosen by the encoder to optimize the spectral performance of the generated sequence. Obviously, we can multiplex, either at data or channel level, the data stream with the dc-control bits. Alternatively, a promising method for adapting an RLL code is *Guided Scrambling (GS)* [1]. In GS, each information word can be represented by a member of a selection set consisting of $L = 2^p$, $p \geq 1$, codewords. The encoder generates the selection set, and the "best" (according to a predefined penalty function) codeword in the selection set is selected for transmission. The RLL codes, listed in Table V, will be employed in conjunction with GS for achieving four goals:

- spectral shaping;
- rejection of long runs of '0's: k constraint;
- rejection of long transition runs of '01's (d=1) or '001's (d=2): MTR constraint;
- rejection of predefined sync(hronization) patterns, sync constraint.

The maximum runlength constraint, k, imposed by the GS penalty function can be made smaller than that of the inner RLL code. It has been found that constraining a long repetition of minimum transition runs (MTR constraint), '1010101...' (d=1) or '1001001001...' (d=2), in conjunction with Partial-Response (PR) detection is beneficial to the system margins. Naturally, the GS method cannot fully guarantee the k and MTR constraints, but the probability of occurrence of such vexatious subsequences can be made extremely small.

A. Format

In the GS format, m_1 user bits are multiplexed with p redundant bits, which are a part of the input of the channel encoder. The p redundant bits are used to generate a selection set of size $L = 2^p$. Each member of the selection set is generated and tested by the encoder with respect to the penalty function. In the proposed coding format, the channel encoder input comprises p redundant bits plus m_1 user bits that from a *super* block. The integers p and m_1 are integers chosen such that

$$Km = p + m_1, \tag{2}$$

where K is an integer that denotes the number of m-bit information words in a super block. In a practical environment of a byte-oriented system, m_1 is preferably a multiple of eight, i.e. $m_1 \bmod 8 = 0$. Under the rules of the RLL code, the $p + m_1 = Km$-bit super block is translated into Kn channel bits. Thus, the overall rate, R_o, of the code is

$$R_o = \frac{m_1}{Kn} = \frac{Km - p}{Kn}. \tag{3}$$

B. Results and comparison with prior art methods

The Power Spectral Density (PSD), $H(f)$, and other relevant characteristics can easily be measured using computer simulation. As a typical example, we will show results obtained with the rate 9/13, (1,14) RLL code. Figure 1 shows the spectrum, $H(f)$, versus (channel) frequency, f, for $p = 5$, $k = 10$, and $K = 45$. The overall code rate is $R_o = (Km - p)/Kn = (45 \times 9 - 5)/(45 \times 13) = 0.68376$. Note that the overall code is byte oriented as $Km - p = 400$ is a multiple of eight. In the runlength penalty function, we set the maximum 'zero' runlength to $k = 10$, which means that the code essentially behaves as a $(d = 1, k = 10)$ code. The spectrum, $H(f)$, versus frequency f has a parabolic shape in the low-frequency range, which shows as a straight line as a result of the logarithmic frequency axis used. We can employ the spectral density at a, given, low frequency as the low-frequency (lf) spectral performance yard stick of a DCRLL code. Results are shown in Figure 2 for $p = 5$ and $p = 8$. In order to compare our results with the maximum theoretical performance of DCRLL codes, we invoked the algorithms found in [1, pages 282-286] which compute the *maxentropic* performance of

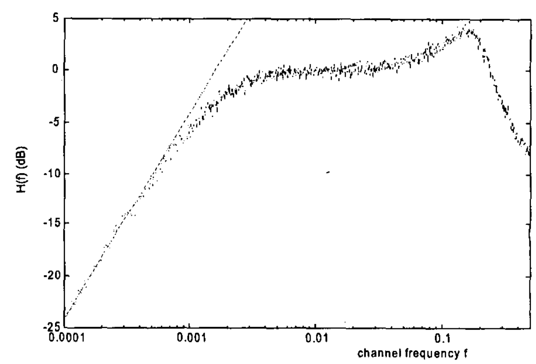

Fig. 1. Simulation results of a PSD function of a $(d = 1, k = 10)$ code of overall rate $R_o = 0.68376$. The straight line is a 'best fit' estimate of the low-frequency part of the spectrum. We simply discern that $H(f = 10^{-4}) = -24.3$ dB

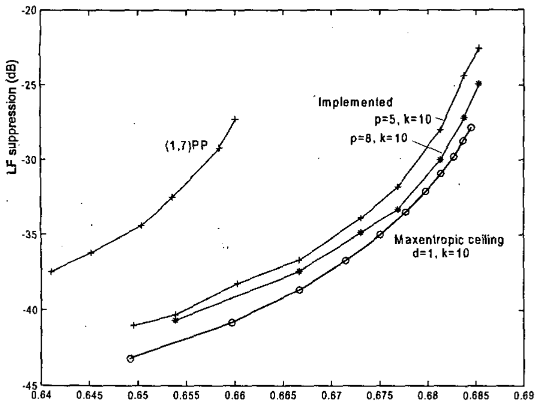

Fig. 2. The two upper curves show the lf suppression, $H(10^{-4})$, as a function of the overall code rate R_o. The upper curve shows results for $p = 5$, and the lower curve for $p = 8$. The maximum imposed runlength for both cases is $k = 10$. The curve denoted by (1,7)PP gives results of a prior art code [8].

(d, k) codes. The maxentropic performance sets a theoretical limit to the performance of any implemented DCRLL code. Figure 2 shows that the implemented codes operate very close to the best theoretical performance. For $p = 5$ the implemented codes are 2-3 dB, (for $p = 8$, 1-2 dB) below the theoretical ceiling. As a further comparison we plotted the performance of a prior art rate 2/3, (1,7) code [4], which is extended with dc-control bits on data sequence level.

Figure 3 shows the lf spectral performance of the rate 6/11, (2,15) code in conjunction with Guided Scrambling. Results are given for $p = 5$ and $p = 8$. As reported in the above $d = 1$ case, the combination of an efficient RLL code and GS works quite satisfactorily as only 2-3 dB can be gained with respect to the theoretical ceiling.

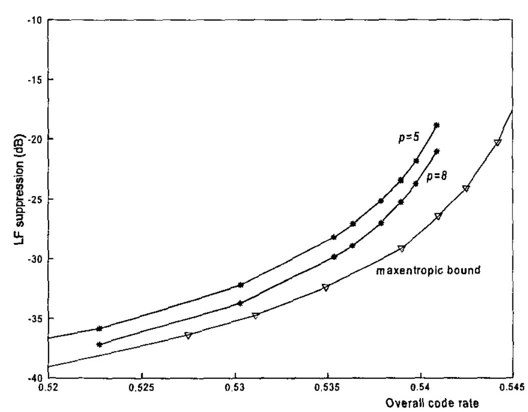

Fig. 3. The two upper curves show the lf suppression, $H(10^{-4})$, as a function of the overall rate R_o. The upper curve is for $p = 5$, and the lower curve is for $p = 8$. The maximum imposed runlength for both cases is $k = 12$. As a comparison we plotted the theoretical ceiling, $H_{\min}(10^{-4})$, of maxentropic $(d = 2, k = 12)$ sequences.

VI. Conclusions

We have studied the construction of extremely efficient runlength-limited (RLL) codes. We have shown that there is a very limited number of pairs of integers m and n, whose quotient m/n form a suitable coding rate for $(d = 1)$ and $(d = 2)$ RLL codes that are more efficient than prior art codes. Suitable values for the rate of a $(d = 1)$ code are 9/13 and 11/16, while for $(d = 2)$ codes we have 11/20, 7/13, and 6/11.

We have disclosed a novel technique for designing very efficient RLL codes. Using the novel technique we constructed a series of new RLL codes, whose rate is only a few tenths below capacity. For example, we have found a 13-state rate 9/13, (1,14) RLL code, whose rate is only 0.2% below channel capacity $C(1, 14)$. In addition, we have constructed a new rate 6/11, (2,15) code, a rate 11/20, (2,23) code, and a rate 7/13, (2,11) code.

The above, and other, RLL codes can be employed in conjunction with Guided Scrambling (GS), or other techniques, to turn them into DC-free RLL codes, which suppress the low frequency (lf) components. Under the rules of the GS algorithm, a selection set of alternative candidate codewords is generated, and the candidate with the least lf spectral content (or other desirable attributes) is transmitted. Results of computer simulations have shown that the arrangement of the newly developed RLL codes in conjunction with GS is extremely efficient in terms of overall rate and spectral performance. With the newly developed rate 9/13, $d = 1$ code as an inner code, we have achieved a 4.5% better overall rate than possible with the prior art (1,7)PP code, and with the newly developed rate 6/11, $d = 2$ code we have achieved a 9.3% higher overall rate than that of EFMPlus.

The new DCRLL codes perform quite well in absolute terms as we have shown that only a few dB in spectral performance can be gained with respect to the theoretical ceiling.

References

[1] K.A.S. Immink, *Codes for Mass Data Storage Systems*, ISBN 90-74249-23-X, Shannon Foundation Publishers, Netherlands, 1999.

[2] K.A.S. Immink and H. Ogawa, 'Method for Encoding Binary Data', US Patent 4,501,000, Feb. 1985.

[3] K.A.S. Immink, 'EFMPlus: The Coding Format of the MultiMedia Compact Disc', *IEEE Trans. Consumer Electr.*, vol. CE-41, pp. 491-497, Aug..1995.

[4] K.A.S. Immink, J.A.H. Kahlman, G. van den Enden, T. Nakagawa, Y. Shinpuku, T. Narahara, and K. Nakamura, 'Apparatus and method for modulation/demodulation with consecutive minimum runlength limitation', Patent Application WO 9963671A1, Issued Dec. 1999.

[5] P.A. Franaszek, 'On Future-dependent Block Coding for Input-restricted Channels', *IBM J. Res. Develop.*, vol. 23, pp. 75-81, 1979.

[6] G.V. Jacoby and R. Kost, 'Binary Two-thirds Rate Code with Full Word Look-Ahead', *IEEE Trans. Magn.*, vol. MAG-20, no. 5, pp. 709-714, Sept. 1984.

[7] M.J. Kim, '7/13 Channel Coding and Decoding Method Using RLL(2,25) code', US Patent 6,188,336, Feb. 2001.

[8] K.A.S. Immink, 'EFM coding: Squeezing the last bits', *IEEE Trans. Consumer Electronics*, vol. CE-43, pp. 491-495, Aug. 1997.

BIOGRAPHY

Kees A. Schouhamer Immink, obtained M.S. and Ph.D degrees at the Eindhoven University of Technology. He is founder and president of Turing Machines Inc. Since 1995, he is an adjunct professor at the Institute for Experimental Mathematics, Essen University, Germany. In addition, he is affiliated with the National University of Singapore.
He has contributed to the design and development of a wide variety of consumer-type audio and video recorders such as the LaserVision video disc, Compact Disc, Compact Disc Video, DAT, DV, DCC, and DVD. He holds 52 issued and pending US patents in various fields.
Dr Immink is an elected member of the Royal Netherlands Academy of Arts and Sciences (KNAW) and holds fellowships of the IEEE, AES, SMPTE, and IEE. For his contributions to the digital audio and video revolution, he received wide recognition such as a Knighthood from Beatrix, Queen of the Netherlands, the 1999 IEEE Edison Medal, AES Gold Medal, IEEE Masaru Ibuka Consumer Electronics Award, and the Golden Jubilee Award for Technological Innovation awarded by the IEEE Information Theory Society in 1998.
He is vice president of the Audio Engineering Society (AES) and a governor of the IEEE Consumer Electronics Society, and a member of the IEEE Fellows Committee.

Jin-Yong Kim received his B.S. degree in electronic engineering from Seoul National University in 1983, his M.S. degree in electrical engineering from KAIST in 1985, and Ph.D degree in electrical engineering from Iowa State University in 1992 respectively. Dr. Kim is currently employed as a Research Fellow at Digital Media Research Laboratory, LG Electronics, Seoul Korea.

Sang-Woon Suh was born in Seoul, Korea, on May 20, 1964. He received the B.S. degree in electronics engineering from Seogang Univeristy, Seoul, Korea, in 1987, and the M.S. degree in Information and Communication engineering form Korea Advances Institute of Science and Technology(KAIST), Korea, in 1997. From 1987 - 1990, he was with Samsung Electronics Korea as a Engineer. From 1990 - present, he was with LG Electronic Inc., Korea as a senior Research Engineer. His current research interests included Optical Data Storage, modulation code for optical discs and physical format for optical discs.

Seong-Keun Ahn received the B.S. and M.S. degree in school of electrical engineering from Seoul National University,Seoul,Korea in 1996 and 1998,respectively. Mr. Ahn is currently a research engineer at Digital Media Laboratory,LG Electronics.