

A Survey of Codes for Optical Disk Recording

Kees A. Schouhamer Immink, *Fellow, IEEE*

Abstract—We report on 20 years of development of codes for optical disk recording systems. A description of the state-of-the-art and feasible options for future extensions and improvements are given.

Index Terms—Constrained code, dc-free code, EFM, optical recording, runlength-limited code.

I. INTRODUCTION

OPTICAL recording, developed in the late 1960s and early 1970s, is the enabling technology of a series of very successful products for digital mass data storage systems such as compact disk (CD), CD-ROM, CD-R, DVD, and many other products that are still in the offing. Eight to fourteen modulation (EFM) developed by Immink and Ogawa in the early 1980s [1] was adopted as the recording code for the CD. Almost 15 years later, the DVD, the successor of the CD, was developed. The DVD uses EFMPlus [2], a code with the same basic parameters as EFM but a slightly (6%) higher rate. Notably, spectral shaping and runlength-limited (RLL) codes have found widespread usage in consumer-type mass storage systems such as CD, DAT, DVD, and so on [3]. The design of codes for optical recording is essentially the design of combined dc-free and runlength-limited (DCRLL) codes. Table I gives a survey of recording codes currently in use by consumer-type optical recording products. An RLL code is very useful in optical recording, where replicas are made for mass distribution. The replication of disks with very small pits and lands turns out to be very difficult, leading to unacceptably high bit error rates. The minimum pit size of RLL sequences is larger than those of uncoded counterparts so that a higher density can be obtained without sacrificing the reliability. A dc-free code makes it possible to use simple servo systems that extract tracking information from the data track without any specific additional considerations. Two design considerations include increased minimum feature size for replication and sufficient rejection of low-frequency components to enable a simple noise-free tracking. As the reading of disks is virtually noiseless, other properties, such as, for example, robustness against additive noise play a minor role. Read errors are mostly caused by imperfections of the disk and can be resolved by sophisticated error-burst correcting Reed–Solomon codes [7].

We start, in the next sections, with an outline of the properties of DCRLL sequences. Thereafter, it will be shown how practical codes can be devised that satisfy the given channel constraints. It will be shown, among others, that industry-standard RLL codes can be supplemented by a simple mechanism with which the

TABLE I
SURVEY OF RECORDING CODES AND THEIR APPLICATION AREA

<i>Device</i>	<i>Code</i>	<i>Type</i>	<i>Ref.</i>
Compact Disc	EFM	RLL, dc-free	[4]
MiniDisc	EFM	RLL, dc-free	[5]
DVD	EFMPlus	RLL, dc-free	[2]
DVR	(1,7)PP	RLL, dc-free	[6]

if-components of the generated sequences can be suppressed. In the remaining part of this article, we will study the construction and performance of alternative EFM-like codes.

We start in the next section, with the definition of (d, k) RLL sequences, and the computation of a number of basic properties of such sequences.

II. PROPERTIES OF RLL AND DCRLL SEQUENCES

A (d, k) RLL sequence is a string of symbols of ones and zeros with at least d and at most k zeros between consecutive ones. Channel codes are needed to translate arbitrary data into (d, k) sequences. In general, a (dk) sequence is not employed in optical or magnetic recording without a simple coding step. A (dk) sequence is converted to an RLL channel sequence in the following way. Let the channel signals be represented by a bipolar sequence $\{y_i\}$, $y_i \in \{-1, 1\}$. The channel signals represent the positive or negative magnetization of the recording medium, or pits or lands when dealing with optical recording. The logical ones in the (dk) sequence indicate the positions of a transition $1 \rightarrow -1$ or $-1 \rightarrow 1$ of the corresponding RLL sequence. The (dk) sequence

0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 1 ...

would be converted to the RLL channel sequence

1 -1 -1 -1 -1 1 1 1 -1 -1 -1 -1 1 -1 -1 1 ...

The mapping of the waveform by this coding step is known as precoding. It can readily be verified that the minimum and maximum distance between consecutive transitions of the RLL sequence derived from a (dk) sequence is $d+1$ and $k+1$ symbols, respectively; or, in other words, the RLL sequence has the virtue that at least $d+1$ and at most $k+1$ consecutive-like symbols (runs) occur.

An encoder translates arbitrary user (or source) information into, in this particular instance, a sequence that satisfies given dk constraints. On the average, m source symbols are translated into n channel symbols. What is the maximum value of $R = m/n$ that can be attained for some specified values of the minimum and maximum runlength d and k ? Using the basic techniques developed by Shannon presented, for example, in

Manuscript received June 20, 2000; revised August 1, 2000.

The author is with the Institute for Experimental Mathematics, 45326 Essen, Germany (e-mail: immink@exp-math.uni-essen.de).

Publisher Item Identifier S 0733-8716(01)01767-X.

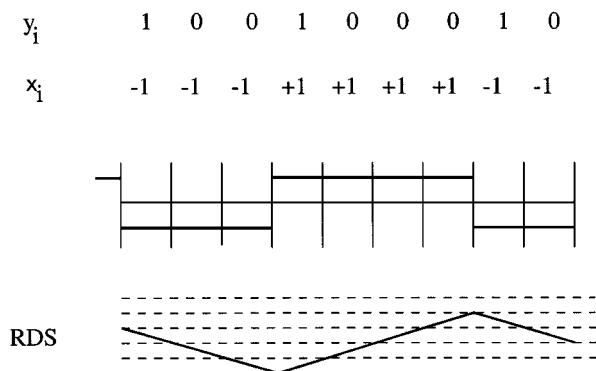


Fig. 1. RDS versus time. Input symbols are translated into the write signal and the channel bits x_i . In this example, the RDS assumes at most seven values.

[3, Ch. 2], it is fairly straightforward to compute the maximum value of R , called capacity, of encoders that generate sequences with a constraint on the minimum and maximum runlength. The capacity $C(d, k)$ of RLL sequences is [3, Ch. 4]

$$C(d, k) = \log_2 \lambda_{dk} \quad (1)$$

where λ_{dk} is the largest real root of the characteristic equation

$$z^{k+1} - z^{k-d} - z^{k-d-1} - \dots - z - 1 = 0, \quad (2)$$

RLL sequences used in optical disk recording systems should satisfy the additional requirement that the low-frequency components are sufficiently small. Sequences with such a property are usually called dc-free sequences. The running digital sum of a sequence, in short, RDS, plays a significant role in the analysis and synthesis of codes whose spectrum vanishes at the low-frequency end. Let

$$\{x_i\} = \{\dots, x_{-1}, x_0, \dots, x_i, \dots\}, \quad x_i \in \{-1, 1\}$$

be a binary sequence. The (running) digital sum z_i is defined as

$$z_i = \sum_{j=-\infty}^i x_j = z_{i-1} + x_i. \quad (3)$$

Fig. 1 portrays the various signals. Chien [8] studied bipolar sequences $\{x_i\}$, $x_i \in \{-1, 1\}$ that assume a finite number of sum values; that is, at any instant i , the RDS z_i of such a sequence meets the condition

$$N_1 \leq z_i \leq N_2,$$

where N_1 and N_2 are two (finite) constants, $N_2 > N_1$. Sequences that have a bound to the number of assumed sum values are termed z -(constrained) or RDS-constrained sequences. The total number of sum values a sequence assumes, denoted by

$$N = N_2 - N_1 + 1, \quad (4)$$

is often called the digital sum variation (DSV). Pierobon [9] showed that the power density function of an encoded sequence $\{x_i\}$ vanishes at zero frequency if, and only if, the encoder is a finite RDS encoder. The capacity $C(0, N-2, N)$ of “pure”

dc-free sequences; i.e., sequences that assume a maximum of N sum values were derived by Chien [8]

$$C(0, N-2, N) = 1 + \log_2 \cos \frac{\pi}{N+1}. \quad (5)$$

The capacity of dc-free RLL sequences has been computed by Norris and Bloomberg [10].

The capacity of an RLL sequence with a bounded RDS is characterized by three parameters d , k , and N , and it will be denoted by $C(d, k, N)$. We will not follow the derivation given by Norris and Bloomberg, but will offer an alternative approach in the next section that will make it possible to efficiently compute the power density function of DCRL sequences as well.

A. Capacity and Spectral Properties of DCRL Sequences

Kerpez [11] presented a description of the combined (d, k) and N constraint in terms of a variable length graph and its adjacency matrix that requires a relatively small number, $N-1-d$, of states.

Let the DCRL message be denoted by $X = \{x_0, x_1, \dots\}$, $x_i \in \{-1, 1\}$. The RLL message is assumed to be composed of runlengths of lengths T_i , $i = 0, 1, \dots$, taken from the set of allowed runlengths $dk = \{d+1, \dots, k+1\}$. As the sequence X has limited DSV, we have

$$\left| \sum_{i=0}^j x_i \right| \leq c,$$

where $N = 2c + 1$. The above constraint can be described in terms of the runlengths. The sequence X is composed of a cascade of runlengths T_i whose symbols have alternate polarity. Transitions of the polarity of the sequence X , i.e., instants where $x_i \neq x_{i+1}$, occur therefore at $t_j = \sum_{i=0}^j T_i$. We simply find

$$\left| \sum_{i=0}^{t_j} x_i \right| = \left| \sum_{i=1}^j (-1)^i T_i \right| = |U_j|,$$

where the sequence $U_j = T_j - U_{j-1}$, $U_0 = 0$. In other words, the DSV constraint is equivalent to

$$|U_j| \leq c \quad \text{for all } j.$$

Thus, a sequence satisfies the $(d, k, N = 2c + 1)$ constraint if and only if the sequence of runlengths $\{T_j\}$ satisfies, for all j

$$d+1 \leq T_j \leq k+1 (\leq 2c) \quad (6)$$

and

$$d+1-c \leq U_j \leq c. \quad (7)$$

The general form of the adjacency matrix D for the $(d, k, N = 2c + 1)$ constraint, derived from (6) and (7), has a regular structure. The matrix has size $(N-1-d) \times (N-1-d)$ and is constant on the anti-diagonals. If the value of k is nontrivial, i.e., $k+1 \leq N-1$, the lower right of the diagonal of the matrix is zero, whereas in the case of $k = N-2$, the lower right corner is filled. Using the above adjacency matrix it is now straightforward to compute the capacity and the spectrum of the maxentropic sequence for large values of N and k . A maxentropic

TABLE II
CAPACITY OF DC-BALANCED RUNLENGTH-LIMITED SEQUENCES

d	k	$N = 5$	$N = 6$	$N = 7$	$N = 8$	$N = 9$
0	1	.6358	.6551	.6662	.6731	.6778
0	2	.7664	.8032	.8244	.8378	.8468
0	3	.7925	.8416	.8704	.8887	.9012
0	4		.8495	.8832	.9048	.9196
0	5			.8858	.9094	.9256
0	6				.9103	.9273
0	7					.9276
1	2	.3471	.3705	.3822	.3889	.3931
1	3	.4248	.4746	.5000	.5145	.5237
1	4		.5018	.5390	.5608	.5746
1	5			.5497	.5772	.5947
1	6				.5816	.6020
1	7					.6039
2	3	.2028	.2457	.2625	.2709	.2757
2	4		.3089	.3471	.3666	.3777
2	5			.3723	.4024	.4199
2	6				.4135	.4366
2	7					.4418
3	4		.1568	.1903	.2035	.2101
3	5			.2434	.2744	.2902
3	6				.2972	.3224
3	7					.3333

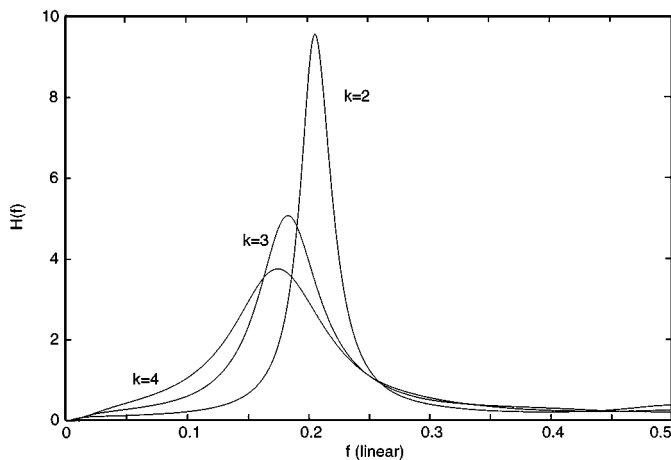


Fig. 2. Power spectral density function of a maxentropic dc-balanced, RLL sequence. DSV $N = 7$, and runlength parameters $d = 1$ and $k = 2, 3, 4$.

sequence is assumed to be generated by a Markov source whose transition matrix is chosen such that the entropy of that source equals capacity. Spectral and other properties of maxentropic sequences are assumed to be a sound predictions of sequences generated by efficient encoders. Table II shows the results of computations for various values of DSV and runlength parameter. The following examples may serve to illustrate the theory. Fig. 2 shows the power spectral density function of maxentropic sequences with $N = 7$, $d = 1$ and the maximum runlength k as a parameter. Apparently, the influence of the maximum runlength parameter is drastic. Most noticeable is the fact that the curves become more peaked with decreasing maximum runlength parameter k . Fig. 3 shows the spectrum for the parameters

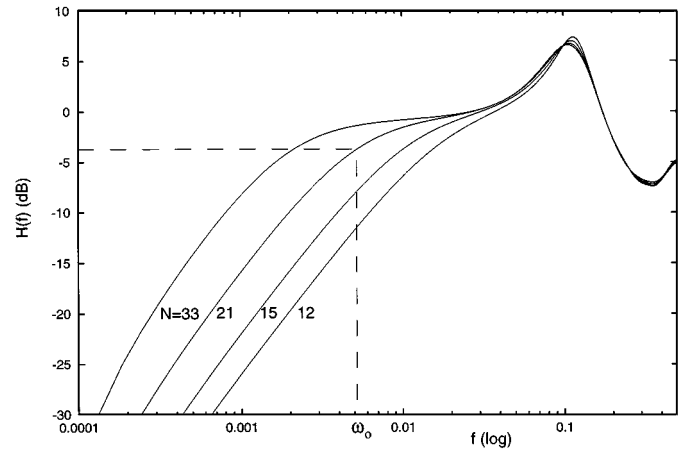


Fig. 3. Power spectral density function of a maxentropic dc-balanced RLL sequence (logarithmic axes). Runlength parameters $d = 2$ and $k = 10$ with N as a parameter. By way of example, the cutoff frequency ω_0 is shown for $d = 2$, $k = 10$, and $N = 21$.

$d = 2$, $k = 10$ and $N = 12, 15, 21$, and $N = 33$ with a logarithmic frequency-axis and a vertical $H(f)$ (dB) axis, where a decibel is defined by $10 \log_{10} H(f)$. The choice of the log axes clearly shows the parabolic relationship

$$H(f) \approx af^2$$

between power and frequency in the low-frequency range. The low-frequency power increases with 6 dB per octave (or 20 dB per decade) frequency increase. We need a sound yardstick for measuring the low-frequency properties of DCRL sequences. The spectral width is usually quantified by a parameter called cutoff frequency ω_0 [3, Ch. 9]. Braun [12] defined the cutoff frequency of DCRL sequences, denoted by ω_0 , by

$$H(\omega_0) = \frac{H_0(d, k)}{2} \quad (8)$$

where $H_0(d, k)$ denotes the spectral density at zero frequency of the maxentropic (d, k) constrained sequence. For $d = 2$ and $k = 10$, the parameters used in Fig. 3, we find $H_0(d, k) = 0.835$ ($= -0.8$ dB).

Braun also studied the relationship between redundancy and cutoff frequency. He defined the extra rate loss, $\rho(d, k, N)$, as the difference between the capacities of the pure RLL channel and the DCRL channel, or

$$\rho(d, k, N) = C(d, k, \infty) - C(d, k, N). \quad (9)$$

The parameter $\rho(d, k, N)$ quantifies the extra rate loss that results from the additional constraint on the RDS N . Braun showed that maxentropic sequences have the property that there is, in good approximation, a linear relationship between cutoff frequency ω_0 and the extra rate loss ρ . The relationship found is given by

$$\omega_0 \approx \rho(d, k, N) \frac{2 \ln 2}{\frac{\pi^2}{6} - 1}, \quad N \gg 1. \quad (10)$$

The constant of proportionality between cutoff frequency and extra rate loss ρ is independent of the d , k , and N constraints.

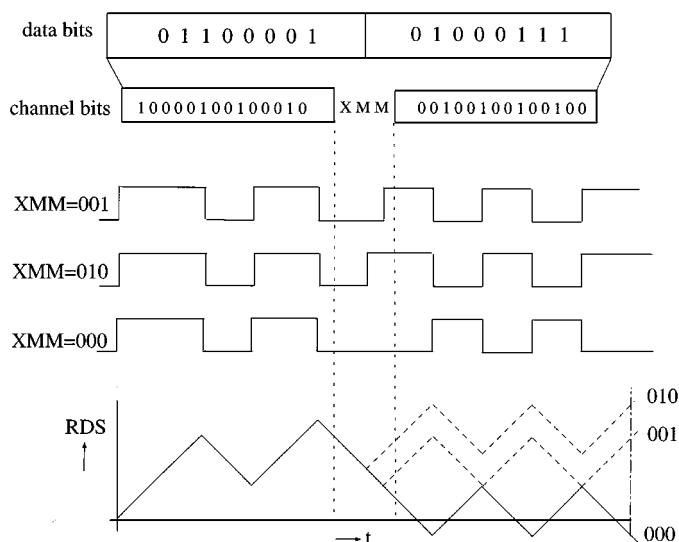


Fig. 4. Strategy for minimizing the RDS.

The constant was derived for pure RDS constrained sequences and is valid if in addition d and k constraints are imposed. Computer simulations revealed that the above relationship appears to be accurate to within 5% for $N > 20$.

In the next section, we will discuss various design methods for constructing DCRL codes that have emerged in the literature.

III. EFM

The main parameters of EFM are $d = 2$, $k = 10$, and rate $R = 8/17$. Detailed information on code tables and so on can be found in the patent granted to Immink and Ogawa [1]. The 8-bit source data are translated into a 14-bit d -constrained word. The 14-bit words are concatenated with 3-bit words, merging words. The 3-bit merging words are selected by the encoder such that the minimum and maximum runlength are guaranteed. There are instances, however, where the merging word is not uniquely governed by the minimum and maximum runlength requirements. This freedom of choice is used for minimizing the power density at the low-frequency end, as will be explained with Fig. 4. Fig. 4 shows an example of the merging process. Eight user bits are translated into 14 channel bits using a look-up table. The 14 bits are merged by means of three merging bits in such a way that the runlength conditions continue to be satisfied. For the case shown in Fig. 4, the condition that there should be at least two zeros between ones requires a zero at the first merging bit position. There are, thus, three alternatives for the merging bits: 000, 010, and 001. The encoder chooses the alternative that gives the lowest absolute value of the RDS at the end of a new codeword, i.e., 100 in this case. In the experimental phase of the CD [13], it was learned that the suppression of low-frequency components, when only two merging bits are used, is not sufficiently effective. Thus, the number of merging bits was increased to three, so providing a greater degree of freedom to set or omit transitions in the merging bits. With three merging bits in 65% of the block concatenations, a transition can be set or omitted freely. The more effective low-frequency control is achieved at the expense of 1/17 of the information rate.

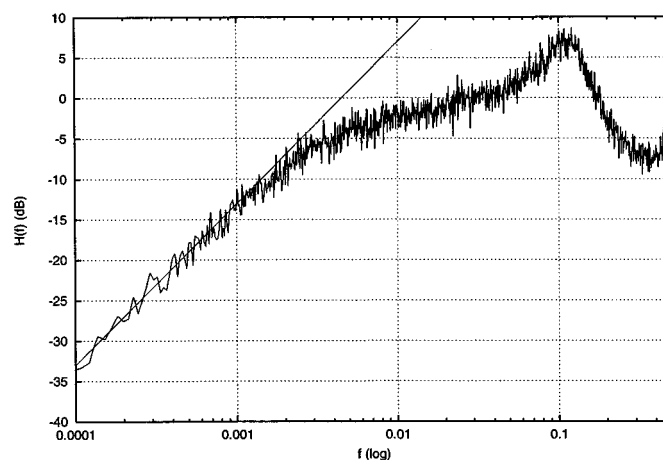


Fig. 5. Spectrum of classic EFM. The straight line is a least-squares mean estimate of the low-frequency part of the spectrum.

In principle, better suppression of the low-frequency components can be obtained, without offending the agreed standard for the CD system, by applying improved merging strategies. For example, by looking more than one symbol ahead, because minimization of the low-frequency content in the short term does not always contribute to longer-term minimization. Improvements of 6–10 dB have been reported [14]. The look-ahead strategy is not used in present equipment. The power spectral density (PSD) function of classic EFM has been obtained by computer simulation. Results are plotted in Fig. 5.

IV. EFMPLUS

The CD and its extensions CD-ROM and CD-V, introduced in the early 1980s, have become a very successful medium for the distribution and storage of audio, MPEG-1 video, and other digital information. Its storage capacity, 680 MByte, is insufficient for graphics-intensive computer applications and high-quality digital video programs. An extension of the CD family, the digital versatile disk (DVD), is a new optical recording medium with a storage capacity seven times higher than the conventional CD. Most of the storage capacity increase is due to improved quality of the light source (red instead of infrared light) and the objective lens. The storage capacity of the DVD is also increased by a complete redesign of the logical format of the disk including a more powerful Reed–Solomon product code (RS-PC) and recording code (EFMPlus). The details of the construction of the rate 8/16, (2,10) EFMPlus code, a sliding-block (d, k) code with suppressed lf-content, will be discussed in the next section.

A. Design Outline

Under EFM rules, the data bits are translated eight at a time into 14 channel bits, with runlength parameters $d = 2$ and $k = 10$. In this section, we will detail a code with the same runlength constraints as EFM, called EFMPlus,¹ having a 6% higher rate than classic EFM. EFMPlus has been adopted in the industry standard of the DVD as the channel modulation scheme. The

¹The name EFMPlus is slightly confusing as the acronym EFM stands for eight to fourteen modulation. In EFMPlus, there is no such mapping, but the dk constraints are the same as in classic EFM.

TABLE III
CODE SIZE M , EFFICIENCY $\eta = 1 - R/C(2, 10, \infty)$, AND $\max\{v_i\}$

M	η	$\max\{v_i\}$
351	0.0246	2
353	0.0237	3
354	0.0232	4
389	0.0075	8
391	0.0067	10
397	0.0041	13
398	0.0037	17
406	0.0000	102

most important design issues of the DVD were that critical parameters such as lf-content and timing should definitely not be compromised. Said parameters are critical as they affect the servos and the timing recovery, which are the Achilles' heels of the optical recording system. EFMPlus is a rate 8/16, sliding-block code with the same runlength parameters as EFM. Dc-control is performed with the surplus words that leave each encoder state. The ACH algorithm is run for a code size M that is as large as possible within the complexity constraints. The code size M is the number of source words (not necessarily a power of two) that can be accommodated by the encoder. The additional source words (>256) that are made possible in this fashion are employed as alternatives for dc-control (see the next section for more details). The complexity of a sliding-block encoder and decoder is essentially governed by the maximum value (weight) of an element of the approximate eigenvector. Table III shows the maximum value (weight) as a function of the code size M . In addition, we listed the parameter η , which denotes the relative redundancy, $\eta = 1 - (1/16) \log_2 M/C(2, 10, \infty)$. Note that the maximum code size that can be accommodated for $n = 16$, $d = 2$, and $k = 10$ is 406.

For the given code parameters, we note that for code size $M < 352$, the maximum weight is two. A one-round split is sufficient to construct the encoder. We also notice in Table III that the maximum weight grows very rapidly with increasing code size M . After many trials and considering the diminishing returns, it was decided for a code size $M = 351$. After an initial merging of the $k + 1 = 11$ states, we obtain a three-state FSM. After a single-state split, this three-state FSM can be transformed into a four-state encoder. Each of the four states of the EFMPlus encoder is characterized by the type of words that enter, or leave, the given state. The states and word sets are characterized as follows.

- Words entering State 1 end with m trailing zeros, $m \in \{0, 1\}$.
- Words entering State 2 and 3 end with m trailing zeros, $m \in \{2, \dots, 5\}$ trailing zeros.
- Words entering State 4 end with m trailing zeros $m \in \{6, \dots, 9\}$ trailing zeros.

The words leaving the states are chosen in such a way that the concatenation of words entering a state and those leaving that state obey the ($d = 2, k = 10$) channel constraints. For example, words leaving State 1 start with a runlength of at least two and at most nine zeros. In an analogous manner, we conclude that words leaving State 4 start with at most one zero.

Obviously, the sets of words leaving State 1 or 4 have no words in common. Words emerging from State 2 and 3 comply with the above runlength constraints, but they also comply with other conditions. Words leaving State 2 have been selected such that the first (msb) bit, x_1 , and the thirteenth bit, x_{13} , are both equal to zero. Words leaving State 3 have $x_1 x_{13} \neq 00$. With a computer, it can easily be verified that from each of the states, at least 351 words are leaving. An encoder is constructed by assigning a source word to each of the 351 edges that leave each state. The encoder requires accommodation for only 256 source words. The excess, 95, words have been used for suppressing the low-frequency power (see the next section), the *dc-control*. More details can be found in [15].

The encoder defined above can freely accommodate 351 source words. In order to make it possible to use a unique 26-bit sync word, seven candidate words were barred, leaving a code size of 344. As we only need accommodation for 256 source words, the surplus words can be exploited for minimizing the power at low frequencies. The suppression of low-frequency components, or dc-control, is done by controlling the RDS. The 88 surplus words are used as an alternative channel representation of the source words $0, \dots, 87$. The full encoder is described by two tables called main and substitute table, respectively. The source words $0, \dots, 87$ can be represented by the designated entries of the main table or, alternatively, by the entries of the substitute table. For source words $0, \dots, 87$, the encoder opts for that particular representation from the main table or the substitute table that minimizes the absolute value of the RDS.

The DVD standard requires an extra rule for dc-control. If the encoder is in State 1, the encoder may use the codeword $h(i, 4)$, $0 \leq i \leq 255$, as an alternative for dc-control, provided the runlength constraints are not violated. Similarly, if the encoder is in State 4, it may use the codeword $h(i, 1)$, $0 \leq i \leq 255$, as an alternative. In other words, codewords pertaining to States 1 and 4, i.e., $h(i, 1)$ and $h(i, 4)$, both in the main and substitute tables, may be used as alternatives for dc-control, provided the runlengths constraints are strictly obeyed. State swapping is allowed as decoding can be accomplished unambiguously. The state swapping offers a 2–3-dB extra reduction of the lf power.

V. ALTERNATIVES TO EFM SCHEMES

It is of some interest to consider the possibility of redesigning the EFM code and its variants of various codes rates and to compare the spectral performance.

The EFM code was designed in 1980 before efficient design algorithms, such as ACH and so on, were developed. A second handicap of the EFM design is that at the time of its conception, every gate used for decoding was one too many.² Let us now for academic interest ignore for the time being the complexity issue, and start from scratch. Essentially, EFMPlus is a redesign of EFM with a rate 8/16 instead of 8/17. Decoding of EFMPlus requires 1000 instead of the 52 gates of EFM.

An obvious alternative of the rate 8/16, EFMPlus code would have been EFM with two instead of three merging bits. The dc-

²Note that this requirement was not imposed for the encoding hardware, as it was anticipated that there would be a very limited number of master and replication plants. Who, at that time, could envisage that there would be EFM encoders in the households in CD-R and CD-RW players as computer peripheral.

TABLE IV
PERFORMANCE OF EFM-LIKE CODES

Code	$H(10^{-4})$ (dB)	Sum var.	R
EFM	-33	16	8/17
EFMPlus	-31	19	8/16
EFMPlus*	-29	24	8/16
EFM16a	-20	66	8/16
EFM16b	-28	27	8/16
EFM15	-9	220	8/15

content of the alternative code, *EFM16a* (the name we shall use for the code with two instead of three merging bits), can easily be assessed by computer simulation, and the results are shown in Table IV. The dc-content can be reduced significantly by a re-assignment of the various words that takes into account the following observation. Observe, for example, that in EFM16a, the 16-bit words 0001000100001000 and 1001000100001000 are alternative channel representations. It can easily be verified that the disparity of both words (after precoding, of course) is zero. This, in fact, means that the encoder has no real option to increase or decrease the RDS with the transmission of those codewords. Obviously, it would be much better if we could redesign the code in such a way that as many source words as possible would have channel representations of zero-disparity. Nonzero-disparity codewords of opposite signs should be paired, whereas zero-disparity codewords may remain single.

It is a straightforward exercise, using Gu and Fuja's method [16] to design a block code according to the above design heuristics. We may construct a block-decodable code with a source size of 260 instead of 257 words. A typical result, note there are many possibilities, called EFM16b, offers 8 dB (see Table IV) more reduction at the low-frequency end than does EFM16a. This is a significant improvement, in particular, as the only disadvantage is the extra gate count required for decoding. Note that the EFM16b code requires a full decoding array of 16 bits instead of 14 bits as in EFM16a. An advantage with respect to EFMPlus, which requires a sliding-block decoder of length two, is the absence of error propagation. On the other side of the balance, we have a 3-dB extra reduction of EFMPlus's lf-content (see Table IV).

As $8/15 = 0.5333 \dots$, at least in theory, it is possible to construct a rate $8/15$ (2, 9) code. The EFM15 code [17] is an example of a rate $8/15$, (2, 14) DCRL code. An alternative rate $8/15$ construction [18] is possible that requires, in contrast with classic EFM, only one merging bit. We could, in principle, employ the same 14-bit word assignment as in classic EFM. For an example of such a construction, the reader is referred to the U.S. Patent granted to Tanaka *et al.* [19].

EFM15, which is very similar to EFMPlus, has a codeword length of 15 bits. The code, a typical sliding-block code, has four encoder states, and was constructed using the ACH algorithm after a single split. The number of words that can be accommodated depends on the state, and is at most 270. This leaves at most $270 - 256 = 14$ spare words that can be used for dc-control. Pairing of the alternative representations has been accomplished in such a way that the words that form a pair differ in

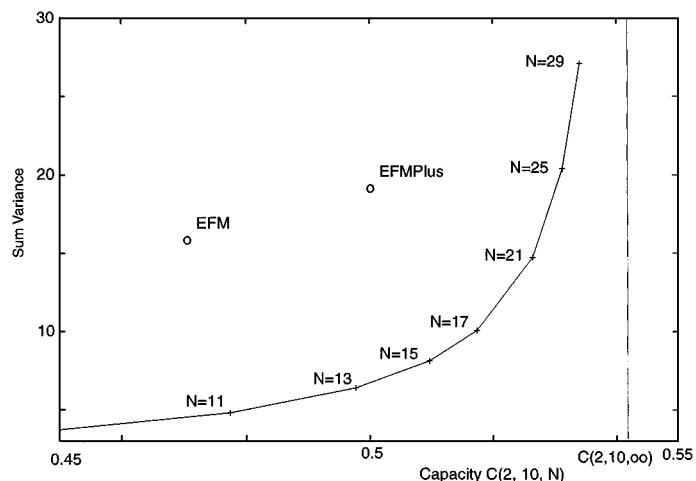


Fig. 6. Sum variance of maxentropic DCRL sequences with parameters $d = 2$ and $k = 10$ and the digital sum variation N as a parameter. As a comparison, we plotted the rate and sum variance of EFM and EFMPlus.

a single position, i.e., have unity Hamming distance. This has the advantage that the decoding operation is simplified and that the alternative representations have an odd or even number of ones, which, as was observed experimentally, has a beneficial effect upon the quality of the dc-control. Further details, such as coding tables and so on, of EFM15 can be found in the U.S. Patent description [17].

VI. PERFORMANCE OF EFM-LIKE CODING SCHEMES

The spectral performance of the various members of the EFM family has been assessed by computer simulation. The outcomes of the simulations have been collected in Table IV.

The lf-suppression, as presented in Table IV, is measured at $10^{-4}f_c$, where the channel bit frequency $f_c = 1$ Hz. If we wish to compare coding schemes of a different rate, it is standard practice to compare the lf-suppression at, say, 0.0001 times the user bit frequency f_b . As the frequency $10^{-4}f_c$ Hz is assumed to be in the range of frequencies, where the spectrum $H(f_c)$ has a parabolic shape, the lf-suppression at $10^{-4}f_b$ can be found by multiplying the lf-suppression measured at $10^{-4}f_c$ by R^3 . For example, if $R = 1/2$, we have to subtract $3 \times 3.1 = 9.3$ dB from the numbers shown in Table IV to obtain the lf-suppression at $10^{-4}f_b$, where $f_b = 1$ Hz. A comparison of the properties of sequences generated under the rules of EFM and EFMPlus with those of maxentropic sequences is shown in Fig. 6. As we can observe, theory predicts there is some room for improvement. For codes of the same rate as EFM and EFMPlus, we could, in theory, construct codes that generate sequences with a factor of three smaller sum variance or, alternatively, a 10-dB extra lf-content suppression.

If, on the other hand, we stipulate that the sum variance and, thus, the lf-content of EFMPlus is adequate, we may conclude from Fig. 6 that a rate 0.53 ($\approx 8/15$) is possible with the sum variance of EFM. The performance of the rate $8/15$, code, EFM15 [17], listed in Table IV, is a far cry from the theoretical bound. Braun *et al.* [20] and Immink [21] presented coding schemes using long block codes with enumerative coding that are very close to the predicted maxentropic performance. The

typical codeword length in their constructions is about 1000 bits, and the hardware required for encoding and decoding is about 5 kB. Other EFM-like codes have been presented by Roth [22].

VII. OTHER EXAMPLES OF DCRLL CODES

The design of any constrained code can, at least in principle, be systematically accomplished by the design techniques that have been developed over the years. Unfortunately, the design of a DCRLL code with a rate close to the Shannon capacity of the constrained channel, is severely hampered by the large number of states of the finite-state machine (FSM), which models the channel constraints at hand. The large number of states of the underlying FSM, can, at least in principle, be handled by buying a sufficiently large computer, but the insight required is too easily lost. The design of DCRLL codes is therefore (still) the province of a plurality of *ad hoc* methods, for example, [23]–[26]. Basically, there are two systematic design approaches that emerged in the literature.

The first method uses the ACH algorithm to design an RLL code. In the final stage of the ACH algorithm, we end with a graph with the property that from any state of the graph, there are at least 2^m (m is assumed to be the source word length) outgoing edges. There are (hopefully) states with a larger number of outgoing edges. These surplus edges are used as alternative codewords that can be used for dc-control. The rate 8/16, (2,10) EFMPlus code, discussed in Section IV, is an example of a DCRLL code used in practice (DVD) that was designed according to these guidelines.

In the second method, dc-control is effectuated by multiplexing the source data or the encoded data with dc-control bits. A given, state-of-the-art RLL code, for example, the rate 2/3, (1,7) code, is used to generate RLL sequences. The sequences generated under the coding rules of said code are multiplexed with channel bits for minimizing the low-frequency components, the dc-control. The user data or, alternatively, the encoded data are partitioned into segments of ν bits. Between two consecutive ν -bit segments β , dc-control bits are inserted, and the β dc-control bits, in turn, are chosen to minimize the low-frequency components. In the experimental phase, we have the freedom to select the parameters ν and β such that the required dc-suppression is reached. There is, in other words, no need to redesign the constituent RLL code.

The success of the design method depends on various factors, such as, for example, how much dc-suppression is required. In most of the practical cases that we encountered, an extra redundancy for the dc-control of 2–3% was sufficient to yield the required dc-suppression. In that case, codes using multiplexing methods offer an excellent performance and flexibility. In the next section, we will present a description of the first design method, where dc-control bits are multiplexed with the user or channel data.

A. Dc-Control on Data Level versus Coded Level

Assume the d and k constraints are given and that an efficient code has been found in the literature or, alternatively, constructed using the various methods offered in the literature.

A straightforward method for extending a standard (d, k) code with dc-suppression is to add (stuff) redundant bits, which can be chosen to reduce the power at low frequencies. Essentially, there are two approaches with which a (d, k) encoder can be extended with multiplexed dc-control: multiplexing can be done at two levels, namely, at the source data level or at the channel data level. Between segments of ν source data or between segments of ν encoded data β , dc-control bits are inserted. In both multiplex formats, the β dc-control bits are chosen to minimize the low-frequency components of the channel sequence generated. This can be accomplished by tallying the RDS at the end of each candidate segment. The encoder transmits that candidate segment whose RDS is closest to zero. At the receiver site, the added dc-control bits, either on the data or channel level, can easily be skipped by the decoder. The two multiplex approaches of dc-control have various distinct features. The β dc-control bits can be freely chosen if they are multiplexed at source data level. Then, the encoder has 2^β possible sequences to be tried. If, on the other hand, the dc-control bits are multiplexed with the (d, k) sequence, the new multiplexed sequence so generated has to obey the (d, k) constraints in force, and as a result, the number of candidate sequences to be tried is less than 2^β . For the dc-control to be effective under all worst-case circumstances, it should guarantee that an (almost) entire segment of ν modulated data bits can be inverted or not inverted. We can easily verify that if the dc-control bits are multiplexed with the (d, k) sequence, that in order to guarantee said worst-case performance, we require at least $(d + 1)$ dc-control bits. Then, the maximum runlength at the segment boundaries will increase from k to $k + 1$. A similar method has been proposed by Odaka [27], Coppersmith and Kitchens [28], and Patel [29]. When, on the other hand, the dc-control bits are multiplexed on the source level, the matter of worst-case performance is much more involved. The encoded segments are both a function of the source data and the encoder state at the start of the segment. It is therefore not recommended to use an industry-standard (d, k) code. A possible solution, using the parity preserving word assignment, will be discussed in the next section.

B. Codes with Parity Preserving Word Assignment

In order to make it possible to efficiently control the dc-content in the source data level mode, we have made the assignment between source words and codewords in such a way that the parity of both the source word and its assigned codeword are the same. The parity P of an n -bit word (x_1, \dots, x_n) , $x_i \in \{0, 1\}$ (either source words or codewords), is defined by

$$P = \sum_{i=1}^n x_i \bmod 2.$$

In other words, if the source word has an even (or odd) number of ones, then its channel representation also has an even (or odd) number of ones. A code with a parity preserving assignment has the virtue that when it is used in conjunction with dc-control bits at the data level, that setting an even (or odd) number of ones at the data level will result in an even (or odd) number of ones at the code level. This leads, as we will demonstrate shortly, to an efficient dc-control.

TABLE V
VARIABLE-LENGTH SYNCHRONOUS RATE 2/3, (1, ∞) CODE WITH PARITY
PRESERVING ASSIGNMENT

<i>Data</i>		<i>Code</i>
00	← →	000
01	← →	010
10	← →	100
1100	← →	001010
1101	← →	001000
1110	← →	101010
1111	← →	101000

TABLE VI
VARIABLE-LENGTH SYNCHRONOUS RATE 1/2, (2, 7) CODE

<i>Data</i>		<i>Code</i>
10	← →	0100
01	← →	1000
001	← →	001000
000	← →	100100
111	← →	000100
1101	← →	00001000
1100	← →	00100100

TABLE VII
BASIC CODING TABLE PARITY PRESERVING (1,8) CODE

<i>Data</i>	<i>Code</i>
00	101
01	100
10	001
11	000

TABLE IX
SUBSTITUTING CODING TABLE II PARITY PRESERVING (1,8) CODE

<i>Data</i>	<i>Code</i>
11.11.11	000.010.010
11.11.10	001.010.010
01.11.10	101.010.010
01.11.11	100.010.010

An example of a variable length rate 2/3, (1, ∞) code that complies with the parity preserving property is shown in Table V. It can easily be verified that indeed the assignment is parity preserving.

A parity preserving assignment of a rate 2/3, (1,8) code, first presented by Kahlman and Immink [30], is based on the look-ahead rate 2/3, (1,7) code described by Jacoby and Kost [31]. Tables VII–IX show the encoding of the new code parity preserving code. The full coding table of the code consists of a main table and two substitute tables instead of a single substitute table. It can easily be verified that the assignment is indeed parity preserving. The code was found by trial and error, as no approach is (yet) available for systematically constructing codes with a parity preserving word assignment.

TABLE VIII
SUBSTITUTING CODING TABLE I PARITY PRESERVING (1,8) CODE

<i>Data</i>	<i>Code</i>
00.00	100.010
00.01	101.010
10.00	000.010
10.01	001.010

The systematic design of RLL codes with a parity preserving word assignment is a challenging task. The above examples show that it is indeed possible and that such codes offer a better performance than do their counterparts. Block codes are by their virtue of simplicity good candidates, but the complexity issue will hamper their design. Variable length synchronous codes seem to be promising candidates. It is not (yet) known how we can design parity preserving codes with the ACH algorithm.

The difference between the quality of the alternative dc-control methods has been assessed by Wang *et al.* [32]. The power density measured at a relatively low-channel frequency $f_c/1000$ was used as a quality criterion. Computer programs have been written for simulating the two coding schemes, where the dc-control bits are multiplexed at source or at channel level, respectively. The code for the channel-level multiplex is the standard, rate 2/3, (1,7) code, whereas the source-level multiplex is the parity preserving, rate 2/3, (1,8) code described in the previous section. The authors observed that the parity preserving code performs 2 dB better than does the standard rate 2/3, (1,7) code used with channel-level multiplex in the range of dc-control redundancy of 1%–4%.

VIII. CONCLUSION

We have given a survey of channel codes for optical disk recording systems. It has been shown that state-of-the-art codes are very close to the bound set by the tenets of information theory.

REFERENCES

- [1] K. A. S. Immink and H. Ogawa, "Method for encoding binary data," U.S. Patent 4.501.000, Feb. 1985.
- [2] K. A. S. Immink, "EFMPlus: The coding format of the multimedia compact disc," *IEEE Trans. Consumer Electron.*, vol. 41, pp. 491–497, Aug. 1995.
- [3] —, *Codes for Mass Data Storage Systems*. The Netherlands: Shannon Foundation, 1999.
- [4] J. P. J. Heemskerck and K. A. S. Immink, "Compact disc: System aspects and modulation," *Philips Technol. Rev.*, vol. 40, no. 6, pp. 157–164, 1982.
- [5] T. Yoshida, "The rewritable minidisc system," *Proc. IEEE*, vol. 82, pp. 1492–1500, Oct. 1994.
- [6] S. Kobayashi, M. Hattori, Y. Shimpuku, G. van den Enden, J. A. Kahlman, M. van Dijk, and R. Woudenberg, "Optical disc system for digital video recording," in *Proc. Joint Int. Symp. Optical Memory and Optical Data Storage*, HI, July 11–15, 1999.
- [7] S. B. Wicker and V. K. Bhargava, Eds., *Reed–Solomon Codes and Their Applications*. Piscataway, NJ: IEEE Press, 1994.
- [8] T. M. Chien, "Upper bound on the efficiency of dc-constrained codes," *Bell Syst. Tech. J.*, vol. 49, pp. 2267–2287, Nov. 1970.
- [9] G. L. Pierobon, "Codes for zero spectral density at zero frequency," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 435–439, Mar. 1984.
- [10] K. Norris and D. S. Bloomberg, "Channel capacity of charge-constrained run-length limited codes," *IEEE Trans. Magn.*, vol. MAG-17, pp. 3452–3455, Nov. 1981.

- [11] K. J. Kerpez, A. Gallopoulos, and C. Heegard, "Maximum entropy charge-constrained run-length codes," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 242–253, Jan. 1992.
- [12] V. Braun, "On modulation, coding and signal processing for optical and magnetic recording systems," doctoral thesis, Institute for Experimental Mathematics, University of Essen, Dusseldorf, Germany, 1997.
- [13] J. B. H. Peek, "Communications aspects of the compact disc digital audio system," *IEEE Commun. Mag.*, vol. 23, pp. 7–15, Feb. 1985.
- [14] K. A. S. Immink and U. Gross, "Optimization of low-frequency properties of eight-to-fourteen modulation," *Radio Electron. Eng.*, vol. 53, pp. 63–66, Feb. 1983.
- [15] K. A. S. Immink, "Method of converting a series of m-bit information words to a modulated signal, method of producing a record carrier, coding device, device, decoding device, recording device, reading device, signal as well as record carrier," U.S. Patent 5,696,505, Dec. 1997.
- [16] J. Gu and T. Fuja, "A new approach to constructing optimal block codes for runlength-limited channels," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 774–785, Mar. 1994.
- [17] K. A. S. Immink, "Method of converting a series of m-bit information words to a modulated signal, method of producing a record carrier, coding device, device, recording device, signal, as well as a record carrier," U.S. Patent 5,790,056, Aug. 1998.
- [18] —, "Constructions of almost block-decodable runlength-limited codes," *IEEE Trans. Inform. Theory*, vol. 41, pp. 284–287, Jan. 1995.
- [19] S. Tanaka, T. Shimada, K. Hirayama, and H. Yamada, "Single merging bit dc-suppressed run length limited coding," U.S. Patent 5,774,078, June 1998.
- [20] V. Braun and K. A. S. Immink, "An enumerative coding technique for dc-free runlength-limited sequences," *IEEE Trans. Commun.*, to be published.
- [21] K. A. S. Immink, "EFM coding: Squeezing the last bits," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 491–495, Aug. 1997.
- [22] R. M. Roth, "On runlength-limited coding with dc-control," *IEEE Trans. Commun.*, vol. 48, pp. 351–358, Mar. 2000.
- [23] T. Uehara, H. Minaguchi, and Y. Oba, "Digital modulation method," U.S. Patent 4,988,999, Jan. 1999.
- [24] S. Tanaka, "Method and apparatus for encoding binary data," U.S. Patent 4,728,929, Mar. 1, 1988.
- [25] J. L. E. Baldwin, "Method and apparatus for processing digital signals prior to recording," EP Patent 193 592, Sept. 1985.
- [26] J. Li and J. Moon, "DC-free run length limited codes for magnetic recording," *IEEE Trans. Magn.*, vol. 33, pp. 868–874, Jan. 1997.
- [27] K. Odaka, "Method and apparatus for encoding binary data," U.S. Patent 4,456,905, June 1984.
- [28] D. Coppersmith and B. P. Kitchens, "Run-length limited code without dc level," U.S. Patent 4,675,650, June 1987.
- [29] A. M. Patel, "Charge-constrained byte-oriented (0,3) code," *IBM Tech. Disclosure Bull.*, vol. 19, pp. 2715–2717, Dec. 1976.
- [30] J. A. H. Kahlman and K. A. S. Immink, "Device for encoding/decoding N -bit source words into corresponding M -bit channel words, and vice versa," U.S. Patent 5,477,222, Dec. 1995.
- [31] G. V. Jacoby and R. Kost, "Binary two-thirds rate code with full word look-ahead," *IEEE Trans. Magn.*, vol. MAG-20, pp. 709–714, Sept. 1984.
- [32] Y. H. Wong, K. A. S. Immink, X. B. Xu, and C. T. Chong, "Comparison of two coding schemes for generating dc-free RLL sequences," in *Proc. Joint Int. Symp. Optical Memory and Optical Data Storage*, HI, July 11–15, 1999.



Kees A. Schouhamer Immink (M'81–SM'86–F'90) was born in Rotterdam, The Netherlands. He received the M.Sc. (EE) and Ph.D. degrees from the Eindhoven University of Technology.

He is President and Founder of Turing Machines Inc. and serves as a Guest Professor at the Institute of Experimental Mathematics, Essen University, Germany, and the National University of Singapore. He has contributed to the design and development of a variety of digital recorders such as the compact disk, compact disk video, DAT, DCC, and recently,

the DVD. Immink was granted 35 U.S. patents, is (co)author of three books and has written numerous papers in the field of digital recorders.

Dr. Immink is Vice-President of the Audio Engineering Society (AES), a Governor of the IEEE Consumer Electronics Society, and Trustee of the Shannon Foundation. He served as Program Chairman and Conference Chairman of various international conferences over the years. He holds fellowships of the AES, IEE, and SMPTE, and is an elected member of the Royal Netherlands Academy of Arts and Sciences. For his part in the digital audio and video revolution, he was honored with the AES Gold and Silver Medal, IEE Thomson Medal, SMPTE Ponyatoff Gold Medal, IEEE Information Theory Society Golden Jubilee Award for Technical Innovation, IEEE Masaru Ibuka consumer electronics award, a Knighthood in the Order of Orange-Nassau, and the IEEE Edison Medal "For a career of creative contributions to the technologies of video, audio, and data recording."