An Enumerative Coding Technique for DC-Free Runlength-Limited Sequences

Volker Braun, *Member, IEEE,* and Kees A. Schouhamer Immink, *Fellow, IEEE*

***Abstract*—We present an enumerative technique for encoding and decoding dc-free runlength-limited sequences. This technique enables the encoding and decoding of sequences approaching the maxentropic performance bounds very closely in terms of code rate and low-frequency suppression capability. Use of finite-precision floating-point notation to express the weight coefficients results in channel encoders and decoders of moderate complexity. For channel constraints of practical interest, the hardware required for implementing such a quasi-maxentropic coding scheme consists mainly of a ROM of at most 5 kB.**

***Index Terms*—Enumeration, modulation coding, recording.**

I. Introduction

DC-FREE runlength-limited (DCRLL) modulation codes have found application in magnetic and optical recording systems, for example, in the compact disk (CD) player [1]. DCRLL codes are used to transform the digital user bit stream into a sequence of binary channel symbols that is suitable for the specific recording requirements. The *runlength* is known as the number of consecutive identical symbols occurring in a sequence. Runlength-limited (RLL) sequences are characterized by two parameters, $l_{\min} = (d+1)$ and $l_{\max} = (k+1)$, denoting the minimum and maximum runlengths occurring in the sequence. DC-free sequences have the properties that their power spectral density (PSD) vanishes at the zero frequency and that there is a region of frequencies close to the zero frequency in which the PSD is low. The restrictions imposed on the DCRLL sequences are usually called the "channel constraints." In optical disk systems, suppression of the low-frequency content of the RLL modulation sequence is employed primarily to circumvent or reduce interaction between the data written on the disc and the servo systems that follow the track [1]. Efficient suppression of the low-frequency components is essential, as error correction is totally useless if track or clock loss occurs [2]. Low-frequency suppression should be achieved at minimum cost in *code rate*, i.e., the ratio between user bit rate and channel bit rate. The maximum code rate, called the *capacity*, is a function of the channel constraints in force. The quotient of rate and capacity is usually called the *rate efficiency*. Implemented DCRLL codes, for example, the EFM code [1] applied in the CD system or the codes designed for magnetic disk or tape drive applications described in [3] and [4], often achieve rate efficiencies in the order of only 90%. The EFMPlus code [2] applied in the digital versatile disk (DVD) system has a rate efficiency of about 92.5%. In other words, many of the conventional DCRLL codes currently implemented can still be improved significantly in terms of code rate.

Enumerative coding techniques [5] make it possible to translate source words into codewords and vice versa by invoking an algorithmic procedure rather than performing the translation with a look-up table [6]. Code rates very close to capacity can be achieved by using enumerative coding and long codewords [6]. Severe error propagation resulting from the use of long codewords can be avoided by reversing the conventional hierarchy of outer error correcting code and inner modulation code [6]. Enumerative decoding is done by forming the weighted sum of the symbols of the codeword received [7]. The integer-valued weights used in forming this sum are a function of the channel constraints in force. Encoding is done with the aid of a method that is similar to decimal-to-binary conversion in which the weights are used instead of the usual powers of 2. The hardware which implements an enumerative coding scheme mainly consists of a ROM to store the weight coefficients, a binary adder, and subtracter. In order to obtain a feasible ROM size, the weight coefficients can, without relevant losses in code rate, be expressed in finite-precision floating-point notation [6].

The outline of the next sections is as follows. In Section II, we will introduce an efficient runlength graph representation of the DCRLL constraints. To enable the enumeration of the DCRLL sequences, knowledge of the number of distinct valid sequences is required. In Section III, we will derive these numbers from the underlying runlength graph. An enumerative technique for encoding and decoding DCRLL sequences will be introduced in Section IV. In Section V, we will discuss the effects of using finite-precision floating-point arithmetic on achievable code rates and on the size of the required weight set. In addition, the low-frequency suppression capabilities of selected enumerative DCRLL codes will be evaluated in computer simulations.

II. Preliminaries

We will assume that binary user information with a bit rate of $f_b = 1/T_b$ is translated into a coded channel sequence having the channel bit rate $1/T_c$, where $R = T_c/T_b$ denotes the rate of the code. Let $\{c_i\} = \{c_1, c_2, \ldots, c_i, \ldots\}$, $c_i \in \{-1, 1\}$,

Paper approved by E. Ayanoglu, the Editor for Communication Theory and Coding Application of the IEEE Communications Society. Manuscript received May 1, 1998; revised October 18, 1999. This paper was presented in part at the 1997 Symposium on Information Theory and Its Applications (SITA'97), Matsuyama, Japan, December 2–5, 1997, and in part at the 1998 IEEE International Symposium on Information Theory, Cambridge, MA, August 16–21, 1998.

V. Braun was with Hitachi Central Research Laboratory, Tokyo 185-8601, Japan. He is now with Alcatel Corporate Research Center, 70499 Stuttgart, Germany.

K. A. S. Immink is with the Digital Communications Group, Institute for Experimental Mathematics, University of Essen, 45326 Essen, Germany.

Publisher Item Identifier S 0090-6778(00)10914-6.

denote the coded channel sequence, representing the positive or negative magnetization of the recording medium, or pits or lands in the case of optical recording. The *running digital sum* (RDS) at time i of the sequence $\{c_i\}$ is defined as

$$z_i = \sum_{j=1}^{i} c_j = z_{i-1} + c_i, \qquad z_0 = 0. \tag{1}$$

A sequence is dc-free if, and only if, the RDS assumes a finite number of values [8]. This number is called the *digital sum variation* (DSV), denoted by N. We will start by confining ourselves to odd values of DSV, so $|z_i| \leq (N-1)/2$. DCRLL constraints will be characterized by the three integer parameters (d, k, N). As a finite RDS implies a constraint on the maximum runlength, these parameters satisfy $0 \leq d < k \leq N-2$.

RLL sequences are often encoded in two consecutive steps. In the first step, the binary user information bits are translated into a sequence $\{x_i\}$, $x_i \in \{0, 1\}$, having at least d and at most k "zeros" (i.e., $x_i = 0$) between consecutive "ones" (i.e., $x_i = 1$). The sequence $\{x_i\}$ is often called a (d, k) sequence or a runlength-limited sequence in nonreturn-to-zero-inverse (NRZI) format. Prior to the recording, the (d, k) sequence $\{x_i\}$ is converted into the bipolar RLL channel sequence $\{c_i\}$ such that the logical "ones" in the (d, k) sequence indicate the positions of a $1 \rightarrow -1$ or $-1 \rightarrow 1$ transition of the corresponding RLL channel sequence. This conversion step is commonly called *precoding*. The $(d, k) = (1, 3)$ sequence

$$\{x_i\} = \{1, 0, 1, 0, 0, 0, 1, 0, \ldots\} \tag{2}$$

would be converted, e.g., into the RLL channel sequence

$$\{c_i\} = \{-1, -1, 1, 1, 1, 1, -1, -1, \ldots\}. \tag{3}$$

Another representation of the (d, k) sequence can be given as a sequence of runlengths $\{l_j\} = \{l_1, l_2, \ldots, l_j, \ldots\}$, where $d+1 \leq l_j \leq k+1$. We define a "run" in the (d, k) sequence $\{x_i\}$ as a logical "one" followed by a sequence of "zeros." The runlength sequence corresponding to the (d, k) sequence in (2) would be $\{l_j\} = \{2, 4, 2, \ldots\}$. Kerpez *et al.* [9] introduced another sequence $\{U_j\}$, defined by $U_j = l_j - U_{j-1}$, where $U_0 = 0$. The sequence $\{U_j\}$ corresponding to the (d, k) sequence (2) would be $\{U_j\} = \{2, 2, 0, \ldots\}$. Let Z_j denote the RDS of the channel sequence $\{c_i\}$ after the jth run in the corresponding sequence $\{x_i\}$. The sequence $\{Z_j\}$ corresponding to the RLL channel sequence (3) would be $\{Z_j\} = \{-2, 2, 0, \ldots\}$. Between U_j and Z_j, we find the relation $U_j = (-1)^j Z_j$, and hence $|U_j| \leq (N-1)/2$. As $U_j + U_{j-1} = l_j \geq d+1$, we find for all j

$$d + 1 - \frac{N-1}{2} \leq U_j \leq \frac{N-1}{2} \tag{4}$$

i.e., $\{U_j\}$ assumes a finite number M of values

$$M = N - 1 - d. \tag{5}$$

A compact description of the (d, k, N) constraints by means of a "runlength graph" has been presented by Kerpez *et al.* [9]. The M states of this graph are associated with U_j values and the edges with runlengths. As an example, Fig. 1 depicts the runlength graph representing the $(1, 3, 7)$ constraint. Although not formally defined, we will denote the runlength graph underlying the DCRLL constraints by G. The states of G will be denoted by their U_j values, where in the sequel $U_j = d + j - (N-1)/2$, $j = 1, \ldots, M$. Note that the states of G are related with RDS values. The sequences (2) and (3), for example, may emanate from state 0 and terminate in state 0 of the graph in Fig. 1.

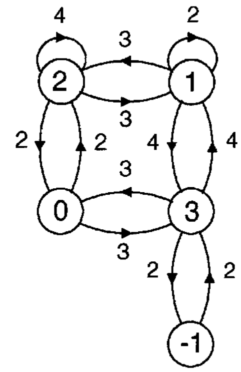

Fig. 1. Runlength graph representing the $(1, 3, 7)$ constraint. The states are denoted by their U_j values, and the edges are labeled by their lengths [9].

Associated with the runlength graph G is an $M \times M$ *adjacency matrix* denoted by $A(D)$ (D-transform notation). The adjacency matrix associated with the runlength graph G turns out to have a Hankel structure, i.e., it is constant on the antidiagonals [9]. As an example, we present the adjacency matrix associated with the runlength graph in Fig. 1, given by

$$A(D) = \begin{pmatrix} 0 & 0 & 0 & 0 & D^2 \\ 0 & 0 & 0 & D^2 & D^3 \\ 0 & 0 & D^2 & D^3 & D^4 \\ 0 & D^2 & D^3 & D^4 & 0 \\ D^2 & D^3 & D^4 & 0 & 0 \end{pmatrix}.$$

Note that $A(D)$ is symmetric, and that the largest exponent, denoted by $l_{\max}(j)$, of the nonzero entries of $A(D)$, occurring in the jth row (or column) which corresponds to the state U_j of G, is given by

$$l_{\max}(j) = \min\{k+1, d+j\}.$$

III. Number of DCRLL Sequences

The number of distinct (d, k, N) constrained sequences can be described using recursive relations obtained from the runlength graph G. Let $N(U_\Omega, n, U_\alpha)$ denote the number of distinct sequences of length n emanating from state U_α and terminating in state U_Ω of the graph G. The number of sequences $N(U_\Omega, n, U_\alpha)$ can be determined using the following recursive relations. Let

$$\begin{aligned} N(U_j, i, U_\alpha) &= 0, \qquad i < 0;\ \forall U_j \in \Sigma \\ N(U_\alpha, 0, U_\alpha) &= 1 \\ N(U_j, 0, U_\alpha) &= 0 \qquad \forall U_j \in \Sigma \backslash \{U_\alpha\} \\ N(U_j, i, U_\alpha) &= 0, \qquad 0 < i \leq d;\ \forall U_j \in \Sigma. \end{aligned}$$

For $d+1 \leq i \leq n$ and for all $U_j \in \Sigma$ let

$$N(U_j, i, U_\alpha) = \sum_{l=d+1}^{l_{\max}(j)} N(l - U_j, i - l, U_\alpha) \tag{6}$$

where Σ denotes the set of states of G. Note that $N(U_\Omega, n, U_\alpha) = N(U_\alpha, n, U_\Omega)$ as G is symmetric (we call G symmetric if the adjacency matrix $A(D)$ is symmetric), and

$N(U_\Omega, n, U_\alpha) = 0$ if $(U_\Omega + U_\alpha + n)$ is odd. This property states that if U_α and U_Ω are both even or both odd, then all sequences emanating from state U_α and terminating in state U_Ω of G have an even length n. If U_α is even and U_Ω odd, or vice versa, then all sequences emanating from state U_α and terminating in state U_Ω of G have an odd length n. Knowledge of the numbers $N(U_j, i, U_\alpha)$ enables the application of enumerative coding techniques.

IV. Enumerative Encoding and Decoding

A. Decoding

A general enumerative technique for encoding and decoding binary constrained sequences has been presented by Cover [5]. Let $\{0,1\}^n$ denote the set of binary sequences of length n and let S be any (constrained) subset of $\{0,1\}^n$. We establish a 1-1 mapping from set S onto the set of integers $0, 1, \ldots, |S|-1$, where $|S|$ is the cardinality of S (i.e., the number of distinct sequences in S). Set S can be ordered lexicographically as follows: if $x = (x_1, \ldots, x_n) \in S$ and $y = (y_1, \ldots, y_n) \in S$, then y is called less than x, in short, $y < x$, if there exists an i, $1 \le i \le n$, such that $y_i < x_i$ and $x_j = y_j$, $1 \le j < i$. For example, $00\,101 < 01\,010$. The position of x in the lexicographical ordering of S is defined to be the *rank* of x, denoted by $i_S(x)$, i.e., $i_S(x)$ is the number of all y in S with $y < x$. Let $n_S(x_1, x_2, \ldots, x_u)$ be the number of elements in S for which the first u coordinates are $(x_1, x_2, \ldots, x_u)$. The rank of $x \in S$ can be obtained by using

$$i_S(x) = \sum_{j=1}^{n} x_j n_S(x_1, x_2, \ldots, x_{j-1}, 0). \tag{7}$$

An alternative of Cover's enumeration scheme can be given by counting the number of elements in S that have a lexicographic index *higher* than x, the *inverse rank* of x [7]. The inverse rank of $x \in S$ can be obtained by using

$$i_S^c(x) = \sum_{j=1}^{n} \bar{x}_j n_S(x_1, x_2, \ldots, x_{j-1}, 1) \tag{8}$$

where $\bar{x}_j = 1 - x_j$, the complement of x_j. The algorithms (7) and (8) implement the decoding operation, i.e., given the constrained sequence x, find the corresponding lexicographic index in set S. The inverse rank has the virtue that the same set of weight coefficients can be used for encoding and decoding [7]. We will now consider the inverse rank for enumerative decoding of DCRLL sequences.

Let $x = (x_1, x_2, \ldots, x_n)$ denote a (d,k,N) sequence of length n in NRZI notation emanating from state U_α and terminating in state U_Ω of the graph G. Note that $x_1 = 1$. Let $a_j(x)$ denote the number of trailing "zeros" of the subvector $(x_1, x_2, \ldots, x_{j-1})$, i.e.,

$$a_j(x) = \begin{cases} 0, & j = 1, 2 \\ \min\{(j-i-1) : 1 \le i < j, x_i = 1\}, & j > 2. \end{cases}$$

Let $l_j(x) = a_j(x) + 1$ be the length of the trailing run of the subvector $(x_1, x_2, \ldots, x_{j-1})$. By observing x, we can uniquely determine the sequence of states of graph G corresponding to x. To this end, we define the sequence of states $U_j(x)$ through graph G associated with the sequence x as

$$U_j(x) = \begin{cases} U_\alpha, & \text{if } j = 1, \\ U_{j-1}(x), & \text{if } x_j = 0 \quad \text{and} \quad j > 1 \\ l_j(x) - U_{j-1}(x), & \text{if } x_j = 1 \quad \text{and} \quad j > 1. \end{cases}$$

Proposition: The inverse rank of a (d,k,N) constrained sequence x of length n emanating from state U_α and terminating in state U_Ω of the runlength graph G is

$$i_S^c(x) = \sum_{j=1}^{n} \delta_j(x) N(l_j(x) - U_{j-1}(x), n-j+1, U_\Omega) \tag{9}$$

where

$$\delta_j(x) = \begin{cases} 1, & \text{if } x_j = 0 \quad \text{and} \quad a_j(x) \ge d \quad \text{and} \\ & \quad (U_\Omega + l_j(x) - U_{j-1}(x) + n - j) \text{ odd} \\ 0, & \text{otherwise.} \end{cases}$$

Proof: Let $x_j^1 = (x_1, x_2, \ldots, x_{j-1}, 1)$, $1 \le j \le n$. We can observe that for a given j, $1 \le j \le n$, $n_S(x_j^1)$ can take one of two different values depending on x. If $a_j(x) < d$ then $n_S(x_j^1) = 0$, because in this case the number of "zeros" between the last two "ones" of x_j^1 is less than d, violating the d constraint. Therefore, no (d,k,N) constrained sequence x can begin with x_j^1, so $n_S(x_j^1) = 0$. If $a_j(x) \ge d$, then x_j^1 does not violate the d constraint. In this case, the sequence x_j^1 leads graph G to the state $l_j(x) - U_{j-1}(x)$, so that $n_S(x_j^1)$ equals the number of sequences of length $n-j+1$ (including the symbol $x_j = 1$) emanating from the state $l_j(x) - U_{j-1}(x)$ of G and terminating in U_Ω, i.e., $n_S(x_j^1) = N(U_\Omega, n-j+1, l_j(x) - U_{j-1}(x))$. Using the symmetry of G, we obtain $n_S(x_j^1) = N(l_j(x) - U_{j-1}(x), n-j+1, U_\Omega)$. If $(U_\Omega + l_j(x) - U_{j-1}(x) + n - j)$ is even, then $N(l_j(x) - U_{j-1}(x), n-j+1, U_\Omega) = 0$. In this case, we do not perform an addition in (9). □

B. Encoding

The encoding operation, i.e., given the inverse lexicographic index I, $0 \le I < |S|$, find the corresponding x, is described by the following algorithm:

$\hat{I} := I, U := U_\alpha, x_1 := 1, a := 0;$
for $j = 2$ **to** n **do**
  if $(U_\Omega + a + 1 - U + n - j)$ odd **and** $a \ge d$
    then $\hat{N} := N(a + 1 - U, n - j + 1, U_\Omega);$
    else $\hat{N} := 0;$
  end if
  if $\hat{I} \ge \hat{N}$
    then $x_j := 0, \hat{I} := \hat{I} - \hat{N};$
    else $x_j := 1, U := a + 1 - U, a =: -1;$
  end if
  $a := a + 1;$
end for

C. Multiple State Coding

The ranking procedure can be generalized for a set of (d,k,N) sequences of length n that emanate from a common state, denoted by U_α, and terminate in a state which is a member of a predefined set of states. Consider L terminating states given by $\Sigma_L = \{U_{i_1}, \ldots, U_{i_L}\} \subset \Sigma$. Ranking of this set of (d, k, N) sequences can be accomplished by

$$i_S^c(x) = \sum_{j=1}^{n} \delta_j(x) \sum_{m=1}^{L} N\left(l_j(x) - U_{j-1}(x), n-j+1, U_{i_m}\right). \tag{10}$$

Equation (10) holds because the sets of sequences emanating from a common state and terminating in U_{i_m}, $m = 1, \ldots, L$, are disjoint.

The enumerative technique described above enables the design of channel encoders and decoders of moderate complexity. Storage capacity is required for approximately $Nn/2$ nonzero weight coefficients. The storage capacity required for implementing the presented enumerative coding scheme for combined dc-free runlength-limited sequences is about the same as that required for implementing the enumerative coding scheme for pure dc-free sequences [1].

D. Even Values of DSV

So far we have confined ourselves to DCRLL constraints having odd values of DSV. For N even, a runlength graph can be derived from a Hankel-type adjacency matrix having $N-1-d$ states, and enumerative encoding and decoding algorithms are similar as described above. However, there may exist paths of even length and paths of odd length emanating from a state U_α and terminating in a state U_Ω of the corresponding runlength graph. Thus, storage capacity is required for approximately Nn nonzero weight coefficients.

V. Implementation Aspects

In the following, we will consider coding schemes which translate m user information bits into codewords of an even length n. Using the technique described in [10], the code construction can be extended to odd codeword lengths. The ratio $R = m/n \leq C$ is called the code rate, and C denotes the capacity. We will distinguish two separate coding schemes. In the first coding scheme, denoted by C_1, we assume that all codewords emanate from a state U_α and terminate in the same state U_α of graph G. We will call the state U_α the principal state of the coding scheme C_1. In the second coding scheme, denoted by C_2, we assume that either of the codewords emanates from a state which is a member of the set of states $\Sigma_L = \{U_{i_1}, \ldots, U_{i_L}\} \subset \Sigma$ and terminates in a state which is a member of the same set Σ_L. We will call Σ_L the set of principal states of the coding scheme C_2. In both cases, transmission errors that may occur in the DCRLL channel sequence must be corrected before the decoding in order to prevent severe error propagation. In the case of coding scheme C_2, transmission errors can cause catastrophic error propagation, which is why the encoder must regularly be forced into a predefined principal state.

A. Principal States and Feasible Code Rates

We will start by determining the principal state U_α of coding scheme C_1 and the set of principal states Σ_L of coding scheme C_2 that result in the maximum number of codewords. We will also investigate at what codeword lengths the code rate approaches the capacity of the DCRLL constraints to within a small value, say 1%, and we will determine feasible code rates for (d, k, N) constraints of practical interest.

Let $N_i(n)$ denote the number of distinct sequences of length n emanating from the state U_i of G. In a coding scheme that enables m bit encoding, $N_i(n) \geq 2^m$ must be satisfied for all principal states U_i of this scheme. For large n, $N_i(n)$ can be approximated by

$$N_i(n) \simeq y_i \lambda^n, \qquad n \gg 1$$

where λ is the largest real root of the characteristic polynomial $\det(A(\lambda^{-1}) - I)$, I denotes the $M \times M$ identity matrix, and $y = (y_1, \ldots, y_M)$ is a positive eigenvector of the matrix $A(\lambda^{-1})$ associated with eigenvalue 1, i.e., $yA(\lambda^{-1}) = y$ [1]. The number of distinct DCRLL sequences of length n emanating from a state U_i and terminating in a state U_j of G can hence be approximated by

$$N(U_j, n, U_i) \simeq N_i(n/2) N_j(n/2) = y_i y_j \lambda^n, \qquad n \gg 1 \tag{11}$$

(if $U_i + U_j + n$ even, else 0). The accuracy of this approximation has been computed for (d, k, N) constraints of practical interest and codeword lengths $200 \leq n \leq 1000$, where the magnitude of vector y has been evaluated numerically from the actual number of sequences of a certain length n ($n = 200$). In all the computed examples, approximation (11) was accurate to within 0.15%. Assuming equality in (11), it follows that the principal state U_α of coding scheme C_1 that results in maximum cardinality is associated with the maximum entry $\max_i y_i$ of vector y. We define $A_{C_1} = (\max_i y_i)^2$ and $N_{C_1}(n) = A_{C_1} \lambda^n \simeq N(U_\alpha, n, U_\alpha)$. In coding scheme C_2, we intend to determine the set of principal states $\Sigma_L \subset \Sigma$ so that the minimum number of valid codewords emanating from a state $U_i \in \Sigma_L$ is maximum among all possible sets Σ_L. Assuming equality in (11), it follows that this set Σ_L is associated with:

$$A_{C_2} = \max_{\Sigma_L \subset \Sigma} \min_{\Sigma_L} y_i \sum_{\Sigma_L} y_j.$$

The optimum set Σ_L can be determined by starting with $\Sigma^{(1)} = \{U_i \in \Sigma : U_i \text{ odd}\}$, and by iteratively removing the state associated with $\min_i y_i$ from this set. The procedure is then repeated starting with $\Sigma^{(2)} = \{U_i \in \Sigma : U_i \text{ even}\}$. Finally, the optimum set Σ_L is selected from the best subsets of $\Sigma^{(1)}$ and $\Sigma^{(2)}$. The minimum number of valid codewords of length $n \gg 1$ emanating from a state that is a member of the set of principal states Σ_L can thus be approximated by $N_{C_2}(n) \simeq A_{C_2} \lambda^n$. The rate of an implemented code is $R = \lfloor \log_2 N_{C_i}(n) \rfloor / n$, and the capacity is given by $C = \log_2 \lambda$. The difference between code rate and capacity is

$$\frac{1}{n}\left(-1 + \log_2 A_{C_i}\right) \leq R - C \leq \frac{1}{n} \log_2 A_{C_i}.$$

Examples of optimum principal state configurations are shown in Table I for coding scheme C_1 and in Table II for coding scheme C_2. Also presented are the corresponding values $\log_2 A_{C_i}$ and the codeword lengths $n_{C_i} = 100(1 - \log_2 A_{C_i})$. The codeword length n_{C_i} ensures that $C - R \leq 0.01$. Table I also shows the capacities of the corresponding (d, k, N) constraints. We see that for many DCRLL constraints of practical

TABLE I
C_1 CODING SCHEME: CAPACITY C, OPTIMUM PRINCIPAL STATE U_α, $\log_2 A_{C_1}$, CODEWORD LENGTH n_{C_1}, FEASIBLE CODE RATES m/n FOR $m = 256$, RATE EFFICIENCIES η, NUMBER OF BITS $p_{\min}$, AND APPROXIMATE STORAGE CAPACITY C_S IN kB FOR SEVERAL (d, k, N) CONSTRAINTS OF PRACTICAL INTEREST

					$m = 256$			
(d,k,N)	C	U_α	$-\log_2 A_{C_1}$	n_{C_1}	n	η	$p_{\min}$	C_S
$(0,5,7)$	.8858	1	1.77	277	292	.9897	8	0.5
$(0,7,9)$	.9276	1	2.18	318	280	.9856	7	1.0
$(0,11,13)$	.9634	2	3.48	448	270	.9842	7	0.9
$(1,7,9)$	.6039	2	2.84	384	430	.9858	8	1.2
$(1,7,15)$	.6540	2	3.60	460	398	.9835	8	0.9
$(1,7,19)$	.6639	2	3.96	496	392	.9837	9	2.8
$(2,10,15)$	.5097	3	4.08	508	512	.9810	7	3.4
$(2,10,29)$	.5340	2	5.07	607	490	.9784	8	5.0
$(2,15,17)$	.5199	3	4.34	534	502	.9809	8	3.0
$(2,15,29)$	.5400	3	5.16	616	484	.9795	9	5.0

TABLE II
C_2 CODING SCHEME: OPTIMUM PRINCIPAL STATE SET Σ_L, $\log_2 A_{C_2}$, CODEWORD LENGTH n_{C_2}, FEASIBLE CODE RATES m/n FOR $m = 256$, RATE EFFICIENCIES η, NUMBER OF BITS $p_{\min}$, AND APPROXIMATE STORAGE CAPACITY C_S IN kB FOR SEVERAL (d, k, N) CONSTRAINTS OF PRACTICAL INTEREST. $\{U_i{:}U_j\}$ DENOTES THE PRINCIPAL STATE SET $\{U_i, U_i + 2, \ldots, U_j\}$

				$m = 256$			
(d,k,N)	Σ_L	$-\log_2 A_{C_2}$	n_{C_2}	n	η	$p_{\min}$	C_S
$(0,5,7)$	{0:2}	1.00	200	292	.9897	7	0.5
$(0,7,9)$	{-1:3}	1.09	209	278	.9927	8	1.0
$(0,11,13)$	{-2:4}	1.28	228	268	.9915	9	0.8
$(1,7,9)$	{0:4}	1.88	288	428	.9904	8	0.7
$(1,7,15)$	{-1:5}	2.17	317	396	.9885	8	1.6
$(1,7,19)$	{-2:6}	2.24	324	390	.9887	8	2.5
$(2,10,15)$	{-1:5}	2.70	370	508	.9887	9	4.2
$(2,10,29)$	{-4:8}	2.84	384	486	.9864	8	4.5
$(2,15,17)$	{-1:7}	2.76	376	498	.9888	10	4.4
$(2,15,29)$	{-3:9}	2.89	389	480	.9877	9	4.6

interest a codeword length in the order of 500 bits is sufficient for achieving rate losses versus capacity of less than 1%.

Also shown in Tables I and II are feasible code rates and rate efficiencies, defined by $\eta = R/C$, where we have assumed the optimum principal state configurations and the encoding of 256 bits. We see that 256 bit encoding using coding scheme C_1 results in rate efficiencies in the order of 98%. The rate efficiency can be further increased by using larger codeword lengths or, for fixed m, by applying coding scheme C_2. We add that the same code rates as in Tables I and II can often be achieved by using principal states other than the optimum configurations, for example, by using $U_\alpha = 0$ in coding scheme C_1.

B. Weight Truncation

In order to enable the implementation of the presented DCRLL coding scheme, we will in the following express the weights in truncated radix-2 representation. Let N denote one of the integer weights $N(U_j, n, U_i)$. An integer $N < 2^\nu$ can be uniquely represented by a binary ν-tuple $x = (x_{\nu-1}, \ldots, x_0)$, where $N = \sum_{i=0}^{\nu-1} x_i 2^i$. Let $u = \lfloor \log_2 N \rfloor$ be the position of the leading "one" element or "most significant bit" of x. We define the p-bit truncated representation of N, denoted by $\lfloor N \rfloor_p$, by

$$\lfloor N \rfloor_p = \left\lfloor 2^{-(u+1-p)} N \right\rfloor 2^{u+1-p}. \qquad (12)$$

In words, $\lfloor N \rfloor_p$ is obtained from N by leaving the p most significant bits unchanged and setting all the other trailing bits to 0. The p-bit truncated weight $\lfloor N \rfloor_p$ can be expressed in two-part radix-2 representation $\lfloor N \rfloor_p = (m, e) = m2^e$, where the integers m and e are called the mantissa and exponent of $\lfloor N \rfloor_p$, respectively. Apparently, p bits are required to express the mantissa m. The number of bits required to express the exponent e is in the order of $\log_2 \nu$. However, the exponents e_1 and e_2 of two weight coefficients $N_1 < N_2$ of similar magnitude $N_1 \approx N_2$ can usually be represented much more efficiently, for example, by using the difference $e_2 - e_1$. The capacity of the memory unit for the storage of the exponents of the weight coefficients is hence negligible compared with the storage capacity required for the mantissa. Therefore, the exponents of the weight coefficients will not be considered any further here.

When use is made of truncated weights, denoted by $\hat{N}$, recursion (6) results in

$$\hat{N}(U_j, i, U_\alpha) = \left\lfloor \sum_{l=d+1}^{l_{\max}(j)} \hat{N}(l - U_j, i - l, U_\alpha) \right\rfloor_p .$$

In the enumerative encoding and decoding algorithms, the truncated weights $\hat{N}(U_j, i, U_\Omega)$ are used instead of the full-precision weights $N(U_j, i, U_\Omega)$. The effect on the set of codewords will be that the $N(U_j, i, U_\Omega) - \hat{N}(U_j, i, U_\Omega)$ highest ranking words of length i will be recursively discarded from the set of all the lexicographically ordered DCRLL sequences.

Speaking in terms of runlengths, the effect of using truncated weights will be that short runs will occur more frequently than in the case of untruncated weights. As an illustrative example, Fig. 2 displays the accumulated runlength distributions of two $(2, 10, 15)$ constrained enumerative codes having a rate of $256/512$ and a rate efficiency of 98.1%, using either $p = 8$ or $p = 12$ bits to express the mantissa of the weights. Also shown in Fig. 2 is the accumulated runlength distribution of an ideal, "maxentropic" $(2, 10, 15)$ constrained sequence, i.e., in this case the code rate equals the capacity.

We will briefly consider the effect of using a truncated weight representation on the achievable code rates. We define $p_{\min}$ as the minimum number of bits required to express the mantissa of the truncated weights, so that there is no rate loss as in the full-precision scheme. Values of $p_{\min}$ are collected in Tables I and II for optimum principal state configurations. We see that for (d, k, N) constraints and codeword lengths of practical interest, values of p in the order of 8–10 are usually sufficient to avoid rate losses as in the full-precision scheme.

As shown in [6], the mantissa of the truncated weights $\hat{N}(U_i, n, U_i)$, $U_i \in \Sigma$, will for increasing n become (and remain) periodic. That is, for $n > n_0$, there are integers h and f such that

$$\hat{N}(U_i, n, U_i)2^h = \hat{N}(U_i, n + f, U_i), \quad U_i \in \Sigma, n > n_0.$$

We will call f the period of the mantissa of $\hat{N}(U_i, n, U_i)$ and n_0 the preamble. The period f of the weight coefficients is a

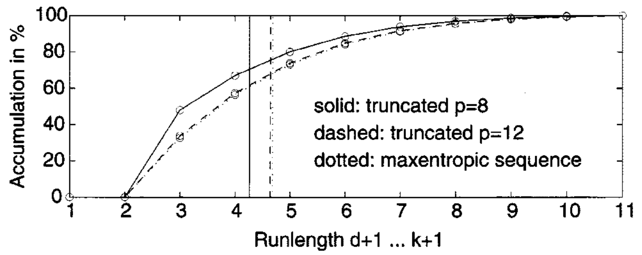

Fig. 2. Accumulated runlength distributions of two $(2, 10, 15)$ constrained enumerative codes having a rate of $256/512$ (obtained in computer simulations; solid, dashed) and the accumulated runlength distribution of the corresponding maxentropic sequence (obtained with the method developed by Kerpez *et al.* [9], denoted by the dotted line). The step functions indicate the corresponding average runlengths.

function of the (d, k, N) constraints and the number of bits p used to express the mantissa of the weights. The preamble n_0 also depends on the set of principal states used for coding. By exploiting the periodicity of the weight coefficients, a significant saving in storage hardware can often be realized [6]. The ROM size required for the storage of the mantissa of the weight coefficients is approximately given by

$$C_S = \begin{cases} \frac{1}{2}(M-1)pn_S, & \text{if } N \text{ odd} \\ (M-1)pn_S, & \text{if } N \text{ even} \end{cases} \tag{13}$$

where $n_S = \min\{n, n_0 + f\}$. The factor $(M-1)$ in (13) is an immediate consequence of the structure of the runlength graph G. We can see in Fig. 1 that $N(-1, i, U_\alpha) = N(3, i-2, U_\alpha)$, i.e., there is no need to store $N(-1, i, U_\alpha)$, $1 \le i < n$. In Tables I and II, useful upper bounds of the ROM size C_S are presented in kilobyte (kB) units, where we assumed $n \ge n_0 + f$ and $7 \le p \le 12$. We can conclude that for many (d, k, N) constraints of practical interest the presented enumerative coding technique can be implemented by using a ROM of at most 5 kB for storing the mantissa of the weight coefficients. In many cases, the required ROM size is significantly lower than 5 kB.

C. Low-Frequency Suppression

Suppression of the low-frequency components is an essential performance criterion of DCRLL codes [1]. The PSD functions at the low-frequency end are depicted in Fig. 3 for several $(2, 10, 15)$ constrained coded sequences. Both axes of the power spectra in Fig. 3 have been normalized for a fixed-user bit rate, i.e., we consider $H^*(fT_b) = RH(2\pi fT_c/R)$. The dotted curve in Fig. 3 represents the PSD of a maxentropic $(2, 10, 15)$ constrained sequence. Maxentropic DCRLL sequences provide an upper bound in low-frequency suppression capability, given a certain (d, k) constraint and code rate [11]. The solid and dashed curves in Fig. 3 represent the PSD functions of $(2, 10, 15)$ constrained sequences generated with the aid of the presented enumerative coding technique. These codes are based on construction C_1 using the principal state $U_\alpha = 0$ and have a rate of $256/512$ and a rate efficiency of 98.1%. The upper curve in Fig. 3 shows the PSD function of the coded sequence when the weights are expressed in radix-2 representation using $p = 8$ bits to express the mantissa. Apparently the use of $p = 8$ results in a loss of about 5 dB in low-frequency suppression relative to the maxentropic bound. When values of p in the order

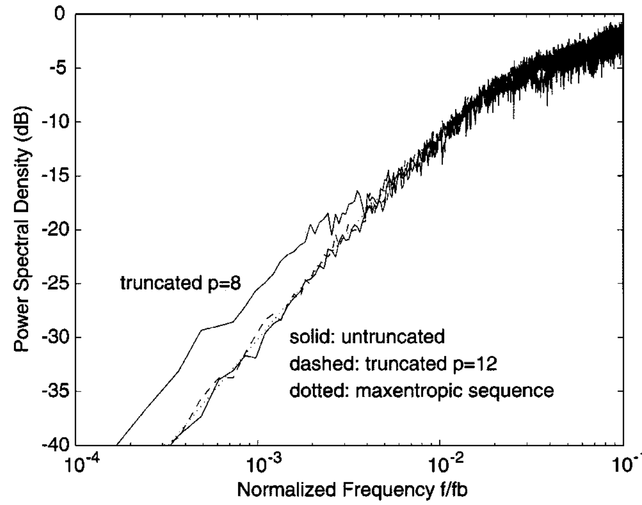

Fig. 3. Power spectra of several $(2, 10, 15)$ constrained enumerative codes having a rate of $256/512$ (obtained in computer simulations; solid, dashed) and power spectrum of the corresponding maxentropic sequence (evaluated using the method developed by Kerpez *et al.* [9], denoted by the dotted line).

of 10–12 or untruncated weights are used, the PSD function of the coded sequence approximates the maxentropic performance bound very closely, as can be seen in the lower solid and dashed curves in Fig. 3. Similar behavior of the PSD function at the low-frequency end has been observed for (d, k, N) constraints other than $(2, 10, 15)$ and for sets of principal states other than $U_\alpha = 0$. We conclude that values of $p = 10 \cdots 12$ bits to express the mantissa of the truncated weights are suitable for obtaining a low-frequency suppression performance very close to the maxentropic bound.

We suppose that the loss in low-frequency suppression versus the maxentropic bound represented by the upper curve in Fig. 3 is a result of the runlength distribution, which, as shown in Fig. 2, differs remarkably from the maxentropic runlength distribution. Given a certain (d, k) constraint and value of p, the loss in low-frequency suppression versus the maxentropic bound tends to become more pronounced with a decreasing DSV or with an increasing codeword length, as we observed in computer simulations. For $p = 12$, we never observed relevant losses in low-frequency suppression versus the maxentropic bound. The variance of the RDS, in short *sum variance*, is often used to characterize the low-frequency characteristic of dc-free sequences [1]. We would like to add that the $(2, 10, 15)$ constrained sequence associated with the upper curve in Fig. 3 exhibits a sum variance that is only about 2.5% larger than for a maxentropic $(2, 10, 15)$ constrained sequence, whereas from the low-frequency characteristic and the theory of maxentropic DCRLL sequences [11], we would roughly expect additional 60%.

D. Performances of Implemented Codes

In the following, we will briefly assess the performances of several selected DCRLL codes with respect to their low-frequency suppression capabilities. In order to obtain a fair comparison of different DCRLL codes, both axes of the power spectra of these codes have been normalized for a fixed-user bit rate. As a performance criterion, we determine the PSD

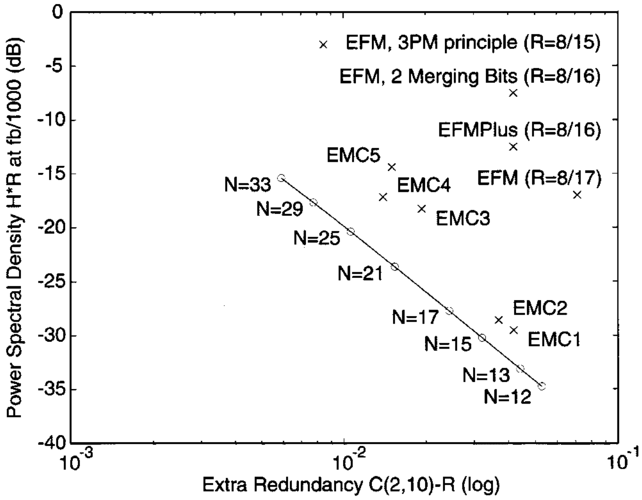

Fig. 4. Comparison of the performance of several $(2, 10, 15)$ constrained modulation codes.

of the coded sequence at a small fraction of the user bit rate, for example, at $f_b/1000$. We will compare the low-frequency suppression capability of the coded sequence with the corresponding maxentropic performance bound.

Fig. 4 displays the PSD $H^*(10^{-3})$ versus a parameter called the *extra redundancy* for several dc-free $(2, 10)$ constrained coded sequences. The extra redundancy has been defined in [11] as the difference between the capacity of the (d, k) constraint, denoted by $C(d, k)$, and the rate of an implemented dc-free code satisfying this runlength constraint. The solid curve in Fig. 4 represents the maxentropic performance bound and the crosses indicate the low-frequency suppression performances of implemented codes. Four conventional codes are considered: the EFM code [1] applied in the CD player; EFMPlus [2] applied in the DVD system; and two other EFM alternatives described in [2]. These four codes all have finite values of DSV and satisfy the $(2, 10)$ runlength constraint. The power spectra of these codes were evaluated in computer simulations [2]. A strategy for improving the EFMPlus low-frequency suppression performance by about 3 dB is presented in [2].

Fig. 4 also shows the low-frequency suppression performances for five DCRLL codes based on the presented enumerative coding technique. All of these five enumerative codes, denoted by EMC1-EMC5, have $k = 10$ and they are based on construction C_1. The design parameters of these codes are listed in Table III. EMC1 and EMC2 have code rates similar to those of EFMPlus. In low-frequency suppression performance, they outperform EFMPlus by about 15 dB. EMC3 and EMC4 have a low-frequency suppression performance similar to that of EFM. They achieve an 11%–12% increase in recording density relative to EFM, but require a rather large ROM for storing the mantissa of the weights. As an alternative, EMC5 achieves a gain of 5.35% in code rate relative to EFMPlus and it has a comparable low-frequency suppression capability.

By relaxing the k constraint from $k = 10$ to $k = 15$, we can achieve a 6.67% gain in code rate relative to EFMPlus by using EMC6. EMC6 is based on construction C_1 and its design parameters are given in Table III. EMC6 achieves $H^*(10^{-3}) \simeq$

TABLE III
DESIGN PARAMETERS OF SEVERAL ENUMERATIVE $(2, k, N)$ CONSTRAINED MODULATION CODES

code	k	N	p	R	η	U_α	C_S (kB)
EMC1	10	15	12	256/512	.9810	3	3.5
EMC2	10	15	12	512/1014	.9906	3	3.5
EMC3	10	29	12	256/490	.9784	2	9
EMC4	10	29	12	512/970	.9885	2	18
EMC5	10	29	7	512/972	.9864	2	5
EMC6	15	29	9	512/960	.9877	3	5
EMC7	15	29	7	512/962	.9856	3	2

-16 dB. The size of the required ROM can be decreased from about 5 kB to about 2 kB by using $p = 7$ instead of $p = 9$. The resulting code, denoted by EMC7, achieves $H^*(10^{-3}) \simeq -14.5$ dB.

VI. CONCLUSIONS

We have presented an enumerative technique for encoding and decoding DCRLL sequences. The size of the full-precision weight set required for implementing the presented coding scheme is about the same as that required for implementing the enumerative coding scheme for pure dc-free sequences [1]. As the greater part of the electronics that implements the enumerative coding technique is taken up by the storage of the weight coefficients, this technique enables the design of channel encoders and decoders of moderate complexity. To enable the implementation of the proposed coding technique, we expressed the weight coefficients in finite-precision floating-point notation. We have shown that the presented enumerative coding technique can be used to encode and decode DCRLL sequences approaching the maxentropic performance bounds very closely in terms of code rate and low-frequency suppression capability. For channel constraints of practical interest, the hardware required for implementing such a quasi-maxentropic coding scheme consists mainly of a ROM of at most 5 kB. In many cases of practical relevance, the size of the required ROM is significantly lower.

ACKNOWLEDGMENT

The authors would like to thank A. J. H. Vinck, H. Morita, and M. Umemoto for useful discussions and comments. The authors would also like to thank the two anonymous reviewers for providing valuable comments on an earlier version of the manuscript.

REFERENCES

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*, Eindhoven, The Netherlands: Shannon Foundation, 1999.
[2] ——, "EFMPlus: The coding format of the multimedia compact disc," *IEEE Trans. Consumer Electron.*, vol. 41, pp. 491–497, Aug. 1995.
[3] T. Uehara and Y. Oba, "A new 8–14 channel coding for D-3 format VTR," *IEEE Trans. Broadcast.*, vol. 39, pp. 265–271, June 1993.
[4] J. Li and J. Moon, "DC-free run-length-limited codes for magnetic recording," *IEEE Trans. Magn.*, vol. 33, pp. 868–874, Jan. 1997.
[5] T. M. Cover, "Enumerative source coding," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 73–77, Jan. 1973.
[6] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1389–1399, Sept. 1997.

[7] L. Pátrovics and K. A. S. Immink, “Encoding of $dklr$-sequences using one weight set,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 1553–1554, Sept. 1996.
[8] G. L. Pierobon, “Codes for zero spectral density at zero frequency,” *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 435–439, Mar. 1984.
[9] K. J. Kerpez, A. Gallopoulos, and C. Heegard, “Maximum entropy charge-constrained run-length codes,” *IEEE J. Select. Areas Commun.*, vol. 10, pp. 242–252, Jan. 1992.
[10] K. A. S. Immink, “DC-free codes of rate $(n-1)/n$, n odd,” in *Proc. 1998 IEEE Int. Symp. Information Theory (ISIT’98)*, Cambridge, MA, Aug. 16–21, 1998, p. 111.
[11] V. Braun and A. J. E. M. Janssen, “On the low-frequency suppression performance of dc-free runlength-limited modulation codes,” *IEEE Trans. Consumer Electron.*, vol. 42, pp. 939–945, Nov. 1996.

Volker Braun (S’93–M’97) was born in Saarland, Germany, in 1967. He received the Dipl.-Ing. degree from the University of Kaiserslautern, Kaiserslautern, Germany, in 1992, and the Dr.-Ing. degree from the University of Essen, Essen, Germany, in 1997.

From 1992 to 1997, he was a member of the Digital Communications Group at the Institute for Experimental Mathematics, University of Essen. During that period, he was a regular Visiting Scientist at Philips Research Laboratories, Eindhoven, The Netherlands, and he was contributing to the ACTS Project SMASH of the European Community. From March 1997 to March 1998, he was with the Hitachi Central Research Laboratories, Tokyo, Japan, as a Visiting Scientist. His research activities from 1992 to 1998 focused on modulation, coding, and signal processing related to magnetic and optical recording systems. In 1998, he joined Alcatel SEL AG, Stuttgart, Germany, where he started working in the Switching Division Hardware Architecture Design Group on optimizing implementation aspects of Alcatel’s MPSR broad-band switch. Since June 1999, he has been with the Radio Communications Department at Alcatel Corporate Research Center, Stuttgart, Germany, where he is involved in research and development activities regarding channel coding and multiple-antenna techniques for applications in second- and third-generation mobile radio communications systems.

Kees A. Schouhamer Immink (SM’86–F’90) received the M.S. and Ph.D. degrees from the Eindhoven University of Technology, Eindhoven, The Netherlands.

He is Director of Turing Machines, Inc. Since 1995, he has been an Adjunct Professor at the Institute for Experimental Mathematics, Essen University, Essen, 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 laser video disc, compact disc, compact disc video, DAT, DV, DCC, and DVD. He holds 42 issued and pending U.S. 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. He is Vice President of the Audio Engineering Society (AES) and a Governor of the IEEE Consumer Electronics Society. He is a member of the IEEE Fellows Committee. For his contributions to the digital audio and video revolution, he received wide recognition, among others 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.