The decoding function can be accomplished with a simple logic array. Note that the code above can easily be transformed into a $(d = 1, k = 11)$ RLL code by representing the source word "2" by "000000" or "111111." If $x_5x_6 = 00$, then represent the source word "2" by "000000." It should be appreciated that the smallest block-decodable conventional rate $2/3$ $(d = 1, k = 11)$ code requires a codeword length of $n = 18$. This clearly shows that a design of an RLL code which is not a (d, k) code plus precoder can be quite advantageous.

IV. Conclusions

We have presented a new rate $4/6$ $(d = 1, k = 11)$ runlength-limited code. The code is block-decodable, and it is particularly attractive as many commercially available Reed–Solomon codes operate in GF (2^8). The encoder can be implemented with a simple 6-bit ROM, and decoding can be accomplished with a logic array.

References

[1] K. A. S. Immink, *Coding Techniques for Digital Recorders*. Englewood Cliffs, NJ: Prentice-Hall Int. (UK) Ltd., 1991.

[2] T. D. Howell, "Statistical properties of selected recording codes," *IBM J. Res. Develop.*, vol. 33, no. 1, pp. 60–73, Jan. 1989.

[3] K. A. S. Immink, "Block-decodable runlength-limited codes via look-ahead technique," *Philips J. Res.*, vol. 46, no. 6, pp. 293–310, 1991.

[4] H. Hollmann, "A block-decodable $(d, k) = (1, 8)$ runlength-limited rate 8/12 code," *IEEE Trans. Inform. Theory*, vol. 40, no. 4, pp. 1292–1295, July 1994.

Encoding of $dklr$-Sequences Using One Weight Set

Levente Pátrovics and Kees A. Schouhamer Immink, *Fellow, IEEE*

***Abstract*—Traditional schemes for encoding and decoding runlength-constrained sequences using the enumeration principle require two sets of weighting coefficients. A new enumeration is presented requiring only one set of coefficients.**

***Index Terms*—Enumerative coding, runlength-limited sequences, dk-constrained codes.**

I. Introduction

Runlength-limited codes have been applied in magnetic and optical recording as well as optical data transmission. A dk-limited (binary) sequence, in short, dk-sequence, satisfies simultaneously the following two conditions: 1) d constraint—two logical "ones" are separated by a run of consecutive "zeros" of length at least d, and 2) k constraint—any run of consecutive "zeros" is of length at most k. If only Condition 1 is satisfied, the sequence is said to be d-limited (with $k = \infty$), and will be termed a d-sequence. Similarly, if only Condition 2 is satisfied, the sequence is said to be k-limited.

In 1965, Kautz [1] described a method of encoding and decoding k-limited codewords by an algebraic approach termed *enumeration*. Enumerative decoding is done by forming the weighted sum of the symbols of the codeword received. The integer-valued weights used in forming this sum are a function of the the channel constraints in force. Encoding is done by a method which is similar to decimal-to-binary conversion where, instead of the usual powers of two, the weights are used. The coding and decoding using enumeration has the virtue that its complexity grows polynomially with the codeword length contrasting the complexity of direct look-up which grows exponentially. Five years later, in 1970, Tang and Bahl [2] presented a classical method for enumerating dk-sequences. Their method requires only one set of weight coefficients for the encoding and decoding of d-limited sequences while the encoding operation of the general case of dk-sequences requires two sets of coefficients. Tang and Bahl also showed how dk-sequences can be cascaded without violating, at the boundaries, the prescribed constraints. A more efficient method for cascading dk-sequences was described by Beenker and Immink [5] using $dklr$-sequences. A $dklr$-limited sequence is a dk-sequence that starts with at most l "zeros" and ends with at most r "zeros." Also for enumerating $dklr$-sequences the prior art requires two sets of coefficients.

We will present a new method for enumerating $dklr$-sequences requiring only one set of coefficients. The major part of the electronics that implements the enumeration technique is taken by the storage of the coefficients. As a result, the new method presented will almost halve the electronics needed.

Manuscript received November 22, 1995.

The authors are with Philips Research Laboratories, 5656 AA Eindhoven, The Netherlands.

Publisher Item Identifier S 0018-9448(96)05853-1.

II. Coding and Decoding Using Enumeration

Tang and Bahl [2] developed a general algebraic technique for encoding and decoding dk-sequences. They defined a 1–1 mapping from the set S of all dk-sequences of length n onto the set of integers $\{0, 1, \cdots, |S| - 1\}$ and they presented an algorithm for converting dk-sequences to integers and *vice versa*. Tang and Bahl's method is briefly described below. Let $\{0, 1\}^n$ denote the set of binary sequences of length n and let S be any subset of $\{0, 1\}^n$. The set S can be ordered lexicographically as follows: if $\boldsymbol{x} = (x_1, \cdots, x_n) \in S$ and $\boldsymbol{y} = (y_1, \cdots, y_n) \in S$, then $\boldsymbol{y}$ is called less than $\boldsymbol{x}$, in short, $\boldsymbol{y} < \boldsymbol{x}$, if there exists an $i, 1 \leq i \leq n$, such that $y_i < x_i$ and $x_j = y_j, 1 \leq j < i$. For example, "00101" < 1 "01010." The position of $\boldsymbol{x}$ in the lexicographical ordering of S is defined to be the *rank* of $\boldsymbol{x}$ denoted by $i_S(\boldsymbol{x})$, i.e., $i_S(\boldsymbol{x})$ is the number of all $\boldsymbol{y}$ in S with $\boldsymbol{y} < \boldsymbol{x}$. Tang and Bahl showed that in the specific case of dk-sequences the rank can be computed by

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j N(n - j) \tag{1}$$

where $N(n)$ is the number of dk-sequences of length n, $n > 0$. By definition we have $N(0) = 1$. The above algorithm is usually called the *decoding* operation. In an implementation of this algorithm, the weight coefficients $N(n)$ can be precalculated and stored in memory or they can be calculated "on the fly." In the more specific case of $dklr$-sequences, a similar relationship can be written down. Translation of an integer $I, 0 \leq I \leq N(n) - 1$, into the corresponding $\boldsymbol{x}$, *the encoding operation*, is done by the following algorithm:

$\hat{I} := a + I;$
for $j = 1$ **to** n **do**
  if $\hat{I} \geq T(n - j)$ **then**
    $x_j := 1,\ \hat{I} := \hat{I} - N(n - j)$ **else** $x_j := 0$.The

integer coefficients a and $T(j)$, $1 \leq j \leq n$, are a function of the runlengh constraints in force, and are not relevant in this context. The interested reader is referred to the literature [3]. In the general dk-limited case, two sets of coefficients are needed for accomplishing the encoding operation. For d-limited sequences we have $a = 0$ and $T(n) = N(n)$, and as a result only one set of coefficients is required for encoding. In the k-limited case we need two coefficient sets. However, as remarked by Tang and Bahl, if, the "complement" form of $\boldsymbol{x}$ is dealt with [1], only one set is required. This motivated us to attempt finding a method for encoding $dklr$-sequences with only one set. The outcome is given below.

III. Description of the New Enumeration Scheme

Another scheme to enumerate vectors of the set $S \in \{0,1\}^n$ was given by Cover [4]. Let $n_s(x_1, x_2, \cdots, x_u)$ be the number of elements in S for which the first k coordinates are $(x_1, x_2, \cdots, x_u)$. The rank of $\boldsymbol{x} \in S$ can be obtained by

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j n_s(x_1, x_2, \cdots, x_{j-1}, 0). \tag{2}$$

Although the above sum looks similar to (1), applying Cover's method for enumerating dk-sequences yields different weights that are dependent of the the sequence coded so far. It can be shown that, as opposed to Tang and Bahl's algorithm, Cover's enumeration needs approximately kn weights for encoding and decoding $dklr$-sequences.

An alternative of Cover's enumeration scheme can be given by counting the number of elements of S that have a *higher* lexicographic index than $\boldsymbol{x}$, the *inverse rank* of $\boldsymbol{x}$.

Proposition 1: The inverse rank of $\boldsymbol{x} \in S$ is given by

$$i_S^c(\boldsymbol{x}) = \sum_{j=1}^{n} \overline{x}_j n_s(x_1, x_2, \cdots, x_{j-1}, 1) \tag{3}$$

where $\overline{x}_j = 1 - x_j$, the complement of x_j.

Proof: Words with prefix $(x_1, x_2, \cdots, x_{j-1}, 0)$ lexicographically precede words with prefix $(x_1, x_2, \cdots, x_{j-1}, 1)$. For each j such that $x_j = 0$, $n_s(x_1, x_2, \cdots, x_{j-1}, 1)$ gives the number of elements of S that first differ from $\boldsymbol{x}$ in the jth term and therefore have a higher lexicographic index. By adding these numbers for $j = 1, 2, \cdots, n$, we eventually count all the elements in S of higher index than $\boldsymbol{x}$.

Given S and the lexicographic index I, the decoding algorithm of Proposition 1 runs as follows:

1) If $I \geq n_s(1)$ set $x_1 = 0$ and set $I = I - n_s(1)$; otherwise set $x_1 = 1$.
2) For $j = 2, \cdots, n$, if $I \geq n_s(x_1, x_2, \cdots, x_{j-1}, 1)$ set $x_j = 0$ and set $I = I - n_s(x_1, x_2, \cdots, x_{j-1}, 1)$; otherwise set $x_j = 1$.

IV. Encoding and Decoding of $dklr$-Sequences

In the following we first give an algorithm to enumerate dkr-constrained sequences, for which the length of the leading zero-run is not constrained. From the results obtained, enumeration of $dklr$-sequences follows easily.

To use Proposition 1, we introduce some notations. Given a dkr-codeword $\boldsymbol{x}$, let $\boldsymbol{x}_j^1 = (x_1, x_2, \cdots, x_{j-1}, 1)$. Denote by $N^0(i)$ the number of dkr-constrained sequences of length i whose first element equals 1. We define the quantity $a_j(\boldsymbol{x})$ as the length of the trailing zero-run of the subvector $(x_1, \cdots, x_{j-1})$ if it is not the all-zero sequence. Hence

$$a_j(\boldsymbol{x}) = \begin{cases} \min\{(j-i-1) : 1 \leq i < j, x_i = 1\}, & \text{if } j > 1 \\ \qquad \text{and } (x_1, \cdots, x_{j-1}) \neq (0, \cdots, 0) & \\ d, & \text{otherwise.} \end{cases}$$

Applying Proposition 1 for dkr-sequences, we can observe that for a given j, $1 \leq j \leq n$, $n_s(\boldsymbol{x}_j^1)$ can take one of two different values, depending on $\boldsymbol{x}$. If $a_j(\boldsymbol{x}) < d$ then $n_s(\boldsymbol{x}_j^1) = 0$, because in this case $\boldsymbol{x}_j^1$ contains at least one "one" in the first $(j-1)$ positions, and the length of the zero-run between the last two 1's of $\boldsymbol{x}_j^1$ is less than d, violating the d-constraint. Therefore, no dkr-codeword can begin with $\boldsymbol{x}_j^1$, so $n_s(\boldsymbol{x}_j^1) = 0$. If $a_j(\boldsymbol{x}) \geq d$, then $\boldsymbol{x}_j^1$ does not violate the d-constraint, so that $n_s(\boldsymbol{x}_j^1)$ equals the number of dkr-sequences beginning with $\boldsymbol{x}_j^1$. Considering that $\boldsymbol{x}_j^1$ ends with a 1, the value of $n_s(\boldsymbol{x}_j^1)$ is independent of $\boldsymbol{x}_j^1$ and is given by $N^0(n-j+1)$. By noting the above we get the next proposition for the enumeration of dkr-sequences.

Proposition 2: The inverse rank of a dkr-sequence $\boldsymbol{x}$ of length n is

$$i_S^c(\boldsymbol{x}) = \sum_{j=1}^{n} \delta_j(\boldsymbol{x}) N^0(n-j+1)$$

where

$$\delta_j(\boldsymbol{x}) = \begin{cases} 1, & \text{if } x_j = 0 \text{ and } a_j(\boldsymbol{x}) \geq d \\ 0, & \text{otherwise.} \end{cases}$$

To put in a simple way, the inverse rank of $\boldsymbol{x}$ can be obtained by summing the weight of each "zero" of $\boldsymbol{x}$ that is either contained in the leading zero-run or is preceded by at least d zeroes.

We can observe that dkr-sequences violating the l-constraint have lower lexicographic indexes than $dklr$-sequences, therefore their inverse rank is higher. If we denote by $\boldsymbol{x}_l$ the $dklr$ codeword having the highest inverse rank in the set of dkr-sequences S, then the set $\{0, \cdots, i_S^c(\boldsymbol{x}_l)\}$ of inverse ranks corresponds to the set of all $dklr$-codewords of length n. This property is very favorable, because using the new algorithm no subtraction of a constant is required during encoding.

Proposition 1 provides the algorithm for decoding $dklr$-sequences. The encoding operation, i.e., given the inverse lexicographic index I find the corresponding $\boldsymbol{x}$, is described by the following program:

```
Î := I, a := d;
for j = 1 to n do
    if Î ≥ N^0(n − j + 1) and a ≥ d
      then x_j := 0, Î := Î − N^0(n − j + 1)
      else if a < d
              then x_j := 0
              else x_j := 1, a := −1;
    a := a + 1;
end for
```

V. Conclusions

We have presented a new method of enumerating runlength-limited sequences that requires only one set of weight coefficients.

References

[1] W. H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 284–292, 1965.

[2] D. T. Tang and L. R. Bahl, "Block codes for a class of constrained noiseless channels," *Inform. Contr.*, vol. 17, pp. 436–461, 1970.

[3] J. Gu and T. Fuja, "A new approach to constructing optimal block codes for runlength-limited channels," *IEEE Trans. Inform. Theory*, vol. 40, no. 3, pp. 774–785, May 1994.

[4] T. M. Cover, "Enumerative source encoding," *IEEE Trans. Inform. Theory*, vol. IT-19, no. 1, pp. 73–77, Jan. 1973.

[5] G. F. M. Beenker and K. A. S. Immink, "A generalized method for encoding and decoding run-length limited sequences," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 5, pp. 751–754, Sept. 1983.