Simple classes of constrained systems with unconstrained positions that outperform the maxentropic bound

Kees A. Schouhamer Immink
Turing Machines Inc.
Willemskade 15b-d, 3016 DK Rotterdam The Netherlands
Email: immink@turing-machines.com

Kui Cai
Data Storage Institute
5 Engineering Drive 1, Republic of Singapore 117608
Email: CaiKui@dsi.a-star.edu.sg

***Abstract*— The Wijngaarden-Immink (WI) scheme is a combined modulation/ECC coding scheme, where arbitrary user data are translated into a constrained sequence in which predefined positions are reserved for ECC parity. Besides offering the benefit of combined modulation/ECC coding, the WI scheme has two extra benefits. They are a) error propagation is limited to the constrained symbols, since symbols on the unconstrained positions are not related, and b) code hardware is limited to a look-up table of the coded part. We will describe classes of simple bit-stuffing schemes that require less redundancy than predicted by the bound based on the performance of maxentropic constrained systems presented by Campello *et al.* [1] and Poo *et al.* [2].**

Key words: magnetic recording, channel capacity, constrained code, runlength-limited, RLL sequence, (d, k) sequence, ECC.

I. Introduction

We will discuss high-rate recording codes for hard disk drives that impose a constraint on the maximum run of transitions in the encoded sequence, i.e. MTR-constrained codes. MTR-constrained codes do not have more than k consecutive 1s, and are therefore related to conventional k-constrained codes that have a maximum run of k consecutive 0's. Properties and constructions of such codes have been collected in [3]. Constrained codes are always used in conjunction with error correcting codes (ECC), and traditionally ECC is followed by the recording code. Combining channel constraints and ECC is a challenging task, and practical combined schemes are therefore hard to find. A recent proposal by Van Wijngaarden and Immink [5] offers promise for the systematic design of constrained ECC codes.

The Wijngaarden-Immink (WI) scheme is a combined modulation/ECC coding scheme, where arbitrary user data are translated into a constrained sequence in which predefined positions are reserved for ECC parity. The parities formed on the basis of the constrained sequence could be generated by any ECC ranging from Reed Solomon codes to LDPC codes. The constrained code is so designed that the bit values in the reserved positions can be arbitrarily chosen, so that the ECC parity can be inserted without violating the channel constraint. The positions in a constrained codeword reserved for parities are called *unconstrained positions*. Following terminology of error correcting codes, the unconstrained part of the constrained codeword is often called the *systematic part* of the constrained codeword. Besides offering the benefit of a combined modulation/ECC coding, the WI scheme has two extra benefits. They are a) error propagation is limited to the constrained symbols, since symbols on the unconstrained positions are not related, and b) code hardware is limited to a look-up table of the coded 'unsystematic' part. Obviously, these benefits are very attractive in practice, and it is therefore of significant interest to further investigate codes based on the WI-scheme.

We start with an overview of results presented by Campello *et al.* [1] and Poo *et al.* [2], who obtained a relationship between code redundancy and *insertion rate* of interleaved schemes, where the *insertion rate* is the quotient of the number of unconstrained positions and the sum of unconstrained and constrained positions. We will present groups of simple bit-stuffing codes that need less redundancy than the maxentropic constrained codes presented in the prior art.

II. Maxentropic Bound on Rate versus Insertion Rate

In this context, it is assumed that a series of m-bit user words are translated into n-bit codewords. The codewords are cascaded to form a string of modulated bits that satisfies the maximum runlength k-constraint, i.e. a sequence having more than k consecutive 0s is not allowed. The quotient of m/n is called the *rate* of the code. The constrained and unconstrained positions in each codeword are given by the binary vector $\boldsymbol{\psi^n}$. Let $\boldsymbol{\psi^n} = (\psi_1, \ldots, \psi_n)$, be a binary vector of length n, where ψ_i = 0 if position i is unconstrained, and ψ_i = 1 if position i is constrained. The key problem in the WI scheme is finding 'good' vectors $\boldsymbol{\psi}$ yielding a high rate and a low weight w of $\boldsymbol{\psi}$ viz. a maximum number of codewords and a maximum number of unconstrained positions. The *insertion rate*, I, is defined as the quotient of the number of unconstrained bits $n - w(\boldsymbol{\psi^n})$ and the codeword length n, or

$$I = 1 - \frac{w(\boldsymbol{\psi^n})}{n}, \tag{1}$$

where the weight $w(\boldsymbol{\psi^n}) = \sum_{i=1}^{n} \psi_i$. It is far from easy to find a vector $\boldsymbol{\psi^n}$ that maximizes code rate for a give w, and simple, non-exhaustive search algorithms have so far not been discovered. Wijngaarden and Immink [5] presented results of exhaustive computer searches for $n < 25$ for codes with rate < 24/25.

Interleaving of constrained symbols generated by a k-constrained code with unconstrained symbols is a straightforward method of implementing a WI scheme, see Campello *et al.* [1] and Poo *et al.* [2]. Interleaved codes of rate including 32/33, 48/49, 56/57, 72/73, 80/81, and others have been published in the magnetic recording literature.

Let X be a k'-constrained sequence, which is chopped into subsequences of length $r = k' + 1$, then, by definition, each subsequence will have at least one '1'. We can interleave the constrained subsequences with unconstrained subsequences of length u, and the resulting interleaved sequence is a $(k' + u)$-constrained sequence, since each encoded subsequence has at least a '1'. Assume a rate m/n, (0,k') base code whose output is interleaved with unconstrained symbols. The code rate, $R_0(k' + u)$, of the interleaved scheme is

$$R_0(k' + u) = \frac{u + (k' + 1)R}{u + k' + 1},$$

and the insertion rate, I_0, of the interleaved scheme is

$$I_0(k' + u) = \frac{u}{u + k' + 1}.$$

If we replace $k' + u$ by k, we obtain a series of points in the rate-insertion rate graph (R, I) with u and k as parameters, namely

$$R(k) = \frac{u + (k - u + 1)C(k - u)}{k + 1} \tag{2}$$

and

$$I(k) = \frac{u}{k + 1}, \quad 0 \le u \le k, \tag{3}$$

where we assumed that the base code is achieving capacity $R = C(k)$. Note that the insertion rate is upper bounded by $k/(k + 1)$, since it is not possible to have more than k consecutive unconstrained positions in a k-constrained code. Eqs (2) and (3) define a series of points in the (R, I) plane. The series of possible points can be extended to a curve in the (R, I) plane. If the desired insertion rate is not a multiple of $1/(k+1)$, we can construct a code via a weighted average of a k-bit-stuffing and a $(k + 1)$-bit-stuffing scheme [1] [2]. Then the points defined in the rate-insertion rate graph above become a piecewise-linear curve connecting the $k + 1$ points.

Figure 1 gives a graphical impression of the relationship between redundancy, $1 - R(k)$, and insertion rate, I, for various values of k, where the points defined by (2) and (3) are connected by lines. The rate versus insertion rate, i.e. the $(R(k)$, $I(k))$ curve, will be called *maxentropic bound* since they are based on the performance of maxentropic k-constrained codes. Campello *et al.* [1] showed that for $k = 1$ and $k = 2$ the weighted average rates of the bit-stuffing schemes are optimal, i.e. achieve the lowest redundancy for a given insertion rate.

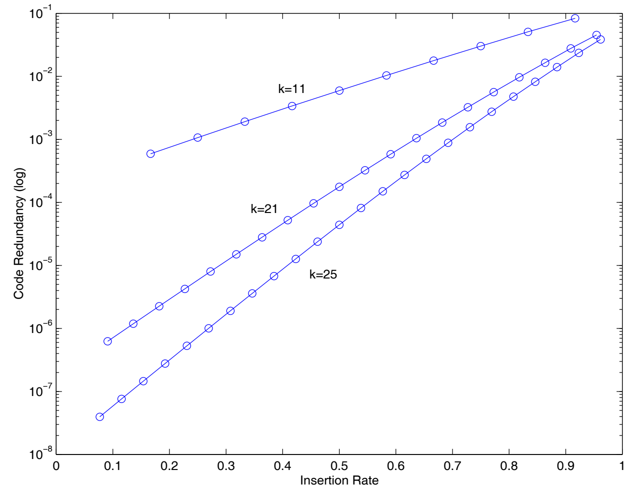

Fig. 1. Maxentropic bound of code redundancy, $1 - R(k)$, (log axis) versus insertion rate, $I(k)$, for three values of k with a maxentropic k-constrained base code.

However, it is false to conclude that for values of $k >$ 2 the piecewise-linear curves shown in Figure 1 bounds the rate, so that, in theory, a higher insertion rate for given code rate is feasible. Campello *et al.* [1] left as an open problem whether or not there is a class of *simple* bit-stuffing schemes that completely describe the optimal coding schemes (for all k). In the next section, we will present a a few examples of simple bit stuffing codes that require a lower redundancy for a given insertion rate than predicted by the maxentropic bound.

III. Improving the Bound

In the previous sections, k'-constrained subsequences of length $r = k' + 1$ are interleaved with unconstrained subsequences of length u, resulting in a $(k' + u)$-constrained sequence. In other words, the concatenation of the coded subsequences form a regular k'-constrained sequence. In this section, alternatively, we will take a look at codes that translate a user word into r-bit nibbles, $r > 1$, where each nibble contains at least one '1' and where the sequence of cascaded r-nibbles is *not* by necessity a $k' = r-1$ constrained sequence. The nibbles are interleaved with unconstrained subsequences of length u.

A. First example

In the simplest case, we have r-bit nibbles, where the all-zero sequence is not allowed, so that the number of allowed r-bit nibbles equals $2^r - 1$. The r-bit nibbles are multiplexed with u-bit uncoded sequences. If $u = k - 2(r - 1)$ then the multiplexed sequence is k-constrained. The rate, R_1, of the code is

$$R_1 = \frac{u + \log_2(2^r - 1)}{u + r} = \frac{k - 2(r - 1) + \log_2(2^r - 1)}{k - r + 2}.$$

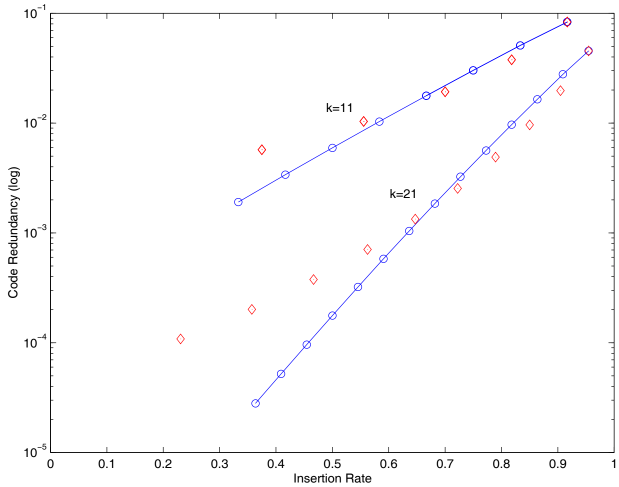

Fig. 2. Redundancy versus insertion rate using independent non-zero r-bit nibbles for $k = 11$ and $k = 21$. The maxentropic bound is plotted as a reference.

The insertion rate, I_1, is

$$I_1 = \frac{u}{u+r} = \frac{k-2r+2}{k-r+2}.$$

Results of computations for $k = 11$ and $k = 21$ are shown in Figure 2. We note that for $k = 11$, there are two points, $r = 2, 3$, and for $k = 21$ there are four points $r = 2, 3, 4, 5$ situated below the maxentropic bound. It does not come as a surprise that there are points below the maxentropic bound, but it is surprising that there is a class of simple bit-stuffing schemes that performs better than the maxentropic bound. In the next section, we will generalize the above idea, and identify more classes of codes.

B. Second example

We start with some definitions. Let D_k be the $(k+1)\times(k+1)$ adjacency matrix that represents the k-constrained channel. The entries of D_k are simply given by

$$\begin{aligned} d_{i1} &= 1, \;\; i \geq 1, \\ d_{ij} &= 1, \;\; j = i+1, \\ d_{ij} &= 0, \;\; \text{otherwise}. \end{aligned} \tag{4}$$

Let D_k^0 be the $(k+1)\times(k+1)$ matrix, whose entries are defined by

$$\begin{aligned} d_{ij}^0 &= 1, \;\; j = i+1, \\ d_{ij}^0 &= 0, \;\; \text{otherwise}. \end{aligned} \tag{5}$$

Note that the entries of D_k^0 equal those of D_k with the exception of the left column that pertains to emitted 'one's. Then, as defined above, let $\boldsymbol{\psi^n} = (\psi_1, \ldots, \psi_n)$, be a binary vector of length n, where $\psi_i = 0$ if position i is unconstrained, and $\psi_i = 1$ if position i is constrained. Then compute recursively, for $i = 1, \ldots, n$

$$A_i = A_{i-1}\{(1-\psi_i)D_k^0 + \psi_i D_k\},$$

where $A_0 = E$, the unity matrix [3]. Let $D_{k,\psi^n} = A_n$, and define the capacity C_{k,ψ^n} as the base-two logarithm of the maximum eigenvalue of D_{k,ψ^n}. Then, we have

$$R_{k,\psi^n} = \frac{n - w(\boldsymbol{\psi^n}) + C_{k,\psi^n}}{n}$$

and

$$I_{k,\psi^n} = 1 - \frac{w(\boldsymbol{\psi^n})}{n}. \tag{6}$$

For $k > 4$, we identified groups of vectors $\boldsymbol{\psi^n}$ such that $(I_{k,\psi^n}, R_{k,\psi^n})$ is below the maxentropic bound. Of great interest are vectors $\boldsymbol{\psi^n}$ having $n-w$ consecutive 'zero's and w consecutive 'one's, i.e. $\psi_i = 1$, $1 \leq i \leq w$, and $\psi_i = 0$, $w+1 \leq i \leq n$.

In the following subsections, we will discuss some results.

1) Case $n = k+1$: In case $n = k+1$, we find

$$D_{k,\psi^n} = D_k^w (D_k^0)^{k+1-w}.$$

It can easily be verified that the columns and rows of D_{k,ψ^n} with index $> w$ are zero, and that the $(w \times w)$ sub-matrix of D_{k,ψ^n} equals D_{w-1}^w. Then, as $C_{k,\psi^n} = wC(w-1)$ we have

$$R_{k,\psi^n} = \frac{k+1-w+wC(w-1)}{k+1}$$

and

$$I_{k,\psi^n} = 1 - \frac{w}{k+1}, 1 \leq w \leq k+1. \tag{7}$$

This is the maxentropic case as discussed above. Of much more interest is the next case, $n = k$.

2) Case $n = k$: If $n = k$, we find

$$D_{k,\psi^n} = D_k^w (D_k^0)^{k-w}.$$

It can be verified that the columns and rows of D_{k,ψ^n} with index $> w$ are zero. Let $\hat{D}_w$ denote the the $w \times w$ sub-matrix of D_{k,ψ^n}. The finite-state machine underlying adjacency matrix $\hat{D}_w$ generates a $k = w$-constrained sequence of concatenated w-tuples, where the all-zero w-tuple is forbidden.

Let $\hat{C}(w)$ denote the base-two logarithm of the maximum eigenvalue of $\hat{D}_w$. Clearly, because of the additional constraint imposed, $\hat{C}(w)/w < C(w)$. A $(w-1)$-constrained channel does not generate all-zero w-tuples, so that we conclude $C(w-1) < \hat{C}(w)/w < C(w)$. Two simple cases can be found by hand, namely $\hat{C}(2) = \log_2(3)$ and $\hat{C}(3) = -1 + \log_2(7+\sqrt{45})$. Table I shows $\hat{C}(w)/w$ and $C(w)$ as a function of w. We notice that the difference between $\hat{C}(w)/w$ and $C(w)$ approaches zero for larger w. We simply find

$$R_{k,\psi^n} = \frac{k-w+\hat{C}(w)}{k}$$

and

$$I_{k,\psi^n} = 1 - \frac{w}{k}, 2 \leq w \leq k. \tag{8}$$

Results of computations for $k = 11$ and $k = 21$ are shown in Figure 3. We notice that for large insertion rates the maxentropic bound can be improved, but the gain all but vanishes for smaller values of the insertion rate.

TABLE I
$\hat{C}(w)/w$ AND $C(w)$ VERSUS w.

w	$C(w)$	$\hat{C}(w)/w$
2	0.8791464	0.7924813
3	0.9467772	0.9256559
4	0.9752253	0.9684265
5	0.9881087	0.9856044
6	0.9941917	0.9931954
7	0.9971343	0.9967182
8	0.9985779	0.9983985
9	0.9992919	0.9992129
10	0.9996468	0.9996114
11	0.9998236	0.9998076

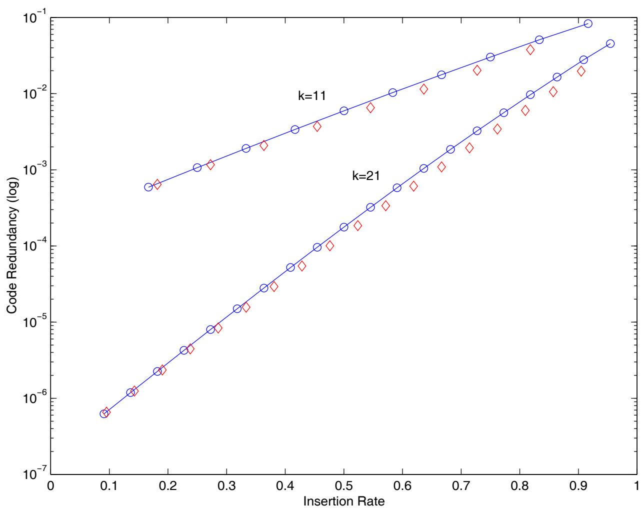

Fig. 3. Redundancy versus insertion rate for the case $n = k$ w-bit nibbles for $k = 11$ and $k = 21$. The maxentropic bound is plotted as a reference.

IV. Conclusions

We have studied constrained systems with unconstrained positions. We have presented two classes of simple bit-stuffing schemes that require less redundancy than predicted by the bound, which is based on the performance of maxentropic k-constrained systems.

References

[1] J. Campello de Souza, B. Marcus, R. New, and B. Wilson, 'Constrained Systems with Unconstrained Positions', *IEEE Trans. Inform. Theory,* vol IT-48, pp. 866-879, April 2002.

[2] T. Lei Poo, P. Chaichanavong, B.H. Marcus, 'Tradeoff Functions for Constrained Systems with Unconstrained Positions', *IEEE Trans. Inform. Theory,* vol. IT-52, pp. 1425-1449, April 2006.

[3] K.A.S. Immink, *Codes for Mass Data Storage Systems,* Second Edition, ISBN 90-74249-27-2, Shannon Foundation Publishers, Eindhoven, Netherlands, 2004.

[4] K.A.S. Immink, 'A Practical Method for Approaching the Channel Capacity of Constrained Channels', *IEEE Trans. Inform. Theory,* vol. IT-43, no. 5, pp. 1389-1399, Sept. 1997.

[5] A.J. de Lind van Wijngaarden and K.A.S. Immink, 'Maximum Runlength Limited Codes with Error Control Capabilities', *IEEE Journal on Selected Areas in Communications,* vol. 19, pp. 602-611, April 2001.