Error Propagation Assessment of Enumerative Coding Schemes

Kees A. Schouhamer Immink, Fellow, IEEE,
and Augustus J.E.M. Janssen, Senior Member, IEEE,*

March 30, 1999

Summary - *Enumerative coding is an attractive algorithmic procedure for translating long source words into codewords and vice versa. The usage of long codewords makes it possible to approach a code rate which is as close as desired to Shannon's noiseless capacity of the constrained channel. Enumerative encoding is prone to massive error propagation as a single bit error could ruin entire decoded words. This contribution will evaluate the effects of error propagation of the enumerative coding of runlength limited sequences.*

Key words: enumerative coding, RLL, error propagation

*Kees A. Schouhamer Immink is with the Institute of Experimental Mathematics, Ellernstrasse 29, 45326 Essen, Germany, (the work was conducted while the author was with Philips Research Labs, Eindhoven), E-mail: immink@exp-math.uni-essen.de. Augustus J.E.M. Janssen is with Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands, Email: janssena@natlab.research.philips.com

1 Introduction

The technique of enumerative coding [1] makes 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. The usage of long codewords makes it possible to approach a code rate which is arbitrarily close to Shannon's noiseless capacity of the constrained channel. The risk of extreme error propagation precluded its usage in practical systems. Single channel bit errors may result in error propagation that could corrupt the entire data in the decoded word, and, of course, the longer the codeword the greater the number of data symbols affected.

This article will evaluate the effects of error propagation of enumerative coding, where it is assumed that the constrained code is used in the conventional code configuration. It will be shown that when certain measures are taken, the average error propagation can be controlled to a level which is quite acceptable for many applications. We start with a basic outline of the enumeration algorithm followed by the error propagation assessment of enumerative schemes applied to the coding of runlength limited sequences.

2 Enumerative encoding

Let $\{0,1\}^n$ denote the set of binary sequences of length n and let S be any (constrained) subset of $\{0,1\}^n$. The set S can be ordered lexicographically as follows: if $\boldsymbol{x} = (x_1, \ldots, x_n) \in S$ and $\boldsymbol{y} = (y_1, \ldots, y_n) \in S$, 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' $<$ '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 $\boldsymbol{y} < \boldsymbol{x}$.

Let $n_s(x_1, x_2, \ldots, x_u)$ be the number of elements in S for which the first u coordinates are $(x_1, x_2, \ldots, x_u)$. Cover [1] showed that the rank of $\boldsymbol{x} \in S$ can be obtained by

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

In Section 3 we will evaluate error propagation effects. It is not a simple matter to analyze the effects of error propagation for a large class of constrained codes. We will focus on the enumeration of the set of runlength limited

(RLL) or (d,k)-constrained sequences, i.e., sequences that have at least d and at most k 'zeros' between consecutive 'ones'. For (d,∞)-constrained sequences (1) simplifies into

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j N(n-j), \tag{2}$$

where $N(i)$ denotes the number of (d,∞)-constrained sequences of length i. Clearly, the rank $i_S(\boldsymbol{x})$ can be found by simply forming the inner product of the received n-vector $\boldsymbol{x}$ and the n-vector of weights $(N(n-1),\ldots,N(0))$. Similar simple enumerative schemes have been derived for $(0,k)$ [4] and (d,k) sequences [5]. The number of (d,∞)-constrained sequences of length n can be found with

$$\begin{array}{rl} (i) & N(n) = n+1,\ \ 0 \le n \le d+1, \\ (ii) & N(n) = N(n-1) + N(n-d-1),\ \ n > d+1. \end{array} \tag{3}$$

Note that if $d=0$, i.e. S is the set of unconstrained sequences, $\{0,1\}^n$, (2) reduces to the familiar binary-to-decimal conversion algorithm

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j 2^{n-j}.$$

For $d>0$, we have n weights of approximately n bits, so that the amount of storage required for storing the weights is proportional to n^2. If we use a floating point representation of the weights, each weight is represented by a fixed number of bits, s. As a result, the hardware required for storage grows linearly with the codeword length n. The finite-precision representation of the weights will entail a (small) loss in code rate [2]. Floating point arithmetic employs a two-part radix-2 representation $I=(m,e)$ to express the weight $I = m\times 2^e$, where I, m, and e are non-negative integers. The two components m and e are usually called *mantissa* and *exponent* of the integer I, respectively. The translation of a weight into (m,e) is easily accomplished with the following procedure which ensures that the mantissa m is represented by a specified number of bits, denoted by q. Let I be a positive integer, and let

$$u = \lfloor \log_2 I \rfloor,$$

then the q-bit truncation of I, denoted by $\lfloor I \rfloor_q$,

$$\lfloor I \rfloor_q = \lfloor 2^{-(u+1-q)} I \rfloor 2^{u+1-q}, \tag{4}$$

can be represented in binary floating-point representation whose mantissa requires at most q non-zero bits.

If the above finite-precision arithmetic is used in the enumeration algorithms we must modify the set of weights, $\{N(i)\}$, developed above. To that end, let $\hat{N}(i)$ denote the number of (d) sequences of length i that can be encoded with a q-bit mantissa representation, then

$$\hat{N}(i) = \begin{cases} i+1, & 1 \le i \le d+1, \\ \lfloor \hat{N}(i-1) + \hat{N}(i-1-d) \rfloor_q, & i > d+1. \end{cases} \tag{5}$$

It is tacitly assumed that $N(i)$, $1 < i \le d+1$, can be represented by a mantissa of q bits, i.e., $d+2 < 2^q$. The enumeration algorithm itself remains unchanged, that is

$$i_S(\boldsymbol{x}) = \sum_{j=1}^{n} x_j \hat{N}(n-j). \tag{6}$$

Note that, though we have not explicitly written down the enumeration algorithms of the general (d,k) case, it can be observed from the results in [2] that the analysis of the error propagation effects given below can be used to this general case.

3 Error propagation

In this section we will investigate the effects of error propagation. It is assumed that a binary source word, $\boldsymbol{b}$ is translated into a binary codeword $\boldsymbol{x}$ using the enumeration algorithm. During transmission of $\boldsymbol{x}$ a single error is made, i.e. we receive $\boldsymbol{x}'$, $d_H(\boldsymbol{x},\boldsymbol{x}')=1$, where $d_H(\boldsymbol{x},\boldsymbol{y})$ denotes Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. Translation using (2) or (6) will result in the word $\boldsymbol{b}' = i_S(\boldsymbol{x}') \neq \boldsymbol{b} = i_S(\boldsymbol{x})$, where $\boldsymbol{b}'$ and $\boldsymbol{b}$ are the binary representations of $i_S(\boldsymbol{x}')$ and $i_S(\boldsymbol{x})$, respectively. In particular we are interested in $d_H(\boldsymbol{b},\boldsymbol{b}')$ and the error burst length distribution. The *error burst length*, b, is defined by $b = n_{\max} - n_{\min} + 1$, where $n_{\min}$ and $n_{\max}$ denote the smallest and largest positions where $\boldsymbol{b}'$ and $\boldsymbol{b}$ differ.

If an error is made at position k of the codeword, then the decoder will invoke (6) and form the inner product

$$\begin{aligned} i_S(\boldsymbol{x}') &= \textstyle\sum_{j=1}^{n} x_j \hat{N}(n-j) + a\hat{N}(n-k) \\ &= i_S(\boldsymbol{x}) + a\hat{N}(n-k), \\ &= \boldsymbol{b} + a\hat{N}(n-k), \end{aligned} \tag{7}$$

where $a = 1$ if $x_k = 0$ or $a = -1$ if $x_k = 1$. All additions (or subtractions) are in binary notation. Clearly, severe error propagation can only occur if the binary addition (or subtraction) of $\boldsymbol{b} = i_S(\boldsymbol{x})$ and $\hat{N}(n-k)$ results in a long carry.

An analysis of the error statistics can be made if we make some assumptions. It is assumed that the source word $\boldsymbol{b}$ is a random binary vector of doubly infinite length. Secondly, the mantissa of a weight $\hat{N}(n-k)$ is the binary q-vector $\boldsymbol{y} = (y_{q-1} \ldots, y_0)$. By definition $y_{q-1} = 1$ and the remaining $(q-1)$ elements are assumed to be random. If the above assumptions hold, the next Theorem provides the error burst length distribution.

Theorem 1 *With the above assumptions on the randomness of* $\boldsymbol{b}$ *and* $\boldsymbol{y}$ *the error burst length distribution,* $p(b)$*, is given by*

$$p(b) = \begin{cases} \left(\frac{1}{2}\right)^{q} & , \quad b = 1, \\ \left(\frac{1}{2}\right)^{q-b+2} & , \quad 2 \le b \le q, \\ \left(\frac{1}{2}\right)^{b-q+1} & , \quad b > q, \end{cases} \tag{8}$$

and $p(b) = 0$ *for* $b \le 0$.

Proof We may assume that we have an addition error (i.e. a=1 in (7)). There holds

$$\begin{aligned} p(b = l) &= \textstyle\sum_k p(n_{\max} = l + k - 1, n_{\min} = k) \\ &= \textstyle\sum_k p(n_{\max} = l + k - 1 | n_{\min} = k) p(n_{\min} = k) \end{aligned} \tag{9}$$

We compute for $k = 0, 1, \ldots, q-1$

$$p(n_{\min} = k) = p(y_0 = y_1 = \ldots = y_{k-1} = 0, y_k = 1) = \left(\frac{1}{2}\right)^{\min(k+1, q-1)}, \tag{10}$$

while $p(n_{\min} = k) = 0$ for other values of k. Furthermore, again for $k = 0, 1, \ldots, q-1$,

$$p(n_{\max} = q - 1 | n_{\min} = k) = 1 - p(\text{carry at } q - 1 | n_{\min} = k), \tag{11}$$

while for $m > q - 1$ there holds

$$\begin{aligned} &p(n_{\max} = m | n_{\min} = k) = \\ &p(\text{carry at } q-1 | n_{\min} = k) p(b_q = \ldots = b_{m-1} = 1, b_m = 0) = \\ &\left(\frac{1}{2}\right)^{m-q+1} p(\text{carry at } q - 1 | n_{\min} = k). \end{aligned} \tag{12}$$

It thus follows that we only need to compute $p(\text{carry at } q-1|n_{\min}=k)$ for $k=0,1,\ldots,q-1$. To do so, we define more generally for $k=0,1,\ldots,q-1$ and $j=k,\ldots,q-1$

$$r_{jk} = p(\text{carry at } j|n_{\min}=k). \tag{13}$$

Now $r_{q-1,q-1}=\frac{1}{2}$, and it is not hard to show for $k=0,1,\ldots,q-2$ there holds

$$\begin{array}{lcl} r_{kk} & = & 1/2, \\ r_{q-1,k} & = & 1/2+(1/2)r_{q-2,k}, \\ r_{jk} & = & 1/4+(1/2)r_{j-1,k}, \quad j=k+1,\ldots,q-2. \end{array} \tag{14}$$

It thus follows that for $k=0,1,\ldots,q-2$

$$r_{kk}=r_{k+1,k}=\ldots=r_{q-2,k}=\frac{1}{2} \text{ and } r_{q-1,k}=\frac{3}{4}. \tag{15}$$

Finally, using (9), (10), (11), (12), (15), and $r_{q-1,q-1}=\frac{1}{2}$, we get

$$p(b=1)=p(n_{\max}=n_{\min}=q-1)=\left(\frac{1}{2}\right)^{q}, \tag{16}$$

$$p(b=q-p)=\frac{1}{4}\left(\frac{1}{2}\right)^{p+1}+\left(\frac{1}{2}\right)^{2q-p-1}+\frac{3}{4}\sum_{k=p+1}^{q-2}\left(\frac{1}{2}\right)^{k-p}\left(\frac{1}{2}\right)^{\min(k+1,q-1)}, \tag{17}$$

for $p=0,1,\ldots,q-2$ and

$$p(b=q+p)=\left(\frac{1}{2}\right)^{2q+p-1}+\frac{3}{4}\sum_{k=0}^{q-2}\left(\frac{1}{2}\right)^{k+p}\left(\frac{1}{2}\right)^{\min(k+1,q-1)}, \tag{18}$$

for $p=1,2,\ldots$. Working out (17) and (18) we find

$$p(b)=\begin{cases} \left(\frac{1}{2}\right)^{q} & , \quad b=1, \\ \left(\frac{1}{2}\right)^{q-b+2} & , \quad 2\le b\le q, \\ \left(\frac{1}{2}\right)^{b-q+1} & , \quad b>q, \end{cases} \tag{19}$$

while $p(b)=0$ for $b\le 0$. This concludes the proof.

From Theorem 1 it is clear that the most likely burst has a length of q, $q+1$

bits with probability $p(q) \approx p(q+1) \approx 1/4$. Error bursts longer or shorter than q or $q+1$ have an exponentially decaying probability. Figure 1 compares results of a typical example of computer simulations and computations using (8). The outcomes of the simulations show a reasonable agreement with the theory developed for the longer bursts and that there is a discrepancy for shorter bursts. In [2] it has been shown that the coefficients are periodic in nature. It has been found that the period lengths for selected values of d, k, and p can be very short and that, therefore, the assumption of randomness of the coefficients, which is essential for the validity of the Theorem, is not valid. However, for the majority of parameters the period length is quite long, and the coefficients show a reasonable "randomness".

Thus we can control the error propagation by a proper choice of q. The choice of q has an effect on the maximum achievable code rate which can simply be approximated by (see [2])

$$\hat{C}(d,k) \approx C(d,k) - 2^{-(q+2)}, \tag{20}$$

where $C(d,k)$ is the capacity of the (d,k)-constrained channel and $\hat{C}(d,k)$ is the capacity of the same channel using enumerative encoding with floating arithmetic. With (8) and (20) a trade-off has to made between, on the one hand, the error propagation effects, which has a bearing on the required capability of the error control code, and on the other hand, the rate of the constrained code. This is a very subtle trade-off requiring a detailed specification of the various coding layers, and has therefore not been pursued.

We present some results that follow from Theorem 1, or the proofs of which are of a similar nature as that of Theorem 1.

1. For the expectation, $E(b)$, and variance, $\sigma^2(b)$, of b we have

$$E(b) = q - \frac{1}{2} + (\frac{1}{2})^q, \quad \sigma^2(b) = 2q + \frac{17}{4} - 2q(\frac{1}{2})^q - (\frac{1}{2})^{2q}. \tag{21}$$

Furthermore, for the number of errors made, denoted by n, i.e. the number of positions k where $x_k^{'} \neq x_k$, we have

$$E(n) = \frac{1}{2}(q+3), \quad \sigma^2(n) = \frac{1}{6}(q+3) + \frac{10}{9} + \frac{8}{9}(\frac{1}{2})^q. \tag{22}$$

2. For the damage done by a fixed additive $\boldsymbol{y} = (y_{q-1}, \ldots, y_0)$ we have the following. Let k be the minimum index i such that $y_i = 1$. Then

$$p(b = q-k|\boldsymbol{y}) = 1 - \sum_{j=0}^{q-1} (\frac{1}{2})^{q-j} y_j, \tag{23}$$

and

$$p(b = l|\boldsymbol{y}) = \sum_{j=0}^{q-1} (\frac{1}{2})^{l+k-j} y_j, \quad l = q - k + 1, \ldots. \tag{24}$$

The above was observed by L.M.G.M. Tolhuizen. Consequently,

$$E(b|\boldsymbol{y}) = q - k + T, \quad \sigma^2(b|\boldsymbol{y}) = \frac{9}{4} - (T - \frac{3}{2})^2, \tag{25}$$

where

$$T = \sum_{j=0}^{q-1} (\frac{1}{2})^{q-1-j} y_j \in [1, 2). \tag{26}$$

Similar, though slightly more complex, expressions involving the correlation function of $\boldsymbol{y}$ can be derived for $E(n|\boldsymbol{y})$ and $\sigma^2(n|\boldsymbol{y})$.

4 Conclusions

We have investigated error propagation effects of enumerative schemes used for the coding of runlength limited sequences. We have given a theoretical expression for the error burst length distribution. The most likely burst has a length of q, $q + 1$ bits. Error bursts longer or shorter than q or $q + 1$ have an exponentially decaying probability. It has been shown that computer simulations compare fairly well with the theoretical results.

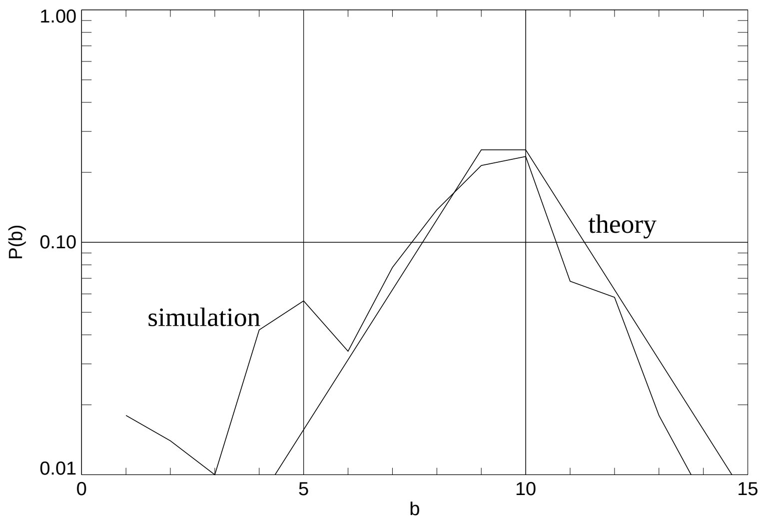

Figure 1: Comparison of error burst distribution for $d = 2$ and $q = 9$.

5 Acknowledgement

We would like to thank Ludo Tolhuizen for his comments on earlier versions of this paper.

References

[1] T.M. Cover, 'Enumerative Source Coding', *IEEE Trans. Inform. Theory,* vol. IT-19, no. 1, pp. 73-77, Jan. 1973.

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

[3] D.T. Tang and L.R. Bahl, 'Block Codes for a Class of Constrained Noiseless Channels', *Information and Control,* vol. 17, pp. 436-461, 1970.

[4] W.H. Kautz, 'Fibonacci Codes for Synchronization Control', *IEEE Trans. Inform. Theory,* vol. IT-11, pp. 284-292, 1965.

[5] L. Patrovics and K.A.S. Immink, 'Encoding of $dklr$-sequences using one weight set', *IEEE Trans. Inform. Theory,* vol. IT-42, no. 5, pp. 1553-1554, Sept. 1996.