CONSTRUCTION OF BINARY DC-CONSTRAINED CODES

by K. A. SCHOUHAMER IMMINK

Philips Research Laboratories, 5600 JA Eindhoven, The Netherlands

Abstract

The systematic design of DC-constrained codes based on codewords of fixed length is considered. Simple recursion relations for enumerating the number of codewords satisfying a constraint on the maximum unbalance of ones and zeros in a codeword are derived. An enumerative scheme for encoding and decoding maximum unbalance constrained codewords with binary symbols is developed. Examples of constructions of transmission systems based on unbalance constrained codewords are given. A worked example of an 8b10b channel code is given being of particular interest because of its practical simplicity and relative efficiency.

ECO: 5.4.

1. Introduction

An important requirement when designing a digital transmission system is the shaping of the power spectral density function of the encoded stream by adding redundancy to match the particular physical properties of the transmission channel. Many practical examples of transmission systems can be mentioned that do not pass the low frequencies with sufficient signal-to-noise ratio. Filtering out the low frequencies can only be done if the encoded signal itself is not seriously distorted by this filtering. Shaping the spectrum of the encoded stream by coding can cope with this phenomenon. Shaping, however, can only be done if some kind of redundancy is added to the source sequence. The field of application of digital channel codes with suppressed low frequency components is quite broad. We find applications in transmission systems over fibre or metallic cable\(^1\)\(^-\)\(^8\) and in storage media such as optical recording\(^4\) and magnetic recording\(^5\).

Transmission systems designed to achieve DC suppression are mostly based on so-called block codes, where the source digits are grouped in source blocks of \(m\) digits. The \(m\) digit blocks are translated using a code book into blocks of \(n\) digits called codewords. Cattermole\(^6\)\(^,\)\(^7\) and Carter\(^8\) gave examples of chan-
Construction of binary DC-constrained codes

nel codes based on block codes with codewords having zero or low disparity, where the disparity of a codeword is defined as the difference of the number of ones and zeros in the codeword.

Justesen\(^9\) showed a close relationship between the low-band cut-off frequency and the variance of the running digital sum of a DC-suppressed encoded stream. The (running) digital sum is defined for a binary stream as the accumulated sum of ones and zeros (a zero counted as \(-1\)) from the beginning of the transmission. Furthermore Chang et al.\(^{10}\) found that the sum variance plays an important role in the bit error rate when the transmission channel is AC-coupled. The simple codes as previously discussed have the disadvantage that codeword length, the sum variance and rate have a fixed relationship.

In this paper the code design is based on codewords having a constraint on the maximum number of assumed sum values, i.e. those codewords are deleted having a relative large contribution to the sum variance.

We start in sec. 2 with a brief description of the properties of sequences that assume a limited number of digital sum values. These properties are used to establish a figure of merit of DC-constrained codes. The actual basis of code design is given in sec. 3. It provides a method for the enumeration of codewords with an unbalance constraint. In general the efficiency of a channel code can be improved if larger codewords are allowed. Unfortunately the complexity of the encoding and decoding hardware increases exponentially with increasing codeword length if a direct method using look-up tables of the source words and their channel representations is used. Section 4 deals with algorithms for encoding and decoding of constrained codewords growing polynomially with increasing codeword length. Finally, in sec. 5 the problem is addressed of selecting sets of codewords to design DC-suppressed channel codes. A worked example of a new \(8b10b\) transmission code is given showing superior features with respect to other codes with rate 8/10.

2. Properties of \(z\)-constrained sequences

This section provides a description of some properties of DC-constrained sequences generated by a Markov source having maximum entropy. This study gives a relation between the redundancy and the sum variance of codes. Using this theory we can derive a new figure of merit taking into account both the redundancy and the resulting frequency range with suppressed components.

Consider binary sequences \(x = (x_1, \ldots, x_i, \ldots)\), \(x \in \{-1, 1\}\). The so-called (running) digital sum of a sequence plays a significant role in the analysis and synthesis of codes of which the spectrum vanishes at the low frequency end. The digital sum \(z_i\) is defined as:
Chien\textsuperscript{11}) studied sequences $x$ assuming a finite number of sum values. Sequences having a constraint on the maximum number of sum values are defined here as $z$ (-constrained) sequences. The total number of sum values a sequence assumes is often called the digital sum variation. Chien determined the information capacity of $z$ sequences as a function of the number of allowed sum values. His results will be briefly discussed in the following.

Taking $z_i$ at any instant $i$ as the state of the signal stream $x$, then the bounds on $z_i$ define a set of allowable states of a Markov source. Each transmission of an additional symbol $x_i$ can be considered as a transition from one state to another. This transition can be represented by a so-called connection matrix. For the $N$ state Markov source an $N \times N$ connection matrix $D$ is defined by $D(i, j) = 1$ if a transition from state $i$ to state $j$ is allowable and $D(i, j) = 0$ otherwise. The connection matrix $D$ is for the $z$-constrained sequences given by

$$D(i + 1, i) = D(i, i + 1) = 1, \quad 1 \leq i \leq N - 1,$$

$$D(i, j) = 0, \text{ otherwise.}$$

Note that $D$ is a symmetric Toeplitz matrix. As an example we have written down the matrix $D$ for $N = 5$.

$$D = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}$$

The above representation is related to the input-restricted noiseless channel studied by Shannon\textsuperscript{12}). The Markov information source model enables us to compute the channel capacity, defined as the number of bits per channel symbol that can maximally be carried by the constrained channel sequence generated by the Markov source. The channel capacity equals the maximum entropy of the Markov source which is given by the base-two logarithm of the largest real eigenvalue of the connection matrix $D$\textsuperscript{13}). According to Chien the maximum real eigenvalue of the Toeplitz matrix is given by

$$\lambda = 2 \cos \left( \frac{\pi}{N + 1} \right),$$

so that the capacity is

$$C(N) = 2 \log \lambda = 1 + 2 \log \cos \left( \frac{\pi}{N + 1} \right).$$
More interesting properties of $z$ sequences were derived by Justesen. He calculated the power density function of $z$ sequences. The transition matrix of a maxentropic $z$-constrained source was determined, i.e. a Markov source with transition probabilities chosen in such a way that the source has maximum entropy. In order to determine the transition probabilities that maximize the entropy of the $N$ state source, one must determine an eigenvector $p_a = (p_a(1), p_a(2), \ldots, p_a(N))$ that satisfies (ref. 14, pp. 210)

$$p_a D = \lambda p_a.$$  (5)

The joint probability $p(i, j)$ of a transition from state $i$ to $j$ of the maxentropic Markov source is given by

$$p(i, j) = \lambda^{-1} D(i, j) \frac{p_a(j)}{p_a(i)}, \quad i, j = 1, 2, \ldots, N.$$  

Using the fact that $D$ is a Toeplitz matrix we find for the stationary probability $p_a(i)$ being in state $i$ \(^9\)

$$p_a(i) = \frac{2}{N + 1} \sin^2 \left( \frac{\pi i}{N + 1} \right), \quad 1 \leq i \leq N.$$  (6)

The cut-off frequency $\omega_0$ of a DC-constrained sequence is defined by \(^9,15\)

$$\frac{H(\omega T)}{T} = \frac{1}{2},$$  (7)

where $H(\omega T)$ is the power density function of the sequence versus frequency and $T$ the time duration of a channel symbol. Justesen found a remarkable and very useful relation

$$\omega_0 T \approx \frac{1}{2s^2},$$  (8)

where $s^2$ is the sum variance of the sequence.

This simple relation is not only restricted to maxentropic sequences; examples of practical channel codes investigated by Justesen show that the reciprocal relation (eq. (8)) between cut-off frequency and sum variance is very accurate. This motivated us to use the sum variance as a criterion of the low-frequency properties of a channel code. This is of practical importance because the sum variance of a sequence is often easier to calculate than the auto-correlation function or spectrum. The sum variance of a maxentropic $z$ sequence can be found using eq. (6)

$$\sigma^2(N) = \frac{2}{N + 1} \sum_{k=1}^{N} \left( \frac{1}{2}(N + 1) - k \right)^2 \sin^2 \left( \frac{\pi k}{N + 1} \right).$$  (9)
TABLE I
Capacity and sum variance of maxentropic $z$ sequences
with digital sum variation $N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$C(N)$</th>
<th>$\sigma^2(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>0.694</td>
<td>0.80</td>
</tr>
<tr>
<td>5</td>
<td>0.792</td>
<td>1.17</td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>1.59</td>
</tr>
<tr>
<td>7</td>
<td>0.886</td>
<td>2.09</td>
</tr>
<tr>
<td>8</td>
<td>0.91</td>
<td>2.64</td>
</tr>
<tr>
<td>9</td>
<td>0.93</td>
<td>3.26</td>
</tr>
</tbody>
</table>

In table I the capacity $C(N)$ and sum variance versus the digital sum variation $N$ are collected (using eqs (4) and (9)).

For large digital sum variation $N$ the sum variance and capacity can be approximated by

$$C(N) \sim 1 - \frac{\pi^2}{(2 \ln 2)(N + 1)^2},$$

$$\sigma^2(N) \sim \left( \frac{1}{12} - \frac{1}{2\pi^2} \right) (N + 1)^2.$$

This leads to following relation between the redundancy $1 - C(N)$ and the sum variance of a maxentropic $z$ sequence

$$0.25 \geq (1 - C(N)) \sigma^2(N) > \frac{(\pi^2 - 1)}{4 \ln 2} = 0.2326.$$

Actually the right-hand bound is within 1% accuracy for $N > 9$. This relation shall be used later to establish a figure of merit of DC-constrained codes.

3. Enumeration of $(N, n)$ sequences

Let $x = (x_1, x_2, \ldots, x_n)$, $x_i \in \{0, 1\}$ be an $(n)$ sequence. We define the digital sum $z_k$ of the $(n)$ sequence

$$z_k = \sum_{i=1}^{k} (2x_i - 1) + z_0, \quad 1 \leq k \leq n.$$  

The rescaling has been done in such a way that 'zeros' are counted as $-1$. The initial sum is denoted by $z_0$. 

---

K. A. Schouhamer Immink
Construction of binary DC-constrained codes

Definition

A binary \((n)\) sequence is called an \((N_1, N_2, n)\) sequence if \(z_k (1 \leq k \leq n)\) is bounded between \(N_1\) and \(N_2 (N_2 > N_1)\).

Without loss of generality choose \(N_1 = 1\) and \(N_2 = N\). We further use the shorthand notation \((N, n)\) sequence to denote a \((1, N, n)\) sequence.

Let \(T = T(z_n, n; N, z_0)\) be the set of \((N, n)\) sequences for given length \(n\), sum constraint \(N\), initial sum \(z_0\) and terminal sum \(z_n\). The cardinality of the set \(T\) can be found by manipulation of the \(N \times N\) connection matrix \(D\).

Let \(|T| = A(z_n, n; N, z_0)\) denote the number of distinct sequences of length \(n\) starting in state \(z_0\) and terminating in state \(z_n\), then

\[
A(z_n, n; N, z_0) = D^n(z_0, z_n),
\]

where \(D^n(i, j), 1 \leq i, j \leq N\), denotes the \(i, j\)-th element of the \(n\)-th power of the matrix \(D\).

In order to avoid many multiplications required in the matrix powers, observe the specific structure of the connection matrix \(D\). We find the following recursive relations for determining the number of sequences.

Let

\[
A(z_0, 0; N, z_0) = 1,
\]

\[
A(i, 0; N, z_0) = 0; \text{ all } i \neq z_0
\]

and

\[
A(0, j; N, z_0) = A(N + 1, j; N, z_0) = 0; \quad 0 \leq j \leq n.
\]

Let further

\[
A(i, j; N, z_0) = A(i - 1, j - 1; N, z_0) + A(i + 1, j - 1; N, z_0); \quad 1 \leq i \leq N; \quad 1 \leq j \leq n.
\]

We can observe that except at the boundaries the recursive relations are similar to the relations for the binomial coefficients in the Pascal triangle.

4. Enumerative coding of \((N, n)\) sequences

In this section a general enumerative technique for decoding \((N, n)\) sequences is developed. To that end we establish a 1-1 mapping from a set \(T\) of \((N, n)\) sequences onto a set of integers \(0, 1, \ldots, |T| - 1\), where \(|T|\) is the cardinality of the set \(T\).

Let \(T(z_n, n; N, z_0)\) be the set of \((N, n)\) sequences for the given constraints. The set \(T\) can be ordered lexicographically as follows:

If \(x = (x_1, \ldots, x_n)\) and \(y = (y_1, \ldots, y_n)\) are elements of the set \(T\) then \(y\) is called less than \(x\), in short \(y <_0 x\), 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’ <\(00110\)’. The position of \(x\) in the lexicographical ordering of \(T\) is defined to be the rank of \(x\) denoted by \(r(x)\), i.e. \(r(x)\) is the number of all \(y\) in \(T\) with \(y <_0 x\).
Theorem 1

The rank \( r(x) \) of the binary sequences \( x \) in \( T \) can be calculated according to

\[
   r(x) = \sum_{k=1}^{n} A(s_k + 1, n - k; N, z_0) \cdot x_k, \tag{13}
\]

where

\[
   s_k = z_n - \sum_{i=1}^{k-1} (2x_i - 1).
\]

Proof

According to Cover \(^{16}\) the lexicographic index \( r(x) \) is given by

\[
   r(x) = \sum_{j=1}^{n} x_j \cdot n_s(x_1, x_2, \ldots, x_{j-1}, 0),
\]

where \( n_s(x_1, x_2, \ldots, x_k) \) denotes the number of elements in set \( T \) for which the first \( k \) coordinates are given by \((x_1, x_2, \ldots, x_k)\). The first \((j - 1)\) coordinates of \( x \) determine the state \( s_j \)

\[
   s_j = z_n - \sum_{i=1}^{j-1} (2x_i - 1),
\]

as

\[
   z_n = z_0 + \sum_{i=1}^{j-1} (2x_i - 1) - 1 + \sum_{i=j+1}^{n} (2x_i - 1).
\]

The number of elements in set \( T \) with prefix \((x_1, x_2, \ldots, x_{j-1}, 0)\) equals the number of \((N, n - j)\) sequences starting with digital sum \( z_0 \) and terminating with \( s_j + 1 \).

In other words

\[
   n_s(x_1, x_2, \ldots, x_{j-1}, 0) = A(s_j + 1, n - j; N, z_0).
\]

Example 1

Consider the set \( T(z_n, n; N, z_0) = T(4, 10; 7, 4) \). It follows from \( z_n = z_0 \) that all elements of the set have an equal number of zeros and ones. It can be verified that the sequence \( x = (1100010110) \) is a member of the set and can be ordered according to theorem 1. The ranking procedure is visualized in fig. 1. The array in fig. 1 has \( N = 7 \) columns and \( n = 10 \) rows. The entries are given by the weighting coefficients \( A(i, j; N, z_0) \). We can easily verify the recursion relations eq. (12).

Ranking proceeds as follows. Start at the bottom at the \( z_n = 4 \)-th column. The left-most symbol \( x_1 \) of the sequence \( x = (1100010110) \) is a 1. Move one
Construction of binary DC-constrained codes

Fig. 1. Generalized Pascal's triangle.

step in the X direction i.e., towards 68, and record the entry at a single step in
the X direction, i.e. 116 in an accumulator. The next symbol is again a 1, so
move again in the X direction towards 34 and add the entry 68 to the current
accumulator value, i.e. 116 + 68 = 184. We further proceed as depicted in the
figure. Eventually we arrive at \( r(1100010110) = 116 + 68 + 4 + 1 + 1 = 190 \).

The procedure given here is a generalization of a procedure given by Schalk-
wijk\(^{17}\) using Pascal's triangle with the binomial coefficients as entries for
ranking codewords with a given disparity.

The inverse function, conversion from a given integer \( I \) to an \((N, n)\) se-
quence with rank \( I \) is also analogous to Schalkwijk's algorithm and can be
carried out as follows.

**Inverse algorithm**

Let an integer \( I (I = 0, 1, \ldots, |T| - 1) \) and the set \( T(z_n, n; N, z_0) \) be given.
The following algorithm finds \( x \) such that \( r(x) = I \) (see Cover\(^{18}\)).

**Step 1**

\( \text{If } A(z_n + 1, n - 1; N, z_0) \leq I \text{ then set } x_1 = 1 \text{ and set } I = I - A(z_n + 1, \)
\( n - 1; N, z_0) \text{ else set } x_1 = 0. \)

**Step 2**

\( \text{For } k = 2, \ldots, n, \text{ if } A(s_k + 1, n - k; N, z_0) \leq I, \text{ where } s_1 = z_0 \text{ and } s_k = s_{k-1} - (2x_{k-1} - 1) \text{ then set } x_k = 1 \text{ and set } I = I - A(s_k + 1, n - k; N, z_0) \)
\( \text{ else set } x_k = 0. \)
Till now the procedure was confined to the ranking of one set \((N, n)\) sequences. By a straightforward generalization it is also possible to rank sequences starting from a predefined set of states and all terminating in the same state. Assume \(L\) codeword subsets starting with sum values given by \(z_i\), \(i = 1, \ldots, L\), i.e. consider the set consisting of the subsets \(T(z_n, n; N, z_i)\), \(i = 1, \ldots, L\). Ranking of the set can be accomplished with

\[
r(x) = \sum_{k=1}^{n} \sum_{i=1}^{L} A(s_k + 1, n-k; N, z_i) \cdot x_k,
\]

where

\[
s_k = z_n - \sum_{j=1}^{k-1} (2x_j - 1).
\]

This follows from the fact that the subsets are disjoint. The inverse function, converting an integer \(I\) to an \((N, n)\) sequence, can also be done in the same manner.

The ranking given in theorem 1 enables the design of channel encoders of moderate complexity. We need storage capacity for approximately \(N/2 \times n\) non-zero weighting coefficients, a full adder and an accumulator to store the intermediate and final results. This has to be compared with a look-up table of \(2^m\) entries if a non-algebraic method of coding is used. The weighting coefficients look-up table can be used for encoding and decoding, which is attractive for application in recording. We do not need a multiplier because \(x\) is binary valued and so the multiplications are simple additions. Lyon\(^{18}\) gave an example of a practical embodiment for the coding of codewords with given disparity using Pascal’s triangle. His circuitry can in principle be used with only a modification of the entries in the look-up table.

5. Transmission codes based on \((N, n)\) sequences

When the state-independent encoding strategy is used transmission codes that satisfy the digital sum constraint can readily be found. Enumerative encoding and decoding can always be used in a straightforward way. A drawback of this simple coding scheme is that generally speaking good coding efficiency will be reached for ‘large’ codewords. A much more powerful method is based on the state-dependent encoding principle. A problem will arise for the assignment of codewords to source words in all encoder states, when the algebraic coding algorithm and state-independent decoding are to be combined.

5.1. State-independent encoding

In the special case of state-independent encoding all codewords start and terminate in the same sum state. Consequently the codewords have zero-dis-
Construction of binary DC-constrained codes

parity. The number of distinct \((N, n)\) sequences can be found by the enumeration method given in the preceding section. Table II lists the number of zero-disparity \((N, n)\) sequences as a function of \(N\) and \(n\). The terminal sum state has been chosen to maximize the number of codewords. Transmission codes can readily be designed with this table.

### TABLE II

Number of zero-disparity \((N, n)\) sequences

<table>
<thead>
<tr>
<th>(n)</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N=3)</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>13</td>
<td>34</td>
<td>89</td>
<td>233</td>
<td>610</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>18</td>
<td>54</td>
<td>162</td>
<td>486</td>
<td>1458</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>19</td>
<td>61</td>
<td>197</td>
<td>638</td>
<td>2069</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>20</td>
<td>68</td>
<td>232</td>
<td>792</td>
<td>2704</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>20</td>
<td>69</td>
<td>241</td>
<td>846</td>
<td>2977</td>
</tr>
</tbody>
</table>

**Example 2**

If \(n = 8\) and \(N = 7\) the table shows that there are 68 distinct \((N, n)\) sequences satisfying the given constraints. This enables the mapping of \(m = 6\) source bits onto the \(n = 8\) channel symbols. Hence the code rate \(R = m/n = 3/4\). We only need 64 codewords, so four codewords are deleted. If the lexicographical largest or smallest codewords are deleted then the enumerative coding scheme can be applied without modification.

The rate efficiency is defined with respect to capacity of the constrained channel\(^{13}\)

\[ e = \frac{R}{C(N)}, \]

where \(N\) is the digital sum variation of the channel code.

The code of example 2 has a rate efficiency (see table I): \(0.75/0.886 = .85\).

The sum variance, adopted in sec. 2 as a low-frequency criterion, could be found by an exhaustive computation of the variance of all codewords. A computation growing polynomially in the codeword length \(n\) can be based on the enumeration method of the preceding section. The probability that the digital sum equals \(i\) at symbol position \(j\) within a codeword is given by the number of paths starting in \(z_0\), passing state \(i\) at position \(j\) and eventually terminating at \(z_n\), divided by the total number of paths from \(z_0\) to \(z_n\).
In the case of zero-disparity codewords we have \( z_0 = z_n \), so that according to the enumeration method the sum variance \( s^2 \) of zero-disparity \((N,n)\) sequences can be found by

\[
s^2 = \frac{1}{n |T|} \sum_{i=1}^{N} \sum_{j=0}^{n-1} (i - \bar{z})^2 A(i, n - j; N, z_0) A(i, j; N, z_0),
\]

where the average digital sum \( \bar{z} \) is given by

\[
\bar{z} = \frac{1}{n |T|} \sum_{i=1}^{N} \sum_{j=0}^{n-1} i A(i, n - j; N, z_0) A(i, j; N, z_0)
\]

and the cardinality \( |T| = A(z_0, n; N, z_0) \).

The code of example 2 has a sum variance \( s^2 = 1.38 \), which amounts to 66\% of the sum variance of a maxentropic sequence with \( N = 7 \). It is clear that the comparison of DC-constrained codes with maxentropic sequences should take into account both the sum variance and the rate. Therefore a new figure of merit, the efficiency \( E \) of DC-constrained codes is introduced. Eq. (10) showed that in the case of maxentropic \( z \) sequences the product of redundancy and sum variance is approximately constant. The efficiency \( E \) is now defined as the ratio of the products of redundancies and sum variances of the maxentropic \( z \) sequence and the actual code, or

\[
E = \frac{[1 - C(N)] \sigma^2(N)}{(1 - R) s^2}.
\]

Fig. 2 shows the efficiency \( E \) of zero-disparity codewords of word length \( n \) with \( N \) as a parameter. The efficiency grows asymptotically to unity with increasing codeword length.
5.2. State-dependent encoding

The case of state-independent encoding is straightforward: all codewords end with the same sum value. More efficient coding, i.e. achieving a larger rate with smaller codeword lengths, is possible if the codewords are allowed to terminate in a predefined set of sum states. The problem is now to establish this set of states and codewords for an optimal transmission. Franaszek\(^{13,19}\) has developed an algorithm for determining the existence of a set of so-called principal states for the design of constrained sequences such as run-length-limited sequences. A necessary and sufficient condition for the existence of a code is the existence of a set of principal states with the property that from each of these principal states there exists a sufficient (with respect to the number of source words) number of distinct codewords to the other principal states. The minimum number of paths from a principal state to another determines the information rate. The method outlined by Franaszek does not guarantee the important feature of state-independent decoding. A code is state-independent decodable\(^{19}\) if every mapping of a source word into a codeword has a unique inverse independent of the encoder code pages, where a code (book) page is the subset of codewords starting in a principal state. This property guarantees that every (noiseless) codeword be correctly decoded even when the actual encoder state is unknown. Franaszek showed that such a unique inverse is always possible if only two principal states are chosen.

We collected in table III some of the results of the computer search for codes using Franaszek’s method with the additional constraint of two allowed

<table>
<thead>
<tr>
<th>(N)</th>
<th>(n)</th>
<th>(m)</th>
<th>(R)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4/5</td>
<td>.95</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>11</td>
<td>14/11</td>
<td>.99</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>19</td>
<td>24/19</td>
<td>.999</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>5/6</td>
<td>.94</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>12</td>
<td>12/14</td>
<td>.97</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>19</td>
<td>19/22</td>
<td>.98</td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td>21</td>
<td>21/24</td>
<td>.99</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>14</td>
<td>14/8</td>
<td>.96</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>16</td>
<td>16/9</td>
<td>.96</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>18</td>
<td>18/10</td>
<td>.97</td>
</tr>
</tbody>
</table>

\(N\) digital sum variation, \(n\) codeword length, \(m\) number of source bits, 
\(R\) rate = \(\frac{m}{n}\), \(e\) rate efficiency = \(R/C(N)\).
principal states. The number of codewords were truncated to the nearest power of two. For this reason the sum variance of the codes is not known and therefore the rate efficiency \( e = R/C(N) \) is used as a figure of merit. The source word length \( m \) is here chosen to optimize the rate efficiency of the code. Note that in practical circumstances we generally do not have such a degree of freedom and consequently some of the efficiency is lost.

We conclude from this table that good rate efficiencies are possible with state-dependent encoding. Most interesting are codes with sum variation \( N = 5 \). In this particular case it can be proved by evaluation of the matrix \( D \) that for any codeword length the number of principal states for optimum transmission is two and that the number \( M \) of sequences from a principal state is \( (n \text{ even}) \)

\[
M = 3^{n/2}.
\]

In the \( n = 2 \) case there are three distinct codewords: the zero-disparity codewords '01' and '10' (i.e. the 'bi-phase' words) and the codewords '00' and '11' to be used alternately. The rate of the \( n = 2 \) code when using a ternary source alphabet is \( 1/2 \log 3 \). According to table I this simple code achieves 100% of the noiseless channel capacity. Also the sum variance of this code equals the sum variance of the maxentropic \( N = 5 \) sequence\(^{26} \). For binary source alphabets it is not possible to reach this efficiency, though according to table III with a codeword length \( n = 24 \) a rate \( R = 10/24 \) can be reached, being 0.1% below the asymptotic bound.

Fig. 3 shows the efficiency \( E \) versus codeword length \( n \) with the digital sum variation \( N \) as a parameter. The sum variance was calculated by an extension

![Fig. 3. Efficiency of state-dependent coding.](image)
Construction of binary DC-constrained codes

of eq. (15). The code efficiency was calculated taking into account all possible codewords (no truncation to a power of two). The efficiency of codes with sum variation $N = 5$ is for arbitrary (even) codeword length equal to unity. We note the improvement in efficiency with respect to zero-disparity codeword based coding.

Till now we did not discuss how to apply the enumerative coding method when state-dependent code book pages will be used. State-independent decoding is not guaranteed if the codewords in the code pages are lexicographically ordered. If the encoder has two pages and $N$ is odd then coding can be done in the following way. Let $A_0$ and $A_1$ be the code book pages. Codewords in $A_0$ have zero or positive disparity. We order one of the pages, say $A_0$, according to sec. 2. A codeword in page $A_1$ is found from a codeword with the same rank in $A_0$ by inverting all bits if the codeword has non zero-disparity. Source words are represented by the same zero-disparity codeword in both pages. The decoder observes the disparity of a received codeword and if needed the codeword can easily be mapped to the $A_0$ ranking. If $N$ is even this simple procedure cannot be applied. In the next section an example is given to solve the $N$ is even case.

5.3. 8b10b channel code, a worked example

In certain applications of channel codes, such as digital recording on magnetic tape or transmission over fibre-optic network it has been found that a ‘DC-balanced’ channel code with rate $R = 0.8$ has attractive features\(^{20-23}\). In this section a design example of such a code is described based on the theory of the preceding section.

The source signal is assumed to have eight parallel bits that are translated by the channel encoder using a code book onto the channel symbols. The channel symbols are sequentially recorded on the recording medium such as for example magnetic tape. In this design example the code book has two pages: $A(S_0)$ and $A(S_1)$, where $S_0$ and $S_1$ are the two encoder states. Each page has 256 entries, enabling a one-to-one state-dependent encoding. Dependent on the particular transmitted codeword the encoder state changes from state $S_0$ to $S_1$ (or from $S_1$ to $S_0$ or it remains in the same state. Both pages are so chosen that:

a) a simple encoding and decoding is possible, permitting a simple low-cost design;
b) the concatenated channel string takes on six digital sum values;
c) the maximum run-length is five channel symbols, i.e. at most five consecutive like symbols can occur;
d) state-independent decoding is possible. Upon decoding no use is made of the digital sum for decoding. In this way error propagation is limited to eight decoded source bits.
From table I it can be concluded that the capacity of a channel that occupies six sum states is $C(6) = 0.85$. In other words the design of a channel code with this constraint and rate $R = 0.8$ is feasible. The capacity of a channel occupying five sum states is $C(5) = 0.792$, so that a code with $R = 0.8$ satisfying the $N = 5$ constraint has not to be attempted.

Fig. 4 shows the unbalance trellis diagram of a code that occupies six sum states. The encoder has two principal states denoted by $S_0$ and $S_1$. The trellis diagram gives a plot of all possible sum values as a function of the symbol position. Using the generalized Pascal triangle defined in the preceding section it can be calculated that the number of codewords satisfying the channel constraints starting in $S_0$ and terminating in $S_0$ and $S_1$ is 197 and 155, respectively. From $S_1$ there are 155 and 131 codewords terminating in $S_0$ and $S_1$, respectively. We conclude that the minimum number of codewords from an initial state is $131 + 155 = 286$, being sufficient for an 8 to 10 mapping.

In order to enable simple coding the sets of codewords with zero-disparity are divided into two subsets:

1) codewords that can be state-independently encoded. We found that the total number of such codewords is 89. These codewords can be found by inspection of the trellis diagram. The zero-disparity state-independent codewords occupy four sum states so that the number of codewords and the codewords themselves can again be evaluated using the generalized Pascal triangle (see table IV);

2) other codewords with zero-disparity.

We now formulate the following rules for encoding and decoding.
Construction of binary DC-constrained codes

TABLE IV
Pascal triangles of 8b10b code

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>21</td>
<td>0</td>
<td>14</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>108</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Encoding rules

Most of the codewords of page $A(S_0)$ can be generated by the enumerative coding method. The page $A(S_1)$ can be found with some 'bit-shuffling' of codewords of page $A(S_0)$. We define the following three subsets of $A(S_0)$, where $I (0 \leq I \leq 255)$ is the rank (or decimal notation) of the source words

Subset $T_0$: $0 \leq I \leq 88$.

The codewords are state-independent encodable, have zero-disparity, and do not change the encoder state.

Subset $T_1$: $89 \leq I \leq 243$.

The codewords have non-zero disparity. By symmetry (see trellis diagram) the codewords in page $A(S_1)$ can be found from codewords in $A(S_0)$ by inverting each symbol and reversing the direction of transmission of the codeword. Example: assume a codeword with rank $I$ in $T_1$ of page $A(S_0)$ to be: '0100101111'. Inverting and reversing results in codeword $A(S_1,I) = '0000101101'$. It can be verified in the trellis diagram that this codeword satisfies the constraints. Whenever a codeword in subset $T_1$ is transmitted the encoder changes its state.

Subset $T_2$: $I > 243$.

The codewords in $T_2$ are chosen from the state-dependent, zero-disparity codeword set in such a way that a codeword $A(S_1,I > 243)$ is found from $A(S_0,I > 243)$ by reversing the direction of transmission of the codeword. We did not found an enumerative method for encoding or decoding the twelve codewords in set $T_2$, so that a direct look-up table is needed.
Decoding rules

At the receiver the codewords in \( A(S_1) \) can with simple hardware be mapped to codewords in \( A(S_0) \) with the same source representation. It thus suffices to have a decoding algorithm for codewords in state \( S_0 \). Due to the partitioning into two subsets two Pascal triangles are needed listed in table IV.

We conclude from this table that the look-up table has \( 21 + 14 = 35 \) non-zero entries. We further need for decoding an 8 bits full adder and additional hardware for distinguishing subsets \( T_0 \) and \( T_1 \) and coding of codewords in \( T_2 \).

Results

Using a computer the following statistical properties of the 8b10b code were calculated (see table V).

**TABLE V**

Statistical properties of 8b10b code

<table>
<thead>
<tr>
<th>sum</th>
<th>prob.</th>
<th>run-length</th>
<th>prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.066</td>
<td>1</td>
<td>.540</td>
</tr>
<tr>
<td>2</td>
<td>.199</td>
<td>2</td>
<td>.303</td>
</tr>
<tr>
<td>1</td>
<td>.286</td>
<td>3</td>
<td>.119</td>
</tr>
<tr>
<td>0</td>
<td>.264</td>
<td>4</td>
<td>.033</td>
</tr>
<tr>
<td>−1</td>
<td>.148</td>
<td>5</td>
<td>.005</td>
</tr>
<tr>
<td>−2</td>
<td>.036</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

average sum = 0.662  
sum variance = 1.534

The run-length of a code is defined as the number of consecutive like symbols. The efficiency \( E \) of the 8b10b code is according to eq. (16) and table I: \( 0.2398/(0.2 \times 1.534) = .78 \).

Table VI lists the main parameters of DC-constrained 8b10b and 16b20b codes that were reported in the literature (see also Tazaki\(^{22} \)).

**TABLE VI**

Main parameters of \( R = 8/10 \) DC-constrained codes

<table>
<thead>
<tr>
<th>reference</th>
<th>( N )</th>
<th>( T_{\text{max}} *) )</th>
<th>note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morizono(^{23})</td>
<td>10</td>
<td>10</td>
<td>**)</td>
</tr>
<tr>
<td>Shirota(^{24})</td>
<td>7</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Widmer(^{20,21})</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Parker(^{25})</td>
<td>6</td>
<td>5</td>
<td>***)</td>
</tr>
<tr>
<td>new</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

*) maximum run-length, **) simple zero-disparity code with 252 codewords, ***\) block code with codeword length \( n = 20. \)
Note that the new code has improved features with respect to the other codes with codeword length \( n = 10 \). The Parker code has the same main parameters, but due to the doubled codeword length it needs considerably more hardware for encoding and decoding.

6. Conclusions

We have presented a systematic approach for designing binary DC-balanced channel codes. DC-balance was achieved by imposing a constraint on the maximum unbalance in the number of positive and negative pulses within codewords. We derived recursion relations for determining the number of codewords satisfying a maximum unbalance constraint. We showed that the efficiency of the new codes is asymptotically optimal, i.e., for long codewords the information rate and sum variance approach the capacity and sum variance of maxentropic \( z \)-constrained sequences. Algorithms for encoding and decoding of maximum unbalance codewords were derived. The hardware needed for these algorithms grow polynomially with the codeword length, so that they are very useful when large codewords are applied. An example of a new 8b10b DC-balanced code assuming only six digital sum values was given.

REFERENCES