Markers to construct DC free \((d, k)\) constrained balanced block codes using Knuth's inversion


In this reported work, Knuth's balancing scheme, which was originally developed for unconstrained binary codewords is adapted. Presented is a simple method to handle the NRZ runlength constrained block codes corresponding to \((d, k)\) constrained NRZI sequences. A short marker violating the maximum runlength or \(k\) constraint is used to indicate the balancing point for Knuth's inversion. The marker requires fewer overhead bits and less implementation complexity than indexing the balancing point's address by mapping it onto a \((d, k)\) runlength constrained prefix, such as when applying Knuth's original scheme more directly. The new code construction may be attractive for future magnetic and especially optical recording schemes. In fact the current optical storage media, such as the CD, DVD and Blue Ray Disc, all attempt to achieve some reduction of low frequency components of the constrained codes, by exploiting a limited degree of freedom within the set of candidate \((d, k)\) words.

**Introduction:** Binary \((d, k)\) constrained codes have been used since the 1960s in magnetic recording and are currently very prominent in optical recording [1]. Here \(d\) denotes the minimum number of 0s and \(k\) the maximum number of 0s between any two 1s in the NRZI representation of the coded sequence. Hence there is a minimum runlength of \(d + 1\) same symbols and maximum runlength of \(k - 1\) same symbols in the NRZ representation of the binary runlength constrained channel sequence. By balancing the binary channel symbols \(0, 1\) as in [2] and mapping them onto bipolar symbols \((-1, +1)\), a balanced and hence DC free code with suppressed low frequency components can be constructed. Previously, investigations of DC free \((d, k)\) constrained codes focused mainly on short finite state machine codes for rotating head magnetic tape recorders as in [3], or medium length block codes as currently implemented in optical storage media [1]. Long block codes as in [4] may suffer from large error propagation. However, long block codes are attractive in order to more closely approach channel capacity and hence increase storage density. There is very little literature on constructing such long block codes. We now present a simple method, requiring fewer overhead bits and also less implementation complexity than the more direct application of Knuth's algorithm in [5].

**Code construction:** We modify and apply Knuth's balancing scheme [2], which was originally developed for unconstrained binary codewords. The main idea in Knuth's construction is to convert the original data sequence into a balanced sequence by inverting all symbols beyond a certain balancing index. This index is represented in a (balanced) prefix, which is also communicated. The receiver obtains the index from the prefix and retrieves the original data sequence by inverting all symbols beyond the balancing index. In the construction for balanced constrained sequences which is presented in this Letter, an alternative to the prefix method is proposed. Specifically, a short marker violating the maximum runlength constraint is now used to indicate the balancing point for Knuth's inversion. Note that while the \(d\) constraint is still critical for recording density and reliability, the \(k\) constraint increased in steps from \(k = 3\) in the 1960s when only simple tank oscillator electronic synchronisation circuitry was available, to \(k = 7\) in the 1970s and later \(k = 11\). So far there has been little interest in further increasing \(k\), since this will only yield an insignificant increase in practical recording density as can be shown by numerical evaluation of the information theoretic channel capacity as in e.g. [1]. With state of the art phase locked loops, occasional violations are tolerable. In fact, it can also be expected that in future synchronisation technology may be further improved. In this Letter we exploit occasional violations to create markers.

We use prior art such as in [1] or [6] to generate a high rate NRZI \((d, k)\) constrained block code. Represent the NRZ \((d + 1, k + 1)\) runlength constrained channel sequence \(x\) with

\[
x_1 \ldots x_j x_{j+1} \ldots x_n, x_i \in \{0, 1\}
\]

where \(j\) is the balancing index. Consequently, the sequence can be balanced by inverting the last \(n-j\) bits \(x_{j+1} \ldots x_n\). Knuth showed that every binary word has at least one such balancing index \(j\), where \(1 \leq j \leq n\). Let \(\tilde{x}_i\) be the binary complement of \(x_i\). We now insert a balanced marker sequence \(M = m_1 \ldots m_n\) of length \(z\) bits and transmit

\[
x_1 \ldots x_j m_1 \ldots m_n \tilde{x}_{j+1} \ldots \tilde{x}_n
\]

We want to extend the last run of the sequence \(x_1 \ldots x_j\) to create a maximum runlength constraint violation, i.e. a run of \(k+2\) bits or longer, without violating the minimum runlength of \(d+1\) channel symbols anywhere. To this end, we propose the following marker. Let \(y^+\) denote a run of length \(\mu\) same symbols \(y\) and set

\[
M = \frac{x_{j+1}^{+1} x_{j+2}^{+1} \ldots x_{j+z}^{+1}}{y^+} \quad \text{if } j = n \quad \text{or} \quad M = \frac{x_1^{+1} x_2^{+1} \ldots x_{d+2}^{+1}}{y^+} \quad \text{if } j < n
\]

If \(j = n\), then we use the first bit of the next word as \(\tilde{x}_{j+1}\). As a simple example for exposition, the set of NRZ candidate markers for \((d, k) = (1, 3)\) code is

\[
\{000011110011, 000011111100, 111100001100, 111100000011\}
\]

Note that \(M\) is balanced and that the overall length of the marker is \(z = 2k + 2d + 4\). The first maximum runlength violation which includes \(\tilde{x}_j\), ranges between \(k + 2\) and \(2k + 2\) symbols. Note that if \(x_j \neq x_{j+1}\), a second maximum runlength violation, of length \(k + d + 2\), occurs inside the marker. A third runlength violation, of length at most \(k + d + 2\), may occur where the marker \(M\) merges with the inverted portion of the sequence. However, the decoder only searches for the first violation. When the first maximum runlength violation is detected the decoder counts back \(k + 1\) bits from the end of this violated run to locate the start of the marker. It then removes the marker, inverts \(x_{j+1} \ldots x_n\) and uses prior art to decode and retrieve the data.

The proposed marker is a very simple one. More advanced markers can be designed for specific purposes, at the expense of a higher complexity. For example, constructions of variable-length markers with a shorter average length are possible by taking into account the lengths of the runs preceding and following the balancing index. Also, the marker construction may be optimised to bring down the maximum runlength violation. Note that for the proposed marker a run of length \(2k + 2\) can occur in the transmitted sequence, which may be considered as being too long. By extending the set of candidate markers the maximum value of this runlength violation may be set closer to \(k + 2\). However, such advanced markers are beyond the scope of this Letter, where we focus on the introduction of the marker concept and the simple design presented in (3).

**Results:** With the above marker construction, the overhead is \(2k + 2d + 4\) bits per codeword. We compare this to the previous construction, presented in [5], where Knuth's algorithm is directly applied to constrained sequences, which requires a fixed length interfix of \(d + 1\) bits as well as a prefix, of length dependent on \((d, k)\) and \(n\), as tabled in [5]. Note that the overhead of the marker construction does not depend on \(n\). Therefore, except for small values of \(n\), the proposed method based on the marker has less overhead as well as lower complexity. For example, if \((d, k) = (1, 3)\) the marker presented here represents less overhead if the prefix in [5] is longer than \((2k + 2d + 4) - (d + 1) = 12 - 2 = 10\) bits.

The power spectral densities for \((d, k) = (1, 3)\) coded sequences with markers corresponding to the proposed construction are compared to the \((d, k) = (1, 3)\) MFM (Miller code, [1]) in Figs. 1 and 2. In Fig. 1, the
influence of the codeword length is shown as well. In Fig. 2, it can be seen how the low frequency components are suppressed.

![Fig. 2](image_url)

**Fig. 2** Low frequency performance comparison for MFM (solid curve) and proposed construction (dotted curve), with \((d, k) = (l, 3)\), for 128 bit code-words as input to balancing

**Conclusion:** We have presented an efficient construction for long balanced and DC free \((d, k)\) constrained block codes. In the context of constrained codes, modifying Knuth’s original scheme by using markers instead of the original Knuth’s prefix – which for \((d, k)\) constraints furthermore mandates an additional interfix – may require considerably fewer overhead symbols and lower implementation complexity.

---

© The Institution of Engineering and Technology 2012
19 April 2012
doi: 10.1049/el.2012.1220

H.C. Ferreira and C.H. Heymann (Department of Electrical and Electronic Engineering Science, University of Johannesburg, APK Campus, Box 524, Auckland Park, 2006, South Africa)
E-mail: hcferreira@uj.ac.za

J.H. Weber (Delft University of Technology, The Netherlands)

K.A.S. Immink (Turing Machines, Rotterdam, The Netherlands)

**References**


