Polarity-Balanced Codes

Jos H. Weber
Delft University of Technology
The Netherlands
j.h.weber@tudelft.nl

Kees A.S. Immink
Turing Machines Inc.
The Netherlands
immink@turing-machines.com

Paul H. Siegel
UC San Diego
USA
psiegel@ucsd.edu

Theo G. Swart
University of Johannesburg
South Africa
tgswart@uj.ac.za

***Abstract*—Balanced bipolar codes consist of sequences in which the symbols '-1' and '$+1$' appear equally often. Several generalizations to larger alphabets have been considered in literature. For example, for the q-ary alphabet $\{-q+1, -q+3, \ldots, q-1\}$, known concepts are symbol balancing, i.e., all alphabet symbols appear equally often in each codeword, and charge balancing, i.e., the symbol sum in each codeword equals zero. These notions are equivalent for the bipolar case, but not for $q > 2$. In this paper, a third perspective is introduced, called polarity balancing, where the number of positive symbols equals the number of negative symbols in each codeword. The minimum redundancy of such codes is determined and a generalization of Knuth's celebrated bipolar balancing algorithm is proposed.**

I. Introduction

Bipolar (or binary) balanced sequences are sequences over the alphabet $\{-1, +1\}$ (or $\{0, 1\}$) in which both alphabet symbols appear equally often. Several extensions of the balancing notion to alphabets with more than two elements have been considered in the literature. Consider, for example, the symmetric alphabets $\mathcal{A}_q = \{-q+1, -q+3, -q+5, \ldots, q-3, q-1\}$ that arise in the context of pulse amplitude modulation (PAM), e.g., $\mathcal{A}_4 = \{-3, -1, +1, +3\}$, $\mathcal{A}_5 = \{-4, -2, 0, +2, +4\}$. It is said that a code is *symbol-balanced* (SB) over $\mathcal{A}_q$ if, in each codeword, all q alphabet symbols appear equally often. A *charge-balanced* (CB) code is one in which the sum of the symbols in each codeword is zero. In this paper, we introduce *polarity-balanced* (PB) codes, for which, in every codeword, the number of positive symbols equals the number of negative symbols. For q odd, this definition does not constrain the number of zero symbols.

It is easy to see that for $q = 2$, i.e., for bipolar sequences of even length n, these three notions of being "balanced" are completely equivalent. For $q = 3$, i.e., for sequences over the alphabet $\{-2, 0, +2\}$, the notions of CB and PB are equivalent, but the SB sequences form a proper subset of the set of CB and PB sequences. For example, the sequence $(-2, -2, +2, 0, -2, +2, +2, +2, -2)$ of length 9 is CB and PB, but not SB. For $q \geq 4$, all three notions are mutually distinct. Any sequence which is SB is also CB and PB, but there do exist sequences which are PB but not CB (e.g., $(-3, -1, +1, +1)$ over $\mathcal{A}_4$) and sequences which are CB but not PB (e.g., $(+3, -1, -1, -1)$ over $\mathcal{A}_4$). Furthermore, there exist sequences which are both CB and PB but not SB (e.g., $(-3, -3, +3, +3)$ over $\mathcal{A}_4$).

Balanced codes have found applications in digital communications and data storage technology [6]. They have been widely studied in the literature, particularly for the binary case, e.g., [1], [3], [4], [8], [14], [15]. Some constructions also take into account error correction capabilities, e.g., [2], [11], [17], [19]. Results for non-binary alphabets have been presented for the SB and CB cases, albeit under different (or no specific) names, e.g., [9], [10] (SB) and [5], [12], [13], [16] (CB). To the best of our knowledge, the PB concept for non-binary sequences is new and has not been studied before. It is of particular interest for applications which demand a balancing of positive and negative symbols, possibly in combination with a charge constraint. In this paper, we determine the number of q-ary PB sequences of length n. From this, we derive expressions for the minimum redundancy of PB codes, which are compared to the corresponding expressions for SB and CB codes.

A celebrated method to generate and decode bipolar balanced sequences of even length n was presented by Knuth [8]. The key idea is to invert the first z symbols of the information sequence such that the resulting sequence is balanced. Knuth showed that it is always possible to find at least one such balancing index z. By communicating the value of z through a (balanced) prefix, decoding can be performed by inverting the first z symbols of the coded sequence. The redundancy of this elegant method is roughly $\log_2(n)$, which is about twice the minimum and can thus be considered as a price to be paid for simplicity. In this paper, we extend Knuth's method, which assumes bipolar sequences, to larger alphabets for the polarity balancing case.

The rest of this paper is organized as follows. In Section II, some definitions and preliminaries are presented. Then, in Section III, we present expressions for the maximum size of q-ary polarity-balanced codes of length n as well as the minimal redundancy of these codes. In Section IV, we describe a Knuth-like construction for polarity-balanced codes. Finally, the paper is concluded in Section V.

II. Preliminaries

A. Alphabets and Balancing

In Section I, we introduced the alphabet

$$\mathcal{A}_q = \{-q+1, -q+3, -q+5, \ldots, q-3, q-1\},$$

where $q \geq 2$. A sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in (\mathcal{A}_q)^n$, with n being a positive integer which is even if q is even, is *polarity balanced* (PB) if the number of positive symbols in

$\mathbf{x}$ equals the number of negative symbols, i.e.,

$$|\{i : x_i > 0\}| = |\{i : x_i < 0\}|.$$

Note that in case q is even and n is odd, there exist no sequences satisfying the desired property. Hence, throughout this paper, we will assume that in case q is even, n is also even.

When studying q-ary balanced codes, other alphabets than $\mathcal{A}_q$ have also been considered in the literature, a prominent example being $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. The choice of the alphabet may influence the balancing notion, in particular for charge balancing. On the other hand, symbol balancing is clearly independent of the symbol representation. The number of symbol-balanced sequences of a certain length n will be the same for any q-ary alphabet. The same conclusion is valid for polarity balancing, as long as we divide the alphabet symbols into two classes of equal size, with one neutral symbol in case q is odd. Therefore, without loss of generality, we will assume that the code alphabet is $\mathcal{A}_q$ throughout the rest of this paper.

B. Codes and Redundancy

A q-ary polarity-balanced code of length n is a set of q-ary polarity-balanced sequences of length n. If the number of codewords is M, the the code's redundancy is $r = n - \log_q M$. The number of q-ary polarity-balanced sequences of length n is denoted by $M(n, q)$. Hence, the minimum redundancy of any q-ary polarity-balanced code of length n is

$$r(n, q) = n - \log_q M(n, q). \tag{1}$$

C. Gaussian Approximation

A tool which we will use in the next section is the following Gaussian approximation technique. We consider the symbols x_i in a sequence $\mathbf{x}$ as n independent random variables which are uniformly drawn from the alphabet $\mathcal{A}_q$. We are interested in the distribution of the sum $\sum_{i=1}^{n} \phi(x_i)$, where ϕ is a function mapping symbols from $\mathcal{A}_q$ to real numbers, which has the property that the possible outcomes of the sum form a set of consecutive integer numbers. Then, by the Central Limit Theorem, the probability that this sum takes the integer value s is approximately

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{s-\mu}{\sigma})^2},$$

with mean

$$\mu = nE[\phi(x)] = \frac{n}{q} \sum_{j=0}^{q-1} \phi(q-1-2j) \tag{2}$$

and variance

$$\begin{aligned} \sigma^2 &= n(E[(\phi(x))^2] - (E[\phi(x)])^2) \\ &= n\left(\left(\frac{1}{q}\sum_{j=0}^{q-1}(\phi(q-1-2j))^2\right) - \left(\frac{\mu}{n}\right)^2\right). \end{aligned} \tag{3}$$

Hence, the number of q-ary sequences of length n with $\sum_{i=1}^{n} \phi(x_i)$ equal to s is approximately

$$q^n \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{s-\mu}{\sigma})^2}. \tag{4}$$

Note that for fixed n and q this expression is maximum if s is equal to μ, which leads to a minimum redundancy of

$$\log_q \sigma + \frac{1}{2}\log_q 2\pi.$$

III. Minimum Redundancy

In this section, we consider the cardinality of q-ary polarity-balanced codes. From the cardinality we derive asymptotic expressions for the minimum redundancy.

When calculating the number of q-ary polarity-balanced sequences of length n, we distinguish between the cases q is even and q is odd, since in the latter case we should take into account the fact that the code alphabet contains the symbol '0' which is of indeterminate polarity. The results are presented in the next theorems, while expressions for the minimum redundancy of PB codes are given in the subsequent corollaries.

Theorem 1. *For any even q and even n, it holds that*

$$\begin{aligned} M(n, q) &= \binom{n}{n/2}\left(\frac{q}{2}\right)^n \qquad (5) \\ &\approx q^n \sqrt{\frac{2}{\pi n}}. \qquad (6) \end{aligned}$$

Proof. The equality (5) follows by observing that there are $\binom{n}{n/2}$ ways to create a balanced polarity pattern over n positions and that for each such pattern we have $q/2$ symbol options for every positions. The approximation can be obtained by applying the Gaussian approximation technique discussed in Subsection II-C. Choosing the function ϕ to be

$$\phi(x) = \begin{cases} -\frac{1}{2}, & \text{if } x < 0, \\ +\frac{1}{2}, & \text{if } x > 0, \end{cases} \tag{7}$$

it follows that the number of q-ary sequences $\mathbf{x}$ of length n with $\sum_{i=1}^{n} \phi(x_i) = s$ is approximately equal to (4) with mean

$$\mu = \frac{n}{q}\sum_{j=0}^{q-1}\phi(q-1-2j) = 0 \tag{8}$$

(from (2) and (7)) and variance

$$\sigma^2 = n\left(\frac{1}{q}\sum_{j=0}^{q-1}(\phi(q-1-2j))^2\right) = \frac{n}{4} \tag{9}$$

(from (3), (7), and (8)). Note that polarity-balanced sequences are characterized by the fact that $s = 0$, and thus substitution of this value in (4), with $\mu = 0$ and $\sigma^2 = n/4$, gives (6). $\square$

Corollary 2. *For any even q and even n, it holds that*

$$r(n, q) \approx \frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{\pi}{2}.$$

Proof. This result follows from (1) and Theorem 1. $\square$

Theorem 3. *For any n and odd q, it holds that*

$$M(n,q) = \sum_{j=0}^{\lfloor n/2 \rfloor} \frac{n!}{j!j!(n-2j)!} \left(\frac{q-1}{2}\right)^{2j} \quad (10)$$

$$\approx q^n \sqrt{\frac{q}{2\pi n(q-1)}}. \quad (11)$$

Proof. The number of q-ary polarity-balanced sequences of length n with j positive symbols, j negative symbols, and thus $n-2j$ neutral symbols, is $\frac{n!}{j!j!(n-2j)!}\left(\frac{q-1}{2}\right)^{2j}$, since there are $\frac{n!}{j!j!(n-2j)!}$ ways to create the positive/negative/neutral pattern over n positions and for each such pattern we have $(q-1)/2$ symbol options for every non-neutral position. Summing over all possible values of j shows (10).

In order to obtain a simple expression for large values of n, we again use the Gaussian approximation technique introduced in Subsection II-C. Proceeding as in the proof of Theorem 1, while replacing the function ϕ by

$$\phi(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ +1, & \text{if } x > 0, \end{cases} \quad (12)$$

giving mean

$$\mu = \frac{n}{q} \sum_{j=0}^{q-1} \phi(q-1-2j) = 0 \quad (13)$$

(from (2) and (12)) and variance

$$\sigma^2 = n\left(\frac{1}{q}\sum_{j=0}^{q-1}(\phi(q-1-2j))^2\right) = \frac{n(q-1)}{q} \quad (14)$$

(from (3), (12) and (13)), we obtain (11). $\square$

Corollary 4. *For any n and odd q, it holds that*

$$r(n,q) \approx \frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{2\pi(q-1)}{q}.$$

Proof. This result follows from (1) and Theorem 3. $\square$

Corollaries 2 and 4 provide expressions for the minimum redundancy of polarity-balanced codes. The corresponding expression for symbol-balanced codes [10] is

$$\frac{q-1}{2}\log_q n + \frac{q-1}{2}\log_q 2\pi - \frac{q}{2},$$

while the corresponding expression for charge-balanced codes [5] is

$$\frac{1}{2}\log_q n + \frac{1}{2}\log_q \frac{\pi(q^2-1)}{6}.$$

Note that for fixed values of q and large values of n, the minimum redundancies of polarity-balanced codes and charge-balanced codes are roughly the same, while the minimum redundancy of symbol-balanced codes is roughly $q-1$ times as large.

IV. Construction of Polarity-Balanced Codes

In the previous section we have determined expressions for the number $M(n,q)$ of q-ary polarity-balanced sequences of length n. From these expressions we calculated the minimum required code redundancy. However, the lists of balanced words come with little structure. Applying table look-up is only feasible for small codes, but for practical implementation of larger codes, we need simple encoding and decoding algorithms. Knuth presented such an algorithm for the case $q = 2$, i.e., for binary/bipolar balanced codes [8]. Here, we will propose an extension of this algorithm to non-binary polarity-balanced codes.

The proposed method takes an approach similar to the original Knuth construction. We make simple and reversible modifications to a q-ary information sequence $\mathbf{u}$ of length k to obtain a q-ary balanced sequence $\mathbf{x}$ of the same length. Next, we create a q-ary balanced prefix $\mathbf{p}$ of length p, which uniquely identifies the modifications. The q-ary balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$ of length $n = p + k$ is then transmitted or stored. The receiver retrieves the modifications from the prefix and applies these in reverse on $\mathbf{x}$ to obtain the original $\mathbf{u}$.

The construction is nice and simple, but not optimal with respect to redundancy. Note that all codewords consist of two parts which are both balanced, and thus words which are balanced overall, but not within these parts, are excluded. Hence, simplicity comes at a price of increased redundancy. In order to still keep the redundancy as small as possible within the construction framework, we should minimize the prefix length p. Since the prefix is much shorter than the information sequence, we will assume that encoding and decoding of the prefix can be done by table look-up or another minimum redundancy achieving method. Let the number of different prefixes required to uniquely identify the modifications be denoted by P. Ignoring balancing, the number of q-ary symbols needed to represent the prefix is thus

$$p' = \log_q P, \quad (15)$$

which we will call the *unbalanced redundancy*. The actual prefix length will be (a little bit) larger, since the prefix needs to be balanced as well. It should be chosen as the smallest integer p such that

$$M(p,q) \geq P. \quad (16)$$

The analysis from the previous section shows that, for fixed q, the extra redundancy to make the prefix balanced is in the order of $\log p'$, i.e.,

$$p = p' + O(\log p').$$

Hence, for rough evaluation purposes, the unbalanced redundancy p', which is easily determined by (15), may serve as a satisfactory approximation of the actual redundancy p, which requires the more cumbersome computation from (16).

Before starting the descriptions of the constructions, we introduce some more notation. The real sum of all symbols in a sequence $\mathbf{y}$ is denoted by $\mathrm{Sum}(\mathbf{y})$, i.e.,

$$\mathrm{Sum}(\mathbf{y}) = \sum_i y_i.$$

Further, let $S_j(\mathbf{y})$ denote the number of appearances of the alphabet symbol j in $\mathbf{y}$, i.e.,

$$S_j(\mathbf{y}) = |\{i : y_i = j\}$$

for any alphabet symbol j.

A. Knuth's Construction

We start by stating Knuth's original construction for bipolar codes [8], as a reference. For any information sequence $\mathbf{u}$ of even length k and any $j \in \{0, 1, \ldots, k\}$, let $\mathbf{u}'_j$ denote the sequence $\mathbf{u}$ with the first j symbols multiplied by -1. A *balancing index* is a number z for which $\mathbf{u}'_z$ is balanced.

Knuth Encoding Procedure

1) Determine a balancing index $z \in \{0, 1, \ldots, k-1\}$ for the information sequence $\mathbf{u}$.
2) Multiply the first z symbols of $\mathbf{u}$ by -1 to obtain the balanced sequence $\mathbf{x}$.
3) Map z to a unique balanced prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

Knuth Decoding Procedure

1) Retrieve the balancing index z from $\mathbf{p}$.
2) Multiply the first z symbols of $\mathbf{x}$ by -1 to retrieve $\mathbf{u}$.

Proof. It is easy to see that the operation in the encoding procedure is properly reversed in the decoding procedure. Hence, we only need to show that for every sequence $\mathbf{u}$ of length k there exists at least one $z \in \{0, 1, \ldots, k-1\}$ such that $\mathbf{u}'_z$ is balanced, i.e., $\mathrm{Sum}(\mathbf{u}'_z) = 0$. This immediately follows from combining the following observations.

1) $\mathrm{Sum}(\mathbf{u}'_0)$ is even.
2) $\mathrm{Sum}(\mathbf{u}'_j) = \mathrm{Sum}(\mathbf{u}'_{j-1}) \pm 2$ for all $j \in \{1, 2, \ldots, k\}$.
3) $\mathrm{Sum}(\mathbf{u}'_k) = -\mathrm{Sum}(\mathbf{u}'_0)$.

□

Since there are k possible values for z, the redundancy, i.e., the length p of the prefix, is a little bit more than $p' = \log_2 k$.

Example 1. For the bipolar sequence

$$\mathbf{u} = (+1, -1, +1, +1, +1, +1)$$

of length 6, encoding goes as follows.

1) Find the balancing index to be $z = 4$.
2) Invert the first 4 positions of $\mathbf{u}$, i.e.,

$$\mathbf{x} = (-1, +1, -1, -1, +1, +1).$$

3) Uniquely map the balancing index 4 to one of the six balanced sequences of length four, e.g.,

$$\mathbf{p} = (+1, -1, -1, +1).$$

Then the balanced transmitted/stored sequence is

$$\mathbf{c} = (\mathbf{p}, \mathbf{x}) = (+1, -1, -1, +1, -1, +1, -1, -1, +1, +1).$$

B. Polarity-Balanced Code Construction

Knuth's original method for generating balanced binary sequences can be adapted to generate q-ary polarity-balanced sequences. This is rather straightforward, although there is a snag if q is odd. In this case, the number of zero-valued symbols in $\mathbf{u}$ may be of different parity than the length k, which results in an odd number of non-zero (either positive or negative) symbols. Since the value zero is (polarity-)neutral, i.e., neither positive nor negative, inversion of any number of symbols in $\mathbf{u}$ will not lead to a polarity-balanced sequence in such a situation. We will solve this by introducing an offset in case q is odd. We propose the following algorithm for sequences over $\mathcal{A}_q$, where $\oplus_{2q}$ denotes addition over the integer numbers, with a reduction modulo $2q$ such that the final outcome is in $\mathcal{A}_q$.

PB Encoding Procedure

1) If q is odd, then determine a symbol a in $\mathcal{A}_q$ such that $S_a(\mathbf{u})$ has the same parity as the length k of $\mathbf{u}$, i.e., $S_a(\mathbf{u})$ and k are either both even or both odd.
2) If q is odd, then compute $\mathbf{u}' = \mathbf{u} \oplus_{2q} (-\mathbf{a})$, where $\mathbf{a} = (a, a, \ldots, a)$ is of length k. If q is even, then $\mathbf{u}' = \mathbf{u}$.
3) Determine a polarity balancing index $z \in \{0, 1, \ldots, k - 1\}$ for $\mathbf{u}'$.
4) Multiply the first z positions of $\mathbf{u}'$ by -1 to obtain the PB sequence $\mathbf{x}$.
5) Map z (if q is even) or (a, z) (if q is odd) to a unique PB prefix $\mathbf{p}$.

Then transmit or store the balanced codeword $\mathbf{c} = (\mathbf{p}, \mathbf{x})$.

PB Decoding Procedure

1) Retrieve the balancing index z from $\mathbf{p}$.
2) Multiply the first z positions of $\mathbf{x}$ by -1 to retrieve $\mathbf{u}$ (if q is even) or $\mathbf{u}'$ (if q is odd).
3) If q is odd, then retrieve a from the prefix $\mathbf{p}$ and compute $\mathbf{u} = \mathbf{u}' \oplus_{2q} \mathbf{a}$.

Proof. It is easy to see that the operations in the encoding procedure are properly reversed in the decoding procedure. Hence, we only need to show the existence of (i) a suitable offset a (in case q odd) and (ii) a suitable polarity balancing index z.

(i) The existence of a can be demonstrated by supposing it does not exist and then deriving a contradiction. If q and k are odd, then $S_j(\mathbf{u})$ is odd for at least one symbol $j \in \mathcal{A}_q$, since all of them being even would imply that $k = \sum_i S_i(\mathbf{u})$ is even. If q is odd and k is even, then $S_j(\mathbf{u})$ is even for at least one $j \in \mathcal{A}_q$, since all of them being odd would imply that $k = \sum_i S_i(\mathbf{u})$, a summation of an odd number of odd terms, is odd.

(ii) The existence of z follows by a similar argument as for the Knuth algorithm. Let $\mathbf{u}'_j$ denote the sequence $\mathbf{u}'$ with the first j symbols multiplied by -1 and let ϕ be defined as in (12). For a PB balancing index z, it must hold that $\mathrm{Sum}(\phi(\mathbf{u}'_j)) = 0$. The existence of a PB balancing index follows by combining the following observations.

1) $\mathrm{Sum}(\phi(\mathbf{u}'_0))$ is even, since the number of non-zero symbols in $\mathbf{u}'$ is even.

2) $\mathrm{Sum}(\phi(\mathbf{u}'_j)) = \mathrm{Sum}(\phi(\mathbf{u}'_{j-1})) + c$ for all $j \in \{1, 2, \ldots, k\}$, where $c \in \{-2, 0, +2\}$.
3) $\mathrm{Sum}(\phi(\mathbf{u}'_k)) = -\mathrm{Sum}(\phi(\mathbf{u}'_0))$.

$\square$

Since there are k possible values for z and q possible values for a, we have $p' = \log_q k$ if q is even and $p' = 1 + \log_q k$ if q is odd.

Example 2. Let $q = 5$. For the sequence

$$\mathbf{u} = (+4, +4, -2, 0, 0, 0, 0) \in (\mathcal{A}_5)^7,$$

encoding goes as follows.

1) Since $q = 5$ and $k = 7$ are odd, identify '-2' as the symbol a with an odd number of appearances in $\mathbf{u}$.
2) Subtract (modulo 10) the value -2 from every symbol in $\mathbf{u}$, resulting in
$$\mathbf{u}' = (-4, -4, 0, +2, +2, +2, +2).$$
3) Find the PB index z to be 6.
4) Multiply the first 6 positions of $\mathbf{u}'$ by -1 to obtain
$$\mathbf{x} = (+4, +4, 0, -2, -2, -2, +2).$$
5) Uniquely map $(a, z) = (-2, 6)$ to one of the PB sequences of length 4, e.g.,
$$\mathbf{p} = (+2, 0, 0, -4).$$

Then the balanced transmitted/stored sequence is

$$\mathbf{c} = (+2, 0, 0, -4, +4, +4, 0, -2, -2, -2, +2).$$

It should be mentioned that this example is misleading in the sense that the redundancy appears to be relatively large, which is due to the fact that extremely short data blocks were used. Four redundant symbols are used for seven data symbols. However, for long codes, the redundancy is only logarithmic in the length of the data block. It is roughly twice the corresponding minimum redundancy derived in Section III.

For the binary case, modifications of Knuth's method have been presented to close the factor of two gap between the redundancy of the original Knuth algorithm and the minimum redundancy, while maintaining sufficient simplicity to enable feasible implementations. In [7], this is done by a more efficient (variable-length) encoding of the prefix. In [18], minimum redundancy is achieved by exploiting the fact that many data sequences have more than one possible balancing index, thus allowing to encode auxiliary data through the choice of the index. It is an interesting research challenge to investigate whether such techniques are also applicable in non-binary cases.

V. Conclusions

In this paper we have considered balancing of q-ary sequences from the polarity perspective, in contrast to the symbol and charge perspectives. We have derived (approximate) expressions for the number of such sequences of a fixed length and for the minimum redundancy. Furthermore, we have presented a q-ary scheme to generate polarity-balanced sequences, in the spirit of the bipolar Knuth algorithm. This scheme allows for simple encoding and decoding, at the price of a redundancy which is twice the minimum required redundancy.

Further discussions on the various balancing perspectives, including an analysis and construction method for codes which satisfy both the polarity and the charge constraints, are provided in [20].

Acknowledgment

P.H. Siegel was supported in part by the Center for Magnetic Recording Research at the University of California, San Diego, and by NSF grant CCF-1116739. Portions of this work were done while J.H. Weber and P.H. Siegel were visiting the Centre Interfacultaire Bernoulli (CIB) at the Ecole Polytechnique Fédérale de Lausanne (EPFL) in Lausanne, Switzerland, in the autumn of 2011.

References

[1] S. Al-Bassam and B. Bose, "On balanced codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 406–408, Oct. 1993.
[2] S. Al-Bassam and B. Bose, "Design of efficient error-correcting balanced codes," *IEEE Trans. Comp.*, vol. 42, no. 10, pp. 1261–1266, Oct. 1993.
[3] S. Al-Bassam and B. Bose, "Design of efficient balanced codes," *IEEE Trans. Comp.*, vol. 43, no. 3, pp. 362–365, Mar. 1994.
[4] N. Alon, E. E. Bergmann, D. Coppersmith and A. M. Odlyzko, "Balancing sets of vectors," *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 129–130, Jan. 1988.
[5] R. M. Capocelli, L. Gargano and U. Vaccaro, "Efficient q-ary immutable codes," *Discrete Applied Mathematics*, vol. 33, pp. 25–41, 1991.
[6] K. A. S. Immink, *Codes for Mass Data Storage Systems,* Second Edition, Shannon Foundation Publishers, Eindhoven, The Netherlands, 2004.
[7] K. A. S. Immink and J. H. Weber, "Very efficient balanced codes," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 188–192, Feb. 2010.
[8] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.
[9] R. Mascella and L. G. Tallini, "On symbol permutation invariant balanced codes," *Proc. IEEE Int. Symp. Inform. Theory*, Adelaide, Australia, pp. 2100–2104, Sept. 4–9, 2005.
[10] R. Mascella and L. G. Tallini, "Efficient m-ary balanced codes which are invariant under symbol permutation," *IEEE Trans. Comp.*, vol. 55, no. 8, pp. 929–946, Aug. 2006.
[11] A. Mazumdar, R. M. Roth, and P. O. Vontobel, "On linear balancing sets," *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, South Korea, pp. 2699–2703, June 28–July 3, 2009.
[12] D. Pelusi, L. G. Tallini, and B. Bose, "On m-ary balanced codes with parallel decoding," *Proc. IEEE Int. Symp. Inform. Theory*, Austin, Texas, USA, pp. 1305–1309, June 13–18, 2010.
[13] T. G. Swart and J. H. Weber, "Efficient balancing of q-ary sequences with parallel decoding," *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, South Korea, pp. 1564–1568, June 28–July 3, 2009.
[14] L. G. Tallini and B. Bose, "Balanced codes with parallel encoding and decoding," *IEEE Trans. Comp.*, vol. 48, no. 8, pp. 794–814, Aug. 1999.
[15] L. G. Tallini, R. M. Capocelli and B. Bose, "Design of some new efficient balanced codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 790–802, May 1996.
[16] L. G. Tallini and U. Vaccaro, "Efficient m-ary balanced codes," *Discrete Applied Mathematics*, vol. 92, pp. 17–56, 1999.
[17] H. van Tilborg and M. Blaum, "On error-correcting balanced codes," *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1091–1095, Sept. 1989.
[18] J. H. Weber and K. A. S. Immink, "Knuth's balanced codes revisited," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1673–1679, Apr. 2010.
[19] J. H. Weber, K. A. S. Immink, and H.C. Ferreira, "Error-correcting balanced Knuth codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 82–89, Jan. 2012.
[20] J. H. Weber, K. A. S. Immink, P. H. Siegel, and T. G. Swart,"Perspectives on Balanced Sequences," submitted to *IEEE Trans. Inf. Theory*, Sept. 2012; available via arXiv:1301.6484 (http://arxiv.org/abs/1301.6484).