On Efficient Second-Order Spectral-Null Codes using Sets of m_1-Balancing Functions

Raffaele Mascella*, Danilo Pelusi*, Laura Pezza†, Samir Elmougy‡, Luca G. Tallini* and Bella Bose§
*Dip. di Scienze della Comunicazione, Università degli Studi di Teramo, Teramo, Italy.
Email: {rmascella,dpelusi,ltallini}@unite.it
†Dip. Me.Mo.Mat., Università degli Studi di Roma "La Sapienza", Roma, Italy.
Email: pezza@dmmm.uniroma1.it
‡Dept. of Computer Science, King Saud University, Riyadh - Saudi Arabia.
E-mail: mougy@ksu.edu.sa
§School of EECS, Oregon State University, Corvallis, OR, USA.
Email: bose@eecs.orst.edu

***Abstract*—A new efficient coding scheme is given for second-order spectral-null (2-OSN) codes. The new method applies the Knuth's optimal parallel decoding scheme for balanced (i.e., 1-OSN) codes to the random walk method introduced by Tallini and Bose to design 2-OSN codes. If $k \in \mathbb{N}$ is the length of a 1-OSN code then the new 2-OSN coding scheme has length $n = k+r \in \mathbb{N}$ with an extra redundancy of $r \gtrsim 2\log_2 k + (1/2)\log_2\log_2 k - 0.674$ check bits. The whole coding process requires $O(n \log n)$ bit operations and $O(n)$ bit memory elements.**

***Index Terms*—high order spectral null codes, balanced codes, Knuth's complementation method, parallel decoding scheme, optical and magnetic recording.**

I. Introduction

Let $\Phi = \{-1, +1\}$ be the bipolar alphabet. The set of qth-order spectral-null words $S(n,q) \subseteq \Phi^n$ is defined as [4], [11], [5] $(Z = z_1 z_2 \ldots z_n)$

$$S(n,q) = \{Z \in \Phi^n : m_i(Z) = 0, \forall i = 0, 1, \ldots, q-1\} \quad (1)$$

where $m_i(Z) \stackrel{\text{def}}{=} \sum_{j=1}^{n} z_j j^i$ is the ith moment (or the m_i-weight) of the word Z, and sums and products are over the real numbers. Any word in $S(n,q)$ is called qth-order spectral-null word (briefly, a q-OSN word). In the following, if $m_i(Z) = 0$ then we will say that the word Z is m_i-balanced. A binary code $\mathcal{C}$ is a qth-order spectral-null code with $\tilde{k}$ information bits and length n (briefly, a q-OSN$(n, \tilde{k})$ code) if, and only if

1) $\mathcal{C}$ is a subset of $S(n,q)$, and
2) $\mathcal{C}$ has, say exactly, $2^{\tilde{k}}$ codewords.

When $q = 1$, these codes coincide with the so-called balanced or DC-free block codes [4], [6], [1], [2], [11], [13], [14], [16], [8], [12], [9], [10]. For values of q greater than 1, the q-OSN$(n, \tilde{k})$ codes are considered for digital recording; these codes are useful in achieving a better rejection of the low frequency components of the transmitted signal and enhancing the error correction capability of codes used in partial-response channels [5], [11]. The q-OSN codes can be also considered over the binary alphabet $\mathbb{Z}_2 = \{0, 1\}$ [15]. In fact, by replacing the symbol -1 with 0 and $+1$ with 1, the set $S(n,q)$ becomes equivalent to the set $S'(n,q) \subseteq \mathbb{Z}_2^n$:

$$S'(n,q) \stackrel{\text{def}}{=} \left\{ Z \in \mathbb{Z}_2^n : m_i(Z) = \frac{1}{2}\sum_{j=1}^{n} j^i, \forall i \in [0, q-1] \right\}.$$

Since $S(n,q)$ and $S'(n,q)$ are equivalent, in the rest of the paper $S(n,q)$ is used for $S'(n,q)$. For example,

$$S(n,2) = \left\{ Z \in \mathbb{Z}_2^n : m_0(Z) = \sum_{j=1}^{n} z_j = \frac{n}{2} \right. \\ \left. \text{and} \quad m_1(Z) = \sum_{j=1}^{n} z_j j = \frac{n(n+1)}{4} \right\} \quad (2)$$

and n is not a multiple of 4 if, and only if, $S(n,2) = \emptyset$ [15].

The code design problem is to convert the information words into qth-order spectral-null words using the minimum possible redundancy. For $q = 2$, this minimum redundancy is $r_{min}(k) = 2\log_2 n - 1.141$ [15]. Also, the conversion should be done so that the encoding and decoding processes are as computationally simple as possible.

In this paper we are concerned with designing 2-OSN codes whose encoding and decoding functions can be computed combining the Knuth's optimal parallel decoding scheme for balanced codes given in [1], [14], [9] with the random walk method for 2-OSN codes in [15], [7], [18]. Here we assume that the information is already m_0-balanced into words of length k (i.e., belongs to $S(k,1)$) by means of a 1-OSN$(k, \tilde{k})$ code. Given $k, M_0, M_1 \in \mathbb{N}$, let

$$S^k_{M_0} = \left\{ X \in \mathbb{Z}_2^k : m_0(X) = M_0 \right\} \subseteq \mathbb{Z}_2^k, \quad \text{and}$$
$$S^k_{M_0,M_1} = \left\{ X \in \mathbb{Z}_2^k : m_0(X) = M_0, m_1(X) = M_1 \right\} \subseteq S^k_{M_0}.$$

Now, given a m_0-balanced word $X = x_1 x_2 \ldots x_k \in S^k_{\lfloor k/2 \rfloor}$ (i.e., such that $m_0(X) = \lfloor k/2 \rfloor$), the idea is to "walk randomly" towards $X^R = x_k x_{k-1} \ldots x_1$ by exchanging adjacent bits, until a word $X^{(d_{h_b})}$ is reached, where this word satisfies a specific property for its 2nd moment. Notice that permutations between bits in the word X do not alter the value of its 1st moment, while its 2nd moment may change with a variation of $+1$, -1, or 0.

A family of sets of check symbols is built so that each set of check symbols identifies a specific step d_h of the random walk. As such, each set of check symbols defines an encoding function f_h. And, as the main theorem states (Theorem 3), there always exists at least one step d_{h_b} such that $X^{(d_{h_b})} = f_{h_b}(X)$ concatenated with one of the check words identifying f_{h_b}, becomes a 2-OSN word. As in [14] for 1-OSN codes, such indices $d_{h_b} = d_{h_b}(X)$ are referred to as the 2-OSN balancing indices of X. Now, the check word $C = C(X) \in S^r_{\lceil r/2 \rceil}$, $r \in \mathbb{N}$, is appended to obtain a $n = k + r$ bit codeword $\mathcal{E}_2(X) = X^{(d_{h_b})}C(X) \in \mathbb{Z}_2^n$ as encoding of X, with n a multiple of 4. Such check C must be chosen so that

1) the codeword $Y = \mathcal{E}_2(X)$ is m_0-balanced (that is, $m_0(\mathcal{E}_2(X)) = m_0\big(X^{(d_{h_b})}\big) + m_0(C) = n/2$),
2) the codeword $Y = \mathcal{E}_2(X)$ is m_1-balanced (that is, $m_1(\mathcal{E}_2(X)) = m_1\big(X^{(d_{h_b})}C\big) = n(n+1)/4$), thus being a 2nd-order spectral-null word, and
3) the original information word X can be recovered from $X^{(d_{h_b})}$ and C.

Here we start with 1-OSN words with either even or odd length, k, which are encoded into 2-OSN words whose length, n, is a multiple of 4 (for simplicity). The coding scheme given in [1] and [14] for balanced code design is adapted to fit the random walk coding scheme given in [15] for 2-OSN codes. The combination of the two methods gives efficient code designs for any value of the parameters k and $r = n-k$, provided that

$$\frac{k(k-1)}{2} \leq \binom{r}{\lceil r/2 \rceil} - 1. \tag{3}$$

Note that (3) implies $r \gtrsim 2\log_2 k + (1/2)\log_2\log_2 k - 0.674$ because of Stirling's approximation. In the full paper we show that the whole coding process can be implemented with $O(n \log n)$ bits operations and using $O(n)$ memory bits.

II. The proposed coding scheme

The main idea of the code design is to convert a balanced data word into an "almost 2-OSN" word, using an appropriate function from a set of "m_1-balancing functions", and append a check symbol. This check word 1) "encodes" which encoding function is used in the encoding process $\mathcal{E}_2$, and 2) corrects any further m_1-imbalance of the almost 2-OSN word. To decode a codeword, the receiver applies, to the m_0-balanced information part, the inverse of the function encoded by the check part.

The coding scheme is based on the concept of the set of m_1-balancing functions. This is explained below.

***Definition** 1 (set of m_1-balancing functions):* Let $k, r \in \mathbb{N}$ be given so that $n \stackrel{\text{def}}{=} k + r \in \mathbb{N}$ be a multiple of 4. Also, let $\mathcal{CS} \stackrel{\text{def}}{=} \{\Gamma_0, \Gamma_1, \ldots, \Gamma_{p-1}\}$, be a family of $p \in \mathbb{N}$ nonempty subsets of $S^r_{\lceil r/2 \rceil}$ (i.e., the set of r bit m_0-balanced check symbols) and, for all $h \in [0, p-1]$, let

$$f_h \stackrel{\text{def}}{=} \langle \Gamma_h \rangle \colon S^k_{\lfloor k/2 \rfloor} \to S^k_{\lfloor k/2 \rfloor}$$

indicate a function from $S^k_{\lfloor k/2 \rfloor}$ (= the set of k bit m_0-balanced data symbols) onto itself. We say that the set of functions $\mathcal{B} \stackrel{\text{def}}{=} \{f_h = \langle \Gamma_h \rangle \colon h \in [0, p-1]\}$ is a set of m_1-balancing functions if, and only if, the following conditions hold.

1) The sets Γ_h's are pair-wise disjoint; i.e., $\Gamma_i \cap \Gamma_j = \emptyset \iff i \neq j$.
2) For all information word $X \in S^k_{\lfloor k/2 \rfloor}$ there exists one (or more) $h_b = h_b(X) \in [0, p-1]$ and $C_{h_b} \in \Gamma_{h_b}$ such that $m_1(\langle \Gamma_{h_b} \rangle(X) C_{h_b})) = n(n+1)/4$. In the sequel, we refer to the index h_b as the/a "m_1-balancing index" of X.
3) For all $h \in [0, p-1]$, the function $\langle \Gamma_h \rangle$ is one-to-one (so, from h and $Z = \langle \Gamma_h \rangle(X)$ it is possible to recover X).

Given a set of m_1-balancing functions, every m_0-balanced data word $X \in S^k_{\lfloor k/2 \rfloor}$ is encoded as $\mathcal{E}_2(X) = \langle \Gamma_{h_b} \rangle(X) C_{h_b}$ where $h_b = h_b(X)$ is the smallest (or simply, an) index such that there exists a check symbol $C_{h_b} \in \Gamma_{h_b}$ which makes

$$m_1(\mathcal{E}_2(X)) = m_1(\langle \Gamma_{h_b} \rangle(X) C_{h_b}) = \frac{n(n+1)}{4}.$$

Now, a possible construction for such balancing functions is explained below. First, given $X = x_1 x_2 x_3 \ldots x_{k-1} x_k$ let $X^R = x_k x_{k-1} \ldots x_3 x_2 x_1$ be the reverse of X, and

$$X^{(i,j)} = x_1 x_2 x_3 \ldots \underline{x_j} \ldots \underline{x_i} \ldots x_{k-1} x_k$$

be the word obtained form X by exchanging the ith with the jth bit. The design is based on the following observations [15].

O1: The weight $m_0(X)$ does not change if the bits of X are permuted.

O2: The relation $m_1(X) + m_1(X^R) = (k+1)m_0(X)$ holds. So, if we assume $m_0(X) = k/2 \in \mathbb{N}$ for simplicity, then $m_1(X) \leq k(k+1)/4 \iff m_1(X^R) \geq k(k+1)/4$.

O3: The following relation holds $m_1\big(X^{(i,i+1)}\big) = m_1(X) + (x_i - x_{i+1})$. This implies that by exchanging two consecutive bits of X, the value of m_1 either decreases by 1 or remains the same or increases by 1.

So, let us consider, for example,

$$\begin{aligned}
X^{(0)} &\stackrel{\text{def}}{=} X,\\
X^{(1)} &\stackrel{\text{def}}{=} \big(X^{(0)}\big)^{(1,2)},\\
X^{(2)} &\stackrel{\text{def}}{=} \big(X^{(1)}\big)^{(2,3)},\\
&\vdots\\
X^{(k-1)} &\stackrel{\text{def}}{=} \big(X^{(k-2)}\big)^{(k-1,k)},\\
X^{(k)} &\stackrel{\text{def}}{=} \big(X^{(k-1)}\big)^{(1,2)},\\
X^{(k+1)} &\stackrel{\text{def}}{=} \big(X^{(k)}\big)^{(2,3)},\\
&\vdots\\
X^{(2k-3)} &\stackrel{\text{def}}{=} \big(X^{(2k-4)}\big)^{(k-2,k-1)},\\
X^{(2k-2)} &\stackrel{\text{def}}{=} \big(X^{(2k-3)}\big)^{(1,2)},\\
X^{(2k-1)} &\stackrel{\text{def}}{=} \big(X^{(2k-2)}\big)^{(2,3)},\\
&\vdots\\
X^{(3k-6)} &\stackrel{\text{def}}{=} \big(X^{(3k-7)}\big)^{(k-3,k-2)},\\
X^{(3k-5)} &\stackrel{\text{def}}{=} \big(X^{(3k-6)}\big)^{(1,2)},\\
&\vdots\\
X^{(k(k-1)/2)-1} &\stackrel{\text{def}}{=} \big(X^{(k(k-1)/2)-2}\big)^{(2,3)},\\
X^{(k(k-1)/2)} &\stackrel{\text{def}}{=} \big(X^{(k(k-1)/2)-1}\big)^{(1,2)}.
\end{aligned}$$

For example, if $k = 4$ and $X = x_1 x_2 x_3 x_4$ then

$$\begin{aligned}
X^{(0)} &= x_1 x_2 x_3 x_4,\\
X^{(1)} &= \underline{x_2 x_1} x_3 x_4,\\
X^{(2)} &= x_2 \underline{x_3 x_1} x_4,\\
X^{(3)} &= x_2 x_3 \underline{x_4 x_1},\\
X^{(4)} &= \underline{x_3 x_2} x_4 x_1,\\
X^{(5)} &= x_3 \underline{x_4 x_2} x_1,\\
X^{(6)} &= \underline{x_4 x_3} x_2 x_1
\end{aligned}$$

The above sequence $\big\{X^{(i)}\big\}_{i=0,\ldots,k(k-1)/2}$ is defined by a particular (say, a bubble-sort like) sequence of exchanging of consecutive bits of X, with $X^{(0)} = X$ and $X^{(k(k-1)/2)} = X^R$. Since $m_1\left(X^{(i+1)}\right) - m_1\left(X^{(i)}\right) \in \{-1, 0, +1\}$ for all $i \in [0, k(k-1)/2]$, it follows that the sequence defined as $\left\{(i, m_1(X^{(i)}))\right\}_{i=0,1,\ldots,k(k-1)/2}$ represents a "random walk" from $(0, m_1(X))$ to $\left(k(k-1)/2, m_1\left(X^R\right)\right)$ and satisfies the following properties.

For all $X \in \mathbb{Z}_2^k$, there always exists $i \in [0, k(k-1)/2-1]$ such that

$$m_1\Big(X^{(i)}\Big) = \left\lfloor \frac{k(k+1)}{4} \right\rfloor \; \left(\text{or } \left\lceil \frac{k(k+1)}{4} \right\rceil\right), \tag{4}$$

and

For all $X \in \mathbb{Z}_2^k$ and for all $i_1, i_2 \in [0, k(k-1)/2]$, $m_1\big(X^{(i_2)}\big) \in$

$$\Big[m_1\Big(X^{(i_1)}\Big) - |i_2 - i_1|, m_1\Big(X^{(i_1)}\Big) + |i_2 - i_1|\Big]. \tag{5}$$

Now we come to the definition of the Γ_i's. Consider the partition $\{\Gamma_i\}_{i=0,\ldots,p-1}$ of $S^r_{\lceil r/2\rceil}$ in non empty sets defined as follows: Γ_0 contains exactly one word of the set $S^r_{\lceil r/2\rceil, M_1} \neq \emptyset$ for each possible value M_1 in the "m_1-image" of $S^r_{\lceil r/2\rceil}$ defined to be $m_1\left(S^r_{\lceil r/2\rceil}\right) \stackrel{\text{def}}{=} \left\{m_1(X) : X \in S^r_{\lceil r/2\rceil}\right\}$. If $S^r_{\lceil r/2\rceil, M_1} - \Gamma_0 \neq \emptyset$ then let Γ_1 contain exactly one word of the set $S^r_{\lceil r/2\rceil, M_1} - \Gamma_0$ for each possible value $M_1 \in m_1(S^r_{\lceil r/2\rceil} - \Gamma_0)$; otherwise, stop. In general, consider the following constructive rule

If $S^r_{\lceil r/2\rceil, M_1} - \bigcup_{j=0}^{h-1} \Gamma_j \neq \emptyset$ then let Γ_h contain exactly one word in $S^r_{\lceil r/2\rceil, M_1} - \bigcup_{j=0}^{h-1} \Gamma_j$ for each possible value $M_1 \in m_1\Big(S^r_{\lceil r/2\rceil, M_1} - \bigcup_{j=0}^{h-1} \Gamma_j\Big)$; otherwise, stop. (6)

Note that such definition implies $p = \max_{M_1} |S^r_{\lceil r/2\rceil, M_1}|$. For example, if $r = 5$ then

$S^5_{3,6} = \{11100\}$, $S^5_{3,10} = \{10011, 01101\}$,
$S^5_{3,7} = \{11010\}$, $S^5_{3,11} = \{01011\}$,
$S^5_{3,8} = \{11001, 10110\}$, $S^5_{3,12} = \{00111\}$
$S^5_{3,9} = \{10101, 01110\}$,

so that, for example,

$\Gamma_0 = \{11100, 11010, 11001, 10101, 10011, 01011, 00111\}$,
$\Gamma_1 = \{10110, 01110, 01101\}$,

and $p = 2$. The following theorem holds.

***Theorem** 1:* Let $c(r, M_0, M_1) \stackrel{\text{def}}{=} \left|S^r_{M_0, M_1}\right|$. The following statements hold for any $M_0 \in [0, r]$.

1) $\alpha \stackrel{\text{def}}{=} \min_{C \in S^r_{M_0}} m_1(C) = M_0(M_0+1)/2$;
2) $\beta \stackrel{\text{def}}{=} \max_{C \in S^r_{M_0}} m_1(C) = M_0(M_0+1)/2 + M_0(r - M_0)$;
3) the sequence $\{c(r, M_0, s) : s = \alpha, \alpha+1, \ldots, \beta\}$ is unimodal and symmetric about $(\alpha+\beta)/2 = M_0(r+1)/2$; namely,

$$\begin{gathered} c(r, M_0, \alpha) \leq c(r, M_0, \alpha+1) \leq \ldots \\ \ldots \leq c(r, M_0, \lfloor M_0(r+1)/2 \rfloor) = \\ = c(r, M_0, \lceil M_0(r+1)/2 \rceil) \leq \ldots \\ \ldots \leq c(r, M_0, \beta - 1) \leq c(r, M_0, \beta) \end{gathered}$$

and $c(r, M_0, \alpha + \mu) = c(r, M_0, \beta - \mu)$, for all integers $\mu \in [0, M_0(r - M_0)]$.

Proof: See Theorem 2, p. 6, in [17]. ■

Note that, if $\{\Gamma_h\}_{h \in [0, p-1]}$ is defined by (6) then Theorem 1 for $M_0 = \lceil r/2 \rceil$ implies that for all integer $h \in [0, p-1]$,

O4: the m_1-image of the words in Γ_h, $m_1(\Gamma_h)$, is an integer interval, say $m_1(\Gamma_h) \stackrel{\text{def}}{=} [\alpha_h, \beta_h]$;

O5: if $h > 0$ then $m_1(\Gamma_h) = [\alpha_h, \beta_h] \subseteq [\alpha_{h-1}, \beta_{h-1}] = m_1(\Gamma_{h-1})$;

O6: $(\alpha_h + \beta_h)/2 = (\alpha + \beta)/2 = \lceil r/2 \rceil (r+1)/2$, $(\beta_h - \alpha_h) = |\Gamma_h| - 1$ and, so, $[\alpha_h, \beta_h] = [(\alpha_h + \beta_h)/2 - (\beta_h - \alpha_h)/2, (\alpha_h + \beta_h)/2 + (\beta_h - \alpha_h)/2] = m_1(\Gamma_h) =$

$$\left[\left\lceil \frac{r}{2} \right\rceil \frac{(r+1)}{2} - \frac{|\Gamma_h| - 1}{2}, \left\lceil \frac{r}{2} \right\rceil \frac{(r+1)}{2} + \frac{|\Gamma_h| - 1}{2}\right]. \tag{7}$$

O7: $p = \left|S^r_{\lceil r/2\rceil, \lfloor \lceil r/2 \rceil (r+1)/2 \rfloor}\right|$.

Finally, given the partition $\{\Gamma_h\}_{h \in [0, p-1]}$, as in [1] and [14], define the following p natural numbers

$$d_h \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } h = 0, \\ d_{h-1} + \lfloor |\Gamma_{h-1}|/2 \rfloor + \lceil |\Gamma_h|/2 \rceil & \text{if } h \in [1, p-1] \end{cases} \tag{8}$$

and the functions $f_h = \langle \Gamma_h \rangle : S^k_{\lfloor k/2 \rfloor} \to S^k_{\lfloor k/2 \rfloor}$ as

$$f_h(X) \stackrel{\text{def}}{=} X^{(d_h)}, \quad \forall h \in [0, p-1]. \tag{9}$$

Note that the functions f_h's are simply, as any step in the random walk, permutations of the components of X.

Also the following theorem holds.

***Theorem** 2:* For all $X \in \mathbb{Z}_2^k$ and $C \in \mathbb{Z}_2^r$, the following relation holds

$$m_1(XC) = m_1(X) + m_1(C) + k m_0(C). \tag{10}$$

Proof: If $X = x_1 x_2 \ldots x_k \in \mathbb{Z}_2^k$ and $C = c_1 c_2 \ldots c_r \in \mathbb{Z}_2^r$ then $m_1(XC) = \sum_{j=1}^k x_j j + \sum_{j=k+1}^{k+r} c_{j-k} j = m_1(X) + \sum_{j=1}^r c_j (k+j) = m_1(X) + k \sum_{j=1}^r c_j + \sum_{j=1}^r c_j j = m_1(X) + k m_0(C) + m_1(C)$. ■

Now we come to the main theorem.

***Theorem** 3:* Assume $k, r \in 2\mathbb{N}$ be such that $k + r = n$ is a multiple of 4, and the relation (3) holds. Assume $\{\Gamma_h\}_{h \in [0, p-1]}$ be a partition of $S^r_{\lceil r/2 \rceil}$ constructed as in (6), and let $\mathcal{B} = \{f_h = \langle \Gamma_h \rangle : h \in [0, p-1]\}$ be the set of functions defined by (8) and (9). Then $\mathcal{B}$ is a set of m_1-balancing functions according to Definition 1.

Sketch of the proof: Property 1) of Definition 1 follows by construction (6). With regard to the property 2), for simplicity, assume k and r be multiple of 4 and note that if there is a "small" m_1-imbalance in the word $f_h(X) = X^{(d_h)}$ so that $m_1\big(X^{(d_h)}\big) = k(k+1)/4 - \mu \in [k(k+1)/4 - (|\Gamma_h| - 1)/2, k(k+1)/4 + (|\Gamma_h| - 1)/2]$ then there exists $C \in \Gamma_h$ with $m_1(C) = r(r+1)/4 + \mu \in m_1(\Gamma_h) = [r(r+1)/4 - (|\Gamma_h| - 1)/2, r(r+1)/4 + (|\Gamma_h| - 1)/2]$ (see (7)) such that

$$m_1\Big(X^{(d_h)} C\Big) = m_1\Big(X^{(d_h)}\Big) + m_1(C) + \frac{kr}{2} =$$
$$\left(\frac{k(k+1)}{4} - \mu\right) + \left(\frac{r(r+1)}{4} + \mu\right) + \frac{kr}{2} = \frac{n(n+1)}{4},$$

because of Theorem 2. So, from Theorem 1, Theorem 2, (4), (5), (8) and (9), property 2) follows as in the analogous Theorem 4 for balanced codes in [14]. Property 3) holds because the functions f_h's in (9) are permutations of the components of X. ∎

Thus, the encoding of an information word is specified by the following algorithm.

***Algorithm** 1 (Encoding Algorithm):*

Input: the information word $Z\in\mathbb{Z}_2^{\tilde{k}}$.

Output: the codeword $YC\in\mathbb{Z}_2^n$ such that $n=k+r$, $Y\in\mathbb{Z}_2^k$, $C\in\mathbb{Z}_2^r$ and $YC=\mathcal{E}_2(\mathcal{E}_1(Z))=(\mathcal{E}_1\circ\mathcal{E}_2)(Z)\in S(n,2)$.

Perform steps S0, S1 and S2.

S0: Apply any m_0-balancing method to Z. For example, a method given in [3], [6], [1], [2], [13], [14], [16] or [9] can be used. Let X be the m_0-balanced word of length k associated with Z, i.e. $X=\mathcal{E}_1(Z)$. In this way, $m_0(X)=\lfloor k/2\rfloor$.

S1: Set $h=0$ and do iteratively steps S2.1, S2.2 and S2.3 until the check symbol C_{h_b} is found.

S1.1: Compute $Y_h=\langle\Gamma_h\rangle(X)=X^{(d_h)}$.

S1.2: Compute $m_1(Y_h)$.

S1.3: If there exists $C\in\Gamma_h$ such that $m_1(C)=n(n+1)/4-k\lceil r/2\rceil-m_1\left(X^{(d_h)}\right)$ then a balancing index is $h_b\overset{\text{def}}{=}h$ and the relative balancing check is $C_{h_b}\overset{\text{def}}{=}C\in\Gamma_{h_b}$. Goto step S2.

S1.3: Otherwise, increase h by 1 and go to step S1.

S2: Select as encoding of the m_0-balanced word X the m_1-balanced word $\mathcal{E}_2(X)=YC=X^{(d_{h_b})}C_{h_b}$. Thus, the information word Z is finally encoded into the 2-OSN word $YC=\mathcal{E}_2(X)=\mathcal{E}_2(\mathcal{E}_1(Z))=(\mathcal{E}_1\circ\mathcal{E}_2)(Z)$.

On the other hand, the decoding is performed as follows.

***Algorithm** 2 (Decoding Algorithm):*

Input: $YC\in\mathbb{Z}_2^n$ such that $n=k+r$, $Y\in\mathbb{Z}_2^k$, $C\in\mathbb{Z}_2^r$ and $YC=\mathcal{E}_2(\mathcal{E}_1(Z))=(\mathcal{E}_1\circ\mathcal{E}_2)(Z)\in S(n,2)$ for some $Z\in\mathbb{Z}_2^{\tilde{k}}$.

Output: $Z=(\mathcal{E}_1\circ\mathcal{E}_2)^{-1}(YC)$.

Perform steps S1, S2 and S3.

S1: Compute the index $h_b\in[0,p-1]$ such that $C\in\Gamma_{h_b}$.

S2: Compute the word $X=\mathcal{E}_2^{-1}(YC)=\langle\Gamma_{h_b}\rangle^{-1}(Y)$.

S3: Undo the m_0-balancing method chosen in step S1 of Algorithm 1 to the word X. Let $Z\in\mathbb{Z}_2^{\tilde{k}}$ be the m_0-unbalanced word associated with X such that $Z=\mathcal{E}_1^{-1}(X)$. Then the word YC is decoded into the word $Z=\mathcal{E}_1^{-1}(X)=\mathcal{E}_1^{-1}(\mathcal{E}_2^{-1}(YC))=(\mathcal{E}_1\circ\mathcal{E}_2)^{-1}(YC)$.

The following example shows the m_1-balancing part of Algorithm 1 and the m_1-unbalancing part of Algorithm 2.

***Example** 1:* Let the number of balanced information bits be $k=15$ so that the number of extra redundant bits can be $r=9$ (see Table I). Let $X=100101001001011$ be the given m_0-balanced (1-OSN) word to be m_1-balanced into a word in $S(24,2)$. In this way, the m_0-weight of every codeword must be equal to $(k+r)/2=12$ and the m_1-weight must be equal to $(k+r)(k+r+1)/4=150$. Note that $k\lceil r/2\rceil=75$ and $|S_{5,25}^{9}|=12$. So, the family of m_1-balancing functions $\mathcal{B}=\{f_h=\langle\Gamma_h\rangle:\ h\in[0,11]\}$ may be defined as follows.

$$
\begin{aligned}
\Gamma_0=\{&111110000,111101000,111100100,111100010,\\
&111100001,111010001,111001001,111000101,\\
&111000011,110100011,110010011,110001011,\\
&110000111,101000111,100100111,100010111,\\
&100001111,010001111,001001111,000101111,\\
&000011111\},
\end{aligned}
$$

$$
\begin{aligned}
\Gamma_1=\{&111011000,111010100,111010010,111001010,\\
&110101010,101101010,110010110,110010101,\\
&101100011,101010011,011010011,010101101,\\
&010101011,010100111,010010111,001010111,\\
&000110111\},\\
\Gamma_2=\{&110111000,110110100,110110010,110110001,\\
&110101001,110100101,101100101,101010101,\\
&101001101,101001011,100101011,100011011,\\
&010011011,001011011,000111011\},\\
\Gamma_3=\{&101111000,101110100,101110010,101110001,\\
&101101001,101011001,100111001,100110101,\\
&100101101,100011101,010011101,001011101,\\
&000111101\},\\
\Gamma_4=\{&111001100,110101100,101101100,011101100,\\
&011101010,011101001,011011001,010110110,\\
&010101110,001101110,001101101,001101011,\\
&001100111\},\\
\Gamma_5=\{&011111000,011110100,011110010,011110001,\\
&010111100,001111100,001111010,100011110,\\
&010011110,001011110,000111110\}\\
\Gamma_6=\{&110011100,110011010,110011001,110001110,\\
&101001110,011001110,100110011,010110011,\\
&001110011\},\\
\Gamma_7=\{&111000110,110100110,101100110,011100110,\\
&011100101,011100011,011001101,011001011,\\
&011000111\},\\
\Gamma_8=\{&101011100,101011010,101010110,100110110,\\
&100101110,010110101,001110101\},\\
\Gamma_9=\{&100111100,100111010,010111010,010111001,\\
&001111001\}\\
\Gamma_{10}=\{&011011100,011011010,011010110,011010101,\\
&001110110\}\\
\Gamma_{11}=\{&110001101\}.
\end{aligned}
$$

Hence, from (8) and (9), $f_h(X)=X^{(d_h)}$, $h\in[0,11]$, with

$$
\begin{array}{llll}
d_0=0, & d_1=19, & d_2=35, & d_3=49,\\
d_4=62, & d_5=74, & d_6=84, & d_7=93,\\
d_8=101, & d_9=107, & d_{10}=112, & d_{11}=115.
\end{array}
$$

Note that, since the random walk length is $k(k-1)/2+1=106$, only the first $p=9$ $(<|S_{5,25}^{9}|=12)$ m_1-balancing functions are needed.

The encoding algorithm proceeds as follows.

0) In the first step, being $d_0=0$ we have

$$\langle\Gamma_0\rangle(X)=X^{(0)}=X=100101001001011,$$

$m_1\left(X^{(0)}\right)=61$ and $n(n+1)/4-k\lceil r/2\rceil-m_1\left(X^{(0)}\right)=150-75-61=14$. In Γ_0 there isn't any checkword whose m_1-weight is 14, i.e. $14\notin m_1(\Gamma_0)=[15,35]$. Thus, Γ_0 cannot encode X. Let's go further.

1) $\langle\Gamma_1\rangle(X)=X^{(19)}=01010\underline{0}01001011\underline{1}$, $m_1\left(X^{(19)}\right)=67$ and $150-75-67=8\notin m_1(\Gamma_1)=[17,33]$. Thus, Γ_1 cannot encode X. Go further.
2) $\langle\Gamma_2\rangle(X)=X^{(35)}=10100100\underline{0}1011\underline{01}$, $m_1\left(X^{(35)}\right)=60$ and $150-75-60=15\notin m_1(\Gamma_2)=[18,32]$. Thus, Γ_2 cannot encode X. Go further.
3) $\langle\Gamma_3\rangle(X)=X^{(49)}=0100100101\underline{1}1\underline{001}$, $m_1\left(X^{(49)}\right)=63$ and $150-75-63=12\notin m_1(\Gamma_3)=[19,31]$. Thus, Γ_3 cannot encode X. Go further.

4) $\langle\Gamma_4\rangle(X) = X^{(62)} = 00\underline{1}1001011\underline{01001}$, $m_1\left(X^{(62)}\right) = 60$ and $150 - 75 - 60 = 15 \notin m_1(\Gamma_4) = [19, 31]$. Thus, Γ_4 cannot encode X. Go further.
5) $\langle\Gamma_5\rangle(X) = X^{(74)} = 01001\underline{0}011\underline{101001}$, $m_1\left(X^{(74)}\right) = 61$ and $150 - 75 - 61 = 14 \notin m_1(\Gamma_5) = [20, 30]$. Thus, Γ_5 cannot encode X. Go further.
6) $\langle\Gamma_6\rangle(X) = X^{(84)} = 1001011\underline{00101001}$, $m_1\left(X^{(84)}\right) = 55$ and $150 - 75 - 55 = 20 \notin m_1(\Gamma_6) = [21, 29]$. Thus, Γ_6 cannot encode X. Go further.
7) $\langle\Gamma_7\rangle(X) = X^{(93)} = 010\underline{0}11\underline{100101001}$, $m_1\left(X^{(93)}\right) = 57$ and $150 - 75 - 57 = 18 \notin m_1(\Gamma_7) = [21, 29]$. Thus, Γ_7 cannot encode X. Go further.
8) $\langle\Gamma_8\rangle(X) = X^{(101)} = 01\underline{1}1\underline{00100101001}$, $m_1\left(X^{(101)}\right) = 53$ and $150 - 75 - 53 = 22 \in m_1(\Gamma_8) = [22, 28]$. Thus, Γ_8 can encode X, so that $h_b = 8$, and the check symbol in Γ_8 whose m_1-weight is 22 is $C_8 = 101011100$.

Finally, the 1-OSN word X is encoded into the 2-OSN word

$$Y = \mathcal{E}_2(X) = X^{(101)} C_8 = 011100100101001\ 101011100.$$

Decoding is straightforward. On receiving the 2-OSN word $YC = 011100100101001101011100$, the sequence of the last $r = 9$ bits (i.e., $C = 101011100$) is the check symbol that allows to identify the encoding function $\langle\Gamma_h\rangle$ used. Being $C \in \Gamma_8$, the word Y is decoded into the m_0-balanced word

$$\mathcal{E}_2^{-1}(YC) = \langle\Gamma_8\rangle^{-1}(Y) = 100101001001011.$$

Note that, even though k and r are both odd, this encoding example works for any $X \in \mathbb{Z}_2^{15}$.

III. Redundancy and complexity

The (extra) m_1-balancing redundancy of the proposed codes is $n - k \simeq 2\log_2 k + (1/2)\log_2\log_2 k - 0.674$ (see (3)). Table I compares the proposed non-recursive encoding to 2-OSN sequences with the optimal encoding and the recursive encoding given in [15]. The full paper shows that, by using an enumerative coding technique [3], the above encoding/decoding algorithms can be implemented with $O(n \log n)$ bit operations and storing $O(n)$ bits. However, note that the decoding can be done in parallel because the check symbol directly identifies the m_1-balancing function used to encode and each balancing function and its inverse are very simple to compute (they are composition of two cyclic shifts). In general, both encoding and decoding may be given as parallel algorithms in line with the algorithms in [14].

TABLE I
Comparisons with the optimal codes and the codes in [15].

n	optimal	new scheme			scheme in [15]		
	$\tilde{k}$	k	$\tilde{k}$	$\Delta\tilde{k}_{op}$	k	$\tilde{k}$	$\Delta\tilde{k}_{[15]}$
16	9	8	6	3	–	9	−3
20	12	12	9	3	–	12	−3
24	15	15	12	3	–	15	−3
28	19	18	15	4	12	9	6
32	23	22	19	4	16	13	6
36	26	25	22	4	20	17	5
40	30	29	26	4	24	21	5
44	34	32	29	5	28	25	4
48	37	36	33	4	32	29	4
52	41	40	37	4	32	29	8
56	45	43	39	6	36	33	6
60	49	47	43	6	40	37	6
64	53	51	47	6	44	40	7
128	115	113	109	6	104	100	9
256	241	238	233	8	232	227	6
512	495	492	487	8	476	471	16
1024	1005	1002	996	9	988	982	14
2048	2027	2024	2018	9	2008	2002	16
4096	4073	4070	4063	10	4056	4049	14
8192	8167	8164	8157	10	8148	8141	16
16384	16357	16354	16346	11	16340	16332	14
32768	32739	32736	32728	11	32720	32712	16
65536	65505	65502	65493	12	65488	65479	14

The 1st column is the overall length of 2-OSN codes; the 2nd column gives the information bits for the optimal code; the 3rd, 4th and 5th refer to the method given in this paper; the 6th, 7th and 8th refer to the method given in [15]. k is the maximum number of m_0-balanced bits that can be m_1-balanced into n bits, with the method indicated above; $\tilde{k}$ is the maximum number of information bits that can be m_0-balanced into k bits; $\Delta\tilde{k}_{op}$ = 2nd minus 4th column; $\Delta\tilde{k}_{[15]}$ = 4th minus 7th column. For $n \geq 128$, the values of the second column are obtained with the approximation $|S(n, 2)| \approx \left\lfloor (4\sqrt{3}/\pi)2^n/n^2 \right\rfloor$ given in [15], [17].

Acknowledgment

This work is supported by the NSF Grant CCF-1117215, by the Italian MIUR Grant PRIN 2009-2009EL7424 and by the King Saud University NPST Project Number 10-INF1165-02.

References

[1] S. Al-Bassam, B. Bose, "On Balanced Codes", *IEEE Trans. Inform. Theory*, vol. 36, pp. 406–408, March 1990.
[2] S. Al-Bassam, B. Bose, "Design of Efficient Balanced Codes", *IEEE Trans. Comput.*, vol. 43, pp. 362–365, March 1994.
[3] T. M. Cover, "Enumerative Source Encoding", *IEEE Trans. Inform. Theory*, vol. 19, pp. 73–77, Jan. 1973.
[4] K. A. S. Immink, "Spectrum Shaping with DC^2-Constrained Channel Codes", *Philips J. Res.*, vol. 40, pp. 40–53, 1985.
[5] K. A. S. Immink, *Codes for Mass Data Storage Systems*, Second Edition. Eindhoven, Netherlands: Shannon Foundation Publishers, 2004.
[6] D. E. Knuth, "Efficient Balanced Codes", *IEEE Trans. Inform. Theory*, vol. 32, pp. 51–53, Jan. 1986.
[7] R. Mascella, L. G. Tallini, "On efficient high-order spectral-null codes over the m-ary alphabet", *J. Discrete Math. Sci. Cryptography*, vol. 8, pp. 459–481, 2005.
[8] R. Mascella, L. G. Tallini, "Efficient m-ary balanced codes which are invariant under symbol permutation", *IEEE Trans. Comput.*, vol. 55, pp. 929–946, Aug. 2006.
[9] D. Pelusi, L. G. Tallini, B. Bose, "On m-ary balanced codes with parallel decoding", *Proc. 2010 IEEE ISIT*, pp. 1305–1309, June 2010. An extended version has been submitted to *IEEE Trans. Inform. Theory*.
[10] L. Pezza, L. G. Tallini, B. Bose, "Variable Length Unordered Codes", *IEEE Trans. Inform. Theory*, vol. 58, pp. 548–569, Feb. 2012.
[11] R. M. Roth, P. H. SIegel, A. Vardy, "High-Order Spectral-Null Codes–Constructions and Bounds", *IEEE Trans. Inform. Theory*, vol. 40, pp. 1826–1840, Nov. 1994.
[12] L. G. Tallini, S. Al-Bassam, B. Bose, "Feedback Codes Achieving the Capacity of the Z-Channel", *IEEE Trans. Inform. Theory*, vol. 54, pp. 1357–1362, March 2008.
[13] L. G. Tallini, R. M. Capocelli, B. Bose, "Design of some new efficient balanced codes", *IEEE Trans. Inform. Theory*, vol. 42, pp. 790–802, May 1996.
[14] L. G. Tallini, B. Bose, "Balanced Codes with Parallel Encoding and Decoding", *IEEE Trans. Comput.*, vol. 48, pp. 794–814, Aug. 1999.
[15] L. G. Tallini, B. Bose, "On Efficient High-Order Spectral-Null Codes", *IEEE Trans. Inform. Theory*, vol. 45, pp. 2594–2601, Nov. 1999.
[16] L. G. Tallini, U. Vaccaro, "Efficient m-ary Balanced Codes", *Discrete Appl. Math.*, vol. 92, pp. 17–56, 1999.
[17] L. G. Tallini, "Geometric properties of high-order spectral-null codes", *Ital. J. Pure Appl. Math.*, n. 14, pp. 149–176, 2003.
[18] C. N. Yang, "Efficient Encoding Algorithm for Second-Order Spectral-Null Codes Using Cyclic Bit Shift", *IEEE Trans. Comput.*, vol. 57, pp. 876–888, July 2008.