Prefixless q-ary Balanced Codes

Kees A. Schouhamer Immink
Turing Machines Inc, Willemskade 15b-d,
3016 DK Rotterdam, The Netherlands
E-mail: immink@turing-machines.com

***Abstract*—We will present a Knuth-like method for balancing q-ary codewords, which is characterized by the absence of a prefix that carries the information on the balancing index. Look-up tables for coding and decoding the prefix are avoided.**

Index Terms - Constrained code, balanced code, running digital sum, Knuth code, error correction.

I. Introduction

Balanced, sometimes called *dc-free*, q-ary sequences have found widespread application in popular optical recording devices such as Compact Disc, DVD and, Blu-Ray [1], cable communication, and recently in non-volatile (Flash) memories [2] [3]. Prior art codes were presented by Capocelli *et al.* [4] and Swart & Weber [5].

Let $\boldsymbol{x} = (x_1, \ldots, x_m)$ be a word of m q-ary symbols taken from the q-ary alphabet $\mathcal{Q} = \{0, 1, \ldots, q-1\}$, where $q \geq 2$ and $m > 1$ positive integers. The integer m will be called *length* of the codeword $\boldsymbol{x}$. The *weight*, or *unbalance*, of $\boldsymbol{x}$, denoted by weight($\boldsymbol{x}$), is simply defined as the sum of the m q-ary symbols, that is,

$$\mathrm{weight}(\boldsymbol{x}) = \sum_{i=1}^{m} x_i.$$

An m-symbol codeword $\boldsymbol{x}$ is said to be balanced if

$$\mathrm{weight}(\boldsymbol{x}) = \frac{m(q-1)}{2}.$$

For certain practical applications, it is a desideratum to generate balanced q-ary sequences. Clearly, look-up tables can be used in case the sequences are not too long. Knuth [6] described a simple encoding technique for generating binary, $q = 2$, balanced codewords, which is capable of handling (very) long binary blocks.

In Knuth's algorithm, an m-bit user word (pay load), m even, is forwarded to the encoder. The encoder splits the user word into a first segment consisting of the first v bits of the user word, and a second segment consisting of the remaining $m - v$ bits. The encoder adds (modulo 2) a '1' to the $m - v$ symbols in the second segment. The index v is chosen in such a way that the modified word is balanced. Knuth showed that such an index v can always be found. In the simplest case, the index v is represented by a balanced word, called *prefix*, of length p. The balanced p-bit prefix is appended to the balanced m-bit user word and transmitted. We simply find that the rate of the code is $m/(m + p)$. After observation of the p-bit prefix, the receiver can retrieve v, and easily undo the modifications made. Note that both encoder and decoder do not require look-up tables for the pay load, and we conclude that Knuth's algorithm is very attractive for constructing long balanced codewords. Note, however, that Knuth's scheme does require look-up tables for encoding and decoding the prefix. Modifications of the generic binary scheme have been discussed by Al-Bassam & Bose [7], Tallini, Capocelli & Bose [8], and Weber & Immink [9].

Capocelli *et al.* [4] and Swart & Weber [5] generalized Knuth's binary scheme to the balancing of q-ary codewords, $q > 2$. In [5], balancing is achieved, as in Knuth's scheme, by splitting the user word into a first and second segment of v and $m - v$ symbols, respectively. The encoder adds (modulo q) an integer $s \in \mathcal{Q}$ to the symbols in the first segment, and an integer $s + 1$ to the symbols in the second segment. Swart & Weber showed that there exists at least one pair of integers s and v such that the modified word is balanced. The integers s and v are represented by a balanced q-ary p-symbol prefix, which is appended to the balanced codeword, and subsequently transmitted to the receiver. The prefix must therefore be sufficiently long to be able of representing qm distinct pairs of integers s and v. As in the binary 'Knuth' case, look-up tables for encoding and decoding the prefix are required which may be prohibitively complex for large values of p.

In Sections II, we start with notational conventions and relevant results from the literature. In Section III, we will present a new method for constructing prefixless q-ary balanced correcting codes. Finally, the results of this paper are discussed in Section IV.

II. Background

We start with some definitions. Let $\boldsymbol{x} = (x_1, \ldots, x_m)$ be a word of m, q-ary symbols, $x_i \in \mathcal{Q}$. The word $\boldsymbol{w} = (w_1, \ldots, w_m)$ is obtained by modulo q *integration* of $\boldsymbol{x}$, that is by the following operation

$$w_i = w_{i-1} \oplus_q x_i, 1 \leq i \leq m, \quad (1)$$

where $w_0 = 0$, and the $\oplus_q$ sign indicates modulo q summation. The above operation, often called *precoding*, will be denoted by the shorthand notation $\boldsymbol{w} = I(\boldsymbol{x})$. Note that the original word $\boldsymbol{x}$ can be uniquely restored by modulo q *differentiation*:

$$x_i = w_i \ominus_q w_{i-1},\ 1 \leq i \leq m, \quad (2)$$

where $\ominus_q$ indicates modulo q subtraction. The above differentiation operation will be denoted by $\boldsymbol{x} = I^{-1}(\boldsymbol{w})$. Clearly,

$$I^{-1}(I(\boldsymbol{x})) = \boldsymbol{x}.$$

Define the binary m-bit word $\boldsymbol{u}_v = (0^{v-1}10^{m-v})$ (that is, $\boldsymbol{u}_v$ consists of '0's except a single '1' at position v), and let $\boldsymbol{x} = (x_1, \ldots, x_m)$ be a word of m q-ary symbols, where q and m are chosen such that $(q-1)m/2$ is an integer. We are now in the position to formulate Theorem 1.

Theorem 1: There is at least one pair of integers, s and v, $s \in \mathcal{Q}$, $v \in \{1, \ldots, m\}$, such that $I(\boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1)$ is balanced, that is weight($I(\boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1)$) = $m(q-1)/2$.
Proof: Trivial considering Swart & Weber's Theorem 1 [5].

Example: Let $q = 5$ and $m = 6$, and let the pay load be $\boldsymbol{x}$ = (4,2,1,0,0,0). After a search, we find $s = 2$ and $v = 5$. Adding $\boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1$ = (2,0,0,0,1,0) to the pay load, yields $\boldsymbol{y}$ = (1,2,1,0,1,0). After precoding $\boldsymbol{y}$, we obtain $\boldsymbol{u} = I$(1,2,1,0,1,0) = (1,3,4,4,0,0). And we may verify that $\boldsymbol{u}$ is balanced since the sum of its entries equals $(q-1)m/2$ = 12.

In order to balance a q-ary word, we must find a pair of integers s and v, $s \in \mathcal{Q}$ and $v \in \{1, \ldots, m\}$. Note that in the binary case, $q = 2$, the search is restricted to finding the balancing index v. There is, except full search, no simple known algorithm available for directly computing s and v. In the prior art construction presented by Swart & Weber, the pair of integers, s and v, is represented as a balanced prefix, which is appended to the balanced codeword, and subsequently transmitted to the receiver. The receiver can uniquely undo the modifications after observation and decoding s and v from the prefix. Note that the encoder requires at least qm distinct balanced prefixes, so that coding of the prefix using look-up tables can be an expensive operation for large q and m. Using the above Theorem, we will show in the next section that the precoding operation in conjunction with a unity-error correcting code may lead to efficient constructions of a balanced codes.

III. PREFIXLESS BALANCED CODES

The next construction exploits Theorem 1 and we will show below that in conjunction with error correcting or detecting codes, it will be possible to efficiently balance q-ary words, and circumvent the encoding and decoding of the prefix in the prior art construction.

Construction 1 We will start by defining a q-ary $(m-1, k)$ linear block code of dimension k and length $m-1$. The encoding function is denoted by ϕ_q. Let $r' = m - 1 - k$ be the redundancy of the block code, and define the $r' \times (m-1)$ matrix $C_{q,r'}$ whose ith column $\boldsymbol{v}_i$ is the q-ary representation of the integer i, $1 \le i \le m-1$, $m \le q^{r'}$. For example, for $q = 3$, $r' = 2$, and $m = 9$ we obtain

$$C_{3,2} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{bmatrix}.$$

We call $C_{q,r'}$ a *check matrix*, for which we have an easy syndrome decoding available similar to that of binary *Hamming* codes [10]. The maximum row length of the check matrix $C_{q,r'}$ is $q^{r'} - 1$, $r' > 1$.

The encoding procedure consists of four steps.

• **Step 1:** the k-symbol pay load $\boldsymbol{u}$ is encoded into the codeword $\boldsymbol{x}' = \{x'_1, \ldots, x'_{m-1}\}$ using the coding function $\boldsymbol{x}' = \phi_q(\boldsymbol{u})$ defined in such a way that $\boldsymbol{x}'$ satisfies

$$C_{q,r'}\boldsymbol{x}'^T = 0.$$

• **Step 2:** the m-symbol word $\boldsymbol{x}$ is obtained by prefixing a redundant '0' to $\boldsymbol{x}'$, that is,

$$\boldsymbol{x} = \{0, x'_1, \ldots, x'_{m-1}\}.$$

• **Step 3:** find a pair of integers, $s \in \mathcal{Q}$ and $v \in \{1, \ldots, m\}$, such that

$$\text{weight}(I(\boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1)) = \frac{m(q-1)}{2}.$$

According to Theorem 1, such a pair of integers s and v can always be found.

• **Step 4:** the balanced m-bit word $I(\boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1)$ is transmitted.

At the receiver's site, the m-symbol word $\boldsymbol{y}$ is retrieved from the received by modulo q differentiation, i.e.

$$\boldsymbol{y} = I^{-1}(I(\boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1)) = \boldsymbol{x} \oplus_q \boldsymbol{u}_v \oplus_q s\boldsymbol{u}_1.$$

We drop the first symbol, $'s'$, of $\boldsymbol{y}$ obtaining the $(m-1)$-symbol $\boldsymbol{y}'$. Then $(m-1)$-symbol words $\boldsymbol{y}'$ and $\boldsymbol{x}'$ differ only at an unknown index position v. As $\boldsymbol{x}'$ satisfies $C_{q,r'}\boldsymbol{x}'^T = 0$, we have

$$C_{q,r'}\boldsymbol{y}'^T = \boldsymbol{v}_v,$$

where $\boldsymbol{v}_v$ is the v-th column of $C_{q,r'}$. So that we can uniquely retrieve the index v, and restore the original word by subtracting '1' from y'_v, that is,

$$\boldsymbol{x}' = \boldsymbol{y}' \ominus_q \boldsymbol{u}_v.$$

By a straightforward reshuffling of the symbols, and removing the redundant symbols, we obtain the original k-symbol pay load $\boldsymbol{u}$. ■

Let r denote the total number of redundant symbols of the balanced code. Since the maximum length of the check matrix $C_{q,r}$ equals $q^{r-1} - 1$, we conclude that the maximum length, $L_q(r)$, of the pay load is

$$L_q(r) = q^{r-1} - r, \quad q > 2. \tag{3}$$

For the binary case, $q = 2$, we deduce that, since the first symbol s can be dropped, and only the index v needs to be encoded

$$L_2(r) = 2^r - r - 1, \tag{4}$$

which is the same value as presented by Knuth [6] using a construction with a prefix. Note that for $q = 2$ the check matrix $C_{2,r}$ defines a regular (binary) Hamming code with redundancy $r' = r$. Swart & Weber proposed a construction with a balanced prefix of length r, where each prefix uniquely

TABLE I
$L_q^{SW}(r)$ AND $L_q(r)$ AS A FUNCTION OF r.

q	r	$L_q^{SW}(r)$	$L_q(r)$
3	4	6	23
3	5	17	76
3	6	47	237
3	7	131	722
3	8	369	2179
3	9	1046	6552
3	10	2984	19672
5	4	17	120
5	5	76	620
5	6	350	3119
5	7	1627	15618
5	8	7633	78117
5	9	36065	390616
5	10	171389	1953115

represents the pair of integers s and v. Let $N_q(r)$ denote the number of distinct q-ary balanced prefixes of length r. Then for Swart and Weber's construction, we require that the length of the pay load, denoted by $L_q^{SW}(r)$, must satisfy

$$L_q^{SW}(r) \leq \left\lfloor \frac{N_q(r)}{q} \right\rfloor .$$

Using generating functions, we can straightforwardly compute the number of distinct q-ary prefixes, $N_q(r)$, of length r. Table I shows, for $q = 3$ and $q = 5$, $L_q(r)$ and $L_q^{SW}(r)$ as a function of r. We conclude from the above table that the redundancy of the new balanced q-ary code is significantly reduced with respect to Swart & Weber's method. Note that Capocelli *et al.* presented a code construction where the length of the pay load is less than

$$\frac{q^r - 1}{q - 1}. \tag{5}$$

For large alphabet q we conclude that the redundancy of the new method is approximately a factor of $q/(q-1)$ higher than that of the prior art construction by Capocelli *et al.*

IV. CONCLUSIONS

We have presented a method for balancing q-ary codewords, where look-up tables for coding and decoding the prefix can be avoided. We have compared the redundancy of the new construction with that of two prior art reference codes. The redundancy of the new construction is much less than that of Swart & Weber's construction, and slightly larger than that of Capocelli *et al.* construction.

REFERENCES

[1] K.A.S. Immink, 'Coding Methods for High-Density Optical Recording', *Philips J. Res.,* vol. 41, pp. 410-430, 1986.

[2] H. Zhou, A. Jiang, and J. Bruck, 'Balanced Modulation for Nonvolatile Memories', Submitted to IEEE Trans. on Inform. Theory, Sept. 2012

[3] J.H. Weber, K.A.S. Immink, P.H. Siegel, and T.G. Swart, 'Polarity-Balanced Codes', 2013 Information Theory and Applications Workshop, San Diego, Feb. 2013.

[4] R.M. Capocelli. L. Gargano, and U. Vaccaro, 'Efficient q-ary immutable codes', *Discrete Applied Mathematics*, vol. 33, pp. 25-41, 1991.

[5] T.G. Swart and J.H. Weber, 'Efficient Balancing of q-ary Sequences with Parallel Decoding', IEEE International Symposium on Information Theory, ISIT2009, pp. 1564-1568, Seoul, June 29 - July 3, 2009.

[6] D.E. Knuth, 'Efficient Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-32, no. 1, pp. 51-53, Jan. 1986.

[7] S. Al-Bassam and B. Bose, 'On Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-36, no. 2, pp. 406-408, March 1990.

[8] L.G. Tallini, R.M. Capocelli, and B. Bose, 'Design of Some New Balanced Codes', *IEEE Trans. Inform. Theory,* vol. IT-42, pp. 790-802, May 1996.

[9] J.H. Weber and K.A.S. Immink, 'Knuth's Balancing of Codewords Revisited', *IEEE Trans Inform Theory,* vol. 56, no. 4, pp.1673-1679, 2010.

[10] R.W. Hamming, *Coding and Information Theory,* Prentice-Hall, Englewood Cliffs, 1986.

[11] S. Al-Bassam and B. Bose, 'Design of Efficient Error-Correcting Balanced Codes', *IEEE Trans. Computers,* vol. 42, no. 10, pp. 1261-1266, October 1993.

[12] J.H. Weber, K.A.S. Immink, and H.C. Ferreira,'Error-Correcting Balanced Knuth Codes', *IEEE Trans. Inform Theory,* vol. IT-58, pp. 82-89, Jan. 2012.

[13] S. Lin and D.J. Costello, *Error Control Coding,* Second Edition, Pearson Prentice-Hall, 2004.