Encoding and Decoding of Balanced q-ary Sequences Using a Gray Code Prefix

Elie N. Mambou and Theo G. Swart
Dept. of Electrical and Electronic Engineering Science, University of Johannesburg
P. O. Box 524, Auckland Park, 2006, South Africa
Email: {emambou, tgswart}@uj.ac.za

***Abstract*—Balancing sequences over a non-binary alphabet is considered, where the algebraic sum of the components (also known as the weight) is equal to some specific value. Various schemes based on Knuth's simple binary balancing algorithm have been proposed. However, these have mostly assumed that the prefix describing the balancing point in the algorithm can easily be encoded. In this paper we show how non-binary Gray codes can be used to generate these prefixes. Together with a non-binary balancing algorithm, this forms a complete balancing system with straightforward and efficient encoding/decoding.**

I. Introduction

Balanced codes have a lot of applications especially in fiber optics, as well as magnetic and optical storage systems [1]. For optical systems, errors occur in the process of recording data, this is due to the low frequencies of interaction between data written on the disc and the servo systems which follow the track. Such errors can be avoided in high-pass filtering by using encoded binary balanced codes as they contain no low frequencies, with a spectral null at the zero frequency. Balanced codes are important in recording systems for tracking and pilot tone insertion. In digital transmission over cables, balanced codes are used to counter the effect of cut-off at low-frequencies due to isolating transformers and coupling effects.

Knuth [2] showed simple and efficient binary balancing schemes, where for a binary sequence $\mathbf{x}$ of even length k, there always exits an integer i, $0 \leq i \leq k-1$, such that by inverting the first i bits the sequence will be balanced, i.e. the same number of zeros and ones. Both serial and parallel schemes were presented. In the serial scheme, a length of $k = 2^r$ information bits is achieved for r redundant bits. The redundant bits are used to indicate the initial weight of the sequence, then the encoding is done by inverting i bits until the balancing value is reached. For the parallel scheme, length $k = 2^r - r - 1$ information bits are used with r redundant bits. Here the redundant bits indicate the number of inverted bits, i.e. the value of i. As soon as i is recovered, decoding can be done by complementing the first i bits simultaneously.

Several algorithms based on Knuth's scheme were developed to balance q-ary sequences. Tallini and Vaccaro [3] developed a generalization of Knuth's serial method, showing that single or double maps can be used to encode sequences that are close to being balanced. Capocelli *et al.* [4] balanced q-ary sequences by partitioning them into different chains with unique weights in each chain. By using two functions that abide to some properties, a prefix is chosen that denotes the original sequence's weight. Another sequence from the chain in which the original sequence is, is then chosen such that the sequence and prefix together is balanced. The maximum size code was obtained when using balanced sequences for the design of q-ary immutable codes. Swart and Weber [5] presented a generalization of Knuth's parallel method, but since we will be making use of this, we will discuss it in more detail in Section II.

A prefixless method was presented in [6]. By using the method from [5] and applying precoding to a very specific error correction code, it was shown that balancing can be achieved without the need for a prefix.

Other related sequence balancing techniques include invariant balancing of codes under symbol permutation [7] and balancing over the m-th roots of unity [8], [9].

All of these schemes in some way rely on a prefix (except for [6]) to send information regarding the balancing point, however this prefix also needs to be balanced. Prior work mostly assumed that this can be done via lookup table or enumerative encoding. In this paper we will present a complete balancing scheme, i.e. balancing of the information together with easy encoding/decoding of the prefix by making use of non-binary Gray codes. In a way our new scheme will be a combination of Knuth's serial and parallel methods.

In Section II we present background on balancing of q-ary sequences, specifically from [5], and non-binary Gray codes. Section III shows our proposed encoding and decoding algorithms, while we investigate the redundancy and complexity in Section IV. Finally, we conclude in Section V.

II. Background

Consider a q-ary information sequence $\mathbf{x} = x_1x_2\ldots x_k$, $x_i \in \{0, 1, \ldots, q-1\}$, of length k. Let the prefix that will be appended to $\mathbf{x}$ be of length r, and let the information and prefix together be denoted by $\mathbf{c} = c_1c_2\ldots c_n$, $c_i \in \{0, 1, \ldots, q-1\}$, of length n, i.e. $n = k + r$. If $w(\mathbf{c})$ represents the weight of $\mathbf{c}$, then the entire sequence is balanced if

$$w(\mathbf{c}) = \sum_{i=1}^{n} c_i = \frac{n(q-1)}{2}.$$

Let β denote this balancing value. This holds for all q and n, except when q is even and n is odd. For the rest of the paper we will not consider sequences with q even and n odd, so that $n(q-1)/2$ is an integer.

In the rest of this section we provide the background of the two essential building blocks of our new scheme: the q-ary balancing scheme from [5] and q-ary Gray codes.

A. Balancing of q-ary Sequences

It has been proven [5] that a q-ary sequence, $\mathbf{x}$, can always be balanced by adding modulo q one sequence from a set of balancing sequences $\mathbf{b}(s,p) = b_1 b_2 \ldots b_k$ generated as follows:

$$b_i = \begin{cases} s, & i-1 \geq p, \\ s+1 \pmod q, & i-1 < p, \end{cases}$$

where s and p are integers with $0 \leq s \leq q-1$, $0 \leq p \leq k-1$. Let z be the iterator through these balancing sequences, with $z = sk + p$, $0 \leq z \leq kq - 1$. We will interchangeably use either $\mathbf{b}(s,p)$ or $\mathbf{b}(z)$ to denote the balancing sequences. Let $\mathbf{y}$ denote the sequence after a balancing sequence is added, i.e. $\mathbf{y} = \mathbf{x} \oplus_q \mathbf{b}(z)$, where $\oplus_q$ denotes modulo q addition. There are kq possible balancing sequences and at least one of them will lead to a balanced output $\mathbf{y}$.

Example 1 For $q = 4$, $n = 4$, consider the sequence 2303. The balancing value is $\beta = 6$.

z	$\mathbf{b}(z)$	$\mathbf{x} \oplus_q \mathbf{b}(z) = \mathbf{y}$	$w(\mathbf{y})$
0	0000	$2303 \oplus_4 0000 = 2303$	8
1	1000	$2303 \oplus_4 1000 = 3303$	9
2	1100	$2303 \oplus_4 1100 = 3003$	6
3	1110	$2303 \oplus_4 1110 = 3013$	7
4	1111	$2303 \oplus_4 1111 = 3010$	4
5	2111	$2303 \oplus_4 2111 = 0010$	1
6	2211	$2303 \oplus_4 2211 = 0110$	2
7	2221	$2303 \oplus_4 2221 = 0120$	3
8	2222	$2303 \oplus_4 2222 = 0121$	4
9	3222	$2303 \oplus_4 3222 = 1121$	5
10	3322	$2303 \oplus_4 3322 = 1221$	6
11	3332	$2303 \oplus_4 3332 = 1231$	7
12	3333	$2303 \oplus_4 3333 = 1232$	8
13	0333	$2303 \oplus_4 0333 = 2232$	9
14	0033	$2303 \oplus_4 0033 = 2332$	10
15	0003	$2303 \oplus_4 0003 = 2302$	7

In this case we also need to send a balanced prefix that represents the index $z = 2$ or $z = 10$, depending on the balancing sequence chosen, so that the receiver can again reconstruct $\mathbf{b}(z)$ to subtract it from $\mathbf{y}$. □

B. q-ary Gray Codes

Gray code theory was invented by Gray [10] and was originally used to solve problems in pulse code communication, before is was extended to several other fields. We consider all q-ary sequences of length r', denoting them by $\mathbf{d} = d_1 d_2 \ldots d_{r'}$, and list them in the normal lexicographic order. These sequences are mapped to Gray code sequences, denoted by $\mathbf{g} = g_1 g_2 \ldots g_{r'}$, in such a way that any two adjacent sequences differ in only one symbol position.

A set of (r', q)-Gray codes is not uniquely defined since a permutation of any column on the set will also result in a new set of (r', q)-Gray codes. However, in this work we consider a specific Gray code as presented by Guan [11], which has the additional property that the difference in weight between any two adjacent sequences is either -1 or $+1$.

Gray code encoding algorithm Let $\mathbf{d} = d_1 d_2 \ldots d_{r'}$ be a q-ary sequence of length r' representing z and $\mathbf{g} = g_1 g_2 \ldots g_{r'}$, its corresponding Gray code sequence. The parity of the sum S_i of the first $i-1$ digits of $\mathbf{g}$ determines the Gray code symbols, where $2 \leq i \leq r'$ and $g_1 = d_1$, then

$$S_i = \sum_{j=1}^{i-1} g_j, \quad \text{and} \quad g_i = \begin{cases} d_i, & \text{if } S_i \text{ is even}, \\ q-1-d_i, & \text{if } S_i \text{ is odd}. \end{cases}$$

Simply put, if S_i is even then the symbol stays the same, else if S_i is odd then the q-ary complement of the symbol is taken.

Table I shows a $(2,4)$-Gray code, where $\mathbf{d}$ is the 4-ary representation of the index z.

TABLE I
EXAMPLE OF $(2,4)$-GRAY CODE

z	$\mathbf{d}$	$\mathbf{g}$	z	$\mathbf{d}$	$\mathbf{g}$	z	$\mathbf{d}$	$\mathbf{g}$	z	$\mathbf{d}$	$\mathbf{g}$
0	00	00	4	10	13	8	20	20	12	30	33
1	01	01	5	11	12	9	21	21	13	31	32
2	02	02	6	12	11	10	22	22	14	32	31
3	03	03	7	13	10	11	23	23	15	33	30

Gray code decoding algorithm If $\mathbf{g} = g_1 g_2 \ldots g_{r'}$ is the Gray code sequence, then $\mathbf{d} = d_1 d_2 \ldots d_{r'}$ is its corresponding q-ary sequence representing the index z to be recovered. As before, the parity of the sum S_i of the first $i-1$ digits of $\mathbf{g}$ determines the original sequence, where $2 \leq i \leq r'$ and $d_1 = g_1$, then

$$S_i = \sum_{j=1}^{i-1} g_j, \quad \text{and} \quad d_i = \begin{cases} g_i, & \text{if } S_i \text{ is even}, \\ q-1-g_i, & \text{if } S_i \text{ is odd}. \end{cases}$$

As an example of how we will use this, Table II presents the decoding of the $(3,3)$-Gray codes to obtain the corresponding balancing sequences.

TABLE II
DECODING OF $(3,3)$-GRAY CODE PREFIXES

Gray code ($\mathbf{g}$)	Sequence ($\mathbf{d}$)	z	s,p	$\mathbf{b}(s,p)$
000	000	0	0,0	000000000
001	001	1	0,1	100000000
002	002	2	0,2	110000000
012	010	3	0,3	111000000
011	011	4	0,4	111100000
010	012	5	0,5	111110000
020	020	6	0,6	111111000
021	021	7	0,7	111111100
022	022	8	0,8	111111110
122	100	9	1,0	111111111
121	101	10	1,1	211111111
120	102	11	1,2	221111111
110	110	12	1,3	222111111
111	111	13	1,4	222211111
112	112	14	1,5	222221111
102	120	15	1,6	222222111
101	121	16	1,7	222222211
100	122	17	1,8	222222221
200	200	18	2,0	222222222
201	201	19	2,1	022222222
202	202	20	2,2	002222222
212	210	21	2,3	000222222
211	211	22	2,4	000022222
210	212	23	2,5	000002222
220	220	24	2,6	000000222
221	221	25	2,7	000000022
222	222	26	2,8	000000002

III. BALANCING WITH GRAY CODE PREFIX

The main idea is for the Gray code to encode the balancing index z, so that it can easily be recovered for decoding. Where the information sequence and prefix both had to be balanced in [5], here we will only require that the information and prefix together be balanced. This is reminiscent of Knuth's serial binary scheme, however, in this case we also retain the advantage of parallel decoding from the parallel scheme.

A. Encoding

We will now formally describe the new encoding scheme. Let a q-ary information sequence, $\mathbf{x}$, of length k be added modulo q to a set of kq balancing sequences, $\mathbf{b}(z)$, with output $\mathbf{y}$ such that $x_1x_2\dots x_k \oplus_q b_1b_2\dots b_k = y_1y_2\dots y_k$. If the Gray code sequence, $g_1g_2\dots g_{r'}$, is then prefixed to all the kq outputs, the final sequence of length $k+r'$ is:

$$\mathbf{c}' = [\,\mathbf{g}\,|\,\mathbf{y}\,] = g_1g_2\dots g_{r'}y_1y_2\dots y_k.$$

We impose a condition on q and k that $k=q^t$, where t is a positive integer, then

$$r' = \log_q(kq) = \log_q(q^{t+1}) = t+1.$$

Therefore it takes at most Gray codes of length $t+1$ to uniquely identify kq balancing sequences. This means that the redundancy is always equivalent to the range of z that we want to encode. This condition is necessary to ensure that the entire range of weights from 0 to $r'(q-1)$ for the Gray code is obtained.

Example 2 For $q=4$, $k=4$, consider the sequence 2302, then with $t=1$, it means we need a Gray code of length $r'=2$.

The cardinality of the $(2,4)$-Gray code is equal to the number of balancing sequences which is 16. The following shows the steps in obtaining the codeword, $\mathbf{c}'$. The Gray code sequence for each z can be verified against Table I.

z	$\mathbf{x}\oplus_q\mathbf{b}(z)=\mathbf{y}$	$\mathbf{c}'$	$w(\mathbf{c}')$
0	$2303 \oplus_4 0000 = 2303$	002303	8
1	$2303 \oplus_4 1000 = 3303$	013303	10
2	$2303 \oplus_4 1100 = 3003$	023003	8
3	$2303 \oplus_4 1110 = 3013$	033013	10
4	$2303 \oplus_4 1111 = 3010$	133010	8
5	$2303 \oplus_4 2111 = 0010$	120010	4
6	$2303 \oplus_4 2211 = 0110$	110110	4
7	$2303 \oplus_4 2221 = 0120$	100120	4
8	$2303 \oplus_4 2222 = 0121$	200121	6
9	$2303 \oplus_4 3222 = 1121$	211121	8
10	$2303 \oplus_4 3322 = 1221$	221221	10
11	$2303 \oplus_4 3332 = 1231$	231231	12
12	$2303 \oplus_4 3333 = 1232$	331232	14
13	$2303 \oplus_4 0333 = 2232$	322232	14
14	$2303 \oplus_4 0033 = 2332$	312332	14
15	$2303 \oplus_4 0003 = 2302$	302302	10

The underlined symbols represent the appended Gray code prefix. □

We note that the first symbol of the Gray code prefix is always equal to s for any balancing sequence $\mathbf{b}(s,p)$, which should be advantageous for implementation.

A (γ,τ)-random walk is defined as a path with increases of γ and decreases of τ. When adding $\mathbf{b}(z)$ to $\mathbf{x}$ and plotting $w(\mathbf{y})$ against z, a $(1,3)$-random walk is obtained for Example 1. In general, a $(1,q-1)$-random walk is obtained, which is always bounded by a minimum below and a maximum above β [5], ensuring that it will pass through β at least once. Now by using the Gray code, the prefix will have increases and decreases of one, and combined with the information sequence's increases and decreases of 1 and $q-1$, respectively, we can control the weight and ensure that it passes through the balancing value.

Returning to Example 1, when the Gray code prefix is appended to the output sequence, it creates a $(\{0;2\},\{2;4\})$-random walk, where there are increases of either 0 or 2 and decreases of either 2 or 4, as shown in the above example. Fig. 1 illustrates the random walks obtained in Examples 1 and 2 before appending the Gray code prefix and after appending it, respectively.

Lemma 1 *Let* $\mathbf{c}' = [\,\mathbf{g}\,|\,\mathbf{y}\,]$ *denote the concatenation of the Gray code prefix with* $\mathbf{y}$. *The weight* $w(\mathbf{c}')$ *plotted against* z *always forms a* $(\{0;2\},\{q-2;q\})$*-random walk.* □

Proof: It has been proven that $w(\mathbf{y})$ vs. z forms a $(1,q-1)$-random walk [5]. Similarly for Gray codes, $w(\mathbf{g})$ vs. z forms a $(1,1)$-random walk. This implies that when combined, $w(\mathbf{c}')$ vs. z forms a $(\{0;2\},\{q-2;q\})$-random walk, with increases of either 0 or 2 and decreases of either $q-2$ or q. ■

For some values of k and q, concatenation of the Gray code prefix with an output sequence does not guarantee the overall sequence to be balanced. This is the case in Example 2, where in Fig. 1(b), the random walk does not pass through the balancing value of $\beta=9$. This happens because the random walk's increases of two will not necessarily go through β. In some cases it will be one unit above or below β.

In order to balance the overall sequence, we append one

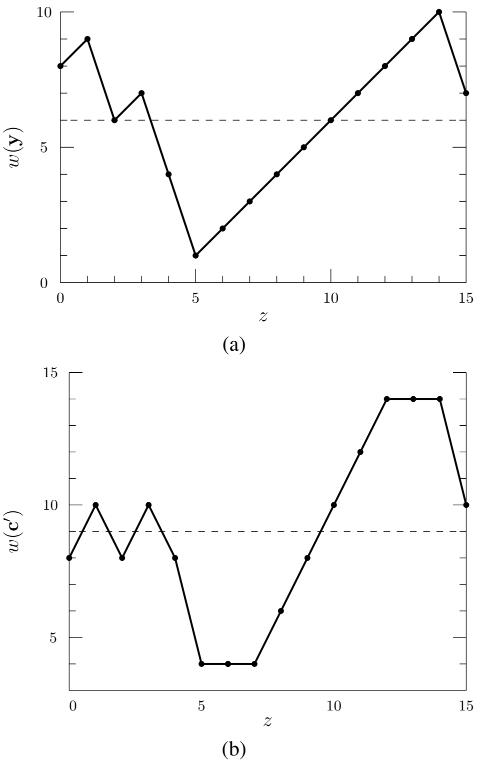

Fig. 1. (a) $(1,3)$-random walk vs. (b) $(\{0;2\},\{2;4\})$-random walk. The dashed line indicates the balancing value.

more redundant symbol, u, to control the weight. For a specific z, if $\beta \geq w(\mathbf{c}')$, then set $u = \beta - w(\mathbf{c}')$, provided that $u \in \{0, 1, \ldots, q-1\}$, otherwise u can be any random symbol. Let $\mathbf{c}$ be the concatenation of u, $\mathbf{g}$ and $\mathbf{y}$:

$$\mathbf{c} = [\, u \,|\, \mathbf{g} \,|\, \mathbf{y}\,] = u\; g_1 g_2 \ldots g_{r'}\; y_1 y_2 \ldots y_k.$$

Now the overall length of the sequence is $n = k + r = k + r' + 1 = k + t + 2$, and the new prefix length and thus the total redundancy is $r = r' + 1 = \log_q k + 2$.

Theorem 1 *Any q-ary sequence of length k can be balanced by adding modulo q an appropriate balancing sequence $\mathbf{b}(z)$, and prefixing a redundant symbol u with a Gray code sequence $\mathbf{g}$ that describes z.* □

Sketch of proof Using the amount of times that each symbol appears in a position, it was shown in [5] that the random walk for the information sequence has an average value of $\frac{k(q-1)}{2}$. A similar approach can be used to show that the random walk for the prefix has an average value of $\frac{r(q-1)}{2}$, considering that every symbol will appear as many times as another in a Gray code, and if we use *all* possible symbols for u. It follows that the average value of the random walk for the information sequence and prefix together is $\frac{(k+r)(q-1)}{2} = \beta$, thus $\min\{w(\mathbf{c})\} \leq \beta \leq \max\{w(\mathbf{c})\}$ and the random walk will pass through β at least once. ■

We are now in a position to present the complete encoding algorithm.

Encoding algorithm The following steps are followed to balance the sequence and find the correct Gray code prefix:

1) Incrementing through z, determine the balancing sequence $\mathbf{b}(s, p)$ and add it to the information sequence $\mathbf{x}$ to obtain $\mathbf{y}$.
2) For each increment, convert z into base q over r' symbols and determine the corresponding Gray code sequence, $\mathbf{g}$, using the Gray code encoding algorithm described in Section II-B.
3) Set $u = \beta - w(\mathbf{c}')$, provided that $u \in \{0, 1, \ldots, q-1\}$, otherwise set $u = 0$.
4) Continue incrementing z until the weight of u, $\mathbf{g}$ and $\mathbf{y}$ together is equal to β.

Example 3 For $q = 3$, $k = 3$, with $t = 1$, we have a Gray code of length two. The total length of the transmitted sequence is $n = 6$ and the balancing value is $\beta = 6$. Now consider encoding the sequence 212:

z	$\mathbf{x} \oplus_q \mathbf{b}(z) = \mathbf{y}$	$\mathbf{c}$	$w(\mathbf{c})$
0	$212 \oplus_3 000 = 212$	$\underline{\mathbf{1}00}212$	6
1	$212 \oplus_3 100 = 012$	$\underline{\mathbf{2}01}012$	6
2	$212 \oplus_3 110 = 022$	$\underline{\mathbf{0}02}022$	6
3	$212 \oplus_3 111 = 020$	$\underline{\mathbf{1}12}020$	6
4	$212 \oplus_3 211 = 120$	$\underline{\mathbf{1}11}120$	6
5	$212 \oplus_3 221 = 100$	$\underline{\mathbf{0}10}100$	2
6	$212 \oplus_3 222 = 101$	$\underline{\mathbf{2}20}101$	6
7	$212 \oplus_3 322 = 201$	$\underline{\mathbf{0}21}201$	6
8	$212 \oplus_3 332 = 211$	$\underline{\mathbf{0}22}211$	8

The underlined symbols represent the appended prefix, and the bold symbol is u, which is chosen such that β is obtained whenever possible. □

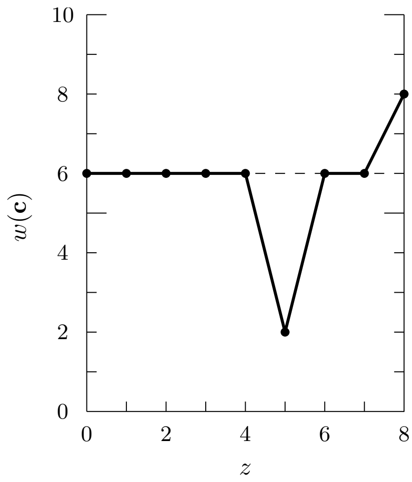

Fig. 2. Adjusted random walk for Example 3. The dashed line indicates the balancing value.

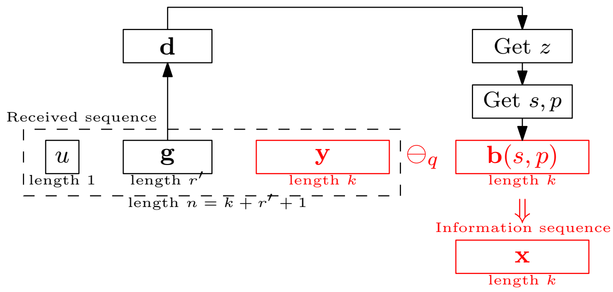

Fig. 3. Flow diagram of the decoding process.

Fig. 2 presents the adjusted random walk of the sequence from Example 3. The flexibility over the digit u will increase the occurrence of balanced outputs. These extra balanced outputs could conceivably be used to send auxiliary data, thus reducing the redundancy. This was proved for the binary case by Weber and Immink [12].

B. Decoding

Fig. 3 shows the decoding process.

Decoding algorithm The following steps are followed to recover the original information from the balanced sequence:

1) Drop the redundant symbol u, then recover z from the Gray code sequence using the decoding algorithm as presented in Section II-B.
2) Use z to determine s and p and then reconstruct $\mathbf{b}(s, p)$.
3) Subtract $\mathbf{b}(s, p)$ from $\mathbf{y}$ to recover the original sequence.

Example 4 Consider the case $q = 3$, $n = 13$, where a sequence was encoded as 1012000122022, with a $(3, 3)$-Gray code (see Table II). The first symbol 1 is dropped, then the Gray code sequence is extracted as 012, and decoded as 010. Thus $z = 3$, leading to $s = 0$ and $p = 3$, and $\mathbf{b}(0, 3) = 111000000$. Finally, the information sequence is recovered as $\mathbf{x} = \mathbf{y} \ominus_q \mathbf{b}(s, p) = 000122022 \ominus_3 111000000 = 222122022$. □

IV. Redundancy and Complexity

A. Redundancy

Let $\mathcal{S}(q,r)$ denote the full set of balanced sequences of length r. For the construction in [5], the balanced prefix is to be chosen from the full balanced set, such that $|\mathcal{S}(q,r)| \leq kq$. Then the information sequence, as a function of r, is

$$k \leq q^{r-1}\sqrt{\frac{6}{\pi r(q^2-1)}}, \tag{1}$$

where we made use of an approximation of $|\mathcal{S}(q,r)|$ from [4].

The first construction in [4] has an information sequence of length

$$k \leq \frac{q^r-1}{r-1}. \tag{2}$$

The second construction, using a similar approach but more complex, results in

$$k \leq 2\frac{q^r-1}{r-1} - r. \tag{3}$$

The construction from [3] makes use of both balancing and compression. In this case the information length is bounded by

$$k \leq \frac{1}{1-2\gamma}\frac{q^r-1}{r-1} - a_1(q,\gamma)r - a_2(q,\gamma),$$

with $\gamma \in [0, \frac{1}{2})$, where a_1 and a_2 are scalars depending of q and γ. If the compression aspect is ignored, the information sequence length is the same as in (3). The prefixless construction from [6] has information sequences of length

$$k = q^{r-1} - r. \tag{4}$$

For our new construction, the redundancy is given by $r = \log_q k + 2$, where $k = q^t$, which results in an information sequence of length

$$k = q^{r-2}. \tag{5}$$

Figure 4 shows a comparison of the information length versus the redundancy for the constructions discussed above. For $q = 3$ we can see that the new algorithm's information length is comparable to those of previous constructions as r increases. However, for $q = 16$ it is only comparable to the construction from [5]. In both cases the new construction is slightly better than the construction it is based on.

B. Complexity

In terms of complexity, previous schemes discussed in [3] and [4], require $\mathcal{O}(qk\log_q k)$ digit operations for the encoding and decoding processes. The scheme in [5] requires $\mathcal{O}(qk\log_q k)$ digit operations for the encoding and $\mathcal{O}(k)$ digit operations for the decoding. In addition to the complexity of the scheme in [5], our new scheme needs $\mathcal{O}(\log_q k)$ to encode/decode the Gray code prefix.

The decoding process requires $\mathcal{O}(k + \log_q k)$ digit operations. In terms of decoding, we still retain the advantage of fast parallel decoding, once $\mathbf{b}(s,p)$ has been determined from the Gray code. Seeing as the algorithm have to test each balancing sequence together with the Gray code until the overall sequence is balanced, the encoding takes longer than decoding. In the worst case where the kq-th balancing sequence and Gray code results in balancing, $\mathcal{O}(qk\log_q k)$ digit operations are needed, since $\mathcal{O}(qk\log_q k) \gg \mathcal{O}(\log_q k)$.

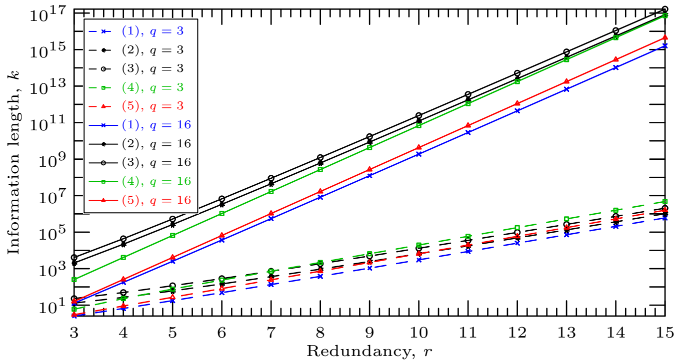

Fig. 4. Comparison of information sequence length vs. redundancy.

V. Conclusion

A simple algorithm was presented to encode and decode balanced q-ary information sequences of length k, where $k = q^t$. The algorithm is based on a Gray code prefix that encodes the balancing index. Both the balancing and Gray code algorithms are efficient in the sense that only simple addition and subtraction operations are needed, and no lookup tables are needed. The majority of the decoding algorithm can also be performed in parallel. Future work includes an investigation into whether the extra symbol u can be eliminated for certain parameters of k and q, and extending the scheme to sequences of length k where $k \neq q^t$.

Acknowledgment

This work is based on research supported in part by the National Research Foundation of South Africa (UID 77596).

References

[1] K. A. S. Immink "Spectral null codes," *IEEE Trans. Magn.*, vol. 26, no. 2, pp. 1130–1135, Mar. 1990.

[2] D. E. Knuth, "Efficient balanced codes," *IEEE Trans. Inform. Theory*, vol. 32, no. 1, pp. 51–53, Jan. 1986.

[3] L. G. Tallini and U. Vaccaro, "Efficient m-ary immutable codes," *Discrete Applied Mathematics*, vol. 92, pp. 17–56, 1999.

[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," in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, Jun. 28 – Jul. 3, 2009, pp. 1564–1568.

[6] T. G. Swart and K. A. S. Immink, "Prefixless q-ary balanced codes with ECC," in *Proc. IEEE Inform. Theory Workshop*, Seville, Spain, Sep. 9–13, 2013.

[7] R. Mascella and L. G. Tallini, "Efficient m-ary balanced codes which are invariant under symbol permutation," *IEEE Trans. Comput.*, vol. 55, no. 8, pp. 929–946, Aug. 2006.

[8] A. Baliga and S. Boztaş, "Balancing sets of non-binary vectors", in *Proc. IEEE Int. Symp. Inform. Theory*, Lausanna, Switzerland, Jun 30–Jul 5, 2002, p. 300.

[9] R. Mascella, L. G. Tallini, S. Al-Bassam and B. Bose, "On efficient balanced codes over the mth roots of unity," *IEEE Trans. Inform. Theory*, vol. 52, no. 5, pp. 2214–2217, May 2006.

[10] F. Gray, "Pulse code communication," *U. S. Patent 2632058*, Mar. 1953.

[11] D.-J. Guan, "Generalized Gray codes with applications," in *Proc. National Science Council, Republic of China, Part A*, vol. 22, no. 6, pp. 841–848, 1998.

[12] 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, Apr. 2010.