Moment Balancing Templates for (d, k) Constrained Codes and Run-Length Limited Sequences

Ling Cheng, Hendrik C. Ferreira and Izak Broere

***Abstract*—The first-order moment of (d, k) constrained codes is investigated in this paper. A generalized moment balancing template is proposed to encode a (d, k) sequence into a single insertion or deletion correcting codeword without losing the constraint property. By relocating 0's in moment balancing runs, which appear in a pairwise manner, of a (d, k) sequence, the first-order moment of this sequence can be modified to satisfy the Varshamov-Tenengolts construction. With a reasonably large base in the modulo system introduced by the Varshamov-Tenengolts construction, this generalized moment balancing template can be applied to run-length limited sequences. The asymptotic bound of the redundancy introduced by the template for (d, k) sequences is of the same order as the universal template for random sequences, and therefore the redundancy is small and suitable for long sequences of practical interest.**

***Index Terms*—(d, k) constrained code, insertions/deletions, moment balancing template, number theoretic codes, run-length limited sequence.**

I. Introduction

The (d, k) constrained codes and run-length limited sequences are widely used in magnetic and optic recording systems. However, as shown in [1], insertion or deletion errors cause catastrophic problems for these systems. To protect a transmission system from a single insertion/deletion error, Ferreira *et al.* [2] presented a moment balancing scheme, in which a linear error correcting codeword is further encoded in order to obtain a single insertion/deletion error correcting capability. However, preserving the constraint properties and keeping the redundancies within a practical-interest range make moment balancing templates of constrained codes more complicated.

Ferreira *et al.* [2] also presented the moment balancing template for *DC*-free sequences and the template for (d, k) sequences with $d = 1$ and $d = 2$. Cheng, Ferreira and Ouahada [3], [4] presented the moment balancing template for further spectral null codes as introduced in [5]. In this paper, we will generalize the template to (d, k) constrained codes and run-length limited sequences.

This paper was presented in part at the IEEE International Symposium on Information Theory, Austin, TX, USA, June 2010.

L. Cheng now is with the School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, Wits. 2050, Johannesburg, South Africa. (email: ling.cheng@wits.ac.za). The work has been partly done while the first author was pursuing a doctoral degree with the University of Johannesburg. H. C. Ferreira is with the Department of Electrical and Electronic Engineering Science, University of Johannesburg, Auckland Park, 2006, South Africa. (e-mail: hcferreira@uj.ac.za). Izak Broere is with the Department of Mathematics, University of Johannesburg, Auckland Park, 2006, South Africa. (e-mail: izakb@uj.ac.za).

The paper is organized as follows. Section II starts with definitions, notations and a brief introduction of (d, k) constrained codes and moment balancing templates. The generalized moment balancing template for (d, k) constrained codes is presented in Section III. In Section IV, an approach to apply this template to run-length limited sequences is presented. The encoding and decoding procedures of implementing this template are presented in Section V. The redundancy of the new templates is discussed in Section VI. We conclude the paper with Section VII.

II. Preliminaries

A. (d, k) constrained codes and run-length limited sequences

Here we define the weight of a sequence as follows.

Definition 1: The *weight* of a binary sequence is equal to the number of 1's in the sequence.

The definition of runs is as follows.

Definition 2: A 0 *run* is a maximal subsequence that consists of consecutive 0's.

A power notation is used to represent a run of symbols, such as 0^a. Here a denotes the number of the repeating 0's.

Following the arguments presented in [6] and [7], we define the *reduced length sequence* of a (d, k) sequence as follows.

Definition 3: The *reduced length sequence* of the w-weight (d, k) sequence $0^{a_1}10^{a_2}1\ldots0^{a_w}1$ is the sequence $a_1a_2\ldots a_w$, where $a_i \in \{d, d+1, \ldots, k\}$ for each i with $1 \le i \le w$.

Note that we assume that all (d, k) sequences we consider in this paper terminate with a 1. Evidently, (d, k) sequences are in a one-to-one correspondence with reduced length sequences.

Let n denote the length of the (d, k) sequence. Based on Definition 3, we have

$$n = \sum_{i=1}^{w}(a_i + 1),$$

and

$$(d+1)w \le n \le (k+1)w. \tag{1}$$

In the modulation stage, a (d, k) sequence is converted to a run-length limited sequence and sent over the channel. In a bipolar modulation system, signals can be represented by sequence $z_1z_2\ldots z_i\ldots$ and $z_i \in \{-1, 1\}$. A bit one in the (d, k) sequence represents a transfer from signals -1 to 1 or 1 to -1, and a bit zero represents a transfer from -1 to -1 or 1 to 1. A (d, k) sequence can be converted into a run-length limited sequence with the same length, if it is assumed that the first bit of the (d, k) sequence indicates the transfer of z_1 from a known signal. Except for the first run of a run-length limited sequence, the run-lengths of this sequence converted from a (d, k) sequence are between $d + 1$ and $k + 1$.

B. Moment Balancing Template

Definition 4: For a binary sequence $\mathbf{v} = v_1 v_2 \dots v_n$, the *first-order moment function* is defined as

$$\sigma(\mathbf{v}) = \sum_{i=1}^{n} i v_i. \tag{2}$$

Given fixed non-negative integers r and m, the Varshamov-Tenengolts codes [8] consist of all binary sequences $\mathbf{v}$ satisfying

$$\sigma(\mathbf{v}) = \sum_{i=1}^{n} i v_i \equiv r \pmod{m}. \tag{3}$$

By taking $m = n + 1$, the corresponding codes can correct one single asymmetrical error. Levenshtein [9] noted that by taking $m \geq n + 1$, the corresponding codes are single insertion or deletion error correcting codes.

Based on the Varshamov-Tenengolts construction in (3), we propose a moment balancing template to encode a binary code C of length K into a single insertion or deletion error correcting code. As shown in [2], [3] and [4], the binary code C can be any one of: an arbitrary random sequence, or an error correcting code, or a DC-free code, or a spectral null code, or a (d, k) constrained sequence with $d = 1$ or $d = 2$. Each codeword $\mathbf{u} = (u_1 u_2 \cdots u_K)$ is encoded into a sequence $\mathbf{v} = (v_1 v_2 \cdots v_n)$, where $n \geq K$. The encoding is done in such a way that, if $\mathbf{u}$ and $\mathbf{u}'$ are different, then the sequences $\mathbf{v}$ and $\mathbf{v}'$ resulting from them are also different. When the first-order moment of the sequence $\mathbf{v}$ satisfies (3) with $m \geq n + 1$, $\mathbf{v}$ can correct a single insertion or deletion error. For practical interest, in the moment balancing templates we propose, bits of $\mathbf{u}$ appear in $\mathbf{v}$ at predetermined positions.

The *moment distribution* of code C can be represented by a generating function as

$$g(X) = n_0 + n_1 X + n_2 X^2 + \cdots + n_r X^r + \cdots + n_{m-1} X^{m-1}.$$

Here n_r is the number of codewords $\mathbf{v}$ in C that have $\sigma(\mathbf{v}) \equiv r \pmod{m}$. A universal moment balancing template is defined in [2] for the code C with an arbitrary moment distribution, in which the number of moment balancing bits $n - K$ is bounded from below by $\lceil \log_2 m \rceil$. For a code having a certain narrower range of moment values, such as one in which for some r's we have $n_r = 0$, we can implement an optimized moment balancing template, which may introduce less redundancy than that of the universal template. In [3] and [4], an optimized template is implemented on spectral null codes. However, when a universal template is applied to a constrained code, the moment balanced codewords do not always maintain the property of the constrained code. It prompts the research on designing certain templates for constrained codes.

III. Moment Balancing Template for (d, k) Constrained Codes

In the sequel, unless stated otherwise, the reduced length sequences will be used to present the properties of the corresponding (d, k) sequences in a moment balancing template.

From (2) it follows that the first-order moment of a (d, k) sequence is completely determined by the indices of the 1's. Given a (d, k) sequence $\mathbf{v}$ with the corresponding reduced length sequence $\mathbf{a} = a_1 a_2 \dots a_w$, we have

$$\begin{aligned} \sigma(\mathbf{v}) &= \sum_{i=1}^{w} \sum_{j=1}^{i} (a_j + 1) \\ &= \sum_{i=1}^{w} \sum_{j=1}^{i} 1 + \sum_{i=1}^{w} \sum_{j=1}^{i} a_j \\ &= \frac{(w+1)w}{2} + \sum_{i=1}^{w} a_i (w - i + 1). \end{aligned} \tag{4}$$

Therefore, the *first order moment function* of a binary (d, k) sequence $\mathbf{v}$ can be presented as the function $\phi(\cdot)$ in terms of the corresponding reduced length sequence $\mathbf{a} = a_1 a_2 \dots a_w$

$$\sigma(\mathbf{v}) = \phi(\mathbf{a}) = \frac{(w+1)w}{2} + \sum_{i=1}^{w} a_i (w - i + 1). \tag{5}$$

We define a relocation of a 0 in a (d, k) sequence represented by the reduced length sequence as follows.

Definition 5: A *relocation* operation of a 0 denoted by $\overset{S,T}{\to}$ with $S \neq T$ inserts a 0 in the T'th run of 0's and deletes a 0 from the S'th run of 0's. It transforms the (d, k) sequence $\dots a_S \dots a_T \dots$ into a sequence $\dots a_S - 1 \dots a_T + 1 \dots$, or it transforms $\dots a_T \dots a_S \dots$ into a sequence $\dots a_T + 1 \dots a_S - 1 \dots$.

Let S and T denote the indices of runs the transform affects where we assume $S \neq T$ and $a_S \geq 1$, since it is impossible for the relocation operation to affect a non-existing 0.

Lemma 1: The first-order moment value defined by (2) of a (d, k) sequence is increased by $S - T$ as a result of a relocation operation $\overset{S,T}{\to}$.

Proof: For $S > T$, let $\mathbf{a} = a_1 a_2 \dots a_T \dots a_S \dots a_w$. After moving a 0 from the S'th run to the T'th run, we have $\mathbf{a}' = a_1 a_2 \dots a_T + 1 \dots a_S - 1 \dots a_w$. According to (5), we have

$$\begin{aligned} \phi(\mathbf{a}') - \phi(\mathbf{a}) &= (a_S - 1)(w - S + 1) + (a_T + 1)(w - T + 1) \\ &\quad - a_S (w - S + 1) - a_T (w - T + 1) \\ &= S - T. \end{aligned}$$

A similar argument is used for $S < T$. ■

Given an original (d, k) sequence $\mathbf{x} = a_1 a_2 \dots a_w$, the *moment balancing template* is an approach to encode $\mathbf{x}$ into $\mathbf{x}' = b_1 b_2 \dots b_{w'}$. Each a_i with $1 \leq i \leq w$ appears in $\mathbf{x}'$ with a predetermined index, which can be represented explicitly by the function

$$\beta : \{1, 2, \dots, w\} \to \{1, 2, \dots, w'\},$$

which is order-preserving, i.e., which satisfies the implication that if $i < j$ in $\{1, 2, \dots, w\}$, then $\beta(i) < \beta(j)$ in $\{1, 2, \dots, w'\}$. Therefore, we have

$$1 \leq \beta(1) < \beta(2) < \cdots < \beta(w) \leq w'.$$

The sequence $\mathbf{x}$ is a subsequence of $\mathbf{x}'$, and the remaining part in $\mathbf{x}'$ is composed of *moment balancing runs*. The moment balancing runs $c_1 c_2 \dots c_{w'-w}$ appear in this order in $\mathbf{x}'$, i.e., with indices which are represented by an order-preserving function

$$\gamma : \{1, 2, \dots, w' - w\} \to \{1, 2, \dots, w'\}.$$

Note that we have

$$1 \leq \gamma(1) < \gamma(2) < \cdots < \gamma(w' - w) \leq w'$$

and that the sets $\{\beta(1), \beta(2), \ldots, \beta(w)\}$ and $\{\gamma(1), \gamma(2), \ldots, \gamma(w' - w)\}$ satisfy

$$\{\beta(1), \beta(2), \ldots, \beta(w)\} \cup \{\gamma(1), \gamma(2), \ldots, \gamma(w'-w)\} = \{1, 2, \ldots, w'\}$$

and

$$\{\beta(1), \beta(2), \ldots, \beta(w)\} \cap \{\gamma(1), \gamma(2), \ldots, \gamma(w' - w)\} = \emptyset.$$

Note that for a given (d, k) sequence $\mathbf{x} = a_1 a_2 \ldots a_w$ and a sequence of balancing runs $\mathbf{c} = c_1 c_2 \ldots c_{w'-w}$, the moment-balanced sequence $\mathbf{x}' = b_1 b_2 \ldots b_{w'}$ with increased weight w' can be generated by a one-to-one mapping relation $\mathbf{x} \to \mathbf{x}'$, where in $\mathbf{x}'$, $b_{\beta(i)} = a_i$ and $b_{\gamma(j)} = c_j$ for each $1 \leq i \leq w$, $1 \leq j \leq w' - w$.

We will show that, for every $\mathbf{x}$ and for any given non-negative integers r and m, one can find a $\mathbf{c}$ such that the first-order moment of $\mathbf{x}'$ satisfies (3).

Without losing generality, we only consider (d, k) sequences with constant weight, in other words, w and w' here are given with $w' > w$. Among w' runs in the above template, there are s pairs of balancing runs, that is, $2s$ runs. The idea behind the moment balancing template we present in this paper is to manipulate the first-order moment value of a (d, k) sequence to a constant positive integer r modulo a constant positive integer $m > n$ by redistributing 0's within the balancing pairs of runs while maintaining the total number of 0's of each balancing-run pair as a constant. We need to consider $m \geq n + 1$, which guarantees a single insertion/deletion correcting capability of a sequence. Note that $m = n + 1$ is the least value which guarantees this error correcting capability, and the cardinality bound from below of the corresponding code is known to drop (see [9]) with higher values of m.

Let $c_{i,1}$ and $c_{i,2}$ denote a pair of balancing runs, where $1 \leq i \leq s$. Using this notation, we have $\mathbf{c} = c_{1,1} c_{2,1} \ldots c_{i,1} \ldots c_{s,1} c_{s,2} \ldots c_{i,2} \ldots c_{2,2} c_{1,2}$. Each pair of balancing runs $c_{i,1}$ and $c_{i,2}$ appears in $\mathbf{c}$ in a mirror-symmetric way. It is evident that in the natural order of the balancing runs, $c_{i,1}$ appears in position i, and $c_{i,2}$ appears in position $2s - i + 1$.

As mentioned earlier, each balancing-run pair has a constant sum. It is evident that when

$$c_{i,1} + c_{i,2} = d + k,$$

$c_{i,1}$ and $c_{i,2}$ reach their maximum dynamic range. Since the moment balance template is composed of the original w runs and $2s$ balancing runs, we have

$$w' = 2s + w.$$

We introduce a symbol for non-negative integer α, of which the value is to be decided later. A two-step moment balancing process is as follows. In the first step the first α pairs of balancing runs are processed with relocation operations, and in the second step, a further ξ pairs of balancing runs are processed. We have

$$s = \alpha + \xi.$$

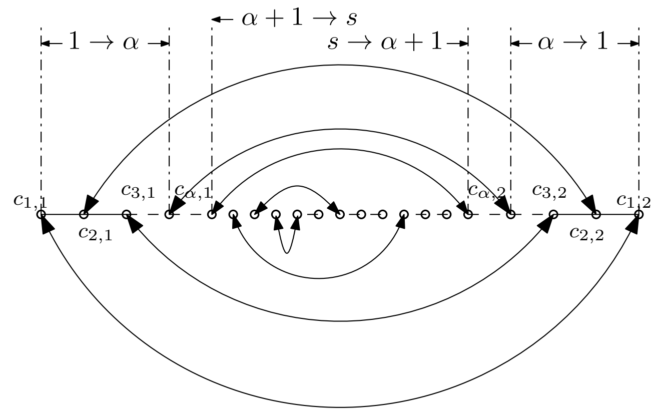

Fig. 1. Layout of balancing runs.

Symbolically, the template can be described in more detail as follows. Given a (d, k) sequence $\mathbf{x} = a_1 a_2 \ldots a_w$ and a sequence of balancing runs $\mathbf{c} = c_{1,1} c_{2,1} \ldots c_{s,1} c_{s,2} \ldots c_{2,2} c_{1,2}$, the moment balanced sequence is

$$\begin{aligned} \mathbf{x}' = {} & c_{1,1} c_{2,1} \ldots c_{s,1} c_{s,2} \\ & a_1 a_2 \ldots a_{k-d+1-3} c_{s-1,2} \\ & a_{k-d+1-2} a_{k-d} \ldots a_{(k-d+1)^2-5} c_{s-2,2} \\ & a_{(k-d+1)^2-4} \ldots a_{(k-d+1)^3-7} c_{s-3,2} \ldots \\ & \ldots a_{(k-d+1)^i-2i} \ldots a_{(k-d+1)^{i+1}-2i-3} c_{s-i-1,2} \ldots \\ & \ldots a_{(k-d+1)^{\xi-2}-2(\xi-2)} \ldots a_{(k-d+1)^{\xi-1}-2(\xi-2)-3} c_{s-\xi+1,2} \\ & a_{(k-d+1)^{\xi-1}-2(\xi-1)} \ldots a_w c_{\alpha,2} c_{\alpha-1,2} \ldots c_{2,2} c_{1,2} \end{aligned} \tag{6}$$

We have the following specifications of the indices of balancing runs in the template

$$\gamma(i) = \begin{cases} i & \text{for } 1 \leq i \leq s \\ 2s - i + 1 + (k - d + 1)^{i-s-1} & \text{for } s + 1 \leq i \leq s + \xi \\ w' + i - 2s & \text{for } s + \xi + 1 \leq i \leq 2s \end{cases} \tag{7}$$

The layout of balancing runs in the template can be illustrated by Fig 1.

An example to describe the moment balancing process follows.

Example 1: The reduced length sequence $\mathbf{x}$ = 16474645247656534377 ($d = 1$, $k = 7$) has length $w = 20$. Let $\alpha = 1$ and $\xi = 2$. Let underlined digits denote the balancing runs. Inserting the balancing reduced length sequence $\mathbf{c} = \underline{111777}$ with indices as described in (7), we obtain $\mathbf{x}' = \underline{1117}1647\underline{7}4645247656534377\underline{7}$ with length $w' = 26$ and length 146. Taking $m = 147$, we have $\phi(\mathbf{x}') \equiv 35 \pmod{147}$. The aim of redistributing 0's in the balancing runs is to obtain $\phi(\mathbf{x}') \equiv 0 \pmod{147}$. There are various choices to make up the deficiency $147 - 35 = 112$. We only present one, i.e., $112 = 4 \times 25 + 1 \times 7 + 5$. As a result of changing $\mathbf{x}'$ into $\underline{5262}1647\underline{6}4645247656534377\underline{3}$, four 0's for the first pair of balancing runs, one 0 for the second pair and five 0's for the third pair are relocated.

Lemma 2: If $c_{i,1} = d$ and $c_{i,2} = k$, where $i \in \{1, 2, \ldots \alpha\}$, by relocating the 0's from $c_{i,2}$ to $c_{i,1}$, the maximum modification in the moment value that can be attained through the processing of α pairs is $(k - d)(\alpha w' - \alpha^2)$.

Proof: Initially, let $c_{i,1} = d$ and $c_{i,2} = k$ for each $i \in \{1, 2, \ldots \alpha\}$.

By relocating the 0's from $c_{i,2}$ to $c_{i,1}$, the maximum modification in the moment value that can be attained through the processing of α pairs is

$$\begin{aligned}\Delta = \phi(\mathbf{x}') - \phi(\mathbf{x}) &\leq (k-d)(w'-1) + (k-d)(w'-3) + \cdots \\ &\quad + (k-d)(w'-2\alpha+1) \\ &= (k-d)(\alpha w' - \alpha^2),\end{aligned}$$

where $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ denote the moment values before and after. Note that to preserve the constraint property of the (d, k) sequence, the maximum number of 0's which can be moved from $c_{i,2}$ to $c_{i,1}$ is $k - d$. It is evident that if the original template starts with $c_{i,1} = d$ and $c_{i,2} = k$, by moving the 0's between pairwise balancing runs, the moment value monotonically increases. Therefore, we have

$$0 \leq \Delta \leq (k-d)(\alpha w' - \alpha^2).$$

■

However, Δ does not take on all the values from 0 to $(k - d)(\alpha w' - \alpha^2)$. The value distribution of Δ can be explicitly described by a generating function as follows

$$\begin{aligned}g(X) &= (1 + X^{w'-1} + X^{2(w'-1)} + \cdots + X^{(k-d)(w'-1)}) \\ &\quad (1 + X^{w'-3} + X^{2(w'-3)} + \cdots + X^{(k-d)(w'-3)}) \\ &\quad \vdots \\ &\quad (1 + X^{w'-2\alpha+1} + X^{2(w'-2\alpha+1)} + \cdots + X^{(k-d)(w'-2\alpha+1)}) \\ &= 1 + X^{w'-2\alpha+1} + X^{w'-2\alpha+3} + \cdots + X^{(k-d)(\alpha w'-\alpha^2)}.\end{aligned} \tag{8}$$

In (8), the last occurrence of $\cdots$ represents all power terms higher than $X^{w'-2\alpha+3}$ and less than $X^{(k-d)(\alpha w'-\alpha^2)}$ with coefficients greater than 0, in increasing order of powers of X. We consider the absolute values of the difference δ, called deficiency, of every two consecutive exponents of the power series, and we investigate the maximum deficiency, $\delta_{\max}$, which cannot be balanced by 2α balancing runs.

Lemma 3: If $\alpha \geq 1$ and $w' \geq 2(k-d)\alpha + 1$, then $\delta_{\max} = w' - 2\alpha + 1$.

Proof: According to (8), it is observed that between 1 and $X^{w'-2\alpha+1}$ there is no term. Therefore we have a deficiency $w' - 2\alpha + 1$. Given a positive integer $1 \leq \lambda \leq k-d$, between $X^{\lambda(w'-1)}$ and $X^{(\lambda-1)(w'-1)}$, it is guaranteed to have $X^{\lambda(w'-3)}$, $X^{\lambda(w'-5)}$, $\ldots$, $X^{\lambda(w'-2\alpha+1)}$. Therefore, between $X^{\lambda(w'-1)}$ and $X^{(\lambda-1)(w'-1)}$, we have maximum deficiency

$$\delta \leq \max(2\lambda, \lambda(w' - 2\alpha + 1) - (\lambda - 1)(w' - 1)).$$

It is evident that we have

$$\delta \leq \lambda(w' - 2\alpha + 1) - (\lambda - 1)(w' - 1),$$

when

$$w' \geq 2(k-d)\alpha + 1.$$

Comparing δ with $w' - 2\alpha + 1$, we have

$$\begin{aligned}w' - 2\alpha + 1 - \delta &\geq w' - 2\alpha + 1 \\ &\quad - (\lambda(w' - 2\alpha + 1) - (\lambda - 1)(w' - 1)) \\ &= 2(\lambda - 1)(\alpha - 1).\end{aligned}$$

Therefore, when $\alpha \geq 1$ and $w' \geq 2(k-d)\alpha + 1$, we have $\delta_{\max} = w' - 2\alpha + 1$. ■

In (6), ξ pairs of balancing runs are implemented to compensate for the deficiency.

Lemma 4: The maximum deficiency $\delta_{\max}$ in (8) can be attained by ξ pairs of balancing runs, where ξ is an integer that satisfies $(k-d+1)^{\xi-1} - 2\xi < w \leq (k-d+1)^{\xi} - 2\xi - 2$.

Proof: Let τ_i denote the number of 1's between the pair of the balancing runs $c_{\alpha+i+1,1}$ and $c_{\alpha+i+1,2}$, where $0 \leq i \leq \xi - 1$. According to (7), we have

$$\tau_i = (k-d+1)^i.$$

Only the 0's in any balancing run can be relocated to the corresponding paired-up balancing run. According to Lemma 1, as a result of the relocations between ξ pairs of balancing runs, the difference of the first-order moment value can be presented as

$$\phi(\mathbf{x}') - \phi(\mathbf{x}) = \sum_{i=0}^{\xi-1} q_i (k-d+1)^i,$$

where $0 \leq q_i \leq k - d$.

Therefore, $\phi(\mathbf{x}') - \phi(\mathbf{x})$ can take on the values of all integers between 0 and $(k-d+1)^{\xi} - 1$.

According to Lemma 3, the maximum first-order moment value deficiency for balancing a moment template is $w' - 2\alpha + 1$. The condition

$$(k-d+1)^{\xi} - 1 \geq w' - 2\alpha + 1 \tag{9}$$

is necessary for the moment balancing template (6) to stand.

From (9), we have

$$w' \leq 2\alpha + (k-d+1)^{\xi} - 2. \tag{10}$$

Furthermore, all indices of balancing runs need to be disjoint. Based on the observation of (7), $\gamma(i)$ increases monotonically for $1 \leq i \leq s$. For $s + 1 \leq i \leq s + \xi$,

$$\gamma(i+1) - \gamma(i) = (k-d)(k-d+1)^{i-s-1} - 1.$$

This shows that the indices of balancing runs specified by (7) also increase monotonically, except that, when $k - d = 1$, in which case $\gamma(s+1) = \gamma(s+2)$. In this case, we can move the s'th pair of balancing runs to any consecutive positions that are not occupied by other balancing runs. For $s+\xi+1 \leq i \leq 2s$, $\gamma(i)$ decreases monotonically.

Moreover, it is evident that $\gamma(s+1) > \gamma(s)$. If

$$\gamma(s + \xi + 1) > \gamma(s + \xi), \tag{11}$$

all indices of balancing runs are disjoint. From (11), we can obtain

$$w' + s + \xi + 1 - 2s > 2s - (s + \xi) + 1 + (k-d+1)^{\xi-1},$$

and

$$w' > 2\alpha + (k-d+1)^{\xi-1}. \tag{12}$$

Therefore, based on (10) and (12), the necessary condition for this template to stand is

$$2\alpha + (k-d+1)^{\xi-1} < w' \leq 2\alpha + (k-d+1)^{\xi} - 2. \tag{13}$$

Since

$$w' = w + 2s = w + 2(\alpha + \xi), \tag{14}$$

Substituting (14) into (13), we have

$$(k - d + 1)^{\xi - 1} - 2\xi < w \leq (k - d + 1)^{\xi} - 2\xi - 2. \tag{15}$$

■

Theorem 1: Let α and ξ pairs of balancing runs be chosen to satisfy (15) and

$$(k - d)(\alpha w' - \alpha^2) + (k - d + 1)^{\xi} - 1 \geq (k + 1)w'. \tag{16}$$

Then, the sequence (6) is a valid moment balancing template that can be used to encode any (d, k) sequence of weight w and $k - d \neq 1$, by relocating 0's among balancing runs, into a (d, k) constrained code of weight w' that can correct a single insertion or deletion error.

Proof: According to Lemma 2, 3 and 4, the first-order moment value of the template (6) can be adjusted for a difference ranging from 0 to $(k-d)(\alpha w' - \alpha^2) + (k-d+1)^{\xi} - 1$ by relocating 0's among balancing runs.

In (3), we choose $m = n + 1$. Then, $m \leq (k + 1)w' + 1$. If $(k - d)(\alpha w' - \alpha^2) + (k - d + 1)^{\xi} - 1 \geq (k + 1)w'$, we have $(k-d)(\alpha w' - \alpha^2) + (k-d+1)^{\xi} - 1 \geq m - 1$. Therefore, (3) can be satisfied for any given value of r. ■

Note that, in the case of $k - d = 1$, Theorem 1 still stands if the s'th balancing pair in (6) can be placed in any two consecutive positions that are not occupied by other balancing runs.

IV. Moment Balancing Template for Run-Length Limited Sequences

Due to the importance of run-length limited sequences in practice, there is great interest in proposing an adaptive approach to implement the moment balancing templates on this type of sequences. As mentioned before, a (d, k) sequence is converted into a $(d + 1, k + 1)$ run-length limited sequence and sent over the channel. Here the analogous notation, *i.e.*, $(d + 1, k + 1)$, is used to indicate that the run-lengths of the sequences over the channel are between $d + 1$ and $k + 1$. The following example enumerates all possible scenarios for a single insertion or deletion error occurring in a run-length limited sequence. Note that, in this section, the (d, k) sequences are presented in binary format. It is convenient to assume that all (d, k) sequences in the following example start with a run of zeros. For example, the first run of the (d, k) sequence 111001 is a zero run with length 0 ($d = 0$).

Example 2: Given a (d, k) sequence 111001, if it is assumed that the known (dummy) signal before the first signal is 1, the (d, k) sequence is converted into the run-length limited sequence $-1, 1, -1, -1, -1, 1$ and sent over the channel.

1) If the received sequence is $-1, 1, -1, -1, 1$ as a result of the deletion of the third signal, the corresponding corrupted (d, k) sequence is 11101. Due to the deletion of a signal from a run of repeating signals with length not less than two ($d \geq 1$), one of the zeros is deleted in the (d, k) sequence.
2) If the received sequence is $-1, -1, -1, -1, 1$ as a result of the deletion of the second signal, the corresponding corrupted (d, k) sequence is 10001. Due to the deletion of a signal from a run of repeating signals with length one ($d = 0$), the (d, k) sequence has a deletion error and an adjacent substitution error. A subsequence in the original (d, k) sequence 11 turns into a subsequence 0 in the corrupted (d, k) sequence. Moreover, it can be considered to be the same scenario that if the received sequence is $1, -1, -1, -1, 1$ as a result of the deletion of the first signal, the corresponding corrupted (d, k) sequence is 01001. Due to the deletion of a signal from a run of repeating signals with length one ($d = 0$), a subsequence in the original (d, k) sequence 11 also turns into a subsequence 0 in the corrupted (d, k) sequence.
3) If the received sequence is $-1, 1, 1, -1, -1, -1, 1$ as a result of the insertion of 1 after the first signal, the corresponding corrupted (d, k) sequence is 1101001. Due to the insertion of a signal between -1 and 1, one zero is inserted in the (d, k) sequence.
4) If the received sequence is $-1, 1, -1, 1, -1, -1, 1$ as a result of the insertion of 1 after the third signal, the corresponding corrupted (d, k) sequence is 1111101. Due to the insertion of a different signal in a run of repeating signals, this inserted signal breaks a run and the corrupted (d, k) sequence has an insertion error and an adjacent substitution error. From the original (d, k) sequence, a subsequence 0 turns into a subsequence 11 in the corrupted (d, k) sequence.

Lemma 5: If $d \geq 1$, a single deletion error in the run-length limited sequence can cause a single deletion of zero in the corresponding (d, k) sequence; if $d = 0$, a single deletion error in the run-length limited sequence can cause a single deletion of zero or a single deletion error and an adjacent substitution error, which turn a subsequence from 11 into 0 in the corresponding (d, k) sequence; for any given d, a single insertion error in the run-length limited sequence can cause a single insertion of zero or an insertion and an adjacent substitution error, which turn a subsequence 0 into 11.

Proof: The proof is straightforward and is well illustrated in Example 2. ■

Based on Lemma 5, the following theorem shows the moment balancing template for (d, k) sequences can be applied to run-length limited sequences.

Theorem 2: A valid moment balancing template for (d, k) sequences with $d \geq 1$ can guarantee single deletion error correction for the corresponding run-length limited sequences; if

$$(k - d)(\alpha w' - \alpha^2) + (k - d + 1)^{\xi} \geq 2(k + 1)w', \tag{17}$$

this template can further guarantee a single deletion error correction for the corresponding run-length limited sequences with $d = 0$; if

$$(k - d)(\alpha w' - \alpha^2) + (k - d + 1)^{\xi} \geq 2(k + 1)w' + 2, \tag{18}$$

it can also guarantee a single insertion error correction for the corresponding run-length limited sequences.

Proof: In the case of Scenario 1 or 3 illustrated by Example 2, there is only a single insertion or deletion error in the corresponding corrupted (d, k) sequences. Therefore, a valid moment balancing template designed to correct a single insertion or deletion error for $(d,\ k)$ sequences also can guarantee the correction of the corresponding run-length limited sequences.

In the case of Scenario 2, the original (d, k) sequence $\mathbf{v}$ turns into $\mathbf{v}'$ with a subsequence 11 replaced by 0. If we choose $m \geq 2n$, the correct sequence can still be retrieved. Without losing generality, we can assume $\sigma(\mathbf{v}) \equiv 0 \pmod{2n}$. Then

$$\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \equiv i + i + 1 + n_R \pmod{2n}, \tag{19}$$

where i is the index of the first bit of the corrupted subsequence 11 and n_R denotes the number of ones after the corrupted subsequence. Choosing $m \geq 2n$ is due to the fact that the maximum value of $\sigma(\mathbf{v}') - \sigma(\mathbf{v})$ is $2n-1$. Let $W(\mathbf{v}')$ denote the weight of the sequence $\mathbf{v}'$. We have $2i + n_R + 1 > W(\mathbf{v}')$, which can be used to distinguish Scenario 2 from Scenario 1, since in Scenario 1 $\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \equiv n_R \pmod{2n}$ and $n_R \leq W(\mathbf{v}')$. From the value of $\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \pmod{2n}$, the solution of index i can be uniquely determined. It can be proved by contradiction: We assume there is another solution $i' \neq i$ that satisfies (19). Then $\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \equiv i' + i' + 1 + n'_R \pmod{2n}$ and $n_R \neq n'_R$. If $n_R < n'_R$, we have $i - i' \geq n'_R - n_R$. If $n_R > n'_R$, we have $i' - i \geq n_R - n'_R$. In either case, it is impossible to satisfy (19). Therefore the solution of i is unique. By reusing (16) with $m = 2n$, the condition (17) ensures all values from 0 to $m-1$ can be enumerated by α and ξ pairs moment balancing runs proposed in (6).

In the case of Scenario 4, the original (d, k) sequence $\mathbf{x}$ turns into $\mathbf{x}'$ with a subsequence 0 replaced by 11. If $m \geq 2n+2$, the error correction is guaranteed. Assume $\sigma(\mathbf{v}) \equiv 0 \pmod{2n+2}$. Then

$$\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \equiv i + i + 1 + n_R \pmod{2n+2}, \tag{20}$$

where i is the index of the corrupted subsequence 0 and n_R denotes the number of ones after the corrupted subsequence. Note that, in this case, the maximum value of $\sigma(\mathbf{v}') - \sigma(\mathbf{v})$ is $2n+1$. Therefore it is necessary to choose $m \geq 2n+2$. Here $2i + n_R + 1 > W(\mathbf{v}')$, which can be used to distinguish Scenario 4 from Scenario 3, since in Scenario 3 $\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \equiv n_R \pmod{2n+2}$ and $n_R \leq W(\mathbf{v}')$. Based on the value of $\sigma(\mathbf{v}') - \sigma(\mathbf{v}) \pmod{2n+2}$, the index i can be derived. A similar argument used in Scenario 2 can prove the uniqueness of the index i. ■

V. Encoding and Decoding Procedures of Implementing Moment Balancing Templates and Their Complexities

A. Encoding and Decoding Procedures

Let $\mathbf{x} = x_1 x_2 \ldots x_w$ be the corresponding reduced length sequence of the original (d, k) constrained sequence, and let $s = \alpha + \xi$ denote the number of pairs of balancing runs. Follow the steps described in Algorithm 1 to create a moment balanced (d, k) constrained sequence with the corresponding reduced length sequence $\mathbf{x}' = x'_1 x'_2 \ldots x'_{w+2s}$.

Algorithm 1: 1) Set $i = 1$, $j = 1$ and $t = 1$.

2) If $t > w + 2s$, go to Step 5).
3) Else,
   a) If $t \leq s$, set $x'_t = d$ and $i = i + 1$.
   b) Else if $t = 2s - i + 1 + (k - d + 1)^{i-s-1}$, set $x'_t = k$ and $i = i + 1$.
   c) Else if $t = w + i$, set $x'_t = k$ and $i = i + 1$.
   d) Else set $x'_t = x_j$ and $j = j + 1$.
4) $t = t + 1$ and go to Step 2).
5) Calculate $\Delta = 0 - \sum_{l=1}^{w+2s} x'_l (w + 2s - l + 1) - \frac{(w+2s+1)(w+2s)}{2} \pmod{\sum_{l=1}^{w+2s} (x'_l + 1) + 1}$.
6) Set $q = 1$.
7) If $q > \alpha$, go to Step 8).
   a) Calculate $p = \left\lfloor \frac{\Delta}{w+2s-2q+1} \right\rfloor$.
   b) If $k - d \geq p > 0$, set $x'_q = x'_q - p$, $x'_{w+2s-q+1} = x'_{w+2s-q+1} + p$, and $\Delta = \Delta - p(w + 2s - 2q + 1)$.
   c) Else if $p > k - d$, set $x'_q = x'_q - k + d$, $x'_{w+2s-q+1} = x'_{w+2s-q+1} + k - d$, and $\Delta = \Delta - (k-d)(w+2s-2q+1)$.
   d) $q = q + 1$ and go to Step 7).
8) Set $p = 1$.
9) If $\Delta \geq (k - d + 1)^{\xi - p}$,
   a) Set $q = \left\lfloor \frac{\Delta}{(k-d+1)^{\xi-p}} \right\rfloor$.
   b) Set $x'_{\alpha+p} = x'_{\alpha+p} - q$, $x'_{\alpha+p+(k-d+1)^{\xi-p}} = x'_{\alpha+p+(k-d+1)^{\xi-p}} + q$, and $\Delta = \Delta - q(k - d + 1)^{\xi-p}$.
10) $p = p + 1$.
11) If $p \leq \xi$, go to Step 9).
12) Else, exit.

From Step 1) to Step 4), Algorithm 1 initiates a moment balancing template according to (6). The initial values of balancing runs have been set according to $c_{i,1} = d$ and $c_{i,2} = k$ for $i = 1, 2, \ldots, s$. In Step 5), the algorithm calculates the deficiency of the moment value of the template with the initial setup. From Step 6) to Step 7), the algorithm adjusts the moment value of the template roughly by relocating 0's in the α pairs of moment balancing runs. From Step 8) to Step 12), the algorithm makes a fine adjustment of the moment value of the template by relocating 0's in the ξ pairs of moment balancing runs.

To make this paper as much self-contained as possible, we also present the decoding algorithms for a single deletion/insertion error. The following decoding algorithms were first presented by Levenshtein [9].

The decoding algorithm of a deletion error for the (d, k) constrained moment balancing template is shown in Algorithm 2. Let $\mathbf{r} = r_1 r_2 \ldots r_{n-1}$ denote the received sequence with one deletion error, and let $N_1(j)$ denote the number of 1's in $\mathbf{r}$ from index j to $n - 1$ and $N_1(n) = 0$, where $j \in \{1, 2, \ldots, n\}$. Note that the length $n - 1$ of the received (d, k) constrained sequences is bounded according to (1).

Algorithm 2: 1) Calculate $\Delta = 0 - \sum_{i=1}^{n-1} i r_i \pmod{n+1}$.

2) If $\Delta \leq N_1(1)$ (a zero is deleted),
   a) Insert a zero after the $(N_1(1) - \Delta)$'th one.
   b) Exit.
3) Else, (a one is deleted)
   a) $j = 1$.
   b) If $N_1(j) + j = \Delta$, insert a one before the j'th bit and exit.
   c) Else $j = j + 1$, go to Step 3.b).

The decoding algorithm of an insertion error for the (d, k) constrained moment balancing template is shown in Algorithm 3. Let $\mathbf{r} = r_1 r_2 \ldots r_{n+1}$ denote the received sequence with one insertion error, and let $N_1(j)$ denote the number of 1's in $\mathbf{r}$ from index j to $n+1$, where $j \in \{1, 2, \ldots, n+1\}$.

Algorithm 3: 1) Calculate $\Delta = 0 - \sum_{i=1}^{n+1} i r_i \pmod{n+1}$.

2) If $\Delta = 0$ (an insertion at the last position)
   a) Delete the last bit.
   b) Exit.
3) Else if $\Delta \geq N_1(1)$ (a zero is inserted),
   a) Delete the zero after the $(n+1-\Delta)$'th one.
   b) Exit.
4) Else, (a one is inserted)
   a) $j = 1$.
   b) If $r_j = 1$ and $n+1-N_1(j+1)-j = \Delta$, delete the one at the j'th position and exit.
   c) Else $j = j+1$, go to Step 4.b).

B. Implementation and Complexities

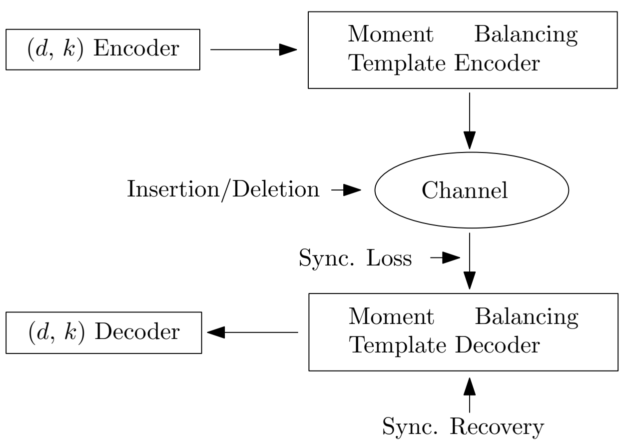

Fig. 2. Implementation of moment balancing template for (d, k) encoding system.

The implementation of the moment balancing template for a (d, k) encoding system is illustrated in Fig. 2. The computational complexity and memory requirements of the template implementation can be estimated as follows.

Both the encoder and the decoder have memory requirements for the encoding and decoding procedures. The memory requirement depends on the length of the template. The template can introduce less redundancy by increasing the template length. As a result, the memory requirement needs to be increased.

The moment balancing template encoding procedure is systematic. In other words, parity bits are located at predetermined indices, which can be separated easily from the original (d, k) sequences. Thus, this property is a benefit to the decoding procedure as well.

We assume multiplication operations have the same computational complexity as division operations and addition operations have the same complexity as subtraction operations. In this paper the computational complexity is estimated by counting how many multiplication and addition operations processed by the procedure. To simplify the estimation, all division operations are counted as multiplication operations, and all subtraction operations are counted as addition operations.

At the encoder, the first-order moment value needs to be calculated once. This calculation needs one multiplication operation and maximum n addition operations, where n is the length of the (d, k) sequence. Note that calculating the first-order moment of a binary sequence requires no multiplication operation but $n-1$ addition operations, and the modulo operation requires one multiplication operation and one addition operation. To balance a template, further maximum $\lceil \log_{k-d+1} n \rceil$ multiplication operations and $\lceil \log_{k-d+1} n \rceil$ addition operations are required. Therefore, the total computational complexity of encoding is maximum $\lceil \log_{k-d+1} n \rceil + 1$ multiplication operations and $\lceil \log_{k-d+1} n \rceil + n$ addition operations.

For the decoding of a template, the first-order moment value also needs to be calculated once. If the inserted or deleted bit is 0, further two addition operations are required. However, when the received sequence has the format of $1010\ldots 10\underline{1}01$ with an inserted 1 at the underlined position, maximum $2\left\lceil \frac{n-1}{2} \right\rceil + 1$ addition operations are required to detect the error. Therefore, the total computational complexity of decoding is maximum one multiplication operation and $2\left\lceil \frac{n-1}{2} \right\rceil + n + 1$ addition operations.

VI. Analysis and Discussion

We assume that the moment balancing encoder can judiciously choose a (d, k) sequence with weight $w = (k-d+1)^{\xi} - 2\xi - 2$, since the weight w can achieve the maximum value to partially satisfy the condition (15) and therefore achieve the minimum redundancy. If $(k-d)(k-d+1)^{\xi-1} - 2 > 0$, the condition (15) is completely satisfied. It is evident that if $k > d$ and $\xi > 1$, $(k-d)(k-d+1)^{\xi-1} - 2 > 0$ always stands.

As $w \to \infty$, for fixed integers $k > d$ we have

$$\begin{aligned} \log_{k-d+1} w - \xi &= \log_{k-d+1} \left((k-d+1)^{\xi} - 2\xi - 2 \right) - \xi \\ &= \log_{k-d+1} \left(1 - \frac{2\xi + 2}{(k-d+1)^{\xi}} \right) \\ &= o(1). \end{aligned} \tag{21}$$

When $w \to \infty$, $\xi \to \infty$. Since $\lim_{\xi \to \infty} \frac{2\xi+2}{(k-d+1)^{\xi}} = 0$, we can obtain the asymptotic result in (21).

Moreover, we can derive an asymptotic lower bound for α as follows.

Lemma 6: Choose $w = (k-d+1)^{\xi} - 2\xi - 2$. As $w \to \infty$, for fixed integers $k > d$,

$$\alpha \geq \frac{k}{k-d} + o(1).$$

Proof:

Substitute w' by $w + 2(\alpha + \xi)$. Then, (16) can be further simplified as

$$(k-d)\alpha^2 + ((k-d)(w+2\xi) - 2k - 2)\,\alpha - kw - 2k\xi + 1 \geq 0. \tag{22}$$

Note that the quadratic polynomial on the left hand side of (22) is a convex function of α, and has two roots: one is negative and the other is positive. We are only interested in the positive root, which decides the least positive number of α that satisfies (16).

Choose $w = (k-d+1)^{\xi} - 2\xi - 2$. Let

$$\rho = (k-d)(w+2\xi) - 2k - 2 \tag{23}$$

and

$$\tau = kw + 2k\xi - 1. \tag{24}$$

From (22), we obtain

$$\begin{aligned}\alpha &\geq \frac{\sqrt{\rho^2 + 4(k-d)\tau} - \rho}{2(k-d)} \\ &= \frac{(\sqrt{\rho^2 + 4(k-d)\tau} - \rho)(\sqrt{\rho^2 + 4(k-d)\tau} + \rho)}{2(k-d)(\sqrt{\rho^2 + 4(k-d)\tau} + \rho)} \\ &= \frac{4(k-d)\tau}{2(k-d)(\sqrt{\rho^2 + 4(k-d)\tau} + \rho)} \\ &= \frac{2\tau}{\sqrt{\rho^2 + 4(k-d)\tau} + \rho}. \end{aligned} \tag{25}$$

According to (21), as $w \to \infty$, for fixed integers $k > d$ $\xi = o(w)$. Then, from (23) and (24), we have

$$\rho = (k-d)w + o(w) \tag{26}$$

and

$$\tau = kw + o(w). \tag{27}$$

Therefore, by substituting (26) and (27) into (25), we obtain

$$\begin{aligned}\alpha &\geq \frac{2kw + o(w)}{\sqrt{(k-d)^2 w^2 + o(w^2)} + (k-d)w + o(w)} \\ &= \frac{2kw + o(w)}{(k-d)w + o(w) + (k-d)w + o(w)} \\ &= \frac{k}{k-d} + o(1). \end{aligned} \tag{28}$$

■

For a sufficiently large w, we have two interesting findings for the moment balancing template of long sequences: first, according to (21), ξ can be completely determined by the weight of the original sequence; second, according to Lemma 6 the minimum value of α depends on the values of d and k, but not on the length or the weight of the original sequence. As w goes to infinity, the minimum value of α tends to $\lceil \frac{k}{k-d} \rceil$.

Now we can derive an asymptotic bound of the redundancy introduced by the moment balancing template (6) as follows.

Theorem 3: Let n denote the length of the moment balanced (d, k) sequence with the corresponding reduced length sequence as shown in (6), and let K denote the length of the original (d, k) sequence. Choose $w = (k-d+1)^{\xi} - 2\xi - 2$. As $w \to \infty$, for fixed integers $k > d$,

$$1 - K/n = \Theta(\frac{\log w}{w}).$$

Proof:

Since

$$1 - K/n = \frac{(d+k+2)(\alpha+\xi)}{n}, \tag{29}$$

we have

$$1 - K/n \leq \frac{(d+k+2)(\alpha+\xi)}{(d+1)w + (d+k+2)(\alpha+\xi)} \tag{30}$$

TABLE I

Redundancy rates of the moment balancing template of a $(1, 7)$ constrained code with $m = n + 1$

Redundancy Rate	w	K	n
42.86–75.00%	5	10–40	40–70
27.27–60.00%	10	20–80	50–110
9.09–28.57%	50	100–400	140–440
4.76–16.67%	100	200–800	240–840
1.23–4.76%	500	1000–4000	1050–4050
0.62–2.44%	1000	2000–8000	2050–8050

and

$$1 - K/n \geq \frac{(d+k+2)(\alpha+\xi)}{(k+1)w + (d+k+2)(\alpha+\xi)}. \tag{31}$$

From (30), we have

$$\begin{aligned}1 - K/n &\leq \frac{(d+k+2)\left(\log_{k-d+1} w + o(\log w)\right)}{(d+1)w + o(w)} \\ &= \frac{d+k+2}{(d+1)\log(k-d+1)} \frac{\log w}{w} + o\left(\frac{\log w}{w}\right). \end{aligned} \tag{32}$$

From (31), we have

$$\begin{aligned}1 - K/n &\geq \frac{(d+k+2)\left(\log_{k-d+1} w + o(\log w)\right)}{(k+1)w + o(w)} \\ &= \frac{d+k+2}{(k+1)\log(k-d+1)} \frac{\log w}{w} + o\left(\frac{\log w}{w}\right). \end{aligned} \tag{33}$$

As $w \to \infty$,

$$g_1 \frac{\log w}{w} \leq 1 - K/n \leq g_2 \frac{\log w}{w}, \tag{34}$$

for some positive g_1 and g_2.

■

For a (d, k) sequence of length n and weight w, we have

$$\frac{\log(d+1)w}{(k+1)w} \leq \frac{\log n}{n} \leq \frac{\log(k+1)w}{(d+1)w}. \tag{35}$$

Hence, we can obtain

$$\frac{\log n}{n} = \Theta\left(\frac{\log w}{w}\right). \tag{36}$$

Recall that the optimal redundancy of a universal moment balancing template of arbitrary random sequences is $1 - K/n \approx \log n/n$ [2]. Therefore, we can conclude that the optimal redundancy introduced by the template for (d, k) sequences is of the same order as that of the universal template.

Table I gives a numerical example of the redundancy rates introduced by the generalized moment balancing template for (d, k) sequences. It shows that the redundancy rate drops dramatically when information length increases. Given a (d, k) sequence with the weight w and the length K, we can decide the number of sufficient balancing runs according to (16). For long sequences, we can decide ξ based on (21), and choose α according to Lemma 6. In Table I, the value ranges of K and n are provided, and the relationship between K and n is $n - K = (\alpha + \xi)(d + k + 2)$.

VII. Conclusion

In this paper, we have extended the investigation of the moment balancing template [2] by applying it to (d, k) constrained codes and run-length limited sequences in order to implement the systematic encoding of certain number theoretic codes. We have also shown how to use the non-binary base to enumerate the contributed moment values. Since the redundancy added by our moment balancing templates may be kept small, it makes the application to longer codes useful and attractive for certain practical applications, e.g., a magnetic recording system having a low insertion/deletion error probability [1].

Acknowledgments

The authors would like to thank the associate editor and the anonymous reviewers for their contributions toward enhancing the quality of this paper.

References

[1] J. Hu, T. M. Duman, E. M. Kurtas, and M. F. Erden, "Bit-patterned media with written-in errors: modeling, detection, and theoretical limits." *IEEE Trans. Magn.*, vol. 43, no. 8, pp. 3517–3524, Aug. 2007.

[2] H. C. Ferreira, K. A. S. Abdel-Ghaffar, L. Cheng, T. G. Swart, and K. Ouahada, "Moment balancing templates: Constructions to add insertion/deletion correction capability to error correcting or constrained codes." *IEEE Trans. Inform. Theory*, vol. 55, no. 8, pp. 3494–3500, Aug. 2009.

[3] L. Cheng, H. C. Ferreira, and K. Ouahada, "*k*-bit grouping moment balancing templates for spectral shaping codes." in *Proc. IEEE Inform. Theory Workshop*, Porto, Portugal, 2008, pp. 426–430.

[4] ——, "Moment balancing templates for spectral null codes." *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 3749–3753, Aug. 2010.

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

[6] H. M. Hilden, D. G. Howe, and E. J. Weldon, Jr., "Shift error correcting modulation codes." *IEEE Trans. Magn.*, vol. 27, no. 6, pp. 4600–4605, Nov. 1991.

[7] T. Kløve, "Codes correcting a single insertion/deletion of a zero or a single peak-shift." *IEEE Trans. Inform. Theory*, vol. 41, no. 1, pp. 279–283, Jan. 1995.

[8] R. P. Varshamov and G. M. Tenengolts, "Correction code for single asymmetric errors." *Automat. Telemekh.*, vol. 26, pp. 288–292, 1965.

[9] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and substitutions of symbols." *Dokl. Akad. Nauk SSSR*, vol. 163, no. 4, pp. 845–848, 1965.

Ling Cheng received the B.Eng. degree in electrical and electronic engineering from the Huazhong University of Science and Technology, Wuhan, Hubei, China, in 1995. From 1995 to 2004, he was with China Mobile Communications Corporation and Telematic Technologies. He received his M. Ing degree and D. Ing. degree in electrical and electronic engineering from the University of Johannesburg, South Africa, in 2005 and in 2010. He is currently a lecturer at the University of the Witwatersrand, Johannesburg, South Africa. His research interests include power-line communications, error correction coding and spectral shaping techniques.

Hendrik C. Ferreira was born and educated in South Africa where he received the D.Sc. (Eng.) degree from the University of Pretoria in 1980.

From 1980 to 1981, he was a post doctoral researcher at the Linkabit Corporation in San Diego, CA, USA. In 1983, he joined the Rand Afrikaans University, Johannesburg, South Africa where he was promoted to professor in 1989 and served two terms as Chairman of the Department of Electrical and Electronic Engineering, from 1994 to 1999. He is currently a research professor at the University of Johannesburg. His research interests are in Digital Communications and Information Theory, especially Coding Techniques, as well as in Power Line Communications.

Dr. Ferreira is a past chairman of the Communications and Signal Processing Chapter and founding chairman of the Information Theory Society Chapter of the IEEE South Africa section. From 1997 to 2006 he was Editor-in-Chief of the Transactions of the South African Institute of Electrical Engineers. He has served as chairman of several international conferences, including the 1999 IEEE Information Theory Workshop in the Kruger National Park, South Africa, the 2010 IEEE African Winter School on Information Theory and Communications, and the 2013 IEEE International Symposium on Power Line Communications.

Izak Broere received the B.Eng. degree in electric and electronic engineering from the University of Khartoum, Sudan, in 1995, the M.Eng. degree in electric and electronic engineering from the Rand Afrikaans University, South Africa, in 2002, and the D.Eng. degree at the University of Johannesburg, South Africa, 2009. His research interests include digital communications, error correction coding and spectral shaping techniques.