RZ 3881 (#ZUR1401-018) 01/17/2014
Electrical Engineering 5 pages

Research Report

Enumerative Modulation Codes Based on Sliding-Window Substitutions

Thomas Mittelholzer and Roy D. Cideciyan

IBM Research – Zurich
8803 Rüschlikon
Switzerland

In: *Proc. 2014 IEEE Int'l Symp. on Information Theory (ISIT),* pp. 1613-1617, June 29-July 4, 2014
http://dx.doi.org/10.1109/ISIT.2014.6875106

Enumerative Modulation Codes Based on Sliding-Window Substitutions

Thomas Mittelholzer and Roy D. Cideciyan
IBM Research – Zurich
8803 Rüschlikon, Switzerland
Email: {tmi,cid}@zurich.ibm.com

***Abstract*—We consider high-rate modulation codes satisfying tight global and interleave (G, I) constraints for magnetic storage systems. We use a novel sliding-window substitution coding technique to improve on known long capacity-efficient $(G = 2\gamma, I = \gamma)$ codes. This coding technique maps one-sided $(G = 2\gamma, I = \gamma)$-constrained sequences into one-sided sequences satisfying a substantially tighter G constraint and a slightly relaxed I constraint. Sliding-window substitution encoding in conjunction with enumerative encoding provides high-rate capacity-efficient codes that are relevant for practical magnetic storage systems.**

I. Introduction

Magnetic storage systems, such as hard disk drives and tape drives, employ binary modulation codes to facilitate timing recovery and data detection during readback [1]. Modulation codes, which impose constraints on the sequences at the channel input to rule out undesired sequences, can be studied in the framework of discrete noiseless channels. In particular, a global runlength constraint, the G-constraint, facilitates timing recovery, whereas a run-length constraint on even and odd interleaves, the I-constraint, allows efficient operation of the detector and eliminates quasi-catastrophic sequences. The corresponding class of codes is known as (G, I) or PRML(G, I) codes.

Based on enumeration techniques, one can construct long capacity-efficient PRML(G, I) codes with $G = 2I$. These codes are obtained by means of an even/odd interleaving construction using generalized Fibonacci codes [2], [3], [4] or enumerative maximum transition run codes [5] in even and odd interleaves. Because of the interleaving construction, the global G constraint is always twice the interleaved constraint I, and no efficient high-rate enumerative code with $G < 2I$ is currently known. However, from an application point of view, it is often desirable to have a smaller G constraint that is of the same order as the I constraint. To construct such a code, the idea is to start with an efficient enumerative PRML(G',I') code and to modify it by a sliding-window substitution map that transforms the (G',I') constraint into a (G,I) constraint with $G < G'$ and $I > I'$.

In Section II, we study the embedding problem of the $(G = 2\gamma, I = \gamma)$ constraint into the $(G = \gamma + 1, I = \infty)$ constraint. In Section III, we define sliding-window substitution maps. In Section IV, we construct block codes, which require additional substitutions at the codeword boundaries.

II. Embedding the $(G = 2\gamma, I = \gamma)$ Constraint

Let $\mathbb{Z}$ be the set of integers and consider the set W_k of binary bi-infinite sequences $w = \{w_i\}_{i\in\mathbb{Z}}$ that satisfy the k-constraint, i.e., w has no more than k consecutive 0s [6]. For instance, the $k = 2$ constraint is characterized by the labeled directed graph D_2 in Fig. 1. Its adjacency matrix is

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

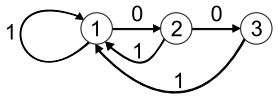

Fig. 1. Labeled directed graph D_2 for the $k = 2$ constraint.

Recall that a binary sequence satisfies a (G, I) constraint, if the maximum number of consecutive zeros is limited to G and the maximum number of consecutive zeros in both the even and odd interleaves of the sequence is limited to I. By interleaving two copies of W_k, one can characterize bi-infinite sequences $X = W_k \times W_k$ satisfying the ($G = 2k$, $I = k$) constraint. In particular, there are no more than $2k$ consecutive zeroes in any interleaved sequence $x = \{x_i\}_{i\in\mathbb{Z}} \in X$ with $\{x_{2i}\}_{i\in\mathbb{Z}} \in W_k$ and $\{x_{2i+1}\}_{i\in\mathbb{Z}} \in W_k$.

The simplest case is an embedding of the shift X for the ($G = 4$, $I = 2$) constraint into the shift Y for the ($G = 3$, $I = \infty$) constraint.

Proposition 1: (i) There exists an embedding of the second power X^2 of the ($G = 4$, $I = 2$) constraint into the second power Y^2 of the ($G = 3$, $I = \infty$) constraint.

(ii) There is no embedding of the ($G = 6$, $I = 3$) constraint into the ($G = 4$, $I = \infty$) constraint.

Proof: The proof of (i) is based on verifying the two conditions of the Embedding Theorem (Theorem 10.1.1 in [6]). To this end, we characterize the two constraints as edge shifts of two labeled graphs.

The ($G = 4$, $I = 2$) constraint X is a two-way interleave of the $k = 2$ constraint W_2. The graph product $D_2 \times D_2$, which is labeled by pairs, generates the second-power shift X^2, which corresponds to the ($G = 4$, $I = 2$) constraint when the basic shift map moves two positions. The adjacency matrix of the graph product is given by the Kronecker product $U = A \otimes A$.

The $k = 3$ constraint W_3 is characterized by a 4-state graph similar to that in Fig. 1 with adjacency matrix

$$A' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The $(G = 3,\, I = \infty)$ constraint, when moving two bits at a time, can be characterized by the second power Y^2 with adjacency matrix $Q = (A')^2$.

By verifying the necessary and sufficient conditions of the Embedding Theorem [6], one can show that there is an embedding of X^2 into Y^2. In particular, the entropy condition $h(X) < h(Y)$ is satisfied because $h(X) = h(W_2) \approx 0.8791$ and $h(Y) = h(W_3) \approx 0.9468$.

To check the condition on the embedding of the periodic points $P(X^2) \hookrightarrow P(Y^2)$, one can consider the technique described in Chap. 10.1 of [6], which is based on the n-th net traces of the adjacency matrices U and Q. By construction, the spectra (i.e., the set of eigenvalues with multiplicities) of U and Q can be expressed in terms of the spectra of A and A', which will be denoted by $sp(A) = \{\lambda_1 \approx 1.8393, \lambda_2 \approx -0.4196 + 0.6063i, \lambda_3 \approx -0.4196 + 0.6063i\}$ and $sp(A') = \{\lambda'_1 \approx 1.9276, \lambda'_2 \approx -0.0764 + 0.8147i, \lambda'_3 \approx -0.0764 - 0.8147i, \lambda'_4 \approx -0.7748\}$, respectively. The spectra of U and Q are

$$sp(U) = \{\lambda_i \lambda_j : \lambda_i, \lambda_j \in sp(A),\, i = 1 \le i, j \le 3\} \quad (1)$$

$$sp(Q) = \{(\lambda'_i)^2 : \lambda'_i \in sp(A'),\, i = 1, \ldots, 4\}. \quad (2)$$

From the spectra, one can compute the traces of U^n and Q^n, derive the n-th net traces, and verify that the sufficient condition in Chap. 10.1 of [6] for the embedding of periodic points holds.

To prove (ii), we show that the embedding condition on periodic points does not hold for the second-power shifts. The second power of the $(G = 6,\, I = 3)$ constraint has 216 4-ary sequences of least period 4 whereas the second power of the $(G = 4,\, I = \infty)$ constraint has only 208 4-ary sequences of least period 4. □

Remark: The $(G = 4, I = 2)$ constraint cannot be embedded into a $(G = 3, I < \infty)$ constraint. This follows by noting that the constraints $(G = 4, I = 2)$ and $(G = 3, I = \infty)$ have the same number of periodic sequences of least period 6. Therefore, all periodic $(G = 3, I = \infty)$ sequences of least period 6 will be images of any embedding. In particular, the periodic sequence $\ldots 1\,0\,1\,0\,0\,0 \ldots$ of period 6 is such a sequence and this sequence has $I = \infty$. Note that one-sided embedding into $(G = 3, I < \infty)$ is possible, and will be discussed in Section III.

The embedding question can be generalized to $(G = 2\gamma, I = \gamma)$ and $(G = \gamma + 1,\, I = \infty)$, $\gamma \ge 2$. Similarly as in part (ii) of the proof, it can be shown that for $I = \gamma = 4$, there is no embedding of $(G = 8, I = 4)$ into $(G = 5, I = \infty)$. It is conjectured that there is no embedding for all $I = \gamma > 2$. For practical values of γ, the entropy condition of the embedding theorem is satisfied, and the condition on periodic points prevents the existence of an embedding. Therefore, it is worthwhile to consider the corresponding embedding problem for one-sided instead of bi-infinite sequences. Furthermore, the embedding in the setting of one-sided sequences is relevant for the construction of modulation codes in practical applications [7], which will be discussed in Section IV.

III. Sliding-Window Substitution

In this section, we define sliding-window substitution (SWS) encoders for one-sided sequences. They provide an efficient means to tighten the global constraint of the one-sided $(G = 2\gamma, I = \gamma)$ constraint from $G = 2\gamma$ to some G with $G < 2\gamma$ while only moderately relaxing the I constraint.

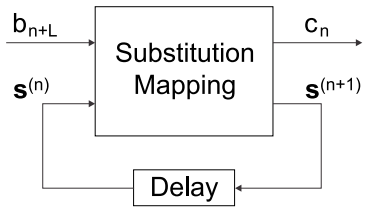

Fig. 2. Sliding-window substitution encoder.

For a bi-infinite shift space B, one defines the one-sided shift by $B^+ = \{b_{[0,\infty)} : b \in B\}$ (see e.g. [6]). A *sliding-window substitution encoder* is a finite-state machine with L-bit states $\mathbf{s}^{(n)} = s_1^{(n)}, s_2^{(n)}, \ldots, s_L^{(n)}$, binary inputs b_{n+L} and binary outputs c_n as illustrated in Fig. 2. These quantities are related by

$$\mathbf{x} = [x_0\, x_1\, \ldots\, x_L] = [\mathbf{s}^{(n)}\, b_{n+L}] \quad (3)$$

$$\mathbf{y} = [y_0\, y_1\, \ldots\, y_L] = [c_n\, \mathbf{s}^{(n+1)}]. \quad (4)$$

The registers $\mathbf{x}$ and $\mathbf{y}$ are related by the substitution map $\phi : \{0,1\}^{L+1} \to \{0,1\}^{L+1}$, $\mathbf{y} = \phi(\mathbf{x})$, which eliminates undesired patterns in the input $\mathbf{x}$. Typically, for most inputs, the mapping ϕ is the identity, and only for a relative small number of inputs, which violate a certain pattern, is the output $\mathbf{y}$ selected such that this pattern is not violated. When applying the SWS encoder to a one-sided shift B^+, one starts at "time 0" and moves from left to right. Given a sequence $b_{[0,\infty)}$ as input, the initial state is set to

$$\mathbf{s}^{(0)} = [b_0\, b_1\, \ldots\, b_{L-1}]$$

and the input at time 0 is b_L. The SWS encoder produces a sequence $\{c_n\}_{n \in \{0,\ldots,\infty\}}$.

Example 1: Consider the one-sided $(G = 4, I = 2)$ constraint B^+. Let $L = 6$ and define the substitution mapping ϕ by

$$\phi(\mathbf{x}) = \begin{cases} [0\, x_6\, 0\, x_4\, 0\, x_5\, 1] & \text{if } [x_0\, x_1\, x_2\, x_3] = [0\,0\,0\,0] \\ \mathbf{x} & \text{otherwise.} \end{cases} \quad (5)$$

Let $b \in B^+$ and apply the SWS to b. The resulting sequence satisfies the $G = 3$ constraint. This can be checked inductively. At time $n = 0$, $[x_0\, x_1\, \ldots\, x_6] = [b_0\, b_1\, \ldots\, b_6]$.

TABLE I
SUBSTITUTION MAP ϕ FOR $G = 12, I = 6$

Case	Input $\mathbf{x} = x_0\, x_1 \ldots x_{15}$	Output $\mathbf{y} = y_0\, y_1 \ldots y_{15}$
1	$0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,x_{11}\,1\,x_{13}\,x_{14}\,x_{15}$	$1\,0\,x_{15}\,0\,x_{11}\,0\,x_{13}\,0\,0\,0\,0\,0\,1\,0\,x_{14}\,1$
2	$0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,x_{11}\,x_{12}\,x_{13}\,x_{14}\,x_{15}$	$1\,0\,x_{15}\,0\,x_{11}\,0\,x_{13}\,0\,0\,0\,1\,0\,x_{12}\,0\,x_{14}\,1$
3	$0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,x_{10}\,x_{11}\,x_{12}\,x_{13}\,x_{14}\,x_{15}$	$1\,0\,x_{11}\,0\,x_{13}\,0\,x_{15}\,0\,1\,0\,x_{10}\,0\,x_{12}\,0\,x_{14}\,1$

If the substitution is applied, i.e., $[x_0\, x_1\, x_2\, x_3] = \mathbf{0}$, then $x_4 = 1 = x_5$ because the sequence b satisfies the $I = 2$ constraint to the right of position $n = 0$ and, therefore, $\mathbf{y} = [0\, x_6\, 0\, x_4\, 0\, x_5\, 1]$ satisfies the $G = 3$ constraint. Thus, the first seven output bits $[c_0\, c_1\, \ldots\, c_6] = \mathbf{y}$ satisfy the $G = 3$ constraint. Morever, after substitution, the $I = 2$ constraint holds in the one-sided sequence $[y_5\, y_6\, b_7\, b_8\, b_9\, \ldots]$, therefore, one can move seven time steps ahead to $n = 7$. After moving 7 steps forward, the **x**-register contains the sequence $b_7\, b_8\, \ldots\, b_{13}$ and the SWS encoder is in a similar condition as at time $n = 0$.

If no substitution occurs, the identity is applied, and then $[y_0\, y_1\, y_2\, y_3] = [x_0\, x_1\, x_2\, x_3]$ satisfies the $G = 3$ constraint by definition. If $y_0 = 1$, then $c_0 = 1$ and one can move one step further to $n = 1$ and ensure that the $G = 3$ constraint holds in the beginning part of the output. If $y_0 = 0$, then one of the other three components must be 1 and, again, one can move one step further to $n = 1$ and ensure that in the beginning part of the output the $G = 3$ constraint holds. After moving 1 step forward, the **x**-register contains the sequence $b_1\, b_2\, \ldots\, b_7$, and the SWS encoder is in a similar condition as at time $n = 0$.

One can verify that the SWS encoder is a one-to-one mapping. If there are no substitutions within some interval, the mapping is one-to-one by definition and, in particular, also the $I = 2$ constraint is maintained. If there is a substitution on a stretch $b_n\, b_{n+1}\, \ldots\, b_{n+6}$, the corresponding output sequence is $[c_n, \mathbf{s}^{(n+1)}] = \mathbf{y}$. This output sequence has a characteristic run of exactly three consecutive zeroes in the interleave $c_n\, c_{n+2}\, c_{n+4}\, c_{n+6}$. Thus, the locations of the substitutions are marked as violation of the $I = 2$ constraint. As the substitution map (5) is one-to-one, there cannot be two input sequences that map to the same output sequence. Moreover, the output satisfies the $I = 3$ constraint. Thus, the SWS gives an embedding of the one-sided $(G = 4, I = 2)$ constraint into the $(G = 3, I = 3)$ constraint.

Theorem 1: Let $B^+_{(G,I)}$ denote the one-sided (G, I) constraint.

(i) There are embeddings $B^+_{(G=2\gamma,I=\gamma)} \hookrightarrow B^+_{(G=\gamma+1,I=\gamma+1)}$ for $\gamma = 2, 3, 4$.

(ii) There is an embedding $B^+_{(G=12,I=6)} \hookrightarrow B^+_{(G=8,I=7)}$.

Proof: We will provide SWS encoders for each case. The case $\gamma = 2$ has already been worked out in Example 1.

Case $(G = 6, I = 3)$: Consider the SWS encoder with a state space of dimension $L = 8$ and window length $L+1 = 9$. The substitution map is given by

$$\phi(\mathbf{x}) = \begin{cases} [0\, x_8\, 0\, x_6\, 0\, x_5\, 0\, x_7\, 1] & \text{if } [x_0\, x_1 \ldots x_4] = [0\,0\,0\,0\,0] \\ \mathbf{x} & \text{otherwise.} \end{cases} \tag{6}$$

The one-to-one property of the mapping and the constraints of the output sequences can be verified similarly as in Example 1.

Case $(G = 8, I = 4)$: We consider an SWS encoder with a state space of dimension $L = 10$, i.e., with a window length of 11. The mapping $\phi : \mathbf{x} \mapsto \mathbf{y}$ is the identity if the first 6 components of $\mathbf{x} = [x_0\, x_2\, \ldots\, x_{10}]$ are not all zero. Otherwise the mapping is given by Table II. The verification of the constraints is similar to Example 1.

TABLE II
SUBSTITUTION MAP ϕ FOR $G = 8, I = 4$

Input $\mathbf{x} = x_0\, x_1 \ldots x_{10}$	Output $\mathbf{y} = y_0\, y_1 \ldots y_{10}$
$0\,0\,0\,0\,0\,0\,x_6\, x_7\, x_8\, x_9\, x_{10}$	$0\,x_{10}\,0\,x_6\,0\,x_8\,0\,x_7\,0\,x_9\,1$

Case $(G = 12, I = 6)$: The SWS encoder has a state space of dimension $L = 15$ and a window length of 16. The mapping ϕ is the identity if the first 9 components of **x** are not all zero. Otherwise, there are three non-identity substitutions specified in Table I. These substitutions are applied to enforce $G = 8$ whenever $G > 8$. In the following, it is assumed that the component b_{n-1} in the input sequence, which is immediately to the left of the length-16 sliding window, equals 1. We will distinguish three cases depending on the number of consecutive leading zeros in **x**. In Case 1 below, component b_{n-2} must be 1 because of the $I = 6$ constraint.

- Case 1 (11 or 12 leading zeros): Note that $I = 6$ implies $x_{12} = 1$ and either x_{11} or x_{13} must be 1. Thus, in the substitution string **y** there can be at most 7 consecutive zeros. As the component b_{n-1} to the left of **x** equals 1, the substitution string **y** satisfies an $I = 7$ constraint.
- Case 2 (10 leading zeros): Because of $I = 6$, either x_{11} or x_{13} must be 1. Thus, in the substitution string **y**, there can be at most 5 consecutive zeros. As $b_{n-1} = 1$, the substitution string **y** satisfies an $I = 7$ constraint.
- Case 3 (9 leading zeros): Because of $I = 6$, either x_{10} or x_{12} must be 1. In the substitution string **y**, there can be at most 7 consecutive zeros.

In all three cases, the substituted string **y** has an interleave constraint of $I = 7$. Moreover, the components y_8 and y_{10} have different pairs of values, which are indicative for each case. This makes the substituted strings unique and allows one to do the proper inverse substitution. In addition, the I and G constraints towards the left and towards the right of the substituted string remain unchanged. At the left side, this is obvious because the first two bits are $(x_0, x_1) = (0, 0)$. At the right side, the G constraint is not increased because the last bit of the substituted string is 1. Furthermore, doing case-by-case checking, one can show that the I constraint at the right side is also maintained.

□

IV. Block Codes Defined by Enumeration and Substitutions

The SWS encoders studied in Section III can embed one-sided shifts with large G constraints into one-sided shifts with substantially reduced G constraints at the expense of a minor increase of the I constraint. In this section, we define additional substitution maps that operate at the codeword boundaries. This SWS encoding technique will be illustrated on a specific high-rate PRML($G = 12, I = 6$) code of length 234.

By using an enumerative MTR(N, j, k) block code of length N in both the even and the odd interleave [5], one obtains a PRML(G, I) block code with $G = 2k$ and $I = k$. The j constraint in each interleave translates into the $M = 2j$ constraint, which limits the runs of alternating $2T$ magnets $\ldots 00110011 \ldots$ in channel input sequences (i.e., after $1/(1 \oplus D^2)$ precoding) to $\lfloor M/2 \rfloor + 1$, where T denotes the symbol duration. The M constraint is also known as VFO (variable-frequency oscillator) constraint. It is a desirable constraint in tape-recording systems, which use a phase-locked loop to acquire timing based on a long alternating $2T$ VFO pattern. The VFO constraint ensures that a modulation-encoded data sequence does not contain a long VFO pattern.

This interleaving construction is illustrated in Fig. 3. For instance, based on an enumerative MTR code of length $N = 117$ and dimension $K = 116$ with constraints $k = 6$, $j = 11$, one obtains a PRML code with constraints $G = 2k = 12$, $I = k = 6$ and $M = 2j = 22$. At the left and right codeword boundaries, the code has the tighter constraints $I_{left} = 3 = I_{right}$ and $M_{left} = 10$, $M_{right} = 12$. Thus, the codewords can be freely concatenated, and the constraints $G = 12, I = 6, M = 22$ are maintained.

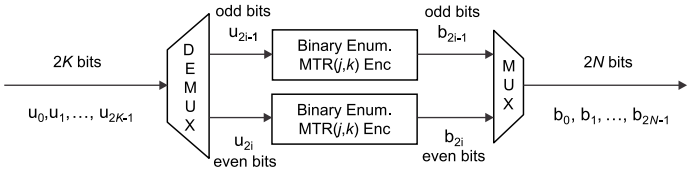

Fig. 3. Encoder for PRML($G = 2k$, $I = k$, $M = 2j$) code.

For this PRML($G = 12, I = 6, M = 22$) code of length $N' = 2N = 234$, we will describe a method to reduce the global constraint from $G = 12$ to 8 while relaxing the interleave constraint from $I = 6$ to 7. There is only a minor capacity increase from the ($G = 12, I = 6$) constraint with capacity 0.994191 (up to 6 digits) to the ($G = 8, I = 7$) constraint with capacity 0.99617 (up to 6 digits). Thus, SWS encoding reduces the efficiency of the rate-232/234 code only slightly, viz., from 99.72% to 99.53%.

To enforce the $G = 8$ constraint, we transform any subsequence with more than 8 consecutive zeros into a subsequence that meets the $G = 8$ constraint. Furthermore, this substitution subsequence is chosen to be unique in the sense that the inverse encoder (decoder), which runs backwards from right to left, can recognize it and reconstruct the original subsequence, which had more than 8 consecutive zeros. Special care is required at the left and right codeword ends to enforce tighter constraints so that the modified codewords can be freely concatenated. A block diagram of this encoder is shown in Fig. 4. The registers $\mathbf{x}$ and $\mathbf{y}$ in the SWS encoder contain $L + 1 = 16$ bits.

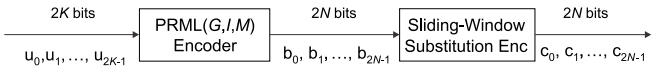

Fig. 4. Encoder for PRML($G = 8$, $I = 7$, $M = 24$) block code.

There are three types of local substitution maps: left-boundary substitution, substitutions within a codeword, and right-boundary substitutions, which will be described next. Substitutions within a codeword are based on the SWS encoder characterized by Table I in Section III. To transform an input sequence $\{b_i\}$ of length $N' = 234$ into an output sequence $\{c_i\}$ of length $N' = 234$, we first apply a 16-bit left-boundary substitution once (Table III), and then $N' - L - 1 = 218$ times 16-bit sliding-window substitutions within the codeword (Table I), followed by a single 15-bit right-boundary substitution (Table IV). Furthermore, this transformation is chosen such that a potential M constraint is only slightly weakened from M to $M + 2$.

A. Left-boundary substitution

The substitution at the left codeword boundary enforces $G_{left} = 4$ whenever $G_{left} > 4$. The non-identity substitution is specified by Table III. Note that $x_6 = 1$ always holds

TABLE III
Left boundary

Input $\mathbf{x} = x_0\, x_1 \ldots x_{15}$	Output $\mathbf{y} = y_0\, y_1 \ldots y_{15}$
$0\,0\,0\,0\,0\, x_5\, 1\, x_7\, x_8\, x_9 \ldots x_{14}\, x_{15}$	$0\, x_8\, 0\, 1\, 0\, x_5\, 0\, x_7\, 1\, x_9 \ldots x_{14}\, x_{15}$

because of $I_{left} = 3$. Only the first nine components are remapped; the last seven components pass unchanged. The even interleave of $\mathbf{y}$ starts with four zeros, which is a violation of $I_{left} = 3$ and thus allows one to detect this substitution during read-back. As the substitution map is one-to-one, it can be inverted by the modulation decoder. Furthermore, the I and G constraints of the original codeword towards the right of position 5 are maintained because the odd interleave is unchanged and x_8 in the even interleave was replaced by a 1. The M constraint towards the right is weakened by at most 1 because of replacing x_8 by 1.

B. Right-boundary substitutions

After completing the sliding-window operations of substitutions within a codeword, the length-16 sliding window has reached the right boundary. At the right boundary, the length-16 output register $\mathbf{y} = y_0 y_1 \ldots y_{14} y_{15}$ satisfies the I and G constraints of the original code towards the right; in particular, the tightened $I_{right} = 3$ constraint holds at the

TABLE IV
RIGHT BOUNDARY

Case	Relation	Input $\mathbf{x} = x_0\, x_1 \ldots x_{14}$	Output $\mathbf{y} = y_0\, y_1 \ldots y_{14}$
1.i	–	$x_0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, x_{11}\, x_{12}\, x_{13}\, x_{14}$	$x_0\, 0\, x_{11}\, x_{12}\, x_{13}\, x_{14}\, 1\, 1\, 0\, 1\, 0\, 0\, 0\, 1\, 0$
1.ii	–	$1\, 1\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, 1$	$1\, 1\, 0\, 0\, 0\, 0\, 1\, 1\, 0\, 1\, 0\, 0\, 0\, 1\, 0$
2.i	$x_{10} \neq x_0$	$x_0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, x_{10}\, x_{11}\, x_{12}\, x_{13}\, x_{14}$	$x_0\, 0\, x_{11}\, x_{12}\, x_{13}\, x_{14}\, 1\, 0\, 0\, 1\, 0\, 0\, 0\, 1\, 0$
2.ii	$x_{12} \neq x_2$	$x_0\, x_1\, x_2\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, x_{12}\, 1\, x_{14}$	$x_0\, x_1\, x_2\, 0\, 1\, x_{14}\, 1\, 1\, 0\, 0\, 0\, 0\, 0\, 1\, 0$
3.i	$x_{14} = 0$	$0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, x_{10}\, x_{11}\, x_{12}\, x_{13}\, 0$	$0\, 1\, x_{10}\, x_{11}\, x_{12}\, x_{13}\, 1\, 0\, 0\, 1\, 0\, 0\, 0\, 1\, 0$
3.ii	$x_{14} = 1$	$0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, x_{10}\, x_{11}\, x_{12}\, x_{13}\, 1$	$0\, x_{10}\, x_{11}\, x_{12}\, 1\, x_{13}\, 1\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, 0$
3.iii	$(x_{11}, x_{13}) \neq (0,0)$	$1\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, x_{11}\, x_{12}\, x_{13}\, x_{14}$	$1\, 0\, x_{11}\, x_{12}\, x_{13}\, x_{14}\, 1\, 0\, 0\, 0\, 0\, 1\, 0\, 1\, 0$
3.iv	$(x_{12}, x_{14}) \neq (0,0)$	$x_0\, 1\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, x_{12}\, x_{13}\, x_{14}$	$x_0\, 1\, 0\, x_{12}\, x_{13}\, x_{14}\, 1\, 0\, 0\, 0\, 0\, 1\, 0\, 1\, 0$
3.v	$x_{12} \neq x_3$	$x_0\, x_1\, x_2\, x_3\, 0\, 0\, 0\, 0\, 0\, 0\, 0\, x_{12}\, 1\, x_{14}$	$x_0\, x_1\, x_2\, x_3\, 0\, x_{14}\, 1\, 0\, 0\, 0\, 0\, 0\, 0\, 1\, 0$
4	$(x_7, x_9) \neq (0,0)$	$x_0\, x_1\, x_2\, x_3\, x_4\, x_5\, x_6\, x_7\, 1\, x_9\, 0\, 0\, 0\, 0\, 0$	$x_0\, x_1\, x_2\, x_3\, x_4\, x_5\, 1\, x_7\, 0\, x_9\, 0\, 1\, 0\, x_6\, 0$

right boundary. Furthermore, the substitutions up to this point ensure that $G = 8$ holds towards the left including the first nine components $y_0 \ldots y_7 y_8$ of the output register. It remains to check the last 15 components of $\mathbf{y}$ for violations of $G = 8$, to specify suitable substitution maps for these cases and, additionally, to enforce the boundary constraint $G_{right} = 4$.

1) Enforcing $G = 8$ within the 15 rightmost components: We consider a length-15 window and initialize the length-15 input register by the last 15 components of the output register, i.e., $\mathbf{x} = y_1 \ldots y_{14} y_{15}$. In particular, if no substitutions have occurred in the last few steps of the sliding-window map, then $\mathbf{x} = b_{N'-14} b_{N'-13} \ldots b_{N'}$ consists of the last 15 bit positions of the original codeword $\{b_i\}$. In any case, the length-15 input register contains a sequence $\mathbf{x}$ that towards the right satisfies the G and I constraints of the original code. Furthermore, when restricting ourselves to the case of $G = 8$ violations, we can assume that the component of the new codeword $\{c_i\}$ at the left side of the length-15 window is $c_{N'-15} = 1$.

We will proceed as in the case of substitutions within codewords and distinguish four cases. The first three cases correspond to violations of the $G = 8$ constraint, namely, the cases in which there are runs of 11 or 12 consecutive zeros, 10 consecutive zeros and 9 consecutive zeros in $\mathbf{x}$. These cases will be further subdivided depending on the position of the first zero of the zero-run. Case 4 corresponds to the violation of the right boundary constraint $G_{right} = 4$. The substitution maps given in Table IV result in local output sequences $\mathbf{y}$ that satisfy the $I_{even} = 4$ constraint at the right boundary because the pattern $1\,0\,0\,0\,0$ is at the end of the even interleave. Moreover, the substitution patterns $\mathbf{y}$=$y_0\, y_1 \ldots y_{14}$ are selected to have a unique characteristic subsequence $y_1\, y_3\, y_4\, y_7\, y_9\, y_{11}$. This property is used for the inverse substitutions.

The column "Relation" in Table IV gives a relation among the input bits in $\mathbf{x}$ for each case that has to be satisfied in order to make a substitution. These relations are important for the inverse encoder (decoder), which applies the inverse substitutions. For example, in Case 2.i, the relation $x_{10} \neq x_0$ ensures that there are exactly ten consecutive zeros in the first 11 components of $\mathbf{x}$. As another example, in Case 4, the input $\mathbf{x}$ satisfies the right boundary condition $I_{right} = 3$ and, therefore, there is the relation $(x_7, x_9) \neq (0,0)$.

In the first three cases and their subcases, the I and G constraints towards the left are maintained because any potential non-zero initial part $x_0 x_1 x_2 x_3$ of the window is substituted by either the same substring or a substring with 1s at some dedicated locations. Furthermore, the M constraint at the left of the substitutions remains unchanged in all cases. In the first three cases, the M constraint is maintained. In case 4, x_6 is substituted by 1 and, thus, the M constraint to the left of this position can be weakened by 1.

2) Enforcing $G_{right} = 4$: To guarantee $G_{right} = 4$ at the right boundary, we check the last nine codeword components, which correspond to the last nine components in $\mathbf{x}$. Clearly, in all cases, the substitutions of Table IV meet the $G_{right} = 4$ constraint.

V. CONCLUSIONS

A sliding-window substitution encoding technique has been introduced to improve on existing enumerative high-rate ($G = 2\gamma, I = \gamma$) codes. It has been shown that the new substitution technique can provide embedding of one-sided ($G = 2\gamma, I = \gamma$)-constrained sequences into one-sided sequences satisfying the constraints $G = \gamma + 1$ and $I = \gamma + 1$. Furthermore, it has been demonstrated that for specific constraints such an embedding is not possible for bi-infinite sequences. High-rate capacity-efficient PRML(G, I) block codes, which impose additional constraints at codeword boundaries by enforcing a separate set of substitutions, have been constructed. Specifically, a rate-232/234 PRML($G = 8, I = 7, M = 24$) code for practical applications has been designed.

REFERENCES

[1] K. A. S. Immink, P. H. Siegel, and J. K. Wolf,"Codes for digital recorders" *IEEE Trans. Inform. Theory*, vol. 44, pp. 2260–2299, Oct. 1998.

[2] W. H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inform. Theory*, vol. 11, pp. 284–292, Apr. 1965.

[3] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1389–1399, Sept. 1997.

[4] M. Blaum, R. D. Cideciyan, E. Eleftheriou, R. Galbraith, K. Lakovic, T. Mittelholzer, T. Oenning and B. Wilson, "Enumerative encoding with non-uniform modulation constraints," *IEEE Proc. Intl. Symp. Inform. Th. (ISIT'07)*, Nice, France, pp. 1831–1835, June 24 - 29, 2007.

[5] T. Mittelholzer, "Enumerative maximum transition run codes," *IEEE Proc. Intl. Symp. Inform. Th. (ISIT'09)*, Seoul, Korea, pp. 1549–1553, June 28 - July 3, 2009.

[6] D. Lind and B. Marcus, *Symbolic Dynamics and Coding*, Cambridge Univ. Press, 1995.

[7] US patent application US20140085114 A1.