A Combinatorial Technique for Constructing High-Rate MTR–RLL Codes
Adriaan J. van Wijngaarden, Member, IEEE, and Emina Soljanin, Member, IEEE

Abstract—We present advanced combinatorial techniques for constructing maximum runlength-limited (RLL) block codes and maximum transition run (MTR) codes. These codes find widespread application in recording systems. The proposed techniques are used to construct a high-rate multipurpose modulation code for recording systems. The code, a rate 16/17, (0, 3, 2, 2) MTR code, that also fulfills (0, 15, 9, 9) RLL constraints is a high-rate distance-enhancing code with additional constraints for improving timing and gain control. The encoder and decoder have a particularly efficient architecture and allow an instantaneous translation of 16-bit source words into 17-bit codewords and vice versa. The code has been implemented in Lucent read-channel chips and has excellent performance.

Index Terms—Constrained coding, intersymbol interference, magnetic recording, runlength codes.

I. INTRODUCTION

A VARIETY of block codes that fulfill certain channel input constraints have been developed for recording channels during the last several decades (see [1]–[4] and references therein). One important category is the class of runlength-limited codes (RLL codes). The codewords of an RLL code are runlength-limited sequences of a fixed length \( n \), also known as \((d, k)\) sequences, that are characterized by the parameters \( d \) and \( k \), which indicate the minimum and maximum number of “zeros” between consecutive “ones” in the binary sequence, respectively. Usually, a “one” and a “zero” correspond to a reversal and a nonreversal in magnetization, respectively. As a consequence, the parameter \( d \) controls the intersymbol interference and the parameter \( k \) determines the self-clocking properties of the sequence.

Current high-density recording systems use a detection technique that combines partial response equalization with maximum-likelihood sequence detection, often referred to as PRML. Intersymbol interference can be mitigated by reconstructing the recorded sequence from sample values of a suitably equalized readback channel. The current systems often use a partial response channel model with transfer function \( H(D) = (1 - D)(1 + D)^N \), which is referred to as a \( E^N - 1 \) PR4 channel [5], [6]. Sequence detection can be enhanced for these channels by reducing the set of possible recording sequences in order to increase the minimum distance. In [6]–[8] and references therein, the pairs of sequences with low minimum distance have been identified for important classes of partial response channels, and codes based on constrained systems have been proposed. For the \( E^2 \) PR4 channel, which is an appropriate model for the current linear recording densities, the constraints that have been specified can be realized by restricting the maximum runlength of consecutive “ones.” The corresponding codes, known as maximum transition run codes (MTR codes), are equivalent to \((0, k)\) RLL codes with “ones” and “zeros” interchanged. It has been shown in [7]–[9] that a \((0, 2)\) MTR code will be sufficient to increase the minimum distance. However, the upper bound on the rate of a \((0, 2)\) code is only 0.8791 [1]. By relaxing the constraints, higher rates can be achieved by specifying constraints that do not remove all undesired pairs but instead significantly reduce the number of undesired pairs of sequences with minimum distance [8], [9]. In this way, a tradeoff is made between code rate, code complexity, and detection performance.

We consider the construction of \((0, k, k_L, k_R)\) block codes, where \( k_L \) and \( k_R \) denote the maximum number of leading and trailing “zeros” of a codeword. Generally, \( k_L + k_R \leq k \), to allow the construction of long \((0, k)\) sequences by concatenating the codewords of a \((0, k, k_L, k_R)\) block code. As indicated, \((0, k, k_L, k_R)\) MTR codes are \((0, k, k_L, k_R)\) RLL codes with “ones” and “zeros” interchanged. Several \((0, k, k_L, k_R)\) codes that are employed in recording systems have been implemented with look-up tables or combinatorial circuitry [1], [3], [5], [10]–[13]. This imposes limitations on the block length and the achievable rate. The majority of codes have a rate \( 8/9 \), and only recently, codes with longer block lengths have been developed, either by using algorithms [4], [14], [15] or by interleaving a short block code with uncoded bits [15]–[17]. The latter methods increase the rate and limit error propagation, but the constraints of the resulting code are certainly not the best possible for the given rate. The combinatorial construction methodology that is proposed here is shown to be a very powerful tool for the design of high-rate constrained codes. Integration of the code into a device is feasible because of its highly parallelized architecture and its small delay and high throughput.

In Section II, we briefly describe some properties of \((0, k, k_L, k_R)\) sets. In Section III, we give an outline of the design methodology for the construction of high-rate runlength-limited codes. We illustrate this methodology in Section IV by detailing the construction of a rate 16/17, \((0, 3, 2, 2)\) MTR code that satisfies \((0, 15, 9, 9)\) RLL constraints. The \((0, 3, 2, 2)\) MTR constraints are the smallest possible for the given code rate, and the \((0, 15, 9, 9)\) RLL constraints are sufficient to provide timing and gain control.

Manuscript received March 3, 2000.

The authors are with the Mathematics of Communications Research Department, Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA.

Publisher Item Identifier S 0733-8716(01)01753-X.
II. Definitions

Let $A^m$ denote the set of binary sequences of length $m$. A sequence $X \in A^m$ is represented as a string of symbols $x_1x_2\cdots x_m$ of length $m$. The concatenation of two sequences $X$ and $Y$ is denoted by $XY$, and $PS^{(m)} = \{PX | X \in S^{(m)} \}$ denotes a set of sequences of length $p + m$ with prefix $P \in A^p$ and suffix $X \in S^{(m)}$, where $S^{(m)} \subseteq A^m$. The null string $\emptyset$ represents a string of length 0 for which $\emptyset X = X \emptyset = X$. A run of $m$ consecutive symbols $q$ is written as $q^m$. We use $*$ and $(s)^m$ to denote any element of $A$ and $A^m$, respectively.

The set $G_{k,\bar{k},l_\ell}$ of length-$n$ sequences that satisfy the $(0, k, \bar{k}, l_\ell)$ constraints on the runs of consecutive ones and its size $G_{k,\bar{k},l_\ell}$ can be determined recursively. For $n \leq \min(k, \bar{k})$, we have $G_{k,\bar{k},l_\ell} = A^k$, and for $n > k$, we have

\[ G_{k,\bar{k},l_\ell} = \bigcup_{i=0}^{l_\ell} \{Q | Q_{(n-i)-1}^i \} . \]

The set $G_{k,\bar{k},l_\ell}$ can be determined using (1) for $t = k + 2$. For $t \leq k$, we have $G_{k,\bar{k},l_\ell} = A^t$, and for $t = k + 1$, we have $G_{k,\bar{k},l_\ell} = A^1 \{1, k+1 \}$. For $k + 2 \leq t \leq k + 1$, it can be easily verified that $G_{k,\bar{k},l_\ell} = G_{k,\bar{k},l_\ell} \{t-1 \} \cup 1G_{k,\bar{k},l_\ell} \{t-1 \}$. For $t \leq k + 1$, we thus have

\[ G_{k,\bar{k},l_\ell} = \begin{cases} 2^t, & \text{if } t \leq k \\ 2^t - k - 1(2^t - 1), & \text{if } k < t \leq k + 1. \end{cases} \]

Because the sets in the union of (1) are mutually disjoint, we obtain the following recursion for $n \geq k + 2$:

\[ G_{k,\bar{k},l_\ell}^{(n)} = \sum_{i=0}^{l_\ell} G_{k,\bar{k},l_\ell}^{(n-i-1)}. \]

We define $F_{k,\bar{k},l_\ell}^{(n)}$ to be the set of length-$n$ sequences that satisfy the $(0, k, \bar{k}, l_\ell)$ constraints on the runs of consecutive zeros and denote by $F_{k,\bar{k},l_\ell}^{(n)}$ the size of $F_{k,\bar{k},l_\ell}^{(n)}$. By reversing the zeros and the ones in the expressions above, we obtain identical results for $F_{k,\bar{k},l_\ell}^{(n)}$ and $F_{k,\bar{k},l_\ell}^{(n)}$. In particular, $F_{k,\bar{k},l_\ell}^{(n)} = F_{k,\bar{k},l_\ell}^{(n)}$. The above expressions will be frequently used to determine the number of (sub)sequences of a given length that satisfy given $(0, k, \bar{k}, l_\ell)$ constraints.

III. Combinatorial Construction Methods

We present an efficient methodology for constructing $(0, k, \bar{k}, l_\ell)$ codes by mapping $(n-1)$-bit source words onto $n$-bit codewords of a $(0, k, \bar{k}, l_\ell)$ code. It improves on techniques presented in [15]. If the number of sequences of length $n$ that satisfy the $(0, k, \bar{k}, l_\ell)$ constraints exceeds $2^{n-l-1}$, it is always possible to perform this mapping, although the complexity may be high. For several codeword lengths, it is possible to realize this mapping procedure using a small, digital combinational circuit.

A sequence $X \in A^m$ is represented as a string of symbols $x_1x_2\cdots x_m$ of length $m$.

The construction methodology has the following characteristics.

- A mapping is defined for a set of source words rather than for one word. Based on the specific constraints, the set of source words and the set of codewords are partitioned into disjoint subsets. For every set, a mapping is specified. The words belonging to the same subset are transformed in the same fashion.
- Symmetry is used whenever possible; i.e., if $Y = \psi(X)$ represents the mapping of $X$ onto $Y$, then $Y'$ and $Y''$, representing the words $X$ and $Y$ in reversed order, satisfy whenever possible the relation $Y'' = \psi(X')$. This significantly simplifies the design.
- The symbols of the source word are, whenever possible, directly mapped onto corresponding codeword positions. Whenever changes are required, the source symbols that cause the constraint to be violated are specified by identifying the type of violation, and the other source symbols that contain information are permuted.
- The translation of $(n-1)$ information bits into $n$-bit codewords and vice versa is done in parallel.
- The combinational circuitry that results from the methodology simplifies the design process and the structure of the digital combinational circuitry.

IV. Design of the Rate 16/17, $(0, 3, 2, 2)$ MTR, $(0, 15, 9, 9)$ RLL Code

In this section, we will describe the construction of the rate 16/17, $(0, 3, 2, 2)$ MTR–RLL code in detail. The objective is to construct a code $C$ that is a subset of $G_{3,2,2}$ and has a cardinality $2^{16}$. Because $G_{3,2,2} \cong 65753$, we have a surplus of 217 codewords. We will show at a later stage that this will allow us to simultaneously fulfill (0, 15, 9, 9) RLL constraints by excluding 199 words to obtain a code $C \subseteq G_{3,2,2} \cap F_{135,9,9}$ with a surplus of 18 candidate words.

We will detail the technique of mapping source words onto codewords that fulfill both the $(0, 3, 2, 2)$ MTR constraints and the $(0, 15, 9, 9)$ RLL constraints. This method inherently specifies the structure of the encoder and decoder that map 16-bit source words in parallel onto 17-bit codewords and vice versa.
A. Partition of the Set of Data Words

We will develop a mapping technique in which the symmetry relative to the central position of the codeword plays an important role. Let \( Y = y_1 y_2 \cdots y_{17} \) be an arbitrary element of \( A^{17} \), and denote \( Y' = y_{17} y_{16} \cdots y_1 \) to be the sequence \( Y \) in reversed order. The set \( A^{17} \) has the property that \( Y \) if and only if \( Y' \). Let \( \psi(X) \) denote the mapping of a 16-bit source word onto a 17-bit codeword. Whenever possible, \( \psi(X) \) will be the codeword associated with \( X \), i.e., \( \psi(Y) = \psi(Y) \). This will help to reduce the complexity of the analysis and the design. The symmetry is taken into account by decomposing \( X \) into two sequences \( A \) and \( B \), where \( A \) and \( B \). Let \( A' \) be the sequence \( A \) in reversed order, i.e., \( A' = x_1 x_2 \cdots x_8 \), \( B \) is given by

\[
X = A' B
\]

where \( y_6 \) denotes the central position of the 17-bit codeword. The index reflects the symmetry relative to the central position of the codeword.

We now consider the 17-bit word \( Y = Y' \quad \psi(X) \), i.e., \( y_i = x_i \), \( y_i = x_i \) for \( 1 \leq i \leq 8 \), and \( y_{10} = 0 \). It can be verified that \( Y \) if and only if \( X' \in C^{(8)} \) if and only if \( X' \in C^{(8)} \), and therefore, \( C^{(8)} \times C^{(8)} = \{0, 1\} \times \{0, 1\} \). The right-most column gives the number of codewords of \( C^{(8)} \).

Let \( V_0 \) denote the set of words of length 8 such that fulfill the (0, 3, 2, 3) MTR constraints. The set \( A^8 \) is partitioned into five disjoint subsets \( V_1, \ldots, V_5 \), which are specified in Table I. Every set \( V_i \), \( 1 \leq i \leq 5 \), consists of words that have in common a substring that violates the imposed (0, 3, 2, 3) MTR constraints. Define \( V \) to be the size of the set \( V_i \). The values of \( V_1 \) up to \( V_5 \) are specified in Table I. The size of \( V_0 \) is given by

\[
V_0 = C^{(8)} = 193 \times 193 = 37249
\]

Because the six sets \( V_1, \ldots, V_5 \) are disjoint and their union equals \( A^8 \), every source word \( X \in A^8 \) can be represented by \( X = X' X'' \), where \( X' \in V_1 \) and \( X'' \in V_2 \). We now introduce the notation \( W_{i,j} = V_i \times V_j \). Because the sets \( V_1, \ldots, V_5 \) are disjoint, the sets \( W_{i,j} \) are consequently disjoint, and therefore, the set of source words \( A^{16} \) is uniquely partitioned. This implies that there are 36 different categories that have to be considered. Define \( W_{i,j} \) to be the number of elements of \( W_{i,j} \). Table II lists the values of \( W_{i,j} \) for each of the 36 sets \( \psi(X) \). The sum of the values \( W_{i,j} \) in Table II is equal to 65 536, as expected. To determine to which set \( V_i \) a word \( X \in A^8 \) belongs, we evaluate the binary equations

\[
\begin{align*}
v_1 & = x_1 \times x_2 \times x_3, \\
v_2 & = \overline{x_1} \times x_2 \times x_3 \times x_4 \times x_5, \\
v_3 & = \overline{x_2} \times x_3 \times x_4 \times x_5 \times x_6, \\
v_4 & = \overline{x_3} \times x_4 \times x_5 \times x_6 \times x_7, \\
v_5 & = \overline{x_4} \times x_5 \times x_6 \times x_7 \times x_8, \\
v_6 & = \overline{x_5} \times \overline{x_6} \times \overline{x_7} \times \overline{x_8} \
\end{align*}
\]

where \( \overline{x} \) denotes the inverse of the binary value \( x \) and the dot-operator denotes the binary AND operation. We can easily verify that \( v_i \) if \( X \in V_i \), and \( v_i = 0 \) otherwise.

In order to determine the set \( W_{i,j} \) a source word \( X = X' X'' \) belongs to, this set of equations is evaluated for \( X' \) and \( X'' \).

B. Partition of the Set of Codewords

The set \( C^{(17)} \) will be partitioned into eight disjoint subsets \( M_0, \ldots, M_7 \), which are specified in Table III. The dots in this table are placeholders for binary symbols of elements of the set specified in the third column of the table. It is easy to see that all possible combinations of the central positions are specified. The right-most column gives the number of codewords of \( M_i \).
The sum of the last column equals \( G_{3,2}^{(47)} = 65,753 \). Because of the symmetry of the central positions, we have for \( 1 \leq m \leq 3 \), \( |\mathcal{M}_{2m-1}| = |\mathcal{M}_{2m}| \). The sets \( \mathcal{M}_i \), where \( 1 \leq i \leq 7 \), automatically fulfill the \((0,15,9,9)\) RLL constraints in addition to the \((0,3,2,2)\) MTR constraints.

### C. Mapping Techniques

Having completed partitioning the set of source words and the set of candidate codewords, we will specify the assignment of codewords to source words in a compact format by mappings that inherently define the encoder and the decoder. It is often necessary to further partition the sets \( \mathcal{W}_{i,j} \) into disjoint subsets, denoted by \( \mathcal{W}_{i,j}^{(p)} \), where \( 1 \leq p \leq P_{i,j} \). These partitions have the property that

\[
\mathcal{W}_{i,j}^{(p)} \subseteq \mathcal{W}_{i,j}^{(p-1)} \bigcup_{s=1}^{p-1} \mathcal{W}_{i,j}^{(s)} \quad \text{and} \quad \mathcal{W}_{i,j} = \bigcup_{p=1}^{P_{i,j}} \mathcal{W}_{i,j}^{(p)}.
\]  

The mapping of a set of source words \( X \in \mathcal{W}_{i,j}^{(p)} \) onto a set of codewords \( Y \in \mathcal{M} \) is specified by the templates \( X_{i,j}^{(p)} \) and \( Y_{i,j} \). The first string \( X_{i,j}^{(p)} \) specifies the set \( \mathcal{W}_{i,j}^{(p)} \) by indicating the relative positions of zeros, ones, and source symbols. The second string \( Y_{i,j} \) specifies the subset of \( \mathcal{M}_g \), onto which the elements of \( \mathcal{W}_{i,j}^{(p)} \) are mapped.

We choose to let the central positions indicate at which side the violation occurred. The elements \( \mathcal{W}_{i,j} \) of the sets \( \mathcal{W}_{i,j} \), where \( 1 \leq i \leq 5 \), will be mapped onto elements of the sets \( \mathcal{M}_1, \mathcal{M}_3, \) and \( \mathcal{M}_5 \). This makes it easy to use the symmetry and to use the central positions in reversed order to handle the violations \( \mathcal{W}_{i,j} \).

The mapping is “symmetric” and marked by \( \Sigma \) if an element \( X \in \mathcal{W}_{i,j}^{(p)} \) is translated into an element \( Y \in \mathcal{M}_{2m-1} \), \( 1 \leq m \leq 3 \), and the element \( X' \in \mathcal{W}_{i,j}^{(p)} \) is translated into \( Y' \in \mathcal{M}_{2m} \). We will often specify one mapping for several sets \( \mathcal{W}_{i,j}^{(p)} \).

We first consider the mapping of the elements \( X \in \mathcal{W}_{i,j}^{(p)} \), specified in a compact format in Table IV. The elements \( X \in \mathcal{W}_{i,j} \) are mapped onto the elements of \( \mathcal{M}_g \), unless the \((0,15,9,9)\) RLL constraints are violated. Table IV specifies the subsets \( \mathcal{W}_{i,j}^{(p)} \) up to \( \mathcal{W}_{i,j}^{(5)} \), for which the \((0,15,9,9)\) RLL constraints would be violated, and the remaining subset \( \mathcal{W}_{i,j}^{(6)} \), which satisfies the \((0,15,9,9)\) RLL constraints. The subsequences that violate the constraints are boldfaced to emphasize the kind of violation that is under consideration. The underlined zeros and ones in the template \( Y \) indicate which positions uniquely specify the mapping. The decoder checks these positions, and if there is a match, the mapping will be reversed to reconstruct the source word.

We now consider the mapping of the elements \( X \in \mathcal{W}_{i,j}^{(1)} \) in more detail. The template \( X \) specifies the additional constraints that are imposed. Because \( X' \in \mathcal{W}_o \), we have \( x_1 x_2 \cdots x_6 \in \mathcal{G}_{3,2}^{(6)} \). In other words, \( X \) specifies \( \mathcal{G}_{3,2}^{(47)} = 48 \) different source words. The template \( Y \) specifies the corresponding codewords. Because \( x_1 x_2 \cdots x_6 \in \mathcal{G}_{3,2}^{(6)} \), we have \( x_1 x_2 x_3 \neq 111 \) and \( x_1 x_2 x_3 \neq 111 \). This guarantees that the words corresponding to template \( Y \) satisfy the imposed constraints.

The five mappings in Table IV specify the transformation of \( \mathcal{W}_{i,j} = 3724/9 \) source words into codewords. The sets \( \mathcal{W}_{i,j}^{(1)} \), \( \mathcal{W}_{i,j}^{(2)} \), \( \mathcal{W}_{i,j}^{(3)} \), \( \mathcal{W}_{i,j}^{(4)} \), and \( \mathcal{W}_{i,j}^{(5)} \) are mapped onto subsets of \( \mathcal{M}_1 \) and \( \mathcal{M}_3 \), and the set \( \mathcal{W}_{i,j}^{(6)} \) is mapped onto \( \mathcal{M}_5 \).

Let \( z' = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 \) and \( z^r = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 \). To determine to which set \( \mathcal{W}_{i,j}^{(p)} \) a word \( X \in \mathcal{W}_{i,j} \) belongs, we can evaluate the following set of equations, which were directly obtained from Table IV:

\[
\begin{align*}
\omega_{0,0} & = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9, \\
\omega_{0,0} & = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9, \\
\omega_{0,0} & = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9, \\
\omega_{0,0} & = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9, \\
\omega_{0,0} & = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9.
\end{align*}
\]

It can be verified that exactly one out of the five binary values \( \omega_{0,0}^{(1)} \), \( \omega_{0,0}^{(2)} \), \( \omega_{0,0}^{(3)} \), \( \omega_{0,0}^{(4)} \), \( \omega_{0,0}^{(5)} \) will be nonzero.

Table V specifies the mapping for the source words \( X \in \mathcal{W}_{i,j} \), \( 1 \leq i \leq 5 \), where \( \mathcal{W}_{i,j} \) is the shorthand notation of \( \mathcal{W}_{i,j}^{(p)} \) that have the property that by reversing the order of \( X \) and \( Y \) the mapping of the
TABLE V

<table>
<thead>
<tr>
<th>( W_{i,j} )</th>
<th>( X )</th>
<th>( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_{1,0} )</td>
<td>1 1 1 1 1 1</td>
<td>1 1 1 1 1 1</td>
</tr>
<tr>
<td>( W_{2,0} )</td>
<td>0 1 1 1 1 1</td>
<td>0 1 1 1 1 1</td>
</tr>
<tr>
<td>( W_{3,0} )</td>
<td>( x_0 ) 0 1 1 1 1</td>
<td>( x_0 ) 0 1 1 1 1</td>
</tr>
<tr>
<td>( W_{4,0} )</td>
<td>( x_0 ) 1 0 1 1 1 1</td>
<td>( x_0 ) 1 0 1 1 1 1</td>
</tr>
<tr>
<td>( W_{5,0} )</td>
<td>( x_0 ) ( x_0 ) 0 1 1 1 1</td>
<td>( x_0 ) ( x_0 ) 0 1 1 1 1</td>
</tr>
<tr>
<td>( W_{6,0} )</td>
<td>( x_0 ) ( x_0 ) 0 1 1 1 1</td>
<td>( x_0 ) ( x_0 ) 0 1 1 1 1</td>
</tr>
</tbody>
</table>

The mapping for the 13 remaining sets will be specified in the next tables.

We first consider the sets \( W_{i,j} \), where \( 1 \leq i \leq 3 \) and \( 1 \leq j \leq 3 \). The mapping, specified in Table VI is of the form \( Y = X \psi(X) \), where \( \psi(X) \) is a mapping that depends on the left part of the codeword \( X \). The same holds for the right part of the codeword, which is given by \( \psi(X') \), depending on the right part of the codeword. As an example, consider the source word \( X = X' X'' \). We therefore obtain \( Y = X \psi(X') \psi(X'') \).

It can be verified that \( |V_1^{[1]}| = 4 \), \( |V_1^{[2]}| = 28 \), \( |V_2^{[1]}| = 2 \), \( |V_2^{[2]}| = 6 \), and \( |V_3| = 8 \). Because the mapping above applies to both the left-hand side and the right-hand side of an element.
The mapping of the words $X \in \mathcal{W}_{4,1}$ is specified in Table VII. The sets $\mathcal{W}_{4,4}$ and $\mathcal{W}_{5,5}$ are the only sets for which the mapping has not been specified yet. As there are only very few candidate words remaining, the sets have to be further partitioned in order to be able to map all of their elements. The resulting mapping is specified in Table VIII. At this stage, the specification is completed. There are 18 surplus candidate words, one or several of which might be used to replace codewords or as special recognition sequences for frame synchronization.

### D. Derived Codes

The rate 16/17 (0, 3, 2, 2) code is suitable for the construction of higher rate codes because of its tight constraints. This is achieved by interspersing the codewords with uncoded source symbols, as described in [15]–[17]. Among others, we obtain a rate 32/33 (0, 7, 6, 6) code and a rate 48/49 (0, 11, 6, 6) code by inserting one and two arbitrary symbols between the bits of the codeword, respectively. Error propagation is limited to two bytes. To further suppress error propagation, we can use a strategy in which the interspersed bits are observed and the source symbols of the underlying constrained code are transformed only if the overall constraints are violated. Consider a rate $(n - 1)/n$, $(0, k_1, k_2, k_3)$ RLL code that is obtained by interspersing a rate $(n' - 1)/n'$, $(0, k'_1, k'_2, k'_3)$ RLL code, as described in [15]–[17]. The mapping $X = X'X'' \in \mathcal{A}^{n'/2}$ onto $Y = X'X''X'$ is used as long as the resulting length-$n$ sequence fulfills the $(0, k_1, k_2, k_3)$ constraints, even if $Y$ does not fulfill the $(0, k'_1, k'_2, k'_3)$ constraints. This strategy has the advantage that source symbols are transformed less often. In general, the probability that the translation of a source word $U = ST \in \mathcal{A}^s \times \mathcal{A}^t$ onto $Z = S1T$ satisfies the $(0, k_1, k_2, k_3)$ constraints is given by

\[
\Pr\left[Z \in \mathcal{F}^{(n)}_{k_1,k_2,k_3}\right] = \frac{\mathcal{F}^{(s)}_{k_2,k_3} \cdot \mathcal{F}^{(t)}_{k_3,k_2}}{2^{s+t}},
\]  

where $n = s + t + 1$. For example, the probability that a source word for the rate 16/17 (0, 3, 2, 2) code needs to be transformed...
is 0.432, whereas the probability that a source word of the rate 48/49 (0, 11, 6, 6) code needs to be transformed is only 0.0185 if the inserted bits surrounding the 17-bits of the 16/17 (0, 3, 2, 2) code are observed. As this resembles the “uncoded” situation, error propagation will consequently be further limited.

V. CONCLUSION

We have presented a methodology for efficiently converting user information into a constrained sequence in which restrictions are imposed on the runs of consecutive-like symbols. The techniques are specifically useful for the construction of MTR codes and RLL codes, which find widespread application in recording systems. The proposed techniques are exemplified by constructing a high-rate multipurpose modulation code that is of interest for novel recording systems. The constructed code, a rate 16/17, (0, 3, 2, 2) MTR code that also fulfills (0, 15, 9, 9) RLL constraints, has been described in detail. This particular code, which has been implemented in a Lucent read-channel chip, has an excellent performance. The MTR constraints and the RLL constraints are the best possible for the given rate 16/17. The code is a very efficient, multipurpose code and is expected to be very useful as a basic code for the construction of higher rate constrained codes by interleaving the codewords with uncoded bits.

REFERENCES


Adriaan J. (de Lind) van Wijngaarden (S’87–M’98) was born in Amsterdam, The Netherlands, in 1968. He received the engineering (M.Sc. equivalent) degree in electrical engineering from the Eindhoven University of Technology, Eindhoven, the Netherlands, in 1992, and the doctorate degree in engineering from the University of Essen, Essen, Germany, in 1998.

From 1992 to 1998, he was a Research Engineer of the Digital Communications Group at the Institute for Experimental Mathematics, University of Essen. During this period, he has also been a Visiting Scientist at the Communications Research Centre, Ottawa, Ont, Canada, and at the University of Electro-Communications, Tokyo, Japan. As of 1998, he is with the Mathematics of Communications Research Department, Bell Laboratories, Lucent Technologies, Murray Hill, NJ. His research interests include communications, in particular, error control coding, synchronization and modulation coding with applications in recording, and optical and wireless communication systems.

Emina Soljanin (S’90–M’92) received the B.S. degree in electrical engineering from Sarajevo University, Bosnia and Herzegovina, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, in 1989 and 1994, respectively.

From 1986 to 1988, she was with the EnergyInvest Company, Bosnia and Herzegovina, developing optimization algorithms and software for power system control. After graduating from Texas A&M, she joined Bell Laboratories, Murray Hill, NJ, as a member of the Technical Staff in the Mathematical Sciences Research Center. Her research interests include communications, information and coding theory, and their applications to storage systems. Dr. Soljanin was the recipient of the 1992 Texas A&M University Electrical Engineering Department Fouraker fellowship. She served as a Technical Proofreader for the IEEE TRANSACTIONS ON INFORMATION THEORY (1990–1992). She is currently the Associate Editor for Coding Techniques for the IEEE TRANSACTIONS ON INFORMATION THEORY.