many valuable comments. A. Vardy wishes to thank Dr. S. Rubinstein, Dr. H. Rinat, and H. Itzkowitz.

REFERENCES


A Coding Scheme for Single Peak-Shift Correction in (d, k)-Constrained Channels

A. V. Kuznetsov and A. J. Han Vinck

Abstract—A two step coding scheme for peak-shift correction in (d, k)-constrained sequences is described. The first step is based on q-ary (q = d–d–1 = 1) block codes that allow correction of specific types of double errors caused by single peak-shifts. The second step is a simple conversion of q-ary symbols to binary strings of the type 00...01. The concatenation of these strings satisfies the (d, k)-constraint within the codeword and in concatenation with neighboring words. The length of the codewords is controlled and, if necessary, can be fixed. The rate Rq of the overall encoding approaches (2log2 (k–d+1))/(k–d+2) for large codeword lengths. Codes for correction of peak-shift, deletions, and insertions of zeros are presented as well. Encoding and decoding are done by simple algorithms without using look-up tables, enumeration or de-numeration procedures and, therefore, the code-length may be large.

Index Terms—Recording, peak-shift correction, (d, k)-constraint.

I. INTRODUCTION

In digital magnetic and optical storage, binary sequences are used as modulation codes, and for the standard modulation scheme known as NRZI the "I's" of the recorded codeword determine the positions on the track of storage media where the magnetization is changed from one state to another one. When data bits are directly used as modulation codes, the number of bits per unit of track length (linear density) is equal to n = 1/δ, where δ is the minimum acceptable distance between two consecutive transitions on the track. It was noted already in the sixties that the same minimum gap δ between transitions can be obtained with modulation codes that have (d + 1) times more bits per unit of track length but that satisfy the condition that the number of zeros between any two consecutive ones is not less than d. Since the number of such sequences equals 2n, N ≈ (d + 1)C(d), where C(d) is the Shannon capacity of the channel with d-constrained input, then the ratio γ = N/n is equal to (d + 1)C(d) and grows with d [1]. It characterizes an improvement in the linear density we gain by coding, and for this reason may be called the coding gain. For clock recovery the number of zeros between any two consecutive ones of the modulation code is upper bounded by another integer parameter k (≥ d + 1). For small values of d = 1, 2 many good recording codes have been constructed in

Manuscript received April 22, 1991; revised December 7, 1992. A. V. Kuznetsov was supported by Eindhoven University and Essen University. This work was presented at the IEEE International Symposium on Information Theory, San Antonio, TX, January 17–22, 1993. This work was done while A. V. Kuznetsov was with Eindhoven University in 1990 and with Essen University in 1991 on leave from the Institute for Problems of Information Transmission (IPPI) of the Academy of Sciences of Russia. A. V. Kuznetsov is with the Institute for Problems of Information Transmission, Academy of Sciences of Russia, Ermolovoy str. 15, Moscow 101447, Russia.

A.J. Han Vinck is with the University of Essen, Institute for Experimental Mathematics, Universitatsstr. 29, 4500 Essen, Germany.

IEEE Log Number 9209661.
the last twenty five years (full references on that subject are given in [2], [3]).

Large values of $d$ could give larger coding gain, but the complexity of encoding and decoding grows with $d$, and problems arise already for $d = 3$ and $4$ when the codewidth is greater than 20. Codes with a larger value of $d$ give larger coding gain, but they are penalized by an increasingly difficult trade-off between the detection window and the density ratio in applications with very high information density and data rates [2]–[4]. Codes detecting and correcting different types of errors in $(d, k)$-constrained sequences were suggested in [5]–[10], but look-up tables or enumeration/denumeration procedures are used for encoding and decoding. A method of using ternary BCH codes to correct multiple peak-shifts that occur in $(d, k)$-sequences, produced by a conventional variable-length code with arbitrary $(d, k)$ constraints, was presented in [11].

In this correspondence, we describe a simple two step coding scheme for the peak-shift correction in $(d, k)$-constrained sequences. It does not use look-up tables or enumeration/denumeration procedures and, therefore, can be used for large values of $d$ and codewidth.

The presented codes, as well as the others known at the moment, are not a "medicine" for all kind of distortions. They are constructed to correct single peak-shifts for $t \leq (k - d)/2$ positions left or right only. However, the proposed coding scheme is more general and can be extended to correct or detect other types of errors as well. Codes for correction of peak-shifts, deletions, and insertions of zeros are described.

The coding scheme can be outlined in the following way. If a one in the $(d, k)$-sequence erroneously appears in an adjacent bit cell, we call this event a peak-shift. A peak-shift leads to two adjacent errors. To correct such errors, a message (data) is first encoded into a q-ary code word of fixed length. Transparent block codes are constructed for this purpose. These codes are similar to q-ary Hamming codes and have simple encoding and decoding algorithms. For the transmission through the $(d, k)$-constrained channel, q-ary codewords are converted to binary sequences satisfying the $(d, k)$-constraint. This is simply done by replacing q-ary components by binary strings of $i$, $d \leq i \leq k$, consecutive zeros followed by a single one. To improve the worst case transmission rate, we control the length of the generated binary sequence. For the transparent codes this can be done without losing the error correction capability, at the cost of approximately one information bit. If necessary, the length of the transmitted binary $(d, k)$-constrained codewords can be fixed.

The maximum length of the binary codewords together with the code redundancy determine the transmission rate. For the given method, the rate approaches $210g_2$ for large values of $d$ and $k$.

II. ENCODING

The general approach we use for transmission is based on the scheme shown in Fig. 1, which includes the following two step encoding procedure.

The first step is necessary to make peak-shift correction possible and is performed by the block encoder. For any given message $m \in \{1, 2, \ldots, M\}$ the block encoder forms a codeword $z = f(m) = (z_1, z_2, \ldots, z_N)$ of length $N$ with components $z_i \in R$, $1 \leq i \leq N$, where $R$ is the output alphabet of the encoder. Since the encoding will be linear, the alphabet $R$ must have the algebraic structure of a ring. Let $R = \{0, 1, 2, \ldots, q - 1\}$ and addition as well as multiplication of the elements of $R$ are performed modulo $q$. In fact, in Section IV, we assume that $R$ is the field GF($q$) with $q = k - d + 1$ elements.

The second step of encoding is a one-to-one mapping $y$ from $R$ to the set $\{0^{d+1}, 0 \leq i \leq q - 1\}$ of $q = k - d + 1 = [R]$, binary strings $0^{d+1}$, the rth of which is a piece of a subset of $d + i$ zeros followed by a single one. It is performed by the q-ary to binary converter called $q/2$-converter (see Fig. 1). For the input sequence $g \in f(m) = (z_1, z_2, \ldots, z_N)$ the $q/2$-converter generates as an output the binary sequence $\phi(g) = (\psi(z_1), \psi(z_2), \ldots, \psi(z_N))$ which is a concatenation of strings $\psi(z_i), 1 \leq i \leq N$. Obviously, all binary sequences $\phi(z)$ satisfy the $(d, k)$-constraint. The total number of binary components of $\phi(z)$, or the length $L(\phi(z))$, depends on $g$, but is always larger than or equal to $(d + 1)N$ and less than or equal to $(k + 1)N$. Using the technique described in Section III, we do the encoding in such a way that the maximum length is

$$L_{\text{max}} = \max_{1 \leq m \leq M} L(\phi(f(m))) \leq \frac{k + d + 2}{2} N. \quad (1)$$

From (1), it follows that the rate of transmission

$$R_t = \frac{\log_2 M}{L_{\text{max}}} \geq \frac{2}{d + k + 2} \log_2 \frac{M}{N}. \quad (2)$$

It is shown in Section III that the method of length control used implies that the number of messages $M = (q^d + 1)/2$ for $q$ odd and $M = q^{d/2}$ when $q$ is even. Thus,

$$R_t \geq \frac{2}{d + k + 2} \left( \frac{1}{N} + \log_2 (k - d + 1) \right) \quad \text{for } N \to \infty \text{ and } \frac{2 \log (k - d + 1)}{d + k + 2} = R(d, k). \quad (3)$$

The codes correcting peak-shifts have smaller values of $M$ than the maximum number, but for large $N$ the rate $R_t$ approaches $R(d, k)$ as well. The rate $R(d, k)$ is less than the capacity $C(d, k)$ of the $(d, k)$-constrained channel. For several values of $d$ and $k$ Table I gives the capacity $C(d, k)$, the potential coding gain $(d + 1)C(d, k)$, as well as the rate $R(d, k)$ and the coding gain $(d + 1)R(d, k)$ achievable by the proposed coding scheme. We may also note that the well-known modified frequency modulation (MFM) coding scheme has the rate $R = 0.5$ for $d = 1$ and $k = 3$. 

<table>
<thead>
<tr>
<th>$d$</th>
<th>$k$</th>
<th>$C(d, k)$</th>
<th>$R(d, k)$</th>
<th>$(d + 1)C(d, k)$</th>
<th>$(d + 1)R(d, k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.5515</td>
<td>0.5263</td>
<td>1.103</td>
<td>1.057</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.4979</td>
<td>0.4664</td>
<td>1.194</td>
<td>1.139</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>0.4057</td>
<td>0.3870</td>
<td>1.622</td>
<td>1.548</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>0.3746</td>
<td>0.3509</td>
<td>1.873</td>
<td>1.755</td>
</tr>
</tbody>
</table>
III. CONTROL OF THE MAXIMUM LENGTH

Since the length of the binary codeword \( \phi(x) \) depend on the \( q \)-ary codeword \( g \), we may expect that for some messages the length \( L(\phi(g)) \) approaches \( (k + 1)N \). This decreases the rate of the transmission \( R_t \) defined by (2). The following method is used in this correspondence to upper bound \( L_{\text{max}} \) as given by (1).

Let \( F \) be a one-to-one mapping of \( R \) onto \( R \) defined by the expression

\[
F(x) = x - 1, \quad x \in R = \{0, 1, 2, \ldots, q - 1\}. \tag{4}
\]

Let \( R^- = \{0, 1, \ldots, (q - 2)/2\}, R^+ = \{(q/2), \ldots, q - 1\} \) for \( q \) even, and \( R^- = \{0, 1, \ldots, (q - 3)/2\}, R^+ = \{(q + 1)/2, \ldots, q - 1\} \) for \( q \) odd. Later we will use the following two properties of the mapping \( F \).

\[
F(x) = x, \quad x \in R. \tag{5}
\]

\[
F(x) \notin \{R^+, R^-\}, \quad x \in R. \tag{6}
\]

We define the mapping \( \varphi \) as

\[
\varphi(x) = 0^{d + 1}(1, q), \quad x \in R. \tag{7}
\]

From the definitions of the mappings \( \varphi \) and \( F \), it follows that for all \( x \in R \)

\[
L(\phi(g)) + L(\phi(F(x))) = d + k + 2. \tag{8}
\]

Let \( C \) be a \( q \)-ary linear code over \( R \) of length \( N \) with \( K \) information symbols. The code is supposed to contain the all 1's word \( \bar{I} \), where \( K \) is the number of ones and are represented respectively as \( (0^{d + 1}0^{q/2} \cdots 0^{q-1}1) \) and \( \alpha_i \) and \( \beta_i \) are positive integers. Therefore, we formally define a single peak-shift of an integer value \( t \) as a distortion such that either

- \( \beta_i = \alpha_i + t, \beta_{i+1} = \alpha_{i+1} - t, \beta_i = \alpha_i \) for \( 1 \leq i \leq j - 1 \) and \( j + 2 \leq i \leq N \), where \( j \leq N - 1 \);
- \( \beta_N = \alpha_N + t, \) and \( \beta_i = \alpha_i \) for \( 1 \leq i \leq N - 1 \).

In fact, the value of \( t \) depends on the position \( j \) and is not greater than \( \alpha_{j+1} \) for \( t \geq 0 \), and is not less than \( -\alpha_j \) for \( t \leq 0 \). Hence, \( \beta_i \geq 0 \) for all \( 1 \leq i \leq N \).

**Proposition 2:** If at most a single peak-shift of value \( t \) occurs, then the output codeword of the encoder \( g \in C \) and the input word \( \bar{x} \) of the decoder are related by

\[
\bar{x} = \bar{x} + g, \tag{10}
\]

where \( + \) is the componentwise addition of integers, and \( g = (e_1, e_2, \ldots, e_N) \) is an error vector with integer components \( e_i \), that belongs to one of the following three classes:

1. \( e_i = 0, \) for \( 1 \leq i \leq N \) (no errors), \( \tag{11} \)
2. \( e_i = 0, \) for \( 1 \leq i \leq N - 1 \) and \( e_N \neq 0 \), \( \tag{12} \)
3. \( e_i = t, e_{j+1} = -t \) for some \( 1 \leq j \leq N - 1 \), \( \tag{13} \)

Therefore, we have related the problem of peak-shift correction to the construction of block codes over the ring of integers modulo \( q \) correcting double errors of the type (13) and a single error in the last component of the codeword. In fact, such double errors are special bursts of length 2. We should note that the deviation \( t \) of the shifted \( j \)-th bit in the \((d, k)\)-sequence determines the absolute value \( |e_j| = |e_{j+1}| \) of the burst, but not its length, which is always equal to 2 for \( j \geq N - 1 \), and 1 for \( j = N \). A class of codes for the correction of a single peak-shift of length \( t \leq (k - d)/2 \) is presented in Section V.

V. LINEAR CODES FOR PEAK-SHIFT CORRECTION

Let \( N = q' \), where \( q = k - d + 1 \geq 5 \) is a prime, and \( r \geq 1 \) is an arbitrary integer. For peak-shift correction we use a \( q \)-ary linear code \( C \) of length \( N \) defined by the parity check matrix \( H = \|h_{ij}\| \) with
two rows of following elements $h_{1, j} \in GF(q) = \{0, 1, \ldots, q - 1\}$ and $h_{2, j} \in GF(q^r)$:

$$h_{1, j} = j \mod q, \quad 1 \leq j \leq N, \quad (14)$$

$$h_{2, j+1} = h_{2, j} + w_j, \quad 1 \leq j \leq N - 1, \quad (15)$$

where $w_1, w_2, \ldots, w_{N-1}$ are distinct nonzero transposed $r$-tuples with components from $GF(q)$, $h_{2, 1}$ is the transposed $r$-tuple $\bar{1} = (1, 0, \ldots, 0)$, and $+$ in (15) represents componentwise modulo $q$ addition of $r$-tuples. It follows from (15) that

$$h_{2, j} = h_{2, 1} + \sum_{i=1}^{j-1} w_i, \quad 2 \leq j \leq N. \quad (16)$$

Since $\sum_{i=1}^{N-1} w_i = \bar{0}$, where $\bar{0}$ is the transposed all-zeros $r$-tuple $(0, 0, \ldots, 0)$, we can see from (16) that the last element $h_{2, N}$ in the second row of $H$ is equal to the first element $h_{2, 1} = \bar{1}$.

Transparency: As was pointed out earlier in Section III, for the maximum length control the code $C$ must be transparent, that is, the all-ones word $\bar{j} = (1, 1, \ldots, 1)$ of length $N$ must belong to the code $C$. In order to satisfy this condition, the elements $w_i$ are chosen in such a way that

$$\sum_{i=1}^{N-1} i w_i = \sum_{i=1}^{N-1} (i \mod q) w_i = \bar{0}, \quad (17)$$

where $iw_i$ represents the summation of $i$ $r$-tuples $w_i$. Using (16), (17), and the equality $Nh_{2, 1} = q'h_{2, 1} = \bar{0}$, we can verify that our code $C$ is transparent, that is $H^t \bar{e} = \bar{0}$. In fact, we have

$$\sum_{i=1}^{N} h_{1, i} = 0, \quad \sum_{i=1}^{N} h_{2, i} = \sum_{i=1}^{N-1} iw_{N-i} + Nh_{2, 1}$$

$$= -\sum_{i=1}^{N-1} (N - i) w_{N-i} = -\sum_{i=1}^{N-1} iw_i = \bar{0}.$$  

Condition (17) can be satisfied in several ways. For example, for any prime $q \geq 3$ and $r \geq 1$ (except the case $q = 3$ and $r = 1$), as an element $w_i$ we may use the ordinary $q$-ary representation of its index $i = 1, 2, \ldots, q - 1$ considered as an integer. In this case the matrix $W = [w_1, w_2, \ldots, w_{N-1}]$ has the following structure

$$W = \begin{bmatrix} 12 \cdots q - 1 & 01 \cdots q - 1 & \cdots & 01 \cdots q - 1 \\ 00 \cdots 0 & 11 \cdots 1 & \cdots & q - 1 \cdots q - 1 \\ 00 \cdots 0 & 00 \cdots 0 & \cdots & q - 1 \cdots q - 1 \end{bmatrix}$$

Let $e = (c_1, c_2, \ldots, c_{N-1})$ be the first row of this matrix $W$. Since $c_i = i \mod q$, $1 \leq i \leq N - 1$, condition (17) is equivalent to $W^t e^t = \bar{0}$. In other words, the first row of the matrix $W$ must be orthogonal to itself and to the other rows of $W$. Since $\sum_{i=1}^{N-1} i^2 = 0 \mod q$ for any $q$, and $\sum_{i=1}^{N-1} i^2 = 0 \mod q$ for any prime $q > 3$, it is a simple matter to verify that $W^t e^t = \bar{0}$ for the matrix $W$ defined by (18). If $q = 3$ and $r \geq 2$, then the product $e \times e^t = q^3 (1^2 + 2^2) = 0 \mod 3$, $a = r - 1 \geq 1$.

**Examples:** The parity check matrices $H_3$ for $N = 9$ ($q = 3$, $r = 2$) and $H_7$ for $N = 7$ ($q = 7$, $r = 1$) constructed according to the described procedure are given below.

$$H_3 = \begin{bmatrix} 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 1 & 2 & 1 & 1 & 2 & 1 & 2 & 1 & 2 \\ 0 & 0 & 1 & 2 & 0 & 2 & 1 & 0 \end{bmatrix}$$

$$H_7 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 1 & 2 & 4 & 0 & 4 & 2 & 1 \end{bmatrix}$$

**TABLE II**

<table>
<thead>
<tr>
<th>Syndrome Values for Different Types of Error Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error type:</td>
</tr>
<tr>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_2$</td>
</tr>
</tbody>
</table>

**TABLE III**

<table>
<thead>
<tr>
<th>Codeword Length and Number of Information Symbols for $q = 3, 5, 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = r + 1$</td>
</tr>
<tr>
<td>$N = q^2$</td>
</tr>
<tr>
<td>$K = N - m$</td>
</tr>
</tbody>
</table>

**Error Correction:** Any error vector $e = (e_1, e_2, \ldots, e_N)$ of the type $(11)-(13)$ can be represented by the position $j$ of its first nonzero component and the type $E = (E_1, E_2)$, where $E_1 = e_j$ and $E_2 = e_{j+1}$. The pair $(j, E)$ is an another representation of the error vector. Proposition 2 partitions all error vectors created by single peak-shifts into three groups $(11)-(13)$ according to the number of errors in the codeword. In fact, the third group $(13)$ is partitioned into $q - 1$ subgroups, so that within one subgroup all-error vectors have the same type $E$ and they differ only by the position $j$ of the first nonzero component. The process of error correction consists in recovering the pair $(j, E)$ on the basis of the syndrome

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = H^t e^t = H e^t,$$

where $S_1 = \sum_{i=1}^{N} h_{1, i} e_i$ and $S_2 = \sum_{i=1}^{N} h_{2, i} e_i$.

For the parity check matrix defined by (14) and (15) the values of $S_1$ and $S_2$ For different types $(11)-(13)$ of error vectors are given in Table II, which is very simple, and in fact is already sufficient for correction of all errors created by a single peak-shift.

We may distinguish the following decoding steps.

1) Calculate the syndrome $S$ by multiplication of the output word $\bar{z}$ of the $2/q$ converter with the parity check matrix $H$. If $S_1 = 0$ and $S_2 = 0$, then let $e_i = 0$ for $1 \leq i \leq N$. If $S_1 = 0$, but $S_2 \neq 0$, then let $e_i = 0$ for $1 \leq i \leq N - 1$ and $e_N = t$, where $r$ is the first nonzero component of $S_2$. If $S_1 \neq 0$ and $S_2 \neq 0$, then first let $t = -S_1$ and find the position $j$ of the first error according to its $q$-ary representation $w_j = (-S_2)^{q-1}$. After that let $e_i = t, e_{i+1} = -t$ and $e_i = 0$ for all other $i$.

2) Make corrections in $\bar{z}$ by forming the word $\bar{z}' = \bar{z} - c$.

3) If the first information symbol of $\bar{z}'$ (used to distinguish between $Z$ and $F(\bar{z})$) belongs to $R^{+}$ then send the information symbols of $\bar{z}'$ to the output of the decoder. If it belongs to $R^{-}$ then first modify them by the mapping $F$.

Finally, let us define the number of $q$-ary information symbols $K$ in codewords $\bar{z}$ of the code $C$. Recalling the definition of the linear code $C$, we should only note that its parity check matrix $H$ has $r + 1$ rows and $N = q^r$ - $q$-ary elements. If the rows are linearly independent, then $K = N - r - 1$, otherwise $K = N - r - 1$. For some values of $q$ and redundancy $m = r + 1$, the values of $N$ and $K$ are given in Table III.

We close this Section by summarizing the main properties of the proposed codes in the following proposition.

**Proposition 3:** For any prime $q \geq 3$, the $q$-ary linear transparent code $C$ of word length $N = q^r \geq 4$, $r \geq 1$, defined by the parity check matrix $H$ as given by (14), (15), and (18), has not less than
$K = q^r - r - 1$ $q$-ary information symbols and corrects all errors vectors of the type \((11)-(13)\).

VI. LINEAR CODES CORRECTING SINGLE PEAK-SHIFTS, DELETIONS, AND INSERTIONS OF ZEROS

The proposed method can be used for the correction of other types of errors. In this section, we present codes that can correct in the \((d, k)\)-sequence a single distortion of the following type:

1) a peak-shift on \((k - d)/2\) or less positions,
2) a deletion of \((k - d)/2\) or less zeros between adjacent one’s,
3) an insertion of \((k - d)/2\) or less zeros between adjacent one’s.

In this case the list of possible types of error vectors \((11)-(13)\) is extended with the following:

\[ e_j = t, \quad \text{for some } 1 \leq j \leq N, \]

\[ e_i = 0, \quad \text{for } 1 \leq i \neq j \leq N. \]  

(19)

In fact, errors of type (12) are particular cases of (19), and for this reason we later consider only three types \((11), (13)\) and (19), Proposition 2 modified in this way will be referred to as Proposition 4.

Let \(N = q^r\), where \(q = k - d + 1 \geq 3\) is a prime, \(r \geq 2\) is an arbitrary integer, and let \(\gamma\) be a primitive element of \(GF(q^r)\) such that the element \(\gamma^{r-1}(1 - \gamma)\) is not an integer in \(GF(q^r)\). For all values of \(q\) and \(r\) such that \(q^r \leq 128\), Tables from [14, ch. 10] can be used to select a primitive element \(\gamma\) that satisfies this condition.

For the correction of errors \((11)-(13)\) and (19) we use a \(q\)-ary linear code \(C\) defined by the parity check matrix \(H\) given in Table IV.

\[ H = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 3 & 4 & \ldots & q-1 & q-1 & 0 & 1 & \ldots & q-1 \\ 0 & \gamma^1 & \gamma^2 & \gamma^3 & \ldots & \gamma^{r-2} & \gamma^{r-1} & \gamma & \gamma^{r+1} & \ldots & \gamma^{N-1} \end{bmatrix} \]  

(20)

with three rows of following elements \(h_{1,j}, h_{2,j} \in GF(q)\) and \(h_{3,j} \in GF(q^r)\):

\[ h_{1,j} = 1, \quad \text{for } 1 \leq j \leq N, \]

\[ h_{2,j} = 2, h_{2,j} = j, \quad \text{for } 2 \leq j \leq q-1, \]

\[ h_{2,j} = q-1, h_{2,j} = j-1 \text{ mod } q, \quad \text{for } q < j \leq N, \]

\[ h_{3,j} = 0, h_{3,j} = \gamma^{r-1}, \quad \text{for } 2 \leq j \leq N. \]  

(21)

The code defined by this parity check matrix is transparent. This follows from the definitions and the fact that the summation of all elements in \(GF(q^r)\) gives 0 for any \(r \geq 1\). As an example for \(q = 3\) and \(r = 2\), the parity check matrix is shown next (\(\gamma = (1, 0)^{t}r\)), and \(GF(2^3)\) is represented as in tables in [14]:

\[ H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 2 & 0 & 2 & 1 & 1 & 0 \end{bmatrix} \]

Error correction is based on the syndrome \(S = (S_1, S_2, S_3)^T = R_z^*\) with the following components:

\[ S_1 = \sum_{j=1}^{N} z_j \text{ mod } q, \]

\[ S_2 = \sum_{j=2}^{N} \gamma^{j-1} z_j \in GF(q^r), \]

\[ S_3 = (2z_1 + (q - 1)z_2 + \sum_{j=2}^{q-1} jz_j + \sum_{j=q+1}^{N} (j-1)z_j) \text{ mod } q. \]

For different types of error vectors the values of \(S_1, S_2, S_3\) are given in Table IV, where \(\square\) denotes the values that are not important for the decoding.
concatenation of strings 01 gives the following important advantage: peak-shifts, insertions, and deletions of zeros in the binary (d, k)-sequence lead to q-ary errors of the type (13) and (19) which are convenient for correction. Block codes with integer components correcting errors of the type (13) and (19) were introduced in this correspondence as a new mathematical object which is important for the correction of peak-shifts, insertions, and deletions of zeros in (d, k)-constrained sequences. Two classes of such codes were constructed, and their parameters are given in Propositions 3, 5, and Table III.

5) A conventional scheme based on the RS-encoder followed by the (d, k)-modulator, is used here for comparisons [10]. When such a system is used for correction of single peak-shifts, an RS-code over GF(2^w), where w usually equals 4, 5, 6, ... corrects a single 2^w-ary error (interleaving of the proper depth λ is applied, if error propagation on the output of the demodulator starts to create double q-ary errors). Therefore, in the conventional scheme a single peak-shift is corrected in the (d, k)-sequence of the length n = λw(2^w - 1)/R_mod that has 2λw/R_mod check bits. For the proposed scheme a single peak-shift is corrected in the binary (d, k)-sequence of length n = N(d + k + 2)/2 (n = d + 1, if dummy bits are used to fix the binary length). Since the q-ary block codes produce different discrete values of binary length n, all comparisons are based on the code rate R = 1/n, where n is the number of information bits encoded in the (d, k)-sequence of length n. For the conventional scheme I = λw(2^w - 3), and for the proposed scheme n = [(N - r - 1) log2 q] + 1, where r = log2 N and [x] is the integer part of x. For a given modulation code the length of error propagation b on the output of its demodulator determines the depth of the interleaving λ. Since a single peak-shift could lead to a burst of b errors at the output of the demodulator, λ must satisfy the inequality b ≤ ν(λ - 1) + 1. In our comparisons, we used the minimal value of λ satisfying this condition.

For the (d = 1, k = 3) and (d = 3, k = 7) constraints some of the values are given in Table V. In the first case, standard MFM is used in the conventional scheme (R_mod = 0.5, b = 1, λ = 1). In the second case modulation codes with the following parameters are considered:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>R</th>
<th>n</th>
<th>d = 1, k = 3, q = 1, λ</th>
<th>d = 3, k = 7, q = 1, λ</th>
<th>R</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>81</td>
<td>0.4531</td>
<td>RS(2^8), MC1</td>
<td>480</td>
<td>0.3250</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>120</td>
<td>0.4533</td>
<td>RS(2^8), MC3</td>
<td>600</td>
<td>0.3466</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>243</td>
<td>0.4897</td>
<td>Proposed, N = 5^7</td>
<td>750</td>
<td>0.3720</td>
<td>4</td>
</tr>
<tr>
<td>Proposed</td>
<td>310</td>
<td>0.4754</td>
<td>RS(2^6), MC1</td>
<td>1240</td>
<td>0.3508</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>729</td>
<td>0.5130</td>
<td>RS(2^6), MC2</td>
<td>1612</td>
<td>0.3598</td>
<td>4</td>
</tr>
<tr>
<td>Proposed</td>
<td>756</td>
<td>0.4841</td>
<td>RS(2^9), MC3</td>
<td>1163</td>
<td>0.3242</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>2187</td>
<td>0.5226</td>
<td>Proposed, N = 5^4</td>
<td>3750</td>
<td>0.3835</td>
<td>4</td>
</tr>
<tr>
<td>Proposed</td>
<td>1778</td>
<td>0.4921</td>
<td>RS(2^9), MC1</td>
<td>3024</td>
<td>0.3631</td>
<td>3</td>
</tr>
<tr>
<td>Proposed</td>
<td>3932</td>
<td>0.5226</td>
<td>RS(2^9), MC2</td>
<td>3932</td>
<td>0.3724</td>
<td>4</td>
</tr>
<tr>
<td>Proposed</td>
<td>2873</td>
<td>0.3873</td>
<td>RS(2^9), MC3</td>
<td>2873</td>
<td>0.3873</td>
<td>3</td>
</tr>
</tbody>
</table>

7) In the proposed scheme, we used fixed-to-variable encoding with maximum length control. In a concatenation of codewords, a shift of the last bit of a codeword also shifts the boundary between codewords. These shifts are corrected and, consequently, the beginning of a codeword is defined correctly.

8) If due to uncorrectable errors, a decoding failure gives rise to a decoding error in the last symbol, the beginning of the next code word is determined erroneously. The error is equivalent to an insertion or a deletion of zeros at the beginning of the word. For the code correcting peak-shifts, such t ≥ 0 insertions (deletions) give a syndrome with S_t = (t - i), S_i = (i - 1). In the absence of other errors in the codeword, this particular syndrome equals that of a peak-shift in one of the first (q - 2) positions and thus does not lead again to a decoding error in the last symbol. Error propagation is limited to one codeword.

9) If we need fixed-to-fixed encoding we extend the codewords to their maximum length L_max with "dummy" strings of the type 01, d ≤ i ≤ k. For the practical values of d ≤ k/2 we are then always able to extend the code word length to L_max + d binary digits without violating the (d, k)-constraint. The 2/q-converter uses only the first N strings 01 of the received sequence and ignores the others. For large values of N, the addition of d + 1 binary digits does not affect the rate of the encoding scheme.

10) For future research, we suggest generalizing Proposition 2 for the case of multiple peak-shifts. Block codes over the ring of integers, correcting multiple bursts of the type (13), can be used for correction of multiple peak-shifts. The number of bursts is equal to the number of shifted 1's in the (d, k)-sequence. A method for the construction of codes correcting multiple bursts of the type (11)-(13) is described in [12]. It is based on ideas used by V. Levenshtein in 1965 for the construction of block codes correcting deletions and insertions [13]. We may also expect that good codes can be constructed for the correction of deleted, inserted, or inverted digits of the binary (d, k)-sequences.

11) As we already mentioned in the Introduction, the presented codes, as well as the others known at the moment, are not a "medicine" for all kind of distortions. They are constructed to correct mainly single peak-shifts for (k - d)/2 or less positions left or right only. However, the proposed coding scheme is more general and can be extended to correct other types of errors as well. Codes for correction of peak-shifts, deletions, and insertions of zeros were presented as well. We avoided generalizations in order to clarify the basic ideas and to make the correspondence as simple as possible.

ACKNOWLEDGMENT

Both authors appreciate the stimulating discussions they had with Dr. Schouhamer Immink from Philips Research in Eindhoven, The Netherlands.
A Nonalgorithmic Maximum Likelihood Decoder for Trellis Codes

Robert C. Davis and Hans-Andrea Loeliger, Member, IEEE

Abstract—A new decoder for trellis codes is presented which is based on a graph model consisting of diodes and switches. Such decoders may prove to be well suited for VLSI.

Index Terms—Trellis codes, decoding, diodes, graph models, networks.

I. INTRODUCTION

This correspondence presents a decoding method for trellis codes or, more generally, a method for finding the shortest path through a directed graph, that is very different from all decoding methods that have so far been used or described in the literature. This decoding method was invented by one of the authors (Davis) in 1983, but it was published only in the form of a patent [1], where, moreover, the basic idea is contained only in the special form of a decision-feedback decoder for convolutional codes. Recently, this idea was independently rediscovered by the other author (Loeliger), which was the incentive to write this correspondence.

The decoder of this correspondence is nonalgorithmic in the sense that, in contrast to all known decoding methods, it does not have a natural description as a sequence of simple steps suitable for execution on a general-purpose digital computer. Since the decoder is essentially an electrical network, it can, of course, be simulated with arbitrary accuracy on such a computer, but such a simulation is unattractive compared to conventional decoding procedures.

We will consider the case where the code trellis is finite, such as the block code in Fig. 1, in contrast to the case where the trellis is repetitively infinite as are, e.g., the trellises of nontruncated convolutional codes. (Trellises of truncated convolutional codes have been known for a long time [2]. The study of trellises for other block codes was begun in [3]; see also [4] and the references therein.) There are, however, well known techniques (e.g., [5] and [6]) for decoding a repetitively infinite trellis code by repeated decoding of finite trellis codes.

Maximum likelihood decoding of a trellis code used with a memoryless channel is well known to be equivalent to finding the shortest path through the trellis when the "length" of a path equals the sum of the weights of its branches and the weight of a branch is proportional to the log-likelihood ratio log \( p_1(r_i | c(b)) \), where the index \( i \) refers to the \( i \)-th step of the trellis, \( c(b) \) is the sequence of channel input symbols with which the branch \( b \) is labeled, \( r_i \) is the corresponding segment of the received data, and the conditional probability \( p_i(\cdot | \cdot) \) is determined by the channel. By adding a suitable constant to all branch weights one can make all branch weights nonnegative. Full accounts of this topic may be found in the literature, e.g., [2], [7], [8].

Once decoding of a trellis code is seen to be a shortest path problem, it is only a small step to the Viterbi algorithm [2], [7]-[9], today's standard decoding procedure, which is a dynamic programming solution [10] to the shortest path problem.

A completely different solution to the shortest path problem was given by Minty [11], [2]: A scale model of the graph is built in which the branches are flexible strings. The shortest path between any two nodes of the model will stretch tight when the two nodes are pulled apart. (Note, however, that Minty's method is confined to undirected graphs. Its application to trellis decoding may produce nonsensical solutions in the form of shortest paths that use a branch in the wrong direction.)

Apart from Forney's discussion of Minty's string model [2] (commented by "...Unfortunately, not well adapted to modern methods of machine computation"), there seems to be no mention of nonalgorithmic decoders (i.e., without a natural description as a program for a general-purpose digital computer) in the literature. The main purpose of this correspondence is to demonstrate by an example that such decoders exist and may prove to be practical.

Like Minty's string model, the decoder of this paper consists essentially of a physical graph model, but the model is electronic rather than mechanical. It is based on the current vs. voltage behavior of semiconductor diodes as shown in Fig. 2, which is characterized by a turn-on voltage \( V_T \) below which there flows hardly any current and above which there flows very much current. The basic operating principle of the new decoder is best understood by the simplified

REFERENCES


IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 4, JULY 1993

A Nonalgorithmic Maximum Likelihood Decoder for Trellis Codes

Robert C. Davis and Hans-Andrea Loeliger, Member, IEEE

Abstract—A new decoder for trellis codes is presented which is based on a graph model consisting of diodes and switches. Such decoders may prove to be well suited for VLSI.

Index Terms—Trellis codes, decoding, diodes, graph models, networks.

I. INTRODUCTION

This correspondence presents a decoding method for trellis codes or, more generally, a method for finding the shortest path through a directed graph, that is very different from all decoding methods that have so far been used or described in the literature. This decoding method was invented by one of the authors (Davis) in 1983, but it was published only in the form of a patent [1], where, moreover, the basic idea is contained only in the special form of a decision-feedback decoder for convolutional codes. Recently, this idea was independently rediscovered by the other author (Loeliger), which was the incentive to write this correspondence.

The decoder of this correspondence is nonalgorithmic in the sense that, in contrast to all known decoding methods, it does not have a natural description as a sequence of simple steps suitable for execution on a general-purpose digital computer. Since the decoder is essentially an electrical network, it can, of course, be simulated with arbitrary accuracy on such a computer, but such a simulation is unattractive compared to conventional decoding procedures.

We will consider the case where the code trellis is finite, such as the block code in Fig. 1, in contrast to the case where the trellis is repetitively infinite as are, e.g., the trellises of nontruncated convolutional codes. (Trellises of truncated convolutional codes have been known for a long time [2]. The study of trellises for other block codes was begun in [3]; see also [4] and the references therein.) There are, however, well known techniques (e.g., [5] and [6]) for decoding a repetitively infinite trellis code by repeated decoding of finite trellis codes.

Maximum likelihood decoding of a trellis code used with a memoryless channel is well known to be equivalent to finding the shortest path through the trellis when the "length" of a path equals the sum of the weights of its branches and the weight of a branch is proportional to the log-likelihood ratio log \( p_1(r_i | c(b)) \), where the index \( i \) refers to the \( i \)-th step of the trellis, \( c(b) \) is the sequence of channel input symbols with which the branch \( b \) is labeled, \( r_i \) is the corresponding segment of the received data, and the conditional probability \( p_i(\cdot | \cdot) \) is determined by the channel. By adding a suitable constant to all branch weights one can make all branch weights nonnegative. Full accounts of this topic may be found in the literature, e.g., [2], [7], [8].

Once decoding of a trellis code is seen to be a shortest path problem, it is only a small step to the Viterbi algorithm [2], [7]-[9], today's standard decoding procedure, which is a dynamic programming solution [10] to the shortest path problem.

A completely different solution to the shortest path problem was given by Minty [11], [2]: A scale model of the graph is built in which the branches are flexible strings. The shortest path between any two nodes of the model will stretch tight when the two nodes are pulled apart. (Note, however, that Minty's method is confined to undirected graphs. Its application to trellis decoding may produce nonsensical solutions in the form of shortest paths that use a branch in the wrong direction.)

Apart from Forney's discussion of Minty's string model [2] (commented by "...Unfortunately, not well adapted to modern methods of machine computation"), there seems to be no mention of nonalgorithmic decoders (i.e., without a natural description as a program for a general-purpose digital computer) in the literature. The main purpose of this correspondence is to demonstrate by an example that such decoders exist and may prove to be practical.

Like Minty's string model, the decoder of this paper consists essentially of a physical graph model, but the model is electronic rather than mechanical. It is based on the current vs. voltage behavior of semiconductor diodes as shown in Fig. 2, which is characterized by a turn-on voltage \( V_T \) below which there flows hardly any current and above which there flows very much current. The basic operating principle of the new decoder is best understood by the simplified