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

REFERENCES

[1] J. P. Adoul and M. Barth, "Nearest neighbor algorithm for spherical codes from the Leech lattice," *IEEE Trans. Inform. Theory*, vol. 34, pt. II, pp. 1188–1202, Sept. 1988.

[2] Y. Be'ery and B. Shahar, "VLSI Architectures for soft decoding of the Leech lattice and the Golay codes," *Proc. IEEE Int. Workshop on Microelectronics in Commun.*, Interlaken, Switzerland, Mar. 1991.

[3] Y. Be'ery, B. Shahar, and J. Snyders "Fast decoding of the Leech lattice," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 959–967, 1989.

[4] J. W. M. Bergmans, S. A. Rajput, and F. A. M. Van de Lear, "Co-designed coding, modulation and equalization," preprint.

[5] A. R. Calderbank and N. J. A. Sloane, "New trellis codes based on lattices and cosets," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 177–195, Mar. 1987.

[6] J. H. Conway, "Hexacode and tetracode—MOG and MINIMOG," in New York: Academic Press, *Computational Group Theory*, 1984, pp. 359–365.

[7] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 211–226, 1982.

[8] —— "On the Voronoi regions of certain lattices", *SIAM J. Algebraic Discrete Methods*, vol. 5, pp. 294–305, 1984.

[9] —— "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 41–50, Jan. 1986.

[10] —— *Sphere Packings, Lattices and Groups.* New York: Springer-Verlag, 1988.

[11] R. T. Curtis, "A new combinatorial approach to M_{24}," *Math Proc. Camb. Phil. Soc.*, vol. 79, pp. 25–41, 1976.

[12] R. de Buda, "Some optimal codes have structure," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 959–967, 1989.

[13] G. D. Forney, Jr., "Coset codes I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, Pt. II, pp. 1123–1151, Sept. 1988.

[14] —— "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pt. II pp. 1152–1187, Sept. 1988.

[15] —— "A bounded distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. Inform. Theory*, vol. 35, pp. 906–909, July 1989.

[16] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 632–647, 1984.

[17] D. E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching.* Reading, MA: Addison-Wesley, 1973.

[18] G. R. Lang and F. M. Longstaff, "A Leech lattice modem," *IEEE J. Select Areas Commun.*, vol. 7, pp. 968–973, 1989.

[19] J. Leech, "Notes on sphere packing," *Can. J. Math.*, vol. 19, pp. 251–267, 1967.

[20] R. E. Peile, Final report on INTEL-963: "Co-designed coding, modulation and equilization," preprint.

[21] V. S. Pless, "Decoding the Golay codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 561–567, July 1986.

[22] R. A. Silverman and M. Balser, "Coding for a constant data rate source," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 50–63, 1954.

[23] N. J. A. Sloane, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 327–338, 1981.

[24] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 963–975, Sept. 1989.

[25] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[26] A. Vardy and Y. Be'ery, "More efficient soft-decision decoding of the Golay codes," *IEEE Trans. Inform. Theory*, vol. 37, pt. I, pp. 667–672, May 1991.

[27] E. F. Assmus, Jr. and H. F. Mattson, Jr., "Algebraic theory of codes II," *Scientific Rep. No. 1*, Contract No. F19628-69-C-0068, Air Force Cambridge Research Labs., 1969.

[28] E. P. Shaughnessy, "Codes with simple automorphism groups," *Arch. Math. (Basel)*, vol. 22, pp. 459–466, 1971.

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 ($q=k-d+1$ is a prime) 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 $00\cdots01$. 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 R_t of the overall encoding approaches $(2\log_2(k-d+1))/(k+d+2)$ 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 denumeration procedures and, therefore, the codelength 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 "1'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 $n = 1/\delta$, 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 2^N, $N \approx (d+1)C(d)n$, where $C(d)$ is the Shannon capacity of the channel with d-constrained input, then the ratio $\gamma = N/n$ is equal to $(d+1)C(d)$ and grows with d [1]. It characterizes an improvement in the linear data 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 ($\geq d+1$). 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 wa 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. 19, Moscow 101447, Russia.

A. J. Han Vinck is with the University of Essen, Institute for Experimental Mathematics, Ellernstr. 29, 4300 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 codelength 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 codelength. 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 $(2\log_2(d-k+1))/(k+d+2)$ for large codeword lengths.

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 $m \in \{1, 2, \cdots, M\}$ the block encoder forms a codeword $\underline{x} = f(m) = (x_1, x_2, \cdots, x_N)$ of length N with components $x_i \in R$, $1 \leq i \leq N$, 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 $R = \{0, 1, 2, \cdots, q-1\}$ 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 φ from R to the set $\{0^{d+i}1,\ 0 \leq i \leq q-1\}$ of $q = k - d + 1 = |R|$ binary strings $0^{d+i}1$, the ith of which is a sequence 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 $\underline{x} = f(m) = (x_1, x_2, \cdots, x_N)$ the $q/2$-converter generates as an output the binary sequence $\phi(\underline{x}) = (\varphi(x_1), \varphi(x_2), \cdots, \varphi(x_N))$ which is a concatenation of strings $\varphi(x_i)$, $1 \leq i \leq N$. Obviously, all binary sequences $\phi(\underline{x})$ satisfy the (d, k)-constraint. The total number of binary components of $\phi(\underline{x})$, or the length $L(\phi(\underline{x}))$, depends on $\underline{x}$, 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

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

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

$$R_t = \frac{\log_2 M}{L_{\max}} \geq \frac{2}{d+k+2} \cdot \frac{\log_2 M}{N}. \tag{2}$$

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

$$R_t \geq \frac{2}{d+k+2}\left(-\frac{1}{N} + \log_2(k-d+1)\right) \underset{N \to \infty}{\longrightarrow} \frac{2\log(k-d+1)}{d+k+2} = R(d, k). \tag{3}$$

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 $k = 3$.

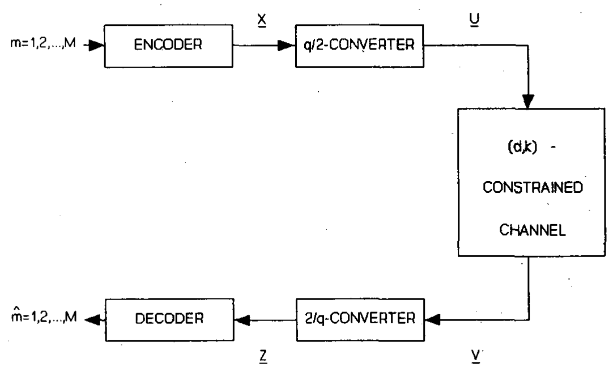

Fig. 1. Configuration of the encoding scheme.

TABLE I
Capacities, Rates, and Coding Gain for (d, k)-Constraint Channel

d	k	$C(d, k)$	$R(d, k)$	$(d+1)C(d, k)$	$(d+1)R(d, k)$
1	3	0.5515	0.5283	1.103	1.057
2	4	0.4057	0.3962	1.217	1.189
2	6	0.4979	0.4644	1.494	1.393
3	7	0.4057	0.3870	1.622	1.548
4	10	0.3746	0.3509	1.873	1.755

III. Control of the Maximum Length

Since the length of the binary codeword $\phi(\underline{x})$ depend on the q-ary codeword $\underline{x}$, we may expect that for some messages the length $L(\phi(\underline{x}))$ 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 $L_{\max}$ as given by (1).

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

$$F(x) = (q-1) - x, \qquad x \in R = \{0, 1, 2, \cdots, q-1\}. \tag{4}$$

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

$$F(F(x)) = x, \qquad \text{for all } x \in R, \tag{5}$$

$$F(x) \in \begin{cases} R^+, & \text{if } x \in R^- \\ R^-, & \text{if } x \in R^+. \end{cases} \tag{6}$$

We define the mapping φ as

$$\varphi(x) = 0^{d+x}1, \qquad x \in R. \tag{7}$$

From the definitions of the mappings φ and F, it follows that for all $x \in R$

$$L(\varphi(x)) + L(\varphi(F(x))) = d + k + 2. \tag{8}$$

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 $\underline{1} = (1, 1, \cdots, 1)$. Such codes are called transparent. It is easy to see that if $\underline{x} = (x_1, x_2, \cdots, x_N) \in C$ and C is transparent then the word $F(\underline{x}) = (F(x_1), F(x_2), \cdots, F(x_N))$ also belongs to C. It follows from (5) and (6) that 1) $F(F(\underline{x})) = \underline{x}$, and 2) $F(\underline{x}) \neq \underline{x}$ for all $\underline{x}$ if q is even, and for $\underline{x} \neq (q-1/2) \cdot \underline{1}$ if q is odd. Therefore, the transparent code C can be partitioned into M pairs of codewords $(\underline{x}, F(\underline{x}))$, $M = (q^K+1)/2$ for q odd and $M = q^K/2$ for q even, such that

$$L(\phi(\underline{x})) + L(\phi(F(\underline{x}))) = N(d+k+2). \tag{9}$$

Note that at least one of the codewords $\underline{x}$ or $F(\underline{x})$ has a binary representation of length $L \leq N(d+k+2)/2$. If both these codewords represent the same message m and the shorter one is used as $f(m)$, then the following proposition holds.

Proposition 1: If a q-ary linear code C of length N with K information symbols is transparent, then $M = q^K/2$ for q even, and $M = (q^K+1)/2$ for q odd, messages m can be encoded into codewords $\underline{x} = f(m) \in C$, which are converted according to (7) to binary codewords $\phi(\underline{x})$ satisfying the (d, k)-constraint and having maximum length $L_{\max} \leq N(k+d+2)/2$.

Remark: The encoder will create only code vectors $\underline{x}$ by operating on those K-element information blocks that have lead elements in R^-, and the code vector $F(\underline{x})$ is sent only when $L(\underline{x}) > N(d+k+2)/2$. The number of possible messages M will be equal to $q^K/2$ for q even and $q^{K-1}(q-1)/2$ for q odd. However, for large K the rate of transmission will be asymptotically the same. The decoder can easily determine from the first component which word $\underline{x}$ or $F(\underline{x})$ was used for transmission.

IV. Peak-Shifts and q-ary Errors

After transmission of the binary sequence $\phi(\underline{x})$ through the channel (storing on some media), some of its symbols might be received erroneously, deleted, inserted or shifted. We consider the distortion, known as peak-shift, defined for channels with input restrictions such that if at time i the channel input is 1, then at the time moments $i+l$, where $-d \leq l \leq d$ and $l \neq 0$, the channel inputs are 0's $(d \geq 1)$. A peak-shift of value t converts the symbol 1 at time i into a 0 and changes the symbol 0 at time $i+t$ into a symbol 1. If t is negative (positive), we speak about a left (right) shift. Hence, if before transmission the channel input looks like $\cdots 010 \cdots$, then after transmission a peak-shift on the one position gives at the output either $\cdots 100 \cdots$ or $\cdots 001 \cdots$.

Since the received binary sequence $\underline{v}$ has a variable length, for error correction it is convenient to convert $\underline{v}$ to the word $\underline{z} = (z_1, z_2, \cdots, z_N)$ of fixed length N with integer components z_i, $1 \leq i \leq N$. Under the condition that the position of the first digit of $\underline{v}$ has been defined correctly we act as follows.

1) Partition the received sequence $\underline{v}$ into binary strings of the type 0^i1, and represent it in the form $\underline{v} = (0^{\beta_1}10^{\beta_2}1 \cdots 0^{\beta_N}1)$, $\beta_i \geq 0$.
2) For $i = 1, 2, \cdots, N$, replace β_i by $z_i = \beta_i - d$ and form the word $\underline{z} = (z_1, z_2, \cdots, z_N)$, which is regarded as the output word of the $2/q$-converter and the input word for the decoder.

Note that a peak-shift can lead to violation of the (d, k)-constraint in $\underline{v}$, and, therefore, it may happen that $z_i < 0$ or $z_i \geq q = k-d+1$ for some $1 \leq i \leq N$.

In this correspondence, input and output sequences $\underline{u}$ and $\underline{v}$ have the same number N of ones and are represented respectively as $(0^{\alpha_1}1\,0^{\alpha_2}1 \cdots 0^{\alpha_N}1)$ and $(0^{\beta_1}1\,0^{\beta_2}1 \cdots 0^{\beta_N}1)$, where α_i and β_j are positive integers. Therefore, we formally define a single peak-shift of an integer value t as a distortion such that either

1) $\beta_j = \alpha_j + t$, $\beta_{j+1} = \alpha_{j+1} - t$, $\beta_i = \alpha_i$ for $1 \leq i \leq j-1$ and $j+2 \leq i \leq N$, where $j \leq N-1$; or
2) $\beta_N = \alpha_N + t$, 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 α_{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 $\underline{x} \in C$ and the input word $\underline{z}$ of the decoder are related by

$$\underline{z} = \underline{x} + \underline{e}, \tag{10}$$

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

$$\begin{array}{lll} 1) & e_i = 0, & \text{for } 1 \leq i \leq N \text{ (no errors)}, \qquad (11)\\ 2) & e_i = 0, & \text{for } 1 \leq i \leq N-1 \text{ and } e_N \neq 0, \qquad (12)\\ 3) & e_j = t, & e_{j+1} = -t \text{ for some } 1 \leq j \leq N-1,\\ & e_i = 0, & \text{for } i \neq j, j+1. \qquad (13) \end{array}$$

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 jth 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 $j \leq N-1$, 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 $N = q^r$, where $q = k-d+1 \geq 3$ 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 $h_{1,j} \in \mathrm{GF}(q) = \{0, 1, \cdots, q-1\}$ and $h_{2,j} \in \mathrm{GF}(q^r)$:

$$h_{1,j} = j \bmod q, \qquad 1 \le j \le N, \tag{14}$$

$$h_{2,j+1} = h_{2,j} + w_j, \qquad 1 \le j \le N-1, \tag{15}$$

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

$$h_{2,j} = h_{2,1} + \sum_{i=1}^{j-1} w_i, \qquad 2 \le j \le N. \tag{16}$$

Since $\sum_{i=1}^{N-1} w_i = \bar{0}$, where $\bar{0}$ is the transposed all-zeros r-tuple $(0, 0, \cdots, 0)$, 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 $\underline{1} = (1, 1, \cdots, 1)$ 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

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

where iw_i represents the summation of i r-tuples w_i. Using (16), (17), and the equality $Nh_{2,1} = q^r h_{2,1} = \bar{0}$, we can verify that our code C is transparent, that is $H\underline{1}^{\mathrm{tr}} = 0$. In fact, we have

$$\sum_{i=1}^{N} h_{1,i} = 0, \ \sum_{i=1}^{N} h_{2,i} = \sum_{i=1}^{N-1} i w_{N-i} + N h_{2,1}$$
$$= -\sum_{i=1}^{N-1} (N-i) w_{N-i} = -\sum_{i=1}^{N-1} i w_i = \bar{0}.$$

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

$$W = \begin{bmatrix} 12\cdots q-1 & 01\cdots q-1 & \cdots & 0 & 1\cdots q-1 \\ 00\cdots 0 & 11\cdots 1 & \cdots & q-1 & q-1\cdots q-1 \\ \cdots\cdots & \cdots\cdots & \cdots & \cdots & \cdots\cdots \\ 00\cdots 0 & 00\cdots 0 & \cdots & q-1 & q-1\cdots q-1 \end{bmatrix}. \tag{18}$$

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

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.

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

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

TABLE II
SYNDROME VALUES FOR DIFFERENT TYPES OF ERROR VECTORS

Error type:	(11)	(12)	(13)
S_1	0	0	$-t$
S_2	0	$t\bar{1}$	$-tw_i$

TABLE III
CODEWORD LENGTH AND NUMBER OF INFORMATION SYMBOLS FOR $q = 3, 5, 7$

	$q=3$				$q=5$			$q=7$	
$m = r+1$	3	4	5	6	3	4	5	3	4
$N = q^r$	9	27	81	243	25	125	625	49	343
$K = N - m$	6	23	76	237	22	121	620	46	339

Error Correction: Any error vector $\underline{e} = (e_1, e_2, \cdots, e_N)$ 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

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = H\underline{z}^{\mathrm{tr}} = H\underline{e}^{\mathrm{tr}},$$
$$\text{where } S_1 = \sum_{i=1}^{N} h_{1,i} e_i \quad \text{and } S_2 = \sum_{i=1}^{N} h_{2,i} e_i.$$

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 $\underline{z}$ 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 $1 \le i \le N$. If $S_1 = 0$, but $S_2 \ne 0$, then let $e_i = 0$ for $1 \le i \le N-1$ and $e_N = t$, where t is the first nonzero component of S_2. If $S_1 \ne 0$ and $S_2 \ne 0$, then first let $t = -S_1$ And find the position j of the first error according to its q-ary representation $w_j = (-S_2)t^{-1}$. After that let $e_j = t$, $e_{j+1} = -t$ and $e_i = 0$ for all other i.
2) Make corrections in $\underline{z}$ by forming the word $\underline{z}' = \underline{z} - \underline{e}$.
3) If the first information symbol of $\underline{z}'$ (used to distinguish between $\underline{x}$ and $F(\underline{x})$) belongs to R^- then send the information symbols of $\underline{z}'$ 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 $\underline{x}$ 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 $N = q^r$ columns of 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 $q \ge 3$, the q-ary linear transparent code C of word length $N = q^r \ge 4$, $r \ge 1$, 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

$$\begin{aligned} e_j &= t, &&\text{for some } 1 \le j \le N,\\ e_i &= 0, &&\text{for } 1 \le i \ne j \le N. \end{aligned} \tag{19}$$

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

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 & \cdots & 1\\ 2 & 2 & 3 & 4 & \cdots & q-1 & q-1 & 0 & 1 & \cdots & q-1\\ 0 & \gamma^1 & \gamma^2 & \gamma^3 & \cdots & \gamma^{q-2} & \gamma^{q-1} & \gamma^q & \gamma^{q+1} & \cdots & \gamma^{N-1} \end{bmatrix} \tag{20}$$

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

$$\begin{aligned} &h_{1,j} = 1, && \text{for } 1 \le j \le N,\\ &h_{2,1} = 2,\ h_{2,j} = j && \text{for } 2 \le j \le q-1,\\ &h_{2,q} = q-1,\ h_{2,j} = j - 1 \bmod q, && \text{for } q < j \le N,\\ &h_{3,1} = 0,\ h_{3,j} = \gamma^{j-1}, && \text{for } 2 \le j \le N. \end{aligned} \tag{21}$$

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 $r \ge 1$. As an example for $q = 3$ and $r = 2$, the parity check matrix is shown next ($\gamma = (1, 0)^{tr}$, and GF(3^2) is represented as in tables in [14]):

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

Error correction is based on the syndrome $S = (S_1, S_2, S_3)^{tr} = H\underline{z}^{tr}$ with the following components:

$$S_1 = \sum_{j=1}^{N} z_j \bmod q, \qquad S_3 = \sum_{j=2}^{N} \gamma^{j-1} z_j \in \text{GF}(q^r),$$

$$S_2 = \left(2z_1 + (q-1)z_q + \sum_{j=2}^{q-1} j z_j + \sum_{j=q+1}^{N} (j-1) z_j\right) \bmod q.$$

For different types of error vectors the values of S_1, S_2, and S_3 are given in Table IV, where □ denotes the values that are not important for the decoding.

TABLE IV
Syndromes for Error Types (11), (13), and (19)

Error type:	(11)	(13)	(19)	Positions j
S_1	0	0	t	$1 \le j \le N$
S_2	0	0	□	$j = 1, q-1$
	0	$-t$	□	$j \ne 1, q-1$
S_3	0	$-t\gamma$	0	$j = 1$
	0	$t\gamma^{j-1}(1-\gamma)$	$t\gamma^{j-1}$	$j \ge 2$

Table IV can be used for correction of all errors of the type (11), (13), and (19). After calculation of the syndrome $S = (S_1, S_2, S_3)^{tr}$ the decoder distinguishes the following cases.

1) If $S = 0$, then $\underline{e} = \underline{0}$.
2) If $S_1 = S_2 = 0$, but $S_3 \ne 0$, then the error vector has type (13), and the position j of the first erroneous component is 1 or $q - 1$. Since by assumption $\gamma^{q-3}(1 - q)$ is not an integer, then different combinations of j and t give different values of S_3, and therefore can be recovered from S_3.
3) If $S_1 = 0$ and $S_2 \ne 0$, then the error vector has type (13), and $j \ne 1$ or $q - 1$. In this case $t = -S_2$, and the position j is defined from equation $S_3 = t\gamma^{j-1}(1 - \gamma)$.
4) If $S_1 \ne 0$, then we have a single error (19) with $t = S_1$. If $S_3 = 0$, then the error occurs at position $j = 1$; otherwise the position j is defined from the equation $S_3 = t\gamma^{j+1}$.

Finally, we summarize the properties of the described codes.

Proposition 5: For any prime $q \ge 3$ and any integer $r \ge 2$ the linear code C of the length $N = q^r$ with the parity check matrix H defined by (21) is transparent, has not less than $K = q^r - r - 2$ q-ary information symbols, and corrects all error vectors of the type (13) and (19).

VII. Comparisons and Conclusion

After the formal description of our approach to peak-shift correction in (d, k)-constrained sequences has been completed, we outline its following main features that are important for comparisons and discussions.

1) The proposed method separates encoding of data for error correction and the fulfilment of the (d, k)-constraint into two independent steps performed by the block encoder and $q/2$-converter, respectively. From this point of view, it may be compared with the conventional scheme using an RS-encoder followed by the (d, k) modulator that converts the q-ary output sequence of the RS-code to a binary (d, k)-sequence according to some modulation code [12].

2) The proposed method does not use modulation codes. The (d, k)-constraint is satisfied by direct concatenation of (d, k)-sequences from $q = k - d + 1$ binary strings of the type 0^i1. This straightforward procedure gives relatively good average binary length. However, for some data it can lead to codewords that have a length twice as long as the average length. To overcome such a negative phenomena, we use it in combination with the so called maximum length control, and make the maximum codeword length equal to the average length. The idea of the length control can be used in other situations as well.

3) The concatenation of phrases 0^i1 can be implemented simpler than encoding based on modulation codes. For some values of d and k it gives good rates $R(d, k)$ (for example, $R(1, 3) = 0.5283$, compared with 0.5 for MFM); sometimes the rates are comparable, as in the case $d = 3$, $k = 7$ (see Table V), and other features must be taken into consideration. For large values of q, the simple length control is not as efficient as for small values of q, like $q = 3$.

4) In channels with distortions such as peak-shifts or insertions and deletions of zeros the generation of (d, k)-sequences by con-

catenation of strings 0^i1 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 $GF(2^\nu)$, where ν usually equals $4, 5, \cdots$, corrects a single 2^ν-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 $n = \lambda\nu(2^\nu - 1)/R_{\text{mod}}$ that has $2\lambda\nu/R_{\text{mod}}$ 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 $R = l/n$, where l is the number of information bits encoded in the (d, k)-sequence of length n. For the conventional scheme $l = \lambda\nu(2^\nu - 3)$, and for the proposed scheme $l = \lfloor(N - r - 1)\log_2 q\rfloor - 1$, where $r = \log_q N$ and $\lfloor x\rfloor$ 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 $b \leq \nu(\lambda - 1) + 1$. 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 values of the rate are given in Table V. In the first case, standard MFM is used in the conventional scheme ($R_{\text{mod}} = 0.5$, $b = 1$, $\lambda = 1$). In the second case modulation codes with the following parameters are considered:

MC1) $R_{\text{mod}} = 3/8$, 8 encoder states, decoder window 2 (8 bit) symbols, $b = 9$;

MC2) $R_{\text{mod}} = 5/13$, 10 encoder states, decoder window 2 (13 bit) symbols, $b = 15$;

MC3) $R_{\text{mod}} = 2/5$, 28 encoder states, decoder window 4 (5 bit) symbols, $b = 10$.

The described codes have reasonable efficiency when $k - d + 1$ is < 6. For larger values of $k - d + 1$, the efficiency of the code is determined by the length control mechanism. The "conventional" length 255 symbol, $GF(2^8)$ Reed Solomon ECC followed by a rate 2/3 (1, 7) RLL modulation code has a coding rate of $R = 0.6614$. The efficiency of our scheme is bounded by the length control, and $R(d, k) = \log_2(7)/5 = 0.5614$, see (3).

6) In [11], a scheme is presented that utilizes conventional ECC's having multiple error correcting capability of q-ary symbols. The scheme in [11] also relates q-ary symbols to runs of the (d, k)-constrained sequences, so that peak-shifts effectively transform into additive errors. However, the main difference is that the coding scheme as presented in [11] has no length control and gives codewords of variable length. Considered as "fixed-to-variable" it has good average rate for modulation codes that give maxentropic (d, k)-sequences [11, (35) and (36)]. The main feature of our scheme is that we designed codes that can be used to bound the maximum binary codeword length. Therefore, with regard to efficiency both schemes are hard to compare.

TABLE V
RATES FOR CONVENTIONAL AND PROPOSED SCHEMES

$d = 1, k = 3, q = k - d + 1 = 3$			$d = 3, k = 7, q = k - d + 1 = 5$			
Scheme	n	R	Scheme	n	R	λ
Proposed	81	0.4321	RS (2^4), MC1	480	0.3250	3
RS (2^4), MFM	120	0.4333	RS (2^4), MC3	600	0.3466	3
Proposed	243	0.4897	Proposed, $N = 5^3$	750	0.3720	
RS (2^5), MFM	310	0.4754	RS (2^5), MC1	1240	0.3508	3
Proposed	729	0.5130	RS (2^5), MC2	1612	0.3598	4
RS (2^6), MFM	756	0.4841	RS (2^5), MC3	1163	0.3742	3
Proposed	2187	0.5226	Proposed, $N = 5^4$	3750	0.3835	
RS (2^7), MFM	1778	0.4921	RS (2^6), MC1	3024	0.3631	3
			RS (2^6), MC2	3932	0.3724	4
			RS (2^6), MC3	2873	0.3873	3

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 $t \geq 0$ insertions (deletions) give a syndrome with $S_1 = t$ $(-t)$, $S_2 = t\bar{1}$ $(-t\bar{1})$. 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 0^i1, $d \leq i \leq k$. For the practical values of $d \leq k/2$ we are then always able to extend the code word length to $L_{\max} + d + 1$ binary digits without violating the (d, k)-constraint. The $2/q$-converter uses only the first N strings 0^i1 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.

REFERENCES

[1] C. E. Shannon, "A Mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
[2] K. A. S. Immink, *Coding Techniques for Digital Recorders.* Englewood Cliffs, NJ: Prentice Hall, 1991.
[3] P. Siegel, "Recording codes for digital magnetic storage," *IEEE Trans. Magn.*, vol. MAG-21, pp. 1344–1349, Sept. 1985.
[4] T. D. Howell, "Statistical properties of selected recording codes," *IBM J. Res. Develop.*, vol. 33, no. 1, pp. 60–73, Jan. 1989.
[5] L. A. Bassalygo, "Correcting codes with an additional property," *Prob. Inform. Transm.*, vol. 4, no. 1, pp. 3–8, 1968.
[6] Y. Lin and J. K. Wolf, "ECC/RLL Codes," *IEEE Trans. on Magn.*, vol. MAG-24, Nov. 1988.
[7] P. Lee and J. K. Wolf, "A general error correcting code construction for run-length limited binary channels," *IEEE Trans. Inform. Theory*, vol. 35, no. 6, pp. 1330–1335, Nov. 1989.
[8] H. C. Ferreira and S. Lin, "ON integer compositions and error correcting (d, k) block codes," *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1399–1408, Sept. 1991.
[9] Ø Ytrehus, "Runlength-limited codes for mixed error channels," *IEEE Trans. Inform. Theory*, vol. 37, no. 6, pp. 1577–1585, Nov. 1991.
[10] M. Blaum, "Combining ECC with modulation: performance comparisons," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, pp. 945–949, May 1991.
[11] H. M. Hilden, D. G. Howe, and E. J. Weldon, Jr., "Shift error correcting modulation codes," *IEEE Trans. Magn.*, vol. 27, no. 6, pp. 4600–4605, Nov. 1991.
[12] V. I. Levenshtein and A. J. Vinck, "Perfect (d, k)-codes capable of correcting peak-shifts," submitted to the *IEEE Trans. Inform. Theory*, 1991.
[13] V. I. Levenshtein, "Binary codes correcting spurious insertions and deletions of ones," *Probl. Inform. Transm.*, vol. 34, no. 1, pp. 45–54, Jan. 1965.
[14] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications.* Cambridge: Cambridge Univ. Press, 1986.

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.

Manuscript received August 26, 1991; revised May 15, 1992. This work was supported by Schweiz. Nationalfonds under Grant 2100-026530 while H.-A. Loeliger was with the Signal Processing Laboratory of ETH Zurich, Switzerland.

R. C. Davis is with ESS, Advanced Technology Department, Harris Corporation, Melbourne, FL 32902.

H.-A. Loeliger is with ISY, Linköping University, S-58183 Linköping, Sweden.

IEEE Log Number 9209600.

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 b is proportional to the log-likelihood ratio $\log p_i(r_i \mid c(b))$, where the index i refers to the ith 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 $p_i(\cdot \mid \cdot)$ 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_f 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