On sequential decoding for the Gilbert channel

Johannesson, Rolf; Sidorenko, Vladimir; Zigangirov, Kamil

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/18.21230

1988

point found at the origin. Both of these features are functions of the design rule chosen to select lattice quantizer code book points. Since the same rules are used to design all the code books of this correspondence, all of the quantizers of Tables I–III demonstrate zero and infinity α performance similar to that indicated in Figs. 5 and 6.

Notice that the SNR curves become more broad and offer less SNR performance as the source pdf changes from normal to Laplacian and then to gamma. The nature of the Laplacian and gamma sources cause the regions of greatest source vector probability to spread through n-space along each of the coordinate axes, while a normal source is spherically symmetric about the origin of its space and show no source vector directional preference. The "tailing" effect of the probability concentrations about the axis is the cause of the peak broadening of the Laplacian and gamma curves. Both the A and D lattices spread their points through n-space in a very symmetric manner and the full shell lattice quantizers of this correspondence are expected to perform less than optimally for a source that is not symmetric because many of the code points are spread throughout the central regions of the source space quadrants and are virtually unused. This causes the mse performance of the above "spherical" code books to be worse than those quantizers whose code books are built with a point placement that shows a preference for the source space coordinate axes. No simulations taking this point into consideration were performed in this work, (see [17] for vector quantizers designed with the LBG algorithm for the Laplacian and gamma pdfs that show "tailing" preferences in the code book point placements).

Since the gamma SNR performance of these quantizers has relatively broad peaks that encompass a large range of α, it is possible to select one of several different entropy rates for coding without suffering a loss in the expected SNR performance of a gamma-based system. This feature is not as significant for the Gaussian and Laplacian cases, where the SNR profiles tend to be more peaked.

IV. Conclusion

The expected mse performance of various low code rate vector quantizers, based upon the low dimension A and D lattices, can be better than that of the best scalar quantizers with equivalent code rates. This can be achieved with lattice-based quantizers that are designed by simple methods that do not guarantee globally optimal results. While the design algorithm presented offers only modest improvements in mse performance for a normally distributed source, improvement gains of up to several decibels can be achieved when coding Laplacian and gamma sources.

Acknowledgment

We are grateful to the anonymous reviewers for their constructive criticisms.

References

[1] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[2] S. Lloyd, "Least squares quantization in PCM," unpublished memorandum, Bell Laboratories, 1957 (see *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982).

[3] W. Adams and C. Giesler, "Quantizing characteristics for signals having Laplacian amplitude probability distribution function," *IEEE Trans. Commun.*, vol. COM-26, pp. 1295–1297, Aug. 1978.

[4] M. Paez and T. Glisson, "Minimum mean-square-error quantization in speech PCM and DPCM systems," *IEEE Trans. Commun.*, vol. COM-20, pp. 225–230, Apr. 1972.

[5] W. Pearlman and G. Senge, "Optimal quantization of the Rayleigh probability distribution," *IEEE Trans. Commun.*, vol. COM-27, pp. 101–122, Jan. 1979.

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

[7] P. Zador, "Asymptotic quantization error of continuous signals and the quantization dimension," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–149, Mar. 1982.

[8] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[9] J. D. Gibson, "The role of lattices in multidimensional quantization," presented at the 29th Midwest Symp. Circuits and Systems, Lincoln, NE, Aug. 1986.

[10] K. Sayood, J. Gibson, and M. Rost, "An algorithm for uniform vector quantizer design," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 805–814, Nov. 1984.

[11] J. Humphreys, *Introduction to Lie Algebras and Representation Theory*. New York: Springer-Verlag, 1972.

[12] R. Gilmore, *Lie Groups, Lie Algebras and Some of Their Applications*. New York: Wiley, 1974.

[13] R. Cahn, *Semi-Simple Lie Algebras and Their Representations*. Menlo Park, CA: Benjamin-Cummings, 1984.

[14] J. Conway and N. Sloane, "Fast quantizing and decoding for lattice quantizers and codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–232, Mar. 1982.

[15] ——, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, Nov. 1983.

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

[17] T. Fischer and R. Dicharry, "Vector quantizer design for memoryless Gaussian, gamma, and Laplacian sources," *IEEE Trans. Commun.*, vol. COM-32, pp. 1065–1069, Sept. 1984.

[18] T. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568–583, July 1986.

[19] P. Swaszek, "Vector quantization for image compression," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, March 1986, pp. 254–259.

[20] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.

[21] P. Noll and R. Zelinski, "Bounds on quantizer performance in the low bit rate region," *IEEE Trans. Commun.*, vol. COM-26, pp. 300–304, Feb. 1978.

On Sequential Decoding for the Gilbert Channel

V. SIDORENKO, ROLF JOHANNESSON, MEMBER, IEEE,
AND K. SH. ZIGANGIROV

***Abstract* —It is well-known that the computational performance of sequential decoding deteriorates drastically when channel errors occur in clusters. Hagenauer suggested a feasible modification of sequential decoding which uses a burst-tracking method for better performance. At each step he considers the channel to be memoryless, but with varying probability of error. A different way of using sequential decoding to exploit the memory of a Gilbert channel is presented. A Fano-like metric matched to this channel is used. The methods are investigated by simulating sequential decoding utilizing the stack algorithm. These simulations confirm the feasibility of the proposed techniques.**

Manuscript received December 15, 1986; revised November 3, 1987. This work was supported in part by the Royal Swedish Academy of Sciences and the Royal Swedish Academy of Engineering Sciences in liaison with the Academy of Sciences of the USSR and in part by the Swedish Board for Technical Development under Grant 85-3303. This work was partially presented at the Second Joint Swedish–Soviet International Workshop on Information Theory, Gränna, Sweden, April 1985, and was presented at the IEEE International Symposium on Information Theory, Ann Arbor, MI, October 6–9, 1986.

V. Sidorenko and K. Sh. Zigangirov are with the Institute for Problems of Information Transmission, USSR Academy of Sciences, Ermolovoy 19, Moscow GSP-4, USSR.

R. Johannesson is with the Department of Information Theory, University of Lund, P.O. Box 118, S-221 00 Lund, Sweden.

IEEE Log Number 8823990.

0018-9448/88/0900-1058$01.00 ©1988 IEEE

I. Introduction

In digital transmission over many real channels, the errors occur in bursts separated by fairly long error-free gaps. Since most error control systems are designed for memoryless channels, interleaving is often used to smear out the errors and create a channel that is effectively memoryless. However, in many applications this causes an unacceptable delay. Since the existence of memory implies more capacity, it is tempting to exploit the memory when designing the error control systems.

Gilbert [1] suggested the following model for a channel with memory: a binary symmetric channel in which the crossover probability is determined by a two-state Markov chain (Fig. 1). In the "good state" G the channel is always error-free. In the "bad" or "burst state" B the crossover probability p is equal to $1/2$.

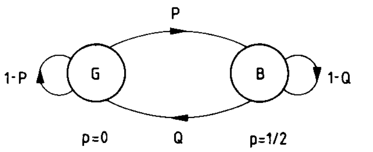

Fig. 1. Gilberts's model for channel with memory.

Standard sequential decoding performs very poorly on a Gilbert channel. To exploit the memory we modify the stack algorithm [2], [3] by combining short interleaving with the use of a Fano-like metric matched to the channel and obtain a substantial improvement in computational performance.

In Section II we give a brief description of Hagenauer's [4] method for sequential decoding on burst-error channels. Also, we suggest a slight modification of this algorithm which gives an improvement when the interleaving is short. In Section III we describe a new algorithm and in Section IV we report on simulations which confirm the superiority of our algorithm when the interleaving is short. An "embryo" of our method was presented in Gränna [5].

II. Hagenauer's Burst-Tracking Method

To improve the computational performance of sequential decoding, Hagenauer suggested a method that, in addition to interleaving, keeps track of a burst during decoding. He used the channel statistics of a four-state Markov model, simulated both modified stack and Fano decoders with high rate codes, and showed the feasibility of his method.

Hagenauer's method is based on interleaving. The de-interleaver matrix is filled from the channel, column by column. The sequential decoder moves row by row. Let n denote the number of channel symbols per branch. Assume that the number of columns in the matrix rn is large enough so that the decoder never backs up more than r subblocks. Hagenauer uses a burst indicator for each column to keep track of whether the channel is in the burst state or in the good state. The probabilities of being in a burst, when the burst indicator is at certain levels, are calculated and used in the evaluation of the Fano metric [6].

If the interleaving is short, the sequential decoder will back up more than r subblocks, and the stack algorithm might get into trouble. To avoid incorrect alteration of the burst indicators, we modify the method and store the set of burst indicators together with each node on the stack. This modified algorithm is, of course, more complex but could be used as a reference when we compare different methods from a computational distribution point of view.

III. Sidorenko–Johannesson–Zigangirov (SJZ) Algorithm

Let us consider a rate $R = k/n$, memory M, binary convolutional code, which we represent by the generator matrix $G(D)$, where

$$G(D) = \begin{pmatrix} G_1^1(D) & G_1^2(D) & \cdots & G_1^n(D) \\ \vdots & \vdots & & \vdots \\ G_k^1(D) & G_k^2(D) & \cdots & G_k^n(D) \end{pmatrix}$$

and

$$G_v^l(D) = g_{0v}^l + g_{1v}^l D + \cdots + g_{Mv}^l D^M$$

where $l = 1, \cdots, n$, and $v = 1, \cdots, k$. The information sequence $i_0^{(1)}, i_0^{(2)}, \cdots, i_0^{(k)}, i_1^{(1)}, i_1^{(2)}, \cdots, i_1^{(k)}, \cdots$ is encoded as the sequence

$$x = x_0^{(1)}, x_0^{(2)}, \cdots, x_0^{(n)}, x_1^{(1)}, x_1^{(2)}, \cdots, x_1^{(n)}, \cdots$$

where

$$x_j^{(l)} = \sum_{v=1}^{k} \sum_{s=0}^{M} i_{j-s}^{(v)} g_{sv}^l \pmod 2.$$

The encoded sequence x is divided into frames. Each frame is divided into blocks of length Lrn. The first block is shown as:

$x_0^{(1)} \cdots x_0^{(n)}$	$x_1^{(1)} \cdots x_1^{(n)}$	$\cdots$	$x_{r-1}^{(1)} \cdots x_{r-1}^{(n)}$
$x_r^{(1)} \cdots x_r^{(n)}$	$x_{r+1}^{(1)} \cdots x_{r+1}^{(n)}$	$\cdots$	$x_{2r-1}^{(1)} \cdots x_{2r-1}^{(n)}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$
$x_{(L-1)r}^{(1)} \cdots x_{(L-1)r}^{(n)}$	$\cdots$	$\cdots$	$x_{Lr-1}^{(1)} \cdots x_{Lr-1}^{(n)}$

The blocks are transmitted columnwise. The received sequence y is de-interleaved in the obvious way. We will choose as our stack decoder output the sequence x, which maximizes the logarithm of the *a posteriori* probability $\log_2 P[x, s|y]$, where s is the state sequence. This is equivalent to maximizing

$$\log_2 \frac{P[y, s|x] P[x]}{P[y]}$$

where $P[x] = 2^{-tR}$ and t is the path length. Following Fano [7], $P[y]$ is approximated by $P_0[y] = 2^{-t}$. Hence we have the diagram in Fig. 2.

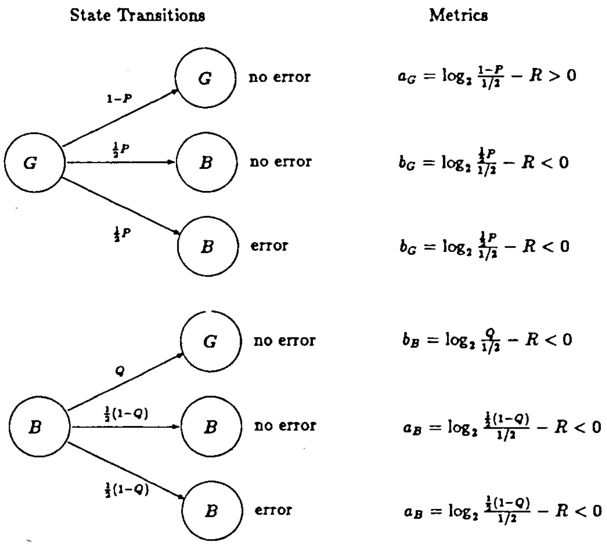
State Transitions
Metrics
G
G
no error
1-P
$a_G = \log_2 \frac{1-P}{1/2} - R > 0$
B
no error
$\frac{1}{2}P$
$b_G = \log_2 \frac{\frac{1}{2}P}{1/2} - R < 0$
B
error
$\frac{1}{2}P$
$b_G = \log_2 \frac{\frac{1}{2}P}{1/2} - R < 0$
B
G
no error
Q
$b_B = \log_2 \frac{Q}{1/2} - R < 0$
B
no error
$\frac{1}{2}(1-Q)$
$a_B = \log_2 \frac{\frac{1}{2}(1-Q)}{1/2} - R < 0$
B
error
$\frac{1}{2}(1-Q)$
$a_B = \log_2 \frac{\frac{1}{2}(1-Q)}{1/2} - R < 0$

Fig. 2. State transitions and corresponding metrics.

Since the stack decoder always follows the path with maximal metric the decoder will never make the transition $G \rightarrow B$, no error. Even if the true state transition is $G \rightarrow B$, no error, the decoder will make the transition $G \rightarrow G$, no error.

Let τ be the largest integer such that

$$b_B + \tau a_G + b_G < (\tau + 2) a_B.$$

The ideal stack decoder will never contain more than τ consecutive state transitions B, no error $\rightarrow B$, no error. Furthermore, it will never contain less than $\tau + 1$ consecutive state transitions G, no error $\rightarrow G$, no error between two B, error states.

To simplify our decoder, for each of the rn columns in the de-interleaver matrix we use a flag to denote when we are in a bad state. The flag will be reset after receiving T consecutive error-free symbols, where $T \leq \tau + 1$. Thus our Fano-like metric m will be calculated for each received symbol according to the rules in Table I. Our Gilbert channel (see Section IV) gives $\tau = 9$ but simulations indicate that we should use $T = 6$.

TABLE I

Situation	State Transition	Flag	Metric
No flag, no error	$G \rightarrow G$	—	$m \leftarrow a_G = \log_2 2(1-P) - R$
No flag, error	$G \rightarrow B$	set flag	$m \leftarrow b_G = \log_2 P - R$
If flag and T consecutive error-free symbols	$B \rightarrow G$	reset flag	$m \leftarrow b_B = \log_2 2Q - R$
Otherwise	$B \rightarrow B$	—	$m \leftarrow a_B = \log_2 (1-Q) - R$

IV. Simulations

To investigate the distribution of the average number of computations we conducted several simulations. Three different codes were tested, viz., an optimum distance profile (ODP) systematic convolutional code with rate $R = 1/3$, memory $M = 19$, $d_{\text{free}} = 24$, and distance profile $\boldsymbol{d} = [3,4,5,6,7,8,9,10,11,11,12,13,13,14,15,15,16,16,17,18]$, [8]; an $R = 2/6$ systematic convolutional code with $M = 9$ and an equivalent initial (over the first seven branches) growth of the column distances $\boldsymbol{d} = [3,5,7,9,11,12,13,14,16,16]$ and with generator (octal notation):

$$G_1 = \begin{pmatrix} 4000 & 0000 & 5660 & 6424 & 3574 & 1340 \\ 0000 & 4000 & 1440 & 2254 & 5764 & 7214 \end{pmatrix}$$

and, finally, an $R = 2/6$ systematic convolutional code with $M = 9$ and an initial growth of the column distances that is optimal over the first four branches, $\boldsymbol{d} = [4,6,8,10,11,12,13,14,15,17]$. This code has generator

$$G_2 = \begin{pmatrix} 4000 & 0000 & 6760 & 3264 & 4034 & 7040 \\ 0000 & 4000 & 1540 & 5014 & 7524 & 6754 \end{pmatrix}.$$

The simulated Gilbert channel had $P = 0.005$ and $Q = 0.05$. Hence we will stay in the burst state $P/(P+Q) \approx 9$ percent of the time. For the three different codes 500 frames each consisting of 1000 information digits (augmented with a tail of dummy zeros) were decoded by the SJZ algorithm. The interleaving was ($Lrn =$) 600 bits. For a given interleaving the value of r that was optimal from a computational distribution point of view was chosen. The computational distributions are shown in Fig. 3 and, as expected, the rate $R = 2/6$ codes gave better computational performances than the $R = 1/3$ code. For this reason we used the rate $R = 2/6$ code with generator G_2 to compare the computational performances of the algorithms. For comparison we also show the computational distribution for the same channel and the interleaving when the classical stack algorithm is used.

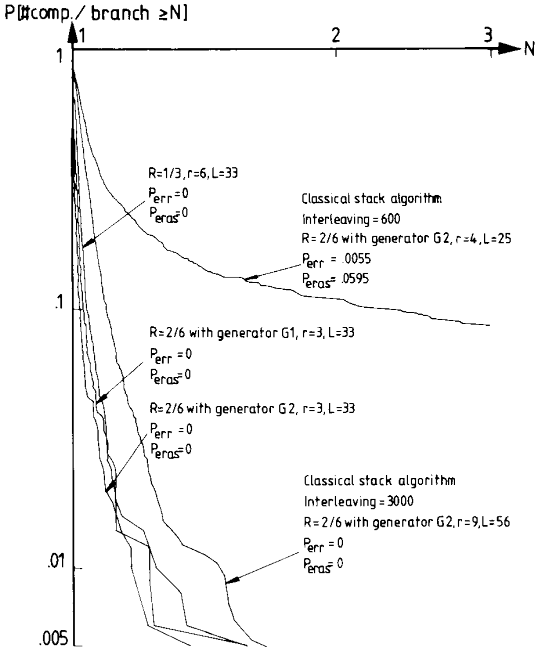

Fig. 3. Computational distributions for three different codes decoded by the SJZ algorithm using interleaving 600 bits. For comparison, performance of classical stack algorithm using interleaving 600 and 3000 bits is shown.

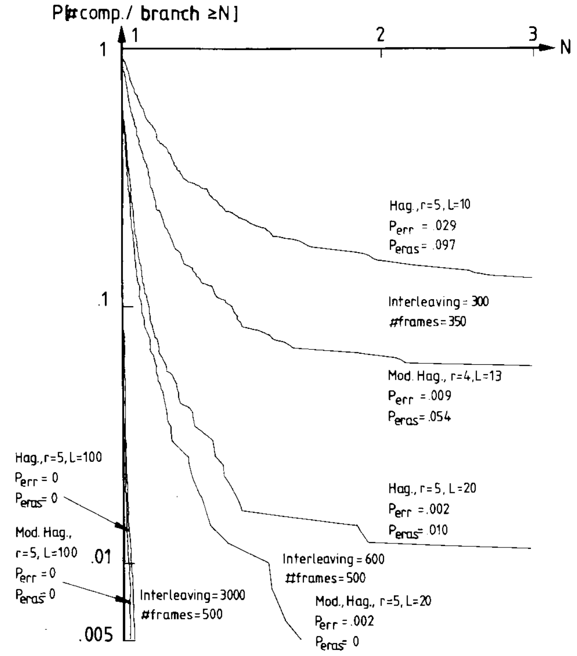

Fig. 4. Comparison of Hagenauer's stack algorithm with our (more complex) modification of his algorithm for three different interleavings: 300, 600, and 3000 bits.

In Fig. 4 we compare the computational distributions obtained by Hagenauer's algorithm with those obtained by our (more complex) modification of his algorithm. Using 3000-bit interleaving the algorithms show similar performances. The modified algorithm is somewhat better with 600-bit interleaving, but for short interleaving (300 bits) the modified algorithm is, of course, much better.

Finally, in Fig. 5 it is shown that the SJZ algorithm gives slightly better performance using short interleaving than the modification of Hagenauer's algorithm does.

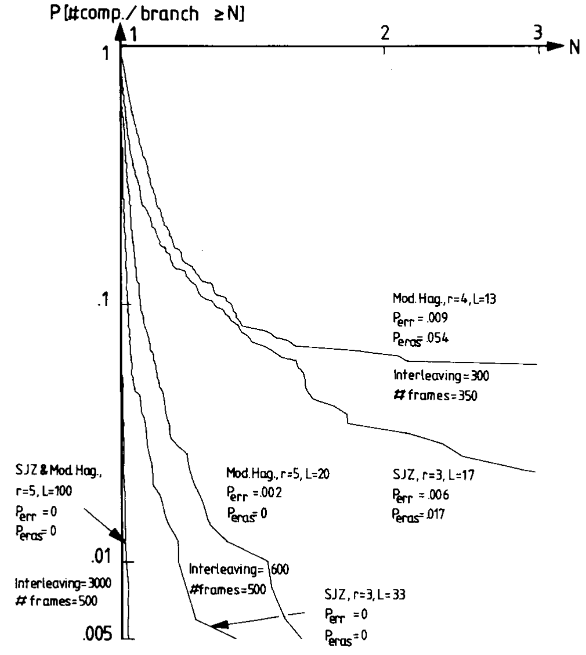

Fig. 5. Comparison of our modification of Hagenauer's algorithm with new SJZ algorithm for three different interleavings: 300, 600, and 3000 bits.

Remark: The modifications of the algorithm necessary to deal with a more general channel are nontrivial. In [9] we presented some preliminary results for the Gilbert–Elliot channel.

References

[1] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1266, Sept. 1960.

[2] K. Sh. Zigangirov, "Some sequential decoding procedures," *Probl. Peredachi Inform.*, vol. 2, pp. 13–25, 1966.

[3] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, vol. 13, pp. 675–685, Nov. 1969.

[4] J. Hagenauer, "Sequential decoding for burst-error-channels," in *Communication Systems and Random Process Theory*, J. K. Skwirzynski, Ed. Alphen am Rhein, The Netherlands: Sijthoff & Noordhoff, 1978, pp. 249–265.

[5] K. Berglund, R. Johannesson, and K. Sh. Zigangirov, "A new sequential decoding algorithm for fading channels," in *Proc. 2nd Joint Swedish-Soviet Int. Workshop Inform. Theory*, Gränna, Sweden, Apr. 1985, pp. 212–216.

[6] J. L. Massey, "Variable-length codes and the Fano metric," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 196–198, Jan. 1972.

[7] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64–74, Apr. 1963.

[8] R. Johannesson, "Some rate 1/3 and 1/4 binary convolutional codes with an optimum distance profile," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 281–283, Mar. 1977.

[9] G. Bratt, R. Johannesson, and K. Sh. Zigangirov, "On sequential decoding for the Gilbert-Elliot channel," in *Proc. 3rd Joint Swedish-Soviet Int. Workshop Inform. Theory*, Sochi, USSR, May 1987, pp. 192–193.

Multiple Burst-Correcting Array Codes

MARIO BLAUM, MEMBER, IEEE, PATRICK G. FARRELL,
MEMBER, IEEE, AND HENK C. A. VAN TILBORG,
SENIOR MEMBER, IEEE

***Abstract*—Two families of binary linear multiple burst-correcting array codes are presented. The codes consist of all possible $n_1 \times n_2$ arrays over GF(2), where the columns have even parity and the rows belong to any given code of length n_2 and minimum distance $2t$. It is shown that if the bits are read out diagonally instead of horizontally, each diagonal followed by the preceding one (viewed cyclically), then the code can correct up to t bursts of length $\leq n_1$ if and only if $n_2 \geq 2tn_1 + 1$. If each diagonal is followed by the next one, the code can correct up to t bursts of length $\leq n_1 - 1$ if and only if $n_2 \geq 2t(n_1 - 2) + 1$. For $t = 1$ some of these results are already known. Decoding algorithms are presented, and the case $t = 1$ is discussed in more detail.**

I. Introduction

Consider the simple binary array code $n_1 \times n_2$ where each row and column has even weight. It is well-known that this code can correct any random error. However, if the bits are read diagonally instead of horizontally, the code can correct a burst of errors [1], [5]. The diagonal readout proceeds as follows. Fix a parameter s, $1 \leq s \leq n_2 - 1$, $\gcd(s, n_2) = 1$; start in the upper left hand corner, and read the corresponding diagonal; then do the same thing to diagonal s, then to diagonal $2s$, etc. [7]. The array is viewed cyclically on n_2, i.e., the last column is followed by the first one. Fig. 1 shows an example of diagonal readout with $s = 3$.

The purpose of this correspondence is to generalize the result [1], that the $n_1 \times n_2$-code with readout $s = 1$ can correct any burst of length up to $n_1 - 1$ if and only if $n_2 \geq 2n_1 - 3$. (Codes with $s = 1$ were investigated also in [2] and [5].)

We study arrays in which columns have even weight and rows are codewords from a code with minimum distance $2t$. It was observed [3], [4], that in some cases one has multiple burst-error correction. The proposed decoding algorithms depended on the specific choice of the code of minimum distance $2t$. Only the readout with $s = 1$ was considered in [3] and [4]. Here we obtain conditions on n_1 and n_2 that are independent of the particular code of minimum distance $2t$ implemented on the rows. We concentrate on the case $s = n_2 - 1$ (denoted $s = -1$), but the analogous result for $s = 1$ is also given. Our main result is proved in the next section and can be stated as follows: Consider the $n_1 \times n_2$ array code with the $s = -1$ readout, each column having even weight and each row being a codeword in a given code of minimum distance $2t$. Then this array code can correct up to t bursts of length $\leq n_1$ if and only if $n_2 \geq 2tn_1 + 1$.

A similar result can be obtained when $s = 1$. In this case, the array code can correct up to t bursts of length $\leq n_1 - 1$ if and only if $n_2 \geq 2t(n_1 - 2) + 1$. Notice that when $t = 1$, we obtain the result of [1].

In Section III we discuss decoding strategies. In Section IV we study the special case $t = 1$ and shortening procedures.

Manuscript received November 7, 1986; revised November 3, 1987.

M. Blaum is with the IBM Almaden Research Center, 650 Harry Road, K65/802, San Jose, CA 95120-6099.

P. G. Farrell was with the California Institute of Technology, Pasadena, CA, on sabbatical. He is now with the Department of Electrical Engineering, University of Manchester, Manchester, UK.

H. C. A. van Tilborg was visiting with the IBM Almaden Research Center, San Jose, CA. He is now with the Department of Mathematics, Eindhoven University of Technology, Eindhoven, The Netherlands.

IEEE Log Number 8823986.

0018-9448/88/0900-1061$01.00 ©1988 IEEE