solid curves indicate the $\max_r B_m(r,\eta)$ versus η with m as a parameter, and the dashed curves show $S_m(r,\eta)$ versus η, for $m = \infty$ and $r = 1$ with N as a parameter. The rapid increase of $\max_r B_m(r,\eta)$, for $\eta \gtrsim m$, is due to the discrete structure of the references, while the mild increase of $\max_r B_m(r,\eta)$ with decreasing η reflects the effect of the "amplitude limiting" [7]. Finally, $S_m(1,\eta)/S_\infty(1,\eta)$ versus η is shown for various values of m in Fig. 6. The statistical error $S_m(r,\eta)$ is not much affected by the discreteness of the references for $\eta \lesssim m$.

As far as the continuously distributed references were concerned, Castanie *et. al.* [7] indicated that, for $1 \lesssim \eta \lesssim 1.5$, the bias error is reasonably acceptable and that the statistical error is of the same order of magnitude as that of direct correlators. Figs. 5 and 6 suggest that, for those values of η, both the bias error and statistical error are quite well within an acceptable value, provided the value of m is chosen to be eight.

Acknowledgment

The computations described here have been performed using the FACOM 230-60 computer at the Computation Center of Nagoya University.

References

[1] T. Sato, "Autocorrelator using random voltage method," *Automat. Contr.* (Trans. Soc. Instrum. and Contr. Engrs. of Japan), vol. 7, pp. 8–12, Jan. 1960.
[2] B. P. T. Veltman and H. Kwakernaak, "Theorie und technik der polaritätskorrelation für die dynamische analyse niederfrequenter signale und systeme," *Regelungstechnik*, vol. 9, pp. 357–364, Sept. 1961.
[3] H. Berndt, "Correlation function estimation by a polarity method using stochastic reference signals," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 796–801, Nov. 1968.
[4] J. B. Knowles and H. T. Tsui, "Correlating devices and their estimation errors," *J. Appl. Phys.*, vol. 38, pp. 607–612, Feb. 1967.
[5] P. J. Kindlman and E. B. Hooper, Jr., "High speed correlator," *Rev. Sci. Instrum.*, vol. 39, pp. 864–872, June 1968.
[6] K. Y. Chang and A. D. Moore, "Modified digital correlator and its estimation errors," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 699–706, Nov. 1970.
[7] F. Castanie, J. C. Hoffmann, and B. Lacaze, "On the performance of a random reference correlator," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-20, pp. 266–269, Mar. 1974.
[8] F. R. Lawson and C. D. McGillem, "The use of a high sampling rate and ternary quantization to improve the performance of the random reference correlator," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-20, pp. 269–271, Mar. 1974.
[9] H. Toda, T. Takeuchi, and T. Shimizu, private communication, 1970.

Concerning a Bound on Undetected Error Probability

S. K. LEUNG-YAN-CHEONG AND MARTIN E. HELLMAN

***Abstract*—In the past, it has generally been assumed that the probability of undetected error for an (n,k) block code, used solely for error detection on a binary symmetric channel, is upperbounded by $2^{-(n-k)}$. In this correspondence, it is shown that Hamming codes do indeed obey this bound, but that the bound is violated by some more general codes. Examples of linear, cyclic, and Bose–Chaudhuri–Hocquenghem (BCH) codes which do not obey the bound are given.**

I. Introduction

An (n,k) block code with minimum distance $d_{\min}$ can be used in an error correcting capacity to correct all error patterns of weight $\lfloor (d_{\min} - 1)/2 \rfloor$ or less [2], where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. Alternatively, it can be used in an error detecting capacity to detect all error patterns of weight $d_{\min} - 1$ or less [2]. In the latter case, the guaranteed performance (i.e., assuming all patterns of weight $d_{\min}$ or greater cause undetected errors) is frequently much inferior to the actual performance. For example, a Hamming (127,120) code with $d_{\min} = 3$ has a guaranteed undetected error rate $P(e) \leq 0.135$ when used on a binary symmetric channel (BSC) with crossover probability $\varepsilon = 0.01$. This guaranteed performance is greatly inferior to the actual performance, and it will be shown that the bound can easily be strengthened to $P(e) < 2^{-7} = 0.0078$ and that the actual value is $P(e) = 0.0011$. The reason for the large discrepancy between the guaranteed and the actual performance is that, although the code cannot detect all error patterns of weight 3, it can detect most errors of this, and even higher, weight.

The simple but fairly tight bound, $P(e) < 2^{-7}$, is a special case of what has often been assumed to be a general rule for block codes, namely, $P(e) < 2^{-p}$, where $p = n - k$ is the number of parity bits [1]. In this correspondence we show that, while a Hamming code used on a BSC with $\varepsilon \leq \frac{1}{2}$ has $P(e) < 2^{-p}$, block codes in general need not obey this bound. We further show that even linear, cyclic, and Bose–Chaudhuri–Hocquenghem (BCH) codes can violate the bound, as can shortened Hamming codes.

The "justification" for the 2^{-p} bound went as follows. Assume that the (n,k) code is used over a totally noisy BSC (i.e., $\varepsilon = \frac{1}{2}$). Then, all 2^n possible received sequences are equally likely, and of these only $2^k - 1$ are codewords other than the one actually transmitted. Therefore, $P(e) = (2^k - 1)/2^n < 2^k/2^n = 2^{-p}$. Since for this worst of all possible channels, $P(e) < 2^{-p}$, it is "reasonable" to assume that for better channels $P(e)$ would still be bounded by 2^{-p}. Unfortunately, as shown below, the above argument is true only for those codes in which the codewords have a very symmetrical distribution in n-space. The question, therefore, arises as to which block codes satisfy the 2^{-p} bound. We attempt to provide some answers in the following.

II. Applicability of the 2^{-p} Bound

Proposition 1: Linear block codes do not necessarily obey the 2^{-p} bound.

Proof: Consider a (1023,1013) code in which the ten parity check bits are identically zero. On a totally noisy channel this code has $P(e) < 2^{-10} = 9.77 \times 10^{-4}$; however, on a BSC with $\varepsilon = 0.01$, we find $P(e) = 0.904$. This is because the probability of some error occurring in the 1013 information bits is essentially one, while the probability of at least one error in the 10 parity bits is only 0.096. However, only when at least one of the parity bits is received in error does the decoder detect an error.

Proposition 2: Cyclic codes do not necessarily obey the 2^{-p} bound.

Proof: Consider the following cyclic code which consists of eight codewords, each formed by repetitions of a three-bit pattern:

000000000 ··· 000
001001001 ··· 001
010010010 ··· 010
011011011 ··· 011
100100100 ··· 100
101101101 ··· 101
110110110 ··· 110
111111111 ··· 111.

Manuscript received March 20, 1975. This work was supported under National Science Foundation Grant GK-33250 and the Joint Services Electronics Program under Contract N00014-67-A-0044.
The authors are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

Suppose we have ten blocks of 3 bits for each codeword. Then $n = 30$, $k = 3$, and $p = 27$.

Clearly on a BSC with crossover probability ε,

$$P(e) = 3\varepsilon^{10}(1-\varepsilon)^{20} + 3\varepsilon^{20}(1-\varepsilon)^{10} + \varepsilon^{30}.$$

When $\varepsilon = 1/3$, $P(e) > 3(1/3)^{10}(2/3)^{20} = 1.53 \times 10^{-8} > 2^{-27} = 7.45 \times 10^{-9}$.

Proposition 3: BCH codes do not necessarily satisfy the 2^{-p} bound.

Proof: The generator polynomial for the (63,24) BCH code is $g(x) = 1 + x^5 + x^8 + x^{11} + x^{17} + x^{22} + x^{23} + x^{25} + x^{27} + x^{28} + x^{31} + x^{33} + x^{34} + x^{36} + x^{37} + x^{38} + x^{39}$. From this, the weight distribution of the code can be computed. Here $p = 63 - 24 = 39$ and $2^{-p} = 1.82 \times 10^{-12}$. On a BSC with crossover probability $\varepsilon = 0.28$, it can be shown that $P(e) = 2.13 \times 10^{-12}$ which exceeds 2^{-p}.

Proposition 4: Hamming codes satisfy the 2^{-p} bound if the BSC has $\varepsilon \leq \frac{1}{2}$.

Comments: The restriction on ε is one which is usually taken for granted in communication problems. Since Hamming codes always contain the all ones codeword (i.e., $111\cdots11$), it is clear that, if there were no restriction on ε, $P(e)$ would be equal to one for $\varepsilon = 1$. To see why the vector $(111\cdots11)$ is a codeword of any Hamming code, note that the generator polynomial $g(x)$ of a Hamming code is primitive and therefore irreducible. Now $(111\cdots11)$ is a codeword, if and only if $g(x)$ divides $(x^{n-1} + x^{n-2} + \cdots + x + 1)$. This, in turn, is equivalent to $(x+1)g(x)$ divides $(x^n + 1)$. However, we know $g(x)$ divides $(x^n + 1)$ (see [2, p. 62, theorem 4.2]), and, since $g(x)$ and $(x+1)$ are relatively prime, $(x+1)g(x)$ divides $(x^n + 1)$.

Proof: We have already shown that the probability of undetected error of a block code on a BSC with $\varepsilon = \frac{1}{2}$ is bounded above by 2^{-p}. Therefore, it is sufficient to show that $P(e)$ is monotonically increasing for $0 \leq \varepsilon \leq \frac{1}{2}$. This is proved in the following lemma.

Lemma: For Hamming codes, $P(e)$ is monotonically increasing in ε.

Proof: From [3], the weight distribution for binary Hamming (n,k) codes is given by

$$A(x) = \sum_{i=0}^{n} A_i x^i = \frac{1}{2^m}\{(1+x)^n + (2^m - 1)(1+x)^{(n-1)/2}(1-x)^{2^{m-1}}\} \tag{1}$$

where A_i denotes the number of codewords of weight i; $n = 2^m - 1$, $k = n - m$. From (1),

$$A(x) = \frac{1}{(n+1)}\{(1+x)^n + n(1+x)^{(n-1)/2}(1-x)^{(n+1)/2}\} \tag{2}$$

and

$$A(x) - 1 = \sum_{i=1}^{n} A_i x^i, \qquad \text{since } A_0 = 1. \tag{3}$$

Since the probability of undetected error is independent of the transmitted codeword, we can assume the all zeros codeword is sent and find

$$\begin{aligned} P(e) &= \sum_{i=1}^{n} A_i \varepsilon^i (1-\varepsilon)^{n-i} \\ &= (1-\varepsilon)^n \sum_{i=1}^{n} A_i \left(\frac{\varepsilon}{1-\varepsilon}\right)^i \\ &= (1-\varepsilon)^n \left[A\left(\frac{\varepsilon}{1-\varepsilon}\right) - 1\right] \\ &= \frac{(1-\varepsilon)^n}{(n+1)}\left[\left(\frac{1}{1-\varepsilon}\right)^n + n\left(\frac{1}{1-\varepsilon}\right)^{(n-1)/2} \cdot \left(\frac{1-2\varepsilon}{1-\varepsilon}\right)^{(n+1)/2} - (n+1)\right] \\ &= \frac{1}{(n+1)}\left[1 + n(1-2\varepsilon)^{(n+1)/2} - (n+1)(1-\varepsilon)^n\right]. \end{aligned} \tag{4}$$

Differentiating $P(e)$ with respect to ε, we obtain

$$\begin{aligned} \frac{dP(e)}{d\varepsilon} &= \frac{1}{(n+1)}\left[n\left(\frac{n+1}{2}\right)(1-2\varepsilon)^{(n-1)/2}(-2) + n(n+1)(1-\varepsilon)^{n-1}\right] \\ &= n[(1-\varepsilon)^{n-1} - (1-2\varepsilon)^{(n-1)/2}] \\ &= (2l+1)[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] \end{aligned} \tag{5}$$

where l is some odd positive integer since $n = 2^m - 1$. Note that

$$\frac{dP(e)}{d\varepsilon} = 0, \qquad \text{at } \varepsilon = 0. \tag{6}$$

We now prove that $[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] > 0$, for $0 < \varepsilon \leq 1$ and l odd. Since l is odd, it is clear that $[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] > 0$, for $\frac{1}{2} \leq \varepsilon \leq 1$. We now prove by induction that $[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] > 0$, for $0 < \varepsilon < \frac{1}{2}$ and $l \geq 1$.

Let

$$f_l(\varepsilon) = (1-\varepsilon)^{2l} - (1-2\varepsilon)^l. \tag{7}$$

Then

$$f_1(\varepsilon) = (1-\varepsilon)^2 - (1-2\varepsilon) = \varepsilon^2 > 0, \qquad \text{since } \varepsilon \neq 0. \tag{8}$$

Now assume

$$f_l(\varepsilon) = (1-\varepsilon)^{2l} - (1-2\varepsilon)^l > 0, \qquad \text{for } 0 < \varepsilon < \tfrac{1}{2}. \tag{9}$$

Since

$$\begin{aligned} f_{l+1}(\varepsilon) &= (1-\varepsilon)^{2(l+1)} - (1-2\varepsilon)^{l+1} \\ &= (1-\varepsilon)^2(1-\varepsilon)^{2l} - (1-2\varepsilon)(1-2\varepsilon)^l \end{aligned} \tag{10}$$

and

$$(1-\varepsilon)^2 = 1 - 2\varepsilon + \varepsilon^2 > 1 - 2\varepsilon > 0, \qquad \text{for } 0 < \varepsilon < \tfrac{1}{2} \tag{11}$$

we have

$$\begin{aligned} f_{l+1}(\varepsilon) &> (1-\varepsilon)^2[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] \\ &= (1-\varepsilon)^2 f_l(\varepsilon) \\ &> 0, \qquad \text{by assumption.} \end{aligned} \tag{12}$$

So, by mathematical induction, $[(1-\varepsilon)^{2l} - (1-2\varepsilon)^l] > 0$, for $0 < \varepsilon < \frac{1}{2}$. This concludes the proof that $P(e)$ is monotonically increasing in ε.

Note: It is easy to extend Proposition 4 to the repetition and Golay codes and thereby to all binary perfect codes.

Proposition 5: Shortened Hamming codes do not necessarily obey the 2^{-p} bound.

Proof: Consider a Hamming (1023,1013) code which has its parity-check bits at positions 2^i, $i = 0, 1, \cdots, 9$ [4]. This implementation is useful for error correction because the syndrome gives the position of the error. If we shorten this code to a (12,2) code by eliminating the leading 1011 information bits, it is easily seen that the codewords for this shortened code are

$$\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0. \end{array}$$

So in this case $P(e) = \varepsilon^3(1-\varepsilon)^9 + \varepsilon^{10}(1-\varepsilon)^2 + \varepsilon^{11}(1-\varepsilon)$. For example, for $\varepsilon = 0.25$, $P(e) = 1.174 \times 10^{-3} > 2^{-10} = 9.766 \times 10^{-4}$.

We might also consider shortening a Hamming code in cyclic form. Recall that an (n, k) Hamming code is a cyclic code whose generator polynomial $g(x)$ is a primitive polynomial of degree m, where $n = 2^m - 1$ and $k = 2^m - m - 1$. From [3] we find that $x^{10} + x^3 + 1$ is a primitive polynomial so that the generator matrix for a cyclic Hamming (1023,1013) code might be

$$\begin{array}{c} \\ 1013 \end{array} \overset{\longleftarrow 1012 \longrightarrow}{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ & \cdot & & & & \cdot & & & & & & & & & & \\ & & \cdot & & & & \cdot & & & & & & & & & \\ & & & \cdot & & & & \cdot & & & & & & & & \end{bmatrix}}.$$

If we shorten the code [3, p. 241] to an (11,1) code, the generator matrix will be just [10010000001]. This gives $P(e) = \varepsilon^3(1-\varepsilon)^8$ which has its maximum at $\varepsilon = 3/11$. At $\varepsilon = 3/11$, $P(e) = 1.59 \times 10^{-3} > 2^{-10} = 9.77 \times 10^{-4}$.

It should be noted that we chose a primitive polynomial which had the minimum number of nonzero coefficients. This reduces the cost for hardware implementation.

III. Conclusion

It has been demonstrated that Hamming codes always obey the 2^{-p} bound on $P(e)$ for $\varepsilon \leq \frac{1}{2}$. However, this is not necessarily true for more general cyclic codes. Therefore, caution is needed when using the bound, especially in the design of systems in which the probability of undetected error is critical. Fortunately, the bound is not badly violated by the cyclic codes we have used as counterexamples, providing hope that a slight overdesign should suffice. More careful analysis or simulation should be used to check out such designs.

Even for Hamming codes, $P(e)$ can be larger than 2^{-p} for $\varepsilon > \frac{1}{2}$. This leads one to wonder if there are any codes which are uniformly good for all $0 \leq \varepsilon \leq 1$. We have been able to show that an (n,k) Hamming code altered to an $(n,k-1) \equiv (n',k')$ code by deleting all codewords of weight greater than $n/2$ has $P(e) < 2^{-(n-k)} = 2^{-(n'-k')+1}$, for all $0 \leq \varepsilon \leq 1$. The proof is straightforward and relies on evaluation of $P(e)$ through use of the weight enumerator polynomial. We have further found, through numerical evaluation, that, when the (7,4) code is thus shortened to a (7,3) code, $P(e) < 2^{-4}$ as opposed to the 2^{-3} value predicted by the above argument, for all $0 \leq \varepsilon \leq 1$. And it is easily shown that the code obtained by affixing an overall parity-check bit to a maximal length shift register code has its maximum undetected error rate at $\varepsilon = \frac{1}{2}$, and thus has $P(e) \leq (2^k - 1)/2^n$, for all $0 \leq \varepsilon \leq 1$. Thus, for these codes, $\varepsilon = \frac{1}{2}$ is the "worst of all possible channels" for error detection. However, we doubt that this is true for codes with more reasonable rates.

Acknowledgment

The authors would like to thank Prof. John T. Gill for his help in computing the weight distribution of some BCH codes.

References

[1] M. E. Hellman, "Error detection made simple," in *Conference Record, Int. Conf. on Communications*, Minneapolis, Minn., June 17–19, 1974, pp. 9A-4.
[2] S. Lin, *An Introduction to Error-Correcting Codes*. Englewood Cliffs, N.J.: Prentice-Hall, 1970.
[3] W. W. Peterson and E. J. Weldon, *Error Correcting Codes*, 2nd ed. Cambridge, Mass.: M.I.T. Press, 1972.
[4] W. Bennett and J. Davey, *Data Transmission*. New York: McGraw-Hill, 1965.

On the Complexity of Decoding Reed–Solomon Codes

JØRN JUSTESEN

***Abstract*—Certain q-ary Reed–Solomon codes can be decoded by an algorithm requiring only $0(q \log^2 q)$ additions and multiplications in $GF(q)$.**

The fast Fourier transform has been used to obtain algorithms of low complexity for multiplying, dividing, and evaluating polynomials and for computing greatest common divisors [1]. When $q - 1$ has many small factors, a fast transform in $GF(q)$ has been suggested for evaluating syndromes of Reed–Solomon codes [2]. The computations are particularly convenient when q is a Fermat prime [3], and we shall simplify the presentation by referring to this particular case.

The fast Fourier transform is based on a factorization of binomials (or other polynomials of low weight) of the type

$$\begin{aligned} x^n - 1 &= (x^{n/2} - 1)(x^{n/2} + 1) \\ &= (x^{n/4} - 1)(x^{n/4} + 1)(x^{n/4} - \alpha^{n/4})(x^{n/4} + \alpha^{n/4}) \\ &= \cdots = \prod_{j=0}^{n-1} (x - \alpha^j) \end{aligned}$$

where α is a primitive nth root of unity. When $q = 2^{2^r} + 1$ is a prime, such a factorization is possible in $GF(q)$ for all n that are powers of 2. A polynomial $a(x)$ is transformed into the values $a(\alpha^j)$. Thus a polynomial can be evaluated for all nonzero values of its argument in $0(n \log n)$ operations. The inverse transform of the product $a(\alpha^j)b(\alpha^j)$ yields the product $a(x)b(x)$ modulo $(x^n - 1)$ in $0(n \log n)$ operations. When this algorithm is applied to Reed–Solomon codes of length n, it is an efficient method for encoding, calculating the syndromes, and computing the error locations and error values. When the error locator $\sigma(x)$ and error evaluator polynomial $\omega(x)$ [4] are known, the values of $\sigma(x)$ and $\omega(x)/\sigma'(x)$ are obtained through the transform.

An algorithm for solving the key equation based on Euclid's algorithm was suggested by Sugiyama *et al.* [5]. Starting from the syndrome polynomial $S(x)$ and x^{2t}, the remainder sequence r_j is computed until $\deg r_{k-1} \geq t$ and $\deg r_k \leq t - 1$. We suggest a modification of this approach based on an algorithm that computes the greatest common divisor in $0(n \log^2 n)$

Manuscript received May 30, 1975.
The author is with the Aalborg Universitets Center, Aalborg, Denmark.