#3

ERROR-FREE CODING

PETER ELIAS

TECHNICAL REPORT 285

SEPTEMBER 22, 1954

RESEARCH LABORATORY OF ELECTRONICS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

The Research Laboratory of Electronics is an interdepartmental laboratory of the Department of Electrical Engineering and the Department of Physics.

The research reported in this document was made possible in part by support extended the Massachusetts Institute of Technology, Research Laboratory of Electronics, jointly by the Army Signal Corps, the Navy Department (Office of Naval Research), and the Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA36-039 sc-42607, Project 132B; Department of the Army Project 3-99-12-022.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

RESEARCH LABORATORY OF ELECTRONICS

Technical Report 285 September 22, 1954

ERROR-FREE CODING

Peter Elias

Abstract

Some simple constructive procedures are given for coding sequences of symbols to be transmitted over noisy channels. A message encoded by such a process transmits a positive amount of information over the channel, with an error probability which the receiver may set to be as small as it pleases, without consulting the transmitter. The amount of information transmitted is less than the channel capacity, so the procedures are not ideal, but they are quite efficient for small error probabilities. It is shown that there exist codes of the same error-free character which transmit information at rates arbitrarily near the channel capacity.

INTRODUCTION

This report (1) describes constructive procedures for encoding messages to be sent over noisy channels so that they may be decoded with an arbitrarily low error rate. The procedures are a kind of iteration of simple error-correcting codes, such as those of Hamming (2) and Golay (3); any additional systematic codes that may be discovered, such as those discussed by Reed (4) and Muller (5), may be iterated in the same way.

The procedures are not ideal (1): the capacity of a noisy channel for the transmission of error-free information using such coding is smaller than information theory says it should be. However, the procedures do permit the transmission of error-free information at a positive rate. They also have these two properties:

(a) The codes are "systematic" in Hamming's sense: They are what Golay calls "symbol codes" rather than "message codes." That is, the transmitted symbols are divided into so-called "information symbols," and "check symbols." The customer who has a message to send supplies the "information symbols," which are transmitted unchanged. Periodically the coder at the transmitter computes some "check symbols," which are functions of past information symbols, and transmits them. The customer with a short message does not have to wait for a long block of symbols to accumulate before coding can proceed, as in the case of codebook coding, nor does the coder need a codebook memory containing all possible symbol sequences. The coder needs only a memory of the past information symbols it has transmitted and a quite simple computer.

(b) The error probability of the received messages is as low as the receiver cares to make it. If the coding process has been properly selected for a given noisy channel, the customer at the receiver can set the probability of error per decoded symbol (or the probability of error for the entire sequence of decoded symbols transmitted up to the present, or the equivocation of part or all of the decoded symbol sequence) at as low a value as he chooses. It will cost him more delay to get a more reliable message, but it will not be necessary to alter the coding and decoding procedure when he raises his standards, nor will it be necessary for less particular and more impatient customers using the same channel to put up with the additional delay. This is again unlike codebook processes, in which the codebook must be rewritten for all customers if any one of them raises his standards.

Perhaps the simplest way to indicate the basic behavior of such codes is to describe how one would work in a commercial telegraph system. A customer entering the telegraph office presents a sequence of symbols that is sent out immediately over a noisy channel to another office, which immediately reproduces the sequence, adds a note "the probability of error per symbol is 10^{-1}, but wait till tomorrow," and sends it off to the recipient. Next day the recipient receives a note saying "For 'sex' read 'six'. The probability of error per symbol is now 10^{-2}, but wait till next week." A week later the recipient gets another note: "For 'lather' read 'gather'. The probability of

error per symbol is now 10^{-4}, but wait till next April." This flow of notes continues, the error probability dropping rapidly from note to note, until the recipient gets tired of the whole business and tells the telegraph company to stop bothering him.

Since these coding procedures are derived by an iteration of simple error-correcting and detecting codes, their performance depends on what kind of code is iterated. For a binary channel with a small and symmetric error probability, the best choice among the available procedures is the Hamming-Golay single-error-correction, double-error-detection code developed by Hamming (2) for the binary case and extended by Golay (3) to the case of symbols selected from an alphabet of M different symbols, where M is any prime number. The analysis of the binary case will be presented in some detail and will be followed by some notes on diverse modifications and generalizations.

ITERATED HAMMING CODES

First-Order Check

Consider a noisy binary channel that transmits each second either a zero or a one, with a probability $(1 - p_o)$ that the symbol will be received as transmitted and a probability p_o that it will be received in error. Error probabilities for successive symbols are assumed to be statistically independent.

Let the receiver divide the received symbol sequence into consecutive blocks, each block consisting of N_1 consecutive symbols. Because of the assumed independence of successive transmission errors, the error distribution in the blocks will be binomial: there will be a probability

$$P(0) = (1 - p_o)^{N_1}$$

that no errors have occurred in a block, and a probability P(i)

$$P(i) = \frac{N_1!}{i!(N_1 - i)!} p_o^i (1 - p_o)^{N_1 - i} \qquad (1)$$

that exactly i errors have occurred.

If the expected number of errors per received block, $N_1 p_o$, is small, then the use of a Hamming error-correction code will produce an average number of errors per block, $N_1 p_1$, after error correction, which is smaller still. Thus p_1, the average probability of error per position after error correction, will be less than p_o. An exact computation of the extent of this reduction is complicated, but some inequalities are easily obtained.

The single-error-correction check digits of the Hamming code give the location of

any single error within the block of N_1 digits, permitting it to be corrected. If more errors have occurred, the check digits give a location which is usually not that of an incorrect digit, so that altering the digit in that location will usually cause one new error, and cannot cause more than one. The double-error-detection check digit tells the receiver whether an even or an odd number of errors has occurred. If an even number has occurred and an error location is indicated, the receiver does not make the indicated correction, and thus avoids what is very probably the addition of a new error.

The single-correction, double-detection code, therefore, will leave error-free blocks alone, will correct single errors, will not alter the number of errors when it is even, and may increase the number by at most one when it is odd and greater than one. This gives for the expected number of errors per block after checking

$$N_1 p_1 \leq \sum_{\text{even } i \geq 2}^{\leq N_1} iP(i) + \sum_{\text{odd } i \geq 3}^{\leq N_1} (i+1)\, P(i)$$

$$\leq P(2) + \sum_{i=3}^{N_1} (i+1)\, P(i)$$

$$\leq \sum_{i=0}^{N} (i+1)\, P(i) - P(0) - 2P(1) - P(2)$$

$$\leq 1 + N_1 p_o - P(0) - 2P(1) - P(2). \qquad (2)$$

Substituting the binomial error probabilities from Eq. 1, expanding, and collecting terms, gives, for $N_1 p_o \leq 3$,

$$N_1 p_1 \leq N_1 (N_1 - 1)\, p_o^2$$

$$p_1 \leq (N_1 - 1)\, p_o^2 < N_1 p_o^2 \qquad (3)$$

The error probability per position can therefore be reduced by making N_1 sufficiently small. The shortest code of this type requires $N_1 = 4$, and the inequality, Eq. 3, suggests that a reduction will therefore not be possible if $p_o \geq 1/3$. The fault is in the inequality, however, and not the code: for $N_1 = 4$ it is a simple majority-rule code that will always produce an improvement for any $p_o < 1/2$.

A Hamming single-correction, double-detection code uses C of the N positions in a block for checking purposes and the remaining N - C positions for the customer's symbols, where

$$C = \left[\log_2 (N-1) + 2\right] \qquad (4)$$

(Here and later, square brackets around a number denote the largest integer which is

less than or equal to the number enclosed. Logarithms will be taken to the base 2 unless otherwise specified.)

Higher Order Checks

After completing the first-order check, the receiver discards the C_1 check digits, leaving only the $N_1 - C_1$ checked information digits, with the reduced error probability p_1 per position. (It can be shown that the error probability after checking is the same for all N_1 positions in the block, so that discarding the check digits does not alter the error probability per position for the information digits.) Now some of these checked digits are made use of for further checking, again with a Hamming code. The receiver divides the checked digits into blocks of N_2; the C_2 checked check digits in each block enable, again, the correction of any single error in the block, although multiple errors may be increased by one in number. In order for the checking to reduce the expected number of errors per second-order block, however, it is necessary to select the locations of the N_2 symbols in the block with some care.

The simplest choice would be to take several consecutive first-order blocks of $N_1 - C_1$ adjacent checked information digits as a second-order block, but this is guaranteed not to work. For if there are any errors at all left in this group of digits after the first-order checking, there are certainly two or more, and the second-order check cannot correct them. In order for the error probability per place after the second-order check to satisfy the analog of Eq. 3, namely,

$$p_j \leqslant (N_j - 1)\, p_{j-1}^2 < N_j p_{j-1}^2 \tag{5}$$

it is necessary for the N_2 positions included in the second-order check to have statistically independent errors after the first check has been completed. This will be true if and only if each position was in a different block of N_1 adjacent symbols for the first-order check.

The simplest way to guarantee this independence is to put each group of $N_1 \times N_2$ successive symbols in a rectangular array, checking each row of N_1 symbols by means of C_1 check digits, and then checking each column of already checked symbols by means of C_2 check digits. The procedure is illustrated in Fig. 1. The transmitter sends the $N_1 - C_1$ information digits in the first row, computes the C_1 check digits and sends them, and proceeds to the next row. This process continues down through row $N_2 - C_2$. Then the transmitter computes the C_2 check digits for each column and writes them down in the last C_2 rows. It transmits one row at a time, using the first $N_1 - C_1$ of the positions in that row for the second-order check, and the last C_1 digits in the row for a first-order check of the second-order check digits.

After the second-order check, then, the inequality, Eq. 5, applies as before, and we have for p_2, the probability of error per position,

$$p_2 < N_2 p_1^2 < N_2 N_1^2 p_o^4 \tag{6}$$

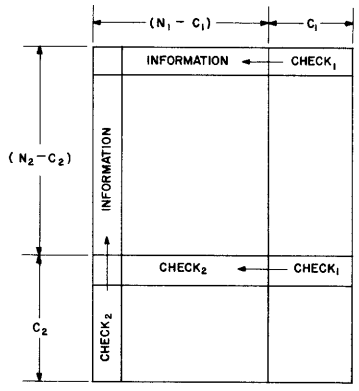

Fig. 1
Organization of first- and second-order check digits.

The N_3 digits to be checked by the third-order check may be taken from corresponding positions in each of N_3 different $N_1 \times N_2$ rectangles, the N_4 digits in a fourth-order block from corresponding positions in N_4 such collections of $N_1 \times N_2 \times N_3$ symbols each, and so on ad infinitum. At the k^{th} stage this gives (see ref. 5)

$$p_k < N_k^{2^0} \cdot N_{k-1}^{2^1} \cdots N_{k-j}^{2^j} \cdots N_1^{2^{k-1}} \cdot p_o^{2^k} \tag{7}$$

It is now necessary to show that not all of the channel is occupied, in the limit, with checking digits of one order or another so that some information can also get through. The fraction of symbols used for information at the first stage is $\left[1 - (C_1/N_1)\right]$. At the k^{th} stage, it is

$$F_k = \prod_1^k \left(1 - \frac{C_j}{N_j}\right) \tag{8}$$

It is now necessary to find a sequence of N_j for which p_k approaches zero and F_k does not, as k increases without bound. A convenient sequence is

$$N_1 = 2^n$$
$$N_j = 2^{j-1} N_1 = 2^{j+n-1} \tag{9}$$

This gives for p_k, from Eq. 7,

$$p_k < \left(N_1 \cdot 2^{k-1}\right)^{2^0} \cdots \left(N_1 \cdot 2^{k-j}\right)^{2^{j-1}} \cdots \left(N_1 \cdot 2^0\right)^{2^{k-1}} p_o^{2^k}$$
$$< \frac{1}{N_1} (2N_1 p_o)^{2^k} \cdot 2^{-(k+1)} \tag{10}$$

The right side of this expression approaches zero as k increases, for any $N_1 p_o \leq 1/2$. Thus the error probability can be made to vanish in the limit. Note that the inequality gives a much weaker kind of approach to zero for the threshold value $N_1 p_o = 1/2$ than for any smaller value of errors per first-order block.

For the same sequence of N_j, a lower bound on F_∞ can be computed. From Eqs. 8 and 4 we have

$$F_\infty = \prod_1^\infty \left(1 - \frac{C_j}{N_j}\right) = \prod_1^\infty \left(1 - \frac{\log_2 N_j + 1}{N_j}\right) = \prod_1^\infty \left(1 - \frac{j+n}{2^{j+n-1}}\right) \tag{11}$$

Let

$$\sigma_j = \frac{C_j}{N_j} \quad \text{and} \quad \sigma = \sum_1^\infty \sigma_j \tag{12}$$

Then σ_j is monotonic decreasing in j and is less than 1 for all constructible Hamming codes, that is, for $N_1 = 2^n \geq 4$. This makes it possible to write the following inequalities:

$$e^{-\sigma} > F_\infty > (1 - \sigma_1)^{\sigma/\sigma_1} > 1 - \sigma \tag{13}$$

Here the last term on the right is one of the Weierstrasse inequalities for an infinite product; the other terms are useful when $\sigma > 1$, and show that F_∞ is strictly positive for $\sigma_1 < 1$ and $\sigma < \infty$.

Evaluating σ in the present case gives

$$\sigma = \sum_{j=1}^\infty \frac{j+n}{2^{j+n-1}} = \frac{n+2}{2^{n-1}} = \frac{2 \log 4N_1}{N_1} \tag{14}$$

At threshold, that is, at $N_1 p_o = 1/2$, this gives

$$\sigma = 4p_o \log \frac{2}{p_o} < 4\left\{p_o \log \frac{1}{p_o} + (1 - p_o) \log \frac{1}{1 - p_o}\right\} = 4E \tag{15}$$

where E is the equivocation of the noisy channel. Thus for p_o small, from Eq. 13 we have

$$F_\infty > 1 - 4E \tag{16}$$

That is, under the specified conditions $\left(N_1 p_o = 1/2,\ N_1 = 2^n \geq 4\right)$ the number of check digits required is never more than four times the number that would be required for an ideal code, provided that an ideal code of the check-digit type exists, which is

not obvious (1). When E is > 1/4, the interior inequality in Eq. 13 shows that F_∞ is still positive.

Equivocation

Feinstein (4) has shown that it is possible to find ideal codes for which not only the probability of error, but the total equivocation, vanishes in the limit as longer and longer symbol sequences are used. This property is also true for the coding processes described here. This is a very important result in the case of codebook codes, where the message becomes infinite in the limit. For the codes under discussion here, it is a less important property, since any finite message can be received without an infinite lag, and its equivocation vanishes with the error probability per position.

The total number of binary digits checked by the k^{th} checking stage is

$$M_k = \prod_1^k N_j \tag{17}$$

Of these, $F_k M_k$ are information digits and the remainder are checks. Using the values of Eq. 9 for the N_j, we have

$$M_k = N_1^k \, 2^{\frac{k(k-1)}{2}} \tag{18}$$

The bound of Eq. 10 limits the probability of error per position. Multiplying this by M_k gives a bound on the mean number of errors per M_k digits, which is also a bound on the fraction of sequences of M_k digits which are in error after checking – a gross bound, since actually any such sequence which is in error must have many errors, and not just one. Thus for Q_k, the probability that a checked group of M_k digits is in error, we have

$$Q_k < p_k M_k \leq \frac{1}{4}\left(\frac{N_1}{2}\right)^{k-1} (2N_1 p_o)^{2^k} \, 2^{\frac{k(k-1)}{2}} \tag{19}$$

At threshold ($N_1 p_o = 1/2$) this inequality does not guarantee convergence, but for $N_1 p_o < 1/2$, Q_k certainly approaches zero as k increases.

The equivocation E_k per sequence of M_k terms is bounded by the value it would have if any error in a block made all possible symbol sequences equally likely at the receiver, that is,

$$E_k < Q_k \log \frac{1}{Q_k} + (1 - Q_k) \log \frac{1}{1 - Q_k} + Q_k M_k \tag{20}$$

Again, at threshold, convergence is not guaranteed, but for $N_1 p_o < 1/2$, E_k, the absolute equivocation of the block, will also vanish as k increases.

Distance Properties

At the k^{th} stage of this coding process, a sequence of M_k binary digits has been selected as a message. Because the check-digit values are determined by the information digit values, there are only $2^{F_k M_k}$ possible message sequences rather than 2^{M_k}. Any two of these possible messages will have a "distance" from one another, defined as the number of positions in which they have different binary symbols; the smallest such distance will be 4^k for the iterated single-correction, double-detection code. This means that by using this set of codes with a codebook, any set of errors less than one-half of the minimum distance in number can be corrected by choosing as the transmitted message the message point nearest to the received sequence.

It is easy to see that for the coding procedure just described this error-correction capability will not be realized. Any set of 2^k errors which are at the corners of a k-dimensional cube in the k-dimensional rectangle of symbol positions will not be corrected by this process, since each check will merely indicate a double error that it cannot correct. By inspecting any two of the sets of check digits at once, these errors could be located, but they will not have been corrected by the process described above. The effective minimum of the maximum number of errors that will be corrected is therefore $2^k - 1$ rather than $2^{2k-1} - 1$.

This shows a loss of error-correction capability because of the strictly sequential use of the checking information. Without going to the extreme memory requirement of codebook techniques, a portion of this loss may be recouped by not throwing the low-order check digits away but using them to recheck after higher order checking has been done. This does not increase the maximum number of errors for which correction is always guaranteed, but it does reduce the average error probability at each stage; the exact amount of this reduction is, unfortunately, difficult to compute. This behavior, however, points up a significant feature of the coding process. If the maximum number of errors for which correction is always guaranteed were the maximum number of errors for which correction was ever guaranteed, the procedure could not transmit information at a nonzero rate; that is, the minimum distance properties of the code are inadequate for the job. It is average error-correction capability that makes transmission at a nonzero rate possible.

The Poisson Limit

Much of this analysis has assumed that $N_j = 2^{j-1} N_1$, and part of it has further assumed that $N_1 = 2^n$. However, any series of N_j which increases rapidly enough so that σ is finite will lead to a coding process that is error-free for sufficiently small values of $N_1 p_o$. In particular, it is possible to use any other approximately geometric series for which

$$N_j \approx b^{j-1} N_1, \quad b > 1 \tag{21}$$

The approximation is necessary if b is not an integer. The expression for p_k analogous to Eq. 10 is then

$$p_k < \frac{1}{N_1} (bN_1p_o)^{2^k} b^{-(k+1)} \tag{22}$$

with a threshold at $N_1p_o = 1/b$. The value of σ can also be bounded for this series. At threshold, the bound corresponding to Eq. 15 is

$$\sigma \leqslant \frac{b^2p_o}{b-1} \log\left(\frac{4}{p_o\, b^{\frac{b-2}{b-1}}}\right) \tag{23}$$

Again, for N_1p_o below threshold, Q_k and E_k approach zero as k increases.

For very small p_o, the value of b that minimizes σ is b = 2. This leads to the maximum value of F_∞ given by Eq. 16. However, for very small p_o, N_1 may be made very large. The distribution of errors in the blocks then approaches the Poisson distribution, for which the probability that just i errors have occurred in a block is

$$P(i) = e^{-N_1p_o} \cdot \frac{(N_1p_o)^i}{i!} \tag{24}$$

This equation may be used to derive an iterative inequality on the mean number of errors per block after single-detection, double-correction coding.

$$N_jp_j \leqslant 1 + N_jp_{j-1} - e^{-N_jp_{j-1}} \left\{1 + 2N_jp_{j-1} + \frac{(N_jp_{j-1})^2}{2!} + \frac{(N_jp_{j-1})^4}{4!} + \ldots\right\}$$

$$\leqslant 1 - N_jp_{j-1}\left(2e^{-N_jp_{j-1}} - 1\right) - \frac{1}{2}\left(1 - e^{-2N_jp_{j-1}}\right) \tag{25}$$

Keeping N_jp_j constant gives the geometric series (21) for N_j. A joint selection of N_1p_o and b for the minimization of the bound on σ gives $N_1p_o \approx 0.75$, $b \approx 1.75$, and an effective channel capacity

$$F_\infty \approx 1 - 3.11E \tag{26}$$

where E is the equivocation of the binary channel. This is an improvement over Eq. 16.

ITERATION OF OTHER CODES

The analysis in the preceding sections has dealt only with iteration of the Hamming single-error-correction, double-error-detection code. Other kinds of codes may also be iterated; nor is it necessary to use the same type of code at each stage in the iterative process. The only requirement is that each code except the first be of the check-digit, or systematic, type, so that its check digits may be computed on the basis of the preceding information digits and added on to the message.

First, the final parity check digit of a Hamming code may be omitted, destroying the double-detection feature of the code. This leads to the inequality

$$p_j \leq \frac{3}{2}(N_j - 1)\, p_{j-1}^2 < \frac{3}{2} N_j p_{j-1}^2 \tag{27}$$

in place of Eq. 5. Iterating this code alone gives a bound on σ that is only slightly smaller than that given in Eq. 15, but the threshold becomes $N_1 p_o = 1/3$ rather than $N_1 p_o = 1/2$, and the effective channel capacity for small p_o is bounded by

$$F_\infty > 1 - 6E \tag{28}$$

where E is the equivocation of the binary channel.

Second, the Golay (3) analogs to both kinds of Hamming code may be constructed, for M-ary channels, where M is a prime number. If there is a probability $(1 - p_o)$ that any symbol will be received correctly, and if the consecutive errors are statistically independent, the results of the binary case carry over quite directly. The inequalities of Eqs. 5 and 27 still hold for the two kinds of codes, since the errors as a whole are still binomially distributed in blocks. At threshold, the inequalities of Eqs. 16 and 28 still hold for the effective channel capacity, where E is now the equivocation of a symmetrical M-ary channel; that is, of a channel in which the probability of an error taking any given symbol into any other different symbol is $p_o/(M-1)$. The result, Eq. 26, for the Poisson limit also applies, with the same interpretation of E.

Third, the Reed (5) -Muller (6) codes may be treated as check-digit codes, and may be iterated to give an error-proof system. For these codes, the average error-reduction capability is not known; only the minimum distance is known. Certain of the codes, such as the triple-correction quadruple-detection code for blocks of 32 binary symbols, might provide a good starting point for an iteration that proceeds by iteration of Hamming codes. The Golay triple-correction, quadruple-detection code for blocks of 24 symbols might be used in the same way. It will take considerable computation to evaluate such mixed iteration schemes.

It is not, at present, profitable to use the Reed-Muller codes for later stages in the iteration. The reason is that an efficient triple-correction, quadruple-detection code should require about C = 2 log N check digits for a block of length N. The Reed-Muller codes require about C = 1 + log N + (1/2) log N (log N-1) check digits for this purpose. For large N, therefore, the effective channel capacity is reduced by the large number of check digits required. There is a similar inefficiency in the Reed-Muller codes with greater error-correction capabilities, which might be removed if the average error-correction capabilities of these codes were known.

NONRECTANGULAR ITERATION

The problem of assuring statistical independence among the N_k digits checked by a k^{th} order check, so that the inequality of Eq. 5 derived on the basis of statistical independence can be used as an iterative inequality, was solved above by what might be called

rectangular iteration. Each of the N_k digit positions in a check group are selected from a different sequence of M_{k-1} consecutive symbols. Thus until the k^{th} order checking has been carried out, no two of them have been associated by lower order checking procedures in any way. This iteration solves the problem, but it makes M_k a function that grows very rapidly with k. When N_j is the geometric series of Eq. 21, then

$$M_k \approx N_1^k \, b^{\frac{1}{2}k(k-1)} \tag{29}$$

This means that p_k, Q_k, and E_k decrease quite rapidly as functions of k, but much more slowly as functions of the length of the message M_k, or its information content $F_k M_k$.

Roughly speaking, if $H_k = F_k M_k$ is the total number of information digits transmitted at the k^{th} stage,

$$p_k \approx Ae^{-2^{a(\log H_k)^{1/2}}}, \quad A > 0, \quad a > 0 \tag{30}$$

This is a much slower decrease of error probability than Feinstein's result (4) which is

$$p_k \approx Be^{-b \cdot 2^{\log H_k}} = Be^{-bH_k}, \quad B > 0, \quad b > 0 \tag{31}$$

A less stringent requirement on the choice of digits checked in a single group is that no two of them have been together in any lower order check group. This requires that there be at least N_k different groups of order k - 1 from which to select digit positions. Thus

$$M_k \geq N_k N_{k-1} \tag{32}$$

If it is possible to approximate equality in Eq. 32, and if the statistical dependence so introduced does not seriously weaken the inequality, Eq. 5, it might be possible to get the result

$$p_k \approx De^{-2^{d \log H_k}}, \quad D > 0, \quad d > 0 \tag{33}$$

which is closer to Feinstein's result.

CONCLUSION

From a practical point of view, this coding procedure has much to recommend it. A question of both theoretical and practical interest is the extent to which the convenience associated with a computable and error-free code is compatible with ideal coding, or the smallest price that must be paid for the convenience if the two are incompatible. One point is clear: the existence of the error-free process, despite its lack of ideality, puts the burden of efficient coding on the first stage of the coding process. For if a coding

process succeeds in reducing the equivocation in a received message to some small but positive value E, the remaining errors may always be eliminated at a cost of 4E (or 3.11 E) in channel capacity: an error-proof termination is available, at a price, to take care of the residual errors left by any other error-correcting scheme.

This means that a restatement of the second coding theorem (3, 8) is possible. Namely, it is possible to transmit information over a noisy binary channel at a rate arbitrarily close to the channel capacity with an error probability per symbol which is as low as the receiver cares to make it. This can be done, for example, by starting with a Shannon or Feinstein code and terminating with an error-free code. Using check-symbol coding at a rate arbitrarily close to the channel capacity as an initial step makes this procedure somewhat more computable, but it is still much less so than the iterated Hamming code. This and related points will be discussed in a future report (1).

Acknowledgment

The iterative approach used in this paper was suggested by a comment of Dr. Victor H. Yngve, of the Research Laboratory of Electronics, M.I.T., on the fact that redundancy in language was added at many different levels, a point that he discusses in reference 7.

References

1. The material in this report was presented at the 1954 Symposium on Information Theory held at M.I.T., September 15-17, 1954, and was published in substantially this form in the Transactions of that meeting, pp. 29-37, publication PGIT-4 of the I.R.E. Professional Group on Information Theory. Since that time a proof has been found that there exist check-symbol codes that transmit information at rates arbitrarily close to the channel capacity. This work will be published as Technical Report 291. It answers some questions raised in the present report, the conclusion of which has been rewritten since the Symposium.
2. R. W. Hamming, Error detecting and error correcting codes, Bell System Tech. J. 29, 147-160 (1950).
3. M. J. E. Golay, Notes on digital coding, Proc. I.R.E. 37, 657 (1949).
4. A. Feinstein, A new basic theorem of information theory, Technical Report No. 282, Research Laboratory of Electronics, M.I.T. (1954).
5. I. S. Reed, A class of multiple-error-correcting codes and the decoding scheme, Technical Report No. 44, Lincoln Laboratory, M.I.T. (1953).
6. D. E. Muller, Metric properties of Boolean algebra and their applications to switching circuits, Report No. 46, Digital Computer Laboratory, University of Illinois (1953).
7. V. H. Yngve, Language as an error-correcting code, Quarterly Progress Report, Research Laboratory of Electronics, M.I.T., April 15, 1954, pp. 73-74.
8. C. E. Shannon, A mathematical theory of communication, Bell System Tech. J. 27, 379-423, 623-656 (1948).