ALGEBRAIC DECODING FOR A BINARY ERASURE CHANNEL

M. A. EPSTEIN

LOAN COPY ONLY

TECHNICAL REPORT 340
MARCH 14, 1958

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 U. S. Army (Signal Corps), the U. S. Navy (Office of Naval Research), and the U. S. Air Force (Office of Scientific Research, Air Research and Development Command), under Signal Corps Contract DA36-039-sc-54637, Department of the Army Task 3-99-06-108 and Project 3-99-06-100.
ALGEBRAIC DECODING FOR A BINARY ERASURE CHANNEL*

M. A. Epstein

Department of Electrical Engineering, Research Laboratory of Electronics, and Lincoln Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

Summary — This paper presents an optimum decoding procedure for parity check codes that have been transmitted through a binary erasure channel. The erased digits are decoded by means of modulo 2 equations generated by the received message and the original parity check equations. Most previous decoding procedures required a number of computations that grew exponentially with code length. At best, the required number of computations grew as a polynomial of code length. The decoding procedure for convolution parity check codes presented here will decode with an average number of computations per digit that is bounded by a finite number, which is independent of code length, for any rate less than capacity.

Introduction

The capacity of a noisy channel is defined as the maximum rate at which information can be transmitted through the channel. Shannon's fundamental theorem on noisy channels states that for a memoryless channel there exist codes that transmit at any rate below capacity with an arbitrarily low probability of error.

A number of papers published since Shannon's original work have shown that the probability of error is significantly dependent on the code length. The results can be summarized as follows. For a given channel and a given rate less than capacity, the probability of error for the optimum code of length n decreases in exponential fashion with increasing code length. Thus, the probability of error for the optimum code of length n is bounded by $c(n)e^{-an}$, where $a$ is a positive constant and $c(n)$ is a function of n that can be treated as a constant for large n. The exponential constant $a$ is small for rates near capacity and is larger for rates much less than capacity. Hence, for a given probability of error, we transmit at rates close to capacity only at the expense of increased code length.

The proof of Shannon's theorem and the proofs of the dependence of the probability of error on code length are existence proofs; that is, they describe the properties of optimum codes without indicating a practical method of finding such codes. But the proofs do show that there exist optimum codes with certain properties and that most codes behave in a fashion similar to that of the optimum codes.

For a given rate of transmission, the probability of error decreases exponentially with code length. However, if the encoding at the transmitter or the decoding at the receiver is done by listing all possible situations, the effort required grows exponentially with code length. Thus, if list coding or decoding is used, the probability of error is decreased only at the expense of large increases in the terminal equipment or in the number of operations needed to code and decode.

Other coding and decoding procedures, which require less average computation, have been found for the binary symmetric channel and the binary
erasure channel (see Fig. 1). The coding procedure was found first. Parity check codes can be encoded with an amount of computation per digit that is proportional to code length. Elias proved that such codes are almost optimum when applied to the binary symmetric channel or the binary erasure channel.

Subsequently, Wozencraft developed a sequential procedure for decoding a binary symmetric channel. For rates not too close to capacity, there are strong indications that the average amount of computation per digit for this procedure increases less rapidly than the square of the code length.

Decoding Procedure for Binary Erasure Channels

We shall describe a decoding procedure for convolution parity check codes that have been transmitted through a binary erasure channel.

The binary erasure channel (hereafter abbreviated as BEC) is a channel with two input symbols - 0, 1 - and three output symbols - 0, X, 1. If a zero or a one is transmitted the probability that the same symbol is received is q and the probability that an X is received is p = 1 - q. There is zero probability that a zero is sent and a one received or that a one is sent and a zero received. The output at a given time depends only on the present input and not on other inputs or outputs. The capacity of this channel is q, the probability that a digit is unerased in transmission. This channel is called an erasure channel, since transmission through the channel is equivalent to erasing input digits at random and replacing the erased digits by X (see Fig. 1b).

The average amount of computation per digit required by our decoding procedure is bounded independently of code length for any given rate and channel capacity. Thus, the best convolution code of any given length requires no more average computation per digit than the bound.

The erased digits are decoded by means of equations generated from the parity check equations for the code. As in Wozencraft's procedure for the binary symmetric channel, the digits are decoded sequentially. This decoding procedure is optimum in the sense that, given any particular parity check code of this form, no other decoding procedure has a lower probability of error.

The bound on the amount of computation is a function of the channel and the rate of transmission of information. At rates close to capacity the bound is large; at rates much less than capacity, it is correspondingly small. Therefore, as might be expected, we can transmit at rates close to capacity only at the expense of increased computation.

The BEC can be thought of as an approximation to certain physical situations, such as a fading teletype channel. It remains to be seen if the procedure described in this paper can be applied not only to the BEC model but also to the physical situations that are approximated by the BEC model.

Block Parity Check Codes

The probability of error can be reduced towards zero only if the rate of transmission is less than channel capacity. One transmits at a given rate by using a coding procedure that selects for each sequence of information digits a longer sequence of digits to be transmitted through the channel. We shall assume that the information digits take on the values zero and one with probability one-half and that they are statistically independent of each other. The rate of transmission R is then defined as the ratio of the number of information digits to the number of transmitted digits.

A parity check code is a code in which the information digits are inserted into the transmitted message, and the remaining digits of the message, called check digits, are determined by parity checks on the information digits. We shall describe two types of parity check codes: block codes and convolution codes.

In a block code of length n, each group of nR information digits is encoded into a message with n digits. The first nR digits of the transmitted message are the information digits, and are denoted by the symbols I₁, I₂, ..., IₙR. The last n(1-R) digits are the check digits. These are
denoted by the symbols \( C_1, C_2, \ldots, C_{n(1-R)} \). The value of each check digit is determined by a parity check on a subset of the information digits. The relation of the check digits to the information digits is described by the following modulo two sums and the matrix \( A \), where each element of the matrix is either a zero or a one (see Fig. 2a). (A modulo two sum is zero if the usual sum is even and one if the usual sum is odd.)

\[
C_i = \sum_{j=1}^{nR} a_{ij} I_j \quad \text{(mod 2)}
\]

where \( i = 1, 2, \ldots, n(1-R) \) (1)

Decoding Block Codes

When a coded message is transmitted through a BEC, erasures occur at random. Usually, some of the erased digits will be information digits and some will be check digits. The decoding problem is that of finding the transmitted message when the received message is known.

Because 0-to-1 and 1-to-0 transitions never occur in the BEC, it is easy to see whether or not a given received message could have come from a given transmitted message. If the given transmitted message has zeros (ones) where the received message has zeros (ones), the received message could have come from the transmitted message. However, if the possible transmitted message has a zero where the received message has a one or a one where the received message has a zero, the received message could not possibly have come from the given transmitted message.

In principle, the received message could be compared with all \( 2^{nR} \) possible transmitted messages. Clearly, the received message will agree in its unerased digits with the actual transmitted message and we can ignore the possibility that the received message agrees with none of the possible transmitted messages. If the received message agrees with only one possible transmitted message then we know that this is the transmitted message. Thus, we can decode correctly. If the received message agrees with more than one possible transmitted message then, to decode optimally, we must choose the a posteriori most probable message. We shall assume that all of the possible transmitted messages are equally probable a priori. Therefore, all possible messages that could have given the received message are equally probable a posteriori and it does not matter which one of these is chosen.

Although the decoding method described above is simple in principle, in practice it requires too much computation or too much memory. (If \( n = 100, R = 0.5 \), then \( 2^{nR} \approx 10^{15} \).) The reason that so much computation or memory is required is that the code is described by listing all possible transmitted messages, and hence the list grows exponentially with increasing code length.

If the code is described by its parity check matrix, we can decode in the following manner with much less computation. We substitute the values of the unerased digits into each parity check equation to form a constant term, and consider the erased digits as the unknowns \( X_1, X_2, \ldots, X_r \), where \( r \) is the number of erased digits. Thus, we get \( n(1-R) \) modulo two equations in the values of the erased digits (see Fig. 3). Because of the properties of modulo two arithmetic, we can attempt to solve these equations by means of a modulo two analogue of the Gauss-Jordan procedure. This procedure is described in Appendix I. The basic idea of this procedure is that we can eliminate each unknown from all but one equation until a solution is found.

If the transmitted message is uniquely determined, the equations have a unique solution. If there are several possible transmitted messages that could have given the received message, the same ambiguity appears in the solution of the equations. Elias has shown that the probability of ambiguity and the probability of error both decrease exponentially with increasing code length. Bounds on these probabilities are described in Appendix II. It is shown in Appendix I that the maximum number of computations required to decode a given received message is bounded by \( \frac{1}{2} n^3 + n^2 \).
The list decoding method and the equation decoding method both perform in optimum fashion and obtain the same solutions. The difference is that the amount of work required to decode a block code by the list decoding method is proportional to $2^{nR}$, while that for the equation decoding method is proportional to $n^3$.

**Convolution Codes**

In a convolution code the information digits are an infinite sequence of digits, and check digits are interleaved with the information digits in some pattern such as information digit, check digit, information digit, check digit, and so on. Each check digit is determined by a parity check on a subset of the preceding $n-1$ digits. For rates $R \geq 1/2$, each check digit checks a fixed pattern of the preceding information digits; for rates $R < 1/2$, each information digit is checked by a fixed pattern of the succeeding check digits (see Fig. 2b).

The value of each check digit is determined by Eq. 2 and matrix $B$, where each element of the matrix is either a zero or a one. The $j^{th}$ transmitted digit is indicated by $d_j$, and the position of the $i^{th}$ check digit by $p(i)$.

$$\sum_{j=p(i)-n+1}^{p(i)} b_{ij} d_j = 0 \pmod{2}$$

where $i = 1, 2, 3, \ldots$ (2)

A convolution code may be used for a very long (infinite) sequence of digits, but the significant length for the error probability is $n$, since this is the largest span of digits related by any one parity check equation. No more than $n$ digits are needed to describe the parity check matrix $B$, since one fixed pattern is used to describe the relation of the information digits to the check digits.

There are other types of parity check codes which behave like convolution codes. Appendix III contains a description of some of these code types.

**Sequential Decoding for Convolution Codes**

In contrast to the block decoding procedure, which decoded all the digits in a block simultaneously, the decoding procedure for convolution codes decodes the received digits sequentially. Thus, digit one of the convolution code is decoded first, then digit two, then digit three, and so on. If the digit being decoded is unerased there is no difficulty because the received digit is the same as the transmitted digit. If the digit being decoded is erased, it is decoded by means of the parity check equations.

The idea of sequential decoding has been used previously by Elias and Wozencraft. The probability of error per digit for this procedure decreases exponentially with code length, as described in Appendix II. However, the average amount of computation per digit is bounded independently of code length.

The sequential procedure inherently assumes that when a digit is being decoded, the previous digits are known. Thus, certain complications arise in the analysis from the fact that some digits cannot be decoded by means of the parity check equations. To simplify the analysis, we will treat the problem of decoding digit $m$ without considering the possibility that the previous digits could not be decoded. This is equivalent to assuming that in the rare cases when a digit cannot be decoded by means of the coding equations, the receiver asks the transmitter for the digit by means of a special noiseless feedback channel, and the transmitter sends the receiver the digit by means of a noiseless feedforward channel parallel to the BEC. Later, this artificial situation will be used to describe the more realistic situation in which no noiseless feedback channels are assumed, and the convolution coding procedure is used for some long but finite code length.

The problem of decoding a complete received message has been divided, by means of the sequential decoding procedure, into a sequence of problems each of which concerns a single digit. This division is useful because the problem
of decoding one digit is the same as the problem of decoding any other digit, as we now prove. The convolution code in Fig. 4 will be used as an example of a convolution coding matrix.

Let us consider the problem of decoding digit \( m \), \( m = 1, 2, \ldots \). If digit \( m \) is not erased, the problem is trivial. If digit \( m \) is erased, it is decoded by means of the parity check equations and those digits that are already known at this stage of the procedure. Since digits are decoded in sequence, digits 1, 2, \ldots, \( m-2, m-1 \) are known when digit \( m \) is being decoded. Furthermore, since the solubility of a set of equations is independent of the coefficients of the known digits, the first \( m-1 \) columns of the coding matrix can be ignored. Likewise, those equations whose check digits precede digit \( m \) have nonzero coefficients only for digits 1, 2, \ldots, \( m-1 \); therefore these equations are of no value because they are relations among known digits only. Figure 4b contains the convolution matrix that is formed from the matrix in Fig. 4a by the removal of the first \( m-1 \) columns and the useless equations for the case of \( m = 5 \). Comparing Fig. 4a and Fig. 4b, we can see that, as a general result, the reduction of the matrix for any step \( m \) leaves a matrix of the same character as the original matrix. (It is assumed that digit \( m \) is not a check digit; check digits are simply related to past digits by the parity check equations.)

In addition to the similarity of the reduced matrix at step \( m \) to the original matrix, the digits \( m, m+1, \ldots \) are unknown at random with probability \( p \) at step \( m \), just as digits 1, 2, \ldots are unknown with probability \( p \) at step 1. Thus the problem of solving each digit in turn is identical with the problem of solving digit 1, and all further discussion will concern the solution of digit 1.

**Decoding Digit 1**

It can be seen from an examination of Fig. 4 that the first \( n(1-R) \) equations can check digit 1, and therefore they are clearly significant for the solution of digit 1. The equations after the first \( n(1-R) \) equations cannot check digit 1. At present, it is not known how much the probability of error can be decreased by the use of these equations. Hence, the following procedure will use the first \( n(1-R) \) equations only. (It is clear that the use of the equations after equation \( n(1-R) \) can only decrease the probability of error.)

Digit 1 is decoded, as follows, in a number of steps. If digit 1 is not erased, the decoding procedure is ended. If digit 1 is erased, an attempt is made to solve digit 1 by using the first equation. If digit 1 is solved, the procedure is ended; if not, we proceed to step 2. In step 2 an attempt is made to solve for digit 1 by using the first two equations. If digit 1 is solved, the procedure is ended; if not, we proceed to step 3. In step m, an attempt is made to solve digit 1 by using the first m equations. If digit 1 is solved, the procedure is ended; if not, we continue to step \( m+1 \). If digit 1 is not decoded by the end of step \( n(1-R) \), the decoding procedure ends, and the special noiseless channels previously assumed are used to determine the digit. The attempts to decode digit 1 at each step use the method described in Appendix I.

**Analysis of the Sequential Decoding Procedure**

**Probability of Ambiguity**

The probability that step \( m \) in the decoding procedure for digit 1 is reached equals the probability that digit 1 is ambiguous at the end of step \( m-1 \) (i.e., that digit 1 is not solved in steps 1, 2, \ldots, \( m-1 \)). Elias has calculated upper bounds on this probability. The bounds apply both to the probability of ambiguity per block in a random block code and to the probability of ambiguity per digit in a random convolution code. Denoting the probability of ambiguity by \( Q \), the form of the bounds for large \( n \) is

\[
Q \leq c(n) e^{-an} \tag{3}
\]

where "\( a \)" is a positive constant determined by the rate of transmission and the channel capacity, and \( c(n) \) is a slowly varying function of \( n \) that can be treated as a constant for large \( n \). A complete description of these bounds is given in Appendix II.

For large \( n \), the main behavior of the bounds is determined by the exponential constant "\( a \)". If the rate of transmission \( R \) is close to channel capacity \( C \), "\( a \)" is small and is approximately equal to...
(C-R)^2/2C(1-C). (For the BEC, channel capacity is equal to q, the probability that a digit is unerased in transmission.) "a" increases monotonically with decreasing rate, and at zero rate "a" = ln(2/(1+p)).

It can be seen from an examination of Fig. 4 that the first m-1 equations span (m-1)/(1-R) digits. Thus, the probability of ambiguity of digit 1 after step m-1 is that of a convolution code of length (m-1)/(1-R) and rate R. Defining Q(m) as the probability of not solving digit 1 by the end of step m, we have

Q(m) = 1 - Q(m-1)

Computation Bound for One Digit

We now evaluate an upper bound on the average amount of computation needed to decode the first digit. This computation bound is evaluated, for reasons of analytical convenience, for a code chosen at random from the ensemble of all convolution codes with the given length and rate of transmission. Since we know that there exists at least one code for which the average amount of computation for the first digit is less than that for a code chosen at random, there must exist at least one particular code that requires an average amount of computation less than the bound.

In the decoding procedure described above, the first digit of a convolution code is decoded in a number of steps. The number of steps is a function of the coding matrix and the pattern of erasures. Usually more computation is required for step m+1 than for step m, but the probability of reaching step m+1 is less than the probability of reaching step m.

At step m, we can use the computation made in step m-1. In step m-1, we attempted to solve for digit 1 by means of the first m-1 equations, and thus the first m-1 equations are in standard form at the start of step m. At step m then, we merely need to make the computation for equation m. This is bounded by 2m^2/(1-R) as proved in Appendix 1.

The average number of computations in step m for a random code is equal to the product of the probability of reaching step m and the average number of computations in step m. Since the average number of computations in step m is bounded by 2m^2/(1-R) and Q(m-1) is the probability of reaching step m, then

The average number of computations in a random code = 2Q(m-1) m^2

for step m.

Since no more than n(1-R) steps are ever used in an attempt to decode digit 1, the average number of computations used in a random code to decode digit 1 is equal to the sum from 1 to n(1-R) of the average number of computations in a random code for step m.

The average number of computations used in a random code to decode digit 1

\[ \sum_{m=1}^{n(1-R)} 2Q(m-1) m^2 \]

Since Q(m) decreases exponentially with increasing m, this sum approaches a limit as n approaches infinity. Therefore W, defined by Eq. 7, is a bound on the average number of computations needed to decode digit 1 in a convolution code of arbitrary length.

A bound on the average number of computations needed to decode digit 1

\[ W = \sum_{m=1}^{\infty} 2Q(m-1) m^2 \]

The bound W is a function of rate and capacity. At rates near capacity, the probability of ambiguity Q(m) decreases slowly with increasing m, and W is very large. At rates close to zero, the probability of ambiguity decreases quickly, and W is small.

If the bounds on the probability of ambiguity previously described are used to bound Q(m), a numerical bound can be found on the average amount of computation needed for a convolution code of rate R. For certain rates and channel capacities, these numerical bounds may overestimate, by a factor of 100 or more, the average computation required. However, these bounds do follow the general behavior of the computation requirements as a function of channel capacity and rate of transmission. Figure 5 contains a graph of the numerical bounds for various rates of transmission and channel capacities. The graph illustrates how the average computation
requirements increase with increasing rate for a fixed probability of erasure.

**Behavior of the Sequential Procedure**

The decoding procedure described above can be easily evaluated under the assumption that digits not decoded by the use of all \(n(1-R)\) parity check equations are decoded by means of the noiseless feedback channel. Since the problem of decoding digit 1 is identical with the problem of decoding any information digit, \(W\) is not only a bound on the number of computations needed by a random code to decode digit 1 but also on the number of computations needed to decode any other information digit. (Parity check digits are determined immediately by the previous digits.) This fact implies that, on the average, a random code needs no more than \(NW\) computations to decode the first \(N\) information digits.

The probability of not solving any information digit is \(Q(n-nR)\). This implies that the average number of digits that cannot be solved by the parity equations in the first \(N\) information digits equals \(NQ(n-nR)\).

The initial assumption of noiseless channels is, of course, unrealistic; it is better to treat a BEC channel without assuming any noiseless channels. A possible procedure is to use a convolution code of length \(n\), which is truncated after the first \(N\) information digits. If one truncates completely after the first \(N\) information digits, the last few information digits have a higher probability of error than the other digits because they are checked by fewer check digits. This can be corrected by adding \(n(1-R)\) check digits to the code so that all the information digits have an equal number of checks (see Fig. 6). Thus, we transmit in blocks of approximately \((N/R) + n(1-R)\) digits containing \(N\) information digits. (This value is approximate because the ratio of the number of information digits to the total number of digits up to information digit \(N\) may differ slightly from \(R\).)

The sequential procedure is used to decode. If a digit cannot be decoded by the parity equations we do not try to decode further, and the digits after a digit that cannot be decoded are also lost. To simplify the following discussion, we shall assume that if one or more digits in a block cannot be decoded the worst possible result occurs, and all digits of the block are lost.

We can evaluate bounds on the probability of losing a block and the average amount of computation per block for this more realistic situation by comparing it with the unrealistic situation first assumed. If a digit cannot be decoded in the second situation, it would be decoded in the first situation by means of the noiseless channels. The average number of times that the noiseless channels are used for the first \(N\) information digits equals \(NQ(n-nR)\), and thus the probability of losing a block by the more realistic procedure is bounded by \(NQ(n-nR)\).

The average amount of computation for the first \(N\) digits decoded by the first procedure has been bounded by \(NW\). If the first \(N\) information digits can be decoded by the parity check equations, both procedures perform the same computation. Otherwise, the second procedure stops after losing the first undecodable digit, while the first procedure performs all of the computation made by the second procedure, and then makes some more on the digits after the first undecodable digit.

Accordingly, \(NW\) also bounds the amount of computation done by the second procedure.

**Equipment Requirements**

The equipment needed to implement the more realistic procedure can be easily described. At the transmitter, the values of the parity check digits must be calculated. Since each information digit can be checked by \(n(1-R)\) check digits, the calculation required per information digit is proportional to code length. Also, \(nR\) information digits must be stored at one time so that the parity check digits can be calculated, and a fixed memory of no more than \(n\) digits is needed to remember the fixed pattern that describes the convolution matrix.

The receiver must evaluate the constant terms in the parity check equations. This evaluation is essentially the same as the evaluation of the parity check digits and requires the same equipment. The receiver will also need to solve the parity check
equations. For this, the receiver will need a computational speed equal to or greater than the average amount of computation needed to solve the equations.

At a fixed rate and channel capacity, the amount of computation required either by the evaluation of the parity check digits in coding or by the evaluation of the constants of the parity check equations in decoding is proportional to code length. Thus, for very long code lengths these evaluations require much computation. On the other hand, the average amount of computation required to solve the parity check equations for one digit is bounded independently of code length. Hence, for very long code lengths, most of the computation consists of the coding computation and the evaluation of the constants in the parity check equations.

In practice, in order to handle peak computational requirements, it is necessary to have more computational capacity than the average speed needed to solve the equations. The peak delay in decoding will be approximately equal to the peak computational requirements divided by the available rate of computation. Any digits that arrive during a period when the decoder is decoding previous digits must be stored. Thus it is only at the expense of added memory that the decoder can have a rate of computation close to the average computation rate. This problem can be treated by the standard techniques of waiting-line theory.

Conclusions

A complete coding and decoding procedure for the BEC has been described. For a given rate of transmission, the probability of error decreases exponentially with code length. The coding effort per digit is proportional to code length. The decoding effort per digit can be divided into two parts: one is related to the coding and is proportional to code length; the other is the amount of computation needed to solve the parity check equations and is bounded independently of code length for fixed rate and channel capacity.

These results are of interest in the theory of transmission through noisy channels. However, it is not known whether or not similar results are true for the computational requirements of all noisy channels. In the BEC, a digit is determined with probability one if it is not erased, or if it is determined unambiguously by the parity check equations. However, in certain channels (such as the binary symmetric channel), a digit is never determined with probability one. Thus it is not clear how much of the computational results for the BEC can be applied to other channels.

The physical systems to which the BEC model forms a good approximation have some small but nonzero probability that a zero is transmitted and a one received, or that a one is transmitted and a zero received. The BEC has zero probability for such transitions. Because of this difference it is not yet known whether or not the procedures described in this paper are practical.

Acknowledgment

The author wishes to acknowledge the assistance of members of the Research Laboratory of Electronics and Lincoln Laboratory. In particular, he is indebted to Professor Peter Elias for his many interesting suggestions and helpful conversations.

Appendix I: The Solution of Simultaneous Modulo 2 Equations

This appendix will first describe a procedure for solving a set of simultaneous linear modulo 2 equations and then evaluate an upper bound on the amount of computation needed for the solution. The procedure is a modulo 2 analogue of the Gauss-Jordan procedure for ordinary simultaneous linear equations.

We formulate the problem as follows. There are \( m \) linear modulo 2 equations in the \( r \) unknowns \( X_1, X_2, \ldots, X_r \). These equations are described by Eq. 1, where the \( b_{ij} \) are the coefficients of the unknowns, and \( b_i \) is the constant term of the \( i^{th} \) equation. All the \( b_{ij} \) and \( b_i \) are either zero or one.
\[ \sum_{j=1}^{r} b_{ij} x_j = b_i \pmod{2} \]

where \( i = 1, 2, \ldots, m \) \hspace{1cm} (1)

The procedure involves \( r \) steps taken in sequence. If the equations have a unique solution this procedure will find the solution. If the equations do not have a unique solution, this procedure will find the value of those unknowns whose values are determined.

In step 1 we try to eliminate the unknown \( X_1 \) from as many equations as possible. First, we search for an equation whose \( X_1 \) coefficient is nonzero. If there is no equation with a nonzero \( X_1 \) coefficient, we go to step 2. If there is such an equation, the equations are reordered so that the first equation has a nonzero coefficient for \( X_1 \). This coefficient is clearly equal to one because the \( b_{ij} \) have either the value zero or the value one. Then the first equation (in the reordered system) is added modulo 2 to every other equation whose coefficient for \( X_1 \) is nonzero. Thus \( X_1 \) is eliminated from every equation but Eq. 1. Then we go to step 2.

We now consider the general step \( p \), where \( p = 1, 2, \ldots, r \). It can be shown by induction that at the beginning of step \( p \) because of the actions of the previous \( p-1 \) steps, the \( m \) equations can be broken into two sets of equations with special properties. The first set of equations consists of Eq. 1 through some equation \( k \) and is such that each equation of the set contains a variable that is not contained in any other equation and is a member of the set \( X_1, X_2, \ldots, X_{p-1} \). The second set of equations consists of equation \( k+1 \) through equation \( m \) and is such that each of the equations in the set has zero coefficients for \( X_1, X_2, \ldots, X_{p-1} \) (see Fig. 7).

Now, our aim in step \( p \) is to eliminate the unknown \( X_p \) from as many equations as possible without disturbing the coefficients of the previous variables. Clearly we cannot use one of the first \( k \) equations to eliminate \( X_p \) from another equation as this will reintroduce that unknown of \( X_1, \ldots, X_{p-1} \), which it alone contains. Thus in step \( p \) we search equations \( k+1, \ldots, m \) to see if one of these equations has a nonzero coefficient for \( X_p \). If there is no such equation, step \( p \) ends. If there is such an equation, then equations \( k+1, k+2, \ldots, m \) are reordered so that equation \( k+1 \) has a nonzero coefficient for \( X_p \). Then \( X_p \) is eliminated from equations \( 1, 2, \ldots, k+2, k+3, \ldots, m \) by adding equation \( k+1 \) modulo 2 to any other equation with a nonzero coefficient for \( X_p \). This ends step \( p \).

The procedure terminates at the end of step \( r \) (see Fig. 8). Now we can immediately check whether or not the value of a given digit was determined by the original equations. Thus, if the \( p \)th unknown was determined by the original equations, the set of equations at the end of the procedure contains an equation of the form \( X_p = b \), and \( X_p \) is solved. If the \( p \)th unknown was not determined there is no such equation. The proof of the above statements is tedious and will not be given here.

A bound will now be found on the amount of computation required by this procedure. First, a bound is found for the computation in step \( p \). The amount of computation in step \( p \) depends on the situation at step \( p \). If equations \( k+1, k+2, \ldots, m \) have zero coefficients for \( X_p \) no computation is made. If one or more of these equations has a nonzero coefficient for \( X_p \), the equations are reordered so that equation \( k+1 \) has a nonzero coefficient for \( X_p \) and equation \( k+1 \) is added to any other equation with a nonzero coefficient for \( X_p \). We shall consider the addition of two coefficients or the addition of two constants as a unit computation. So, the addition of equation \( k+1 \) in step \( p \) to another equation requires \( r-p+2 \) computations. (The coefficients of \( X_1, X_2, \ldots, X_{p-1} \) are zero. We add the constant terms and the coefficients of \( X_{p'}, X_{p'+1}, \ldots, x_r \).) At worst, equation \( k+1 \) is added to the other \( m-1 \) equations and thus step \( p \) never requires more than \( (m-1)(r-p+2) \) computations. Summing over the bounds for each step we get Eq. 2.

The number of \( r \) computations for \( p \) equations and \( m \) equations and \( r \) unknowns

\[ \sum_{p=1}^{r} (m-1)(r-p+2) = \frac{1}{2} (m-1)(r^2+3r) \]

(2)
From this equation it is clear that \( n(1-R) \) equations in \( n \) unknowns require no more than \( \frac{1}{2} n^3(1-R) + n^2 \) computations.

A bound on the required computation can be found if the first \( m-1 \) equations are in standard form. In this case, the only computation required is the addition of some of the first \( m-1 \) equations to equation \( m \) and the addition of the resulting equation \( m \) to some of the first \( m-1 \) equations. The amount of computation required for the addition of one equation to another is bounded by \( r+1 \). Thus the amount of computation required is bounded by \( 2(m-1)(r+1) \) if the first \( m-1 \) equations are in standard form. If \( m \) is less than \( r+1 \), this expression is bounded by the first term, \( 2mr \).

**Appendix II: Bounds on the Probability of Ambiguity**

Equation 1 contains the bounds found by Elias on the probability of ambiguity for block parity check codes of length \( n \) and rate \( R \). \( Q_b \) denotes the probability of ambiguity of the optimum code of length \( n \) and rate \( R \) for a binary erasure channel whose probability of erasure is \( p \). \( Q_{av} \) denotes the average probability of ambiguity over the ensemble of all convolution codes of length \( n \) and rate \( R \).

\[
Q_b < Q_{av}
\]

\[
\leq \sum_{j=1}^{n(1-R)} \binom{n}{j} p^j q^{n-j} 2^{-n(1-R)+j} + \sum_{n(1-R)+1}^{n} \binom{n}{j} p^j q^{n-j} 2^{-n(1-R)+j+1}
\]  

1(a)

\[
Q_b \leq \sum_{n(1-R)+1}^{n} \binom{n}{j} p^j q^{n-j}
\]  

1(b)

These sums are much too complicated to evaluate directly for large \( n \), and hence Elias has calculated the following more convenient bounds. If the rate \( R \) is larger than a critical value, each of the sums in Eq. 1(a) can be bounded by a geometric series. Using the bounding geometric series and Stirling's formula, we find

**Channel Capacity**

\[
C = \text{channel capacity} = q
\]

if \( C > R > q/(1+p) \)

\[
Q_{av} \leq K n^{-1/2} e^{-n \ln(X)}
\]

\[
K = (2\pi R(1-R))^{-1/2} \left( \frac{p R}{C-R} + \frac{1}{1 - q(1-R)} - \frac{1}{2p R} \right)
\]

\[
X = \left( \frac{R}{C} \right)^{R(1-R)}
\]

For rates close to channel capacity or rates near \( q/(1+p) \), \( K \) becomes very large. For such rates and small values of code length it is preferable to bound \( Q_{av} \) by the largest term in Eq. 1(a) multiplied by the number of terms. If Stirling's formula is applied to this bound we obtain Eq. 4.

\[
Q_{av} \leq (2\pi R(1-R))^{-1/2} n^{1/2} \left( \frac{C}{R} \right)^{nR} \left( \frac{1-C}{1-R} \right)^{n(1-R)}
\]

4

For rates below \( q/(1+p) \) and code lengths shorter than \( (1+p)/(2p) \) one geometric series bounds Eq. 1(a).

\[
Q_{av} \leq (2\pi R(1-R))^{-1/2} n^{1/2} \left( \frac{C}{R} \right)^{nR} \left( \frac{1-C}{1-R} \right)^{n(1-R)}
\]

5

For rates below \( q/(1+p) \) and code lengths larger than \( (1+p)/(2p) \), the probability of ambiguity can be bounded by the largest term in Eq. 1(a) multiplied by the number of terms. Applying Stirling's formula, we get,

\[
Q_{av} \leq K n^{1/2} e^{-n \ln(X)}
\]

6

For small code lengths, Eqs. 2-6 are poor approximations to Eq. 1(a), and it may be preferable to evaluate Eq. 1(a) directly. Alternatively, we
can always bound the probability of ambiguity for a single digit by \( p \), since a digit is ambiguous only if it is erased.

**Appendix III: Various Types of Parity Check Code**

There are several types of parity check code whose behavior is similar to that of convolution codes. Diagonal codes are such a code type. (A diagonal code is defined to be a code in which information and check digits are interleaved, and each check digit is a parity check on some of the previous \( n-1 \) digits (see Fig. 9a).) The convolution codes are thus a subset of the diagonal codes.

However, there are diagonal codes which are not convolution codes. This is true because a check digit in a diagonal code can check other check digits as well as information digits and because the pattern of digits checked by a check digit in a diagonal code can differ from the pattern for other check digits of the code.

There are certain interesting subsets of the diagonal codes. One such subset is the set of all diagonal codes in which the check digits check only information digits in the preceding \( n-1 \) digits (see Fig. 9b). Another is the set of all diagonal codes in which each check digit checks the same pattern of the preceding \( n-1 \) digits (see Fig. 9c). Clearly, this last set of codes possesses the property that \( n \) or fewer digits describe the complete coding matrix. Convolution codes also possess this property, but a general diagonal code does not.

It can be proved* that the average probability of ambiguity per digit, for each of the sets of codes described above, is bounded by the bounds described in the text and in Appendix II. Thus, these bounds apply to the average probability of ambiguity for the set of all diagonal codes, the set of all convolution codes, the set of all diagonal codes in which the check digits check only information digits, the set of all diagonal codes in which each check digit checks the same pattern of the preceding \( n-1 \) digits.

* A proof will appear in the thesis entitled "Coding for the Binary Erasure Channel" to be submitted to the Department of Electrical Engineering, M.I.T., in partial fulfillment of the requirements for the degree of Doctor of Science.

**References**

Fig. 1 Transition probabilities for: (a) the binary symmetric channel and (b) the binary erasure channel.

Fig. 2 Coding matrices and coding equations. All equations are modulo two. (a) Block code, length = 7, rate = 4/7 and (b) convolution code, length = 4, rate = 1/2.

Fig. 3 Algebraic decoding procedure for a block code. (a) Block coding matrix, length = 5, rate = 3/5; (b) transmitted message and received message and (c) decoding equations: \( X_1 = I_2 \) and \( X_2 = C_1 \).
Fig. 4 (a) Convolution matrix, length = 8, rate = 1/2 and (b) convolution matrix reduced for the solution of digit 5.

Fig. 5 Bound on the decoding computation per digit for the binary erasure channel. Rate of transmission = 1/4; 1/2; 2/3.

Fig. 6 Convolution code of length 8; rate 1/2 truncated after the fifth information digit.

Fig. 7 A set of modulo two equations at the beginning of step 6; k = 3, m = 5, and r = 7.
Original Equations

End Step 1

End Step 3

\[ X_2 + X_3 + X_4 = 0 \]

\[ X_1 + X_3 + X_4 = 1 \]

\[ X_1 + X_3 = 1 \]

\[ X_3 = 0 \]

\[ X_1 + X_4 = 1 \]

\[ X_2 + X_3 + X_4 = 0 \]

\[ X_2 + X_4 = 0 \]

\[ X_3 = 0 \]

\[ 0 = 0 \]

Fig. 8 Solving four modulo two equations by the procedure outlined in Appendix I. \( X_3 \) alone is determined.

Fig. 9 Various types of diagonal codes. Blank spaces are zero: (a) diagonal code, any digit can be checked; (b) diagonal code, only information digits are checked; (c) diagonal code, same pattern for each check digit and (d) convolution code. For all the codes illustrated above, the code length is 6 and the rate of transmission is 1/3.