INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9315518

Bounds for cylindrical projection code

Yang, Gang, Ph.D.

City University of New York, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A

BOUNDS FOR CYLINDRICAL PROJECTION CODE

BY

GANG YANG

A dissertation submitted to the Graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1993

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/12/93
Date

Chair of Examining Committee

1/25/93
Date

Executive Officer

Joseph Barba

David Manela

Tarek Saadawi

Radimir Bozowic
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

BOUNDS FOR CYLINDRICAL PROJECTION CODES

by

Gang Yang

Advisor: Professor Donald L. Schilling

Both theoretical analysis and simulation results have been obtained. The main features of this research are:

1) The derivation and use of combinatorial generating functions to obtain the weight distribution of a Cylindrical Projection code.

2) The derivation and use of the weight distribution of a Cylindrical Projection code to obtain the upper bound and lower bound of the code.

3) The application of Improved Importance Sampling technique to simulate Cylindrical Projection code when the input error rate is very small.

ACKNOWLEDGEMENT

First, I would like to thank my adviser, Professor Donald L. Schilling, for his invitation to come to America as a visiting scholar which made it possible for me to become a Ph. D student. His valuable help was very much appreciated and his intelligence always impressive.

I would also like to thank the members of my Doctoral Committee: Professor Joseph Barba, Professor Tarek Saadawi, Professor Radimir Bozovic, Professor N. Scheinberg and Professor David Manela for their time and constructive criticism of the dissertation. I would like to thank Dean Lowen and Chairman S. Ahmed for their kindness.

Having suffered a life-threatening illness during my studies, there are a number of people I would like to thank for visiting me during hospital stays: Ms. Nancy Matthews, Mr. David Martin, Mr Qingfan Zhu and his wife, Mrs. Yuming Gong and his husband, and Mrs. Li Zou. I would like to thank Professor Schilling, Dr. Nasser Abdelatif, Dr. Nader Bolourchi, Mr. Kamal Shahrabi, Mrs. Joy Robinson, Dr. Ali Hipshish and Dr. David Manela for their concerns about my health.

I would like to thank City College for its extreme flexibility which was crucial for my recovery.

It is difficult to find appropriate words to thank my partner Dong Li. It was he who accompanied me to hospital, he who exchanged his financial aid source with mine so that I had medical insurance, he who aided me in getting teaching assignments, he who substituted for me while I was hospitalized and it was he who visited me in hospital frequently and asked his wife cook food for me. My parents also wish to thank him for helping their only son to survive.

Finally, I would like to thank my parents for allowing me to leave them living lonely and worrying a lot about me.

Table of Contents

Table of Figures

1 Introduction

In this dissertation, the error performance of binary Cylindrical Projection codes was determined using both a theoretical analysis and a simulation. The theoretical approach is to find the weight distribution of the Projection codes by using a generating function, and then to obtain the upper and lower bound. Simulation results were obtained using the Improved Importance Sampling technique.

1.1 A Simple Retrospect of Coding

A primary concern in any digital communication system is the reliability, that is, to what extent, one can recover the transmitted sequence from the received data sequence. These sequences may be different due to the interference in the channel. Three approaches are available to improve the performance.

The first approach is to improve the transmission medium in order to reduce the interference in the channel. However, one is not always able to obtain a sufficiently good channel, because of the cost, the technology limit, or both. More economical methods have to be considered.

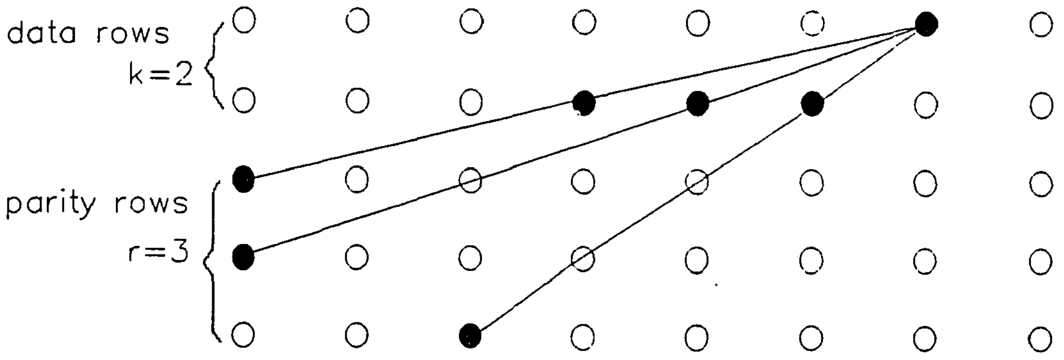
The second one is to increase the transmitting power, to increase the efficient area of antenna, to make both the transmitter and the receiver optimum or least sensitive to the interference and so on. Although a lot of methods have been put forward in this field, better and more economical methods are expected. It is expected that the performance of the system will improve, while the complexity and the cost of the communication system will not increase.

The third one is to apply coding to the communication system. In a code, at the encoder, some redundant bits, called parity bits, are added to the information sequence according to some rules (algorithms); at the decoder, the information sequence is recovered from the received data sequence by making use of the checking ability of parity bits. The more parity bits, the more errors can be corrected. But the number of parity bits can not be arbitrarily large, because the more parity bits per codeword, the less energy per bit, and therefore, the higher of the input bit error rate. So a code is useful only when the error correcting ability can complement the reduction of the energy of a bit. With the rapid development of large scale integrated circuits, better encoding and decoding methods can be realized less costly and more reliably. Hence, today, considerable attention is being focused to the use of the forward error correcting coding (FEC).

There are several different types of codes. Block codes and convolution codes are the two major types. They can be linear codes and nonlinear codes. In the case of block codes, they may be cyclic or non-cyclic. A lot of work has been done in studying all of these various codes. In particular, Shannon's theorem states that if the only interference is Gaussian white noise and the information transmitting rate is less than the capacity of the channel (which is a function of signal to noise ratio), then a code can be found such that the output bit error rate of the decoder can be arbitrary small [4]. This theorem has encouraged people to find such ideal codes. The Projection code is an attempt to obtain an ideal code. It is very easy to encode and decode, and from previous simulation of the code, it shows it is a powerful code. All of these make it a desirable code. However, because of its very abrupt performance curve, it is difficult to investigate and evaluate its performance at reasonable input error rates by classical simulation technique. Therefore, it is necessary to analyze this code theoretically and develop a better simulation technique.

The Projection code has three basic forms. [1][2]

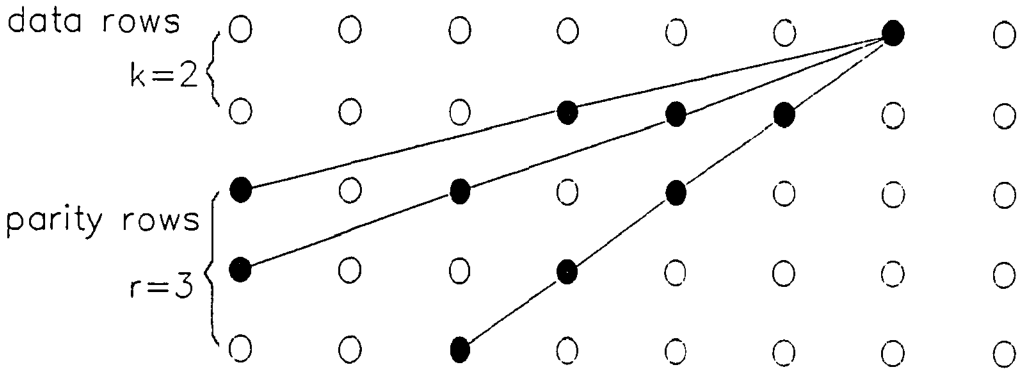
The first one is called the Basic Projection Code , or P1 code. In such a block code, the parity bits check only the information bits on the same parity line and do not check any other parity bits. For example, in Fig.1, there is a P1 code which has two data rows and three parity rows with slopes 1, 1/2 and 1/3. The parity bits on a slope line check only the information bits on the same slope line.



A P1 Projection Code

Fig.1

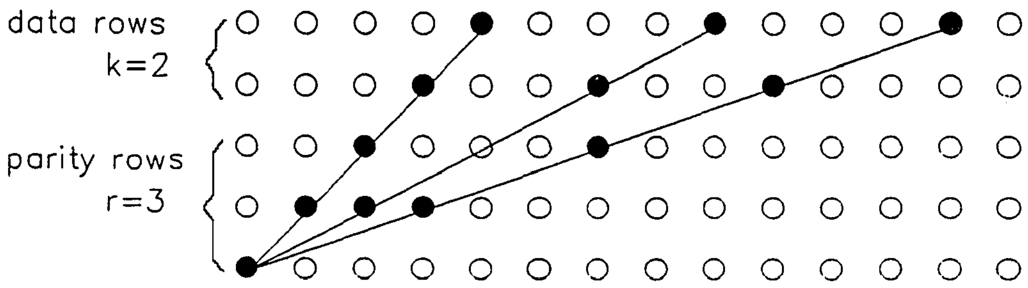
The second one is called the Partial Autoconcatenation Projection Code, or P2 code. In this code, the parity bits not only check the information bits on the same parity check line, but also check the other parity bits on the rows above them and on the same parity check line. A P2 code with slopes 1, 1/2 and 1/3, with two data rows and three parity check rows is shown in Fig.2.



A P2 Projection Code
Fig.2

The third one is called the Total Autoconcatenation Projection Code, or P3 code. In the case of P3, the parity bits check not only the information bits and the parity bits on the rows above them, but also the parity bits on rows below them. In Fig.3, there is a P3 code with slopes

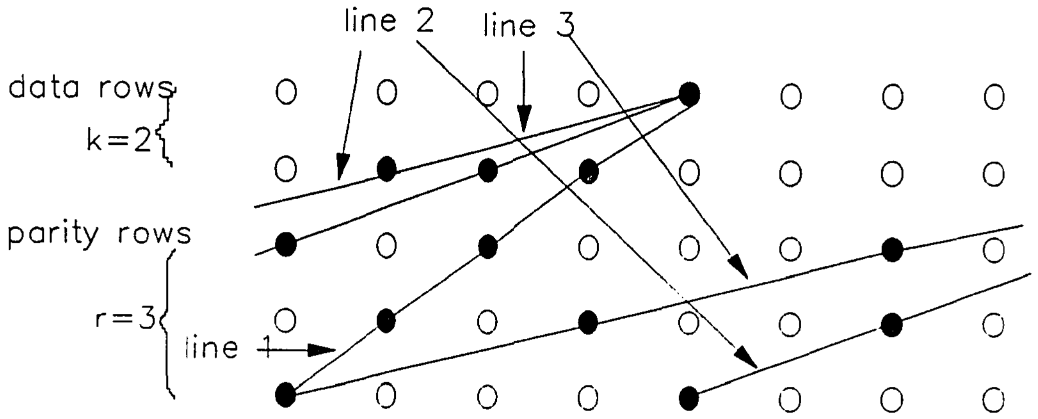
1, 1/2 and 1/3, two data rows and three parity check rows.



A P3 Projection Code

Fig.3

In order to analyze the Projection codes, it is necessary to distinguish the codes further. On one hand, the Projection code can be a convolution-like code, that is, being realized by shift registers and adders and with an infinite block length. On the other hand, the Projection code can adopt the block code form. In this case, it can be either a cylindrical block code, where the number of the bits in a block is finite, but with the end and the head connected together; or a noncylindrical block code, that is, a block code with a beginning and an end. A cylindrical P3 code with block length 10, two data rows, three parity check rows and slopes 1, 1/2 and 1/3 is shown in Fig.4.



A P3 Projection Code
Fig.4

We will focus on the cylindrical block code. The reason is that when the block length is long enough, there is little difference between the error performance of the cylindrical block code and noncylindrical block code. And the convolution-like code is studied by Dong Li.

In this dissertation, the error performance of the P2 and P3 cylindrical Projection code in the binary symmetric channel will be analyzed by obtaining the weight distribution. Simulation result for a P3 code will be given also by using improved importance sampling technique.

1.2 The Existing Error Performance Technology and Its Limits

Although numerous methods for designing encoder and decoder exist, there is no universal method which can be used to analyze the performance of most of the codes. Nor is there a unified method for obtaining the error performance even for the codes in $GF(q^m)$ which have very good mathematical structures. Each code has to be analyzed individually. Further, the approximation method widely used in block codes can't be used in Projection codes. The approximation formula is

$$P_b \doteq \sum_{i=t+1}^{N} \left(\frac{t+1}{N}\right)\binom{N}{i} p_b^i (1-p_b)^{(N-i)} \qquad (1.1)$$

Where

P_b, the output bit error rate of the decoder;

p_b, the input bit error rate of the decoder;

N, the number of bits in a codeword;

t, the guaranteed number of correctable errors.

Usually, the error performance of Projection code can't be estimated by using the above formula. The reason is as follows.

The probability of the number of the errors occurring in a block code follows the binomial distribution. The peak of the probability distribution curve is located near $N p_b$. The larger the input error rate is, the flatter the curve becomes; the larger the block length is, the flatter the curve becomes. In order to have enough precision to calculate the output error rate, it is necessary to consider that enough errors occur in one block. For example, for the P3 block code with equivalent block length 76 and three parity check rows and input error rate 10^{-2}, if only $2^{3-1}=4$ errors are considered, the precision of calculation is not better than $\sum_{i=5}^{76}\binom{76}{i}0.01^{i}0.99^{76-i}=0.00102$. In other words, if the precision of 0.000001 is needed and only 4 errors will be given consideration, the input error rate must be less than 0.0023.

Compared with the ordinary block codes, the Projection codes typically have a much longer equivalent block length and can work at higher error rates. For example, the widely used Hamming code (7,4) has a block length of 7, the Golay code has a block length of 23, but the simplest cylindrical P3 code with one data row, three parity check rows and with slopes 1,1/2 and 1/7 must have a block length no less than 19 in order to make sure that no two lines will meet more than once and therefore, the maximum Hamming distance will be guaranteed. If the code is transmitted row by row, then the equivalent block length is not less than 76. Further, being similar to the convolutional code, the P2 and P3 codes can be used in a much worse environment than ordinary block codes. Because the P2 Projection codes and P3 Projection codes can correct many error patterns consisting of errors which exceed more than half of the Hamming distance in one block, in addition to correcting any $2^{(r-1)}$ errors or less in one block.

Therefore, the above approximation formula can't be simply used to obtain the output error rate of the Projection codes because either N or input error rate is not small. A more precise method has to be put forward to analyze the error rate of the Projection codes. It is necessary for us to review the Projection code in order to find some new methods.

2 Description of the Projection Code

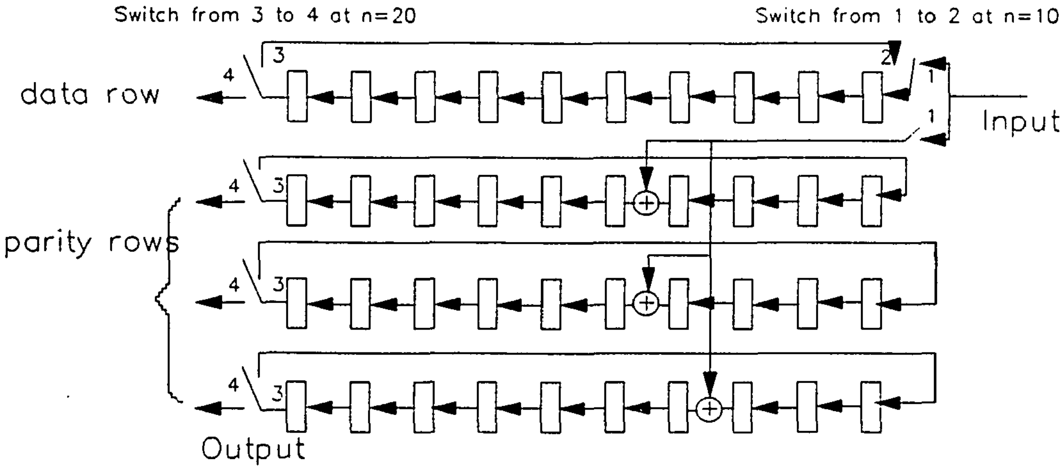
2.1 The Definition of the Projection Code

In a typical block Projection code, there are k data rows and below the data rows, there are r parity check rows, with the block length h. Thus, the code efficiency is k/(k+r). The parity bits can be calculated by many different ways. For example, parity bits can be calculated by the data bits on the same straight line, or on some type of curve as long as there is no contradiction. For simplicity, here the parity bits are calculated from the data bits on the same straight lines. Usually, there are three different algorithms to determine the parity bits. They are called the P1, P2 and P3 Projection code respectively.

2.1.1 The Encoder and Decoder of the P1 Cylindrical Projection Code

Suppose the slopes of the form $1/m_1, 1/m_2, \ldots, 1/m_r$ are chosen. Through each data bit, a line with slope $1/m_j$ is drawn in order to get a parity bit on the jth parity row. The line is extended into the parity check block and is wrapped cyclically when reaching either end of a row. A bit on the jth parity row is equal to the mod-2 sum of all of the data bits on the line which passes through the parity bit and has slope of $1/m_i$. Totally, $h \times r$ lines are drawn to determine all of the parity bits.

The cylindrical encoder for P1 with one data row, three parity rows, slopes 1, 1/2 and 1/4 and block length 10 is shown in Fig.5.



A P1 Cylindrical Encoder

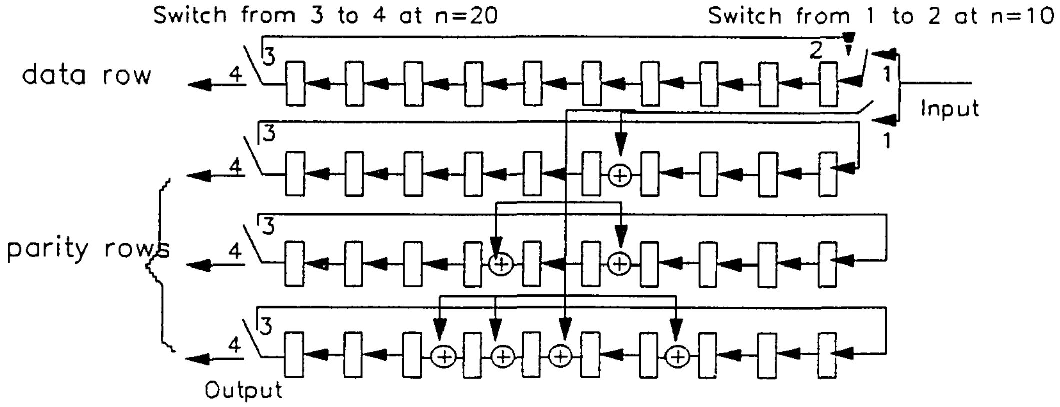
Fig.5

Majority logic is used to realize the decoder. There are r parity lines through each data bit. If in these r lines, there are more than T (decoding threshold, larger than r/2) lines having mistakes, then this data bit is inverted. The decoder can correct [(r-1)/2] errors.

We can obtain the output error performance of P1. The basic idea is to determine T so that the sum of the probability of uncorrected errors and that of generated errors will be smallest, then to obtain the probability of uncorrected errors and the probability of generated errors, and to sum them together.

2.1.2 The Encoder and Decoder of the P2 Cylindrical Projection Code

The parity bits of the P1 code check only data bits and not check other parity bits. Therefore, the error correcting ability of this code is not very powerful. In the P2 code, a parity bit is equal to the mod-2 sum of both the data bits and the parity bits which are above this parity check bit and on the same parity line as the parity bit. Compared to the P1 encoder, the P2 encoder is somewhat more complex. An encoder for the P2 cylindrical code with one data row, three parity check rows, slopes 1, 1/2, 1/7 and block length 19 is shown in Fig.6.

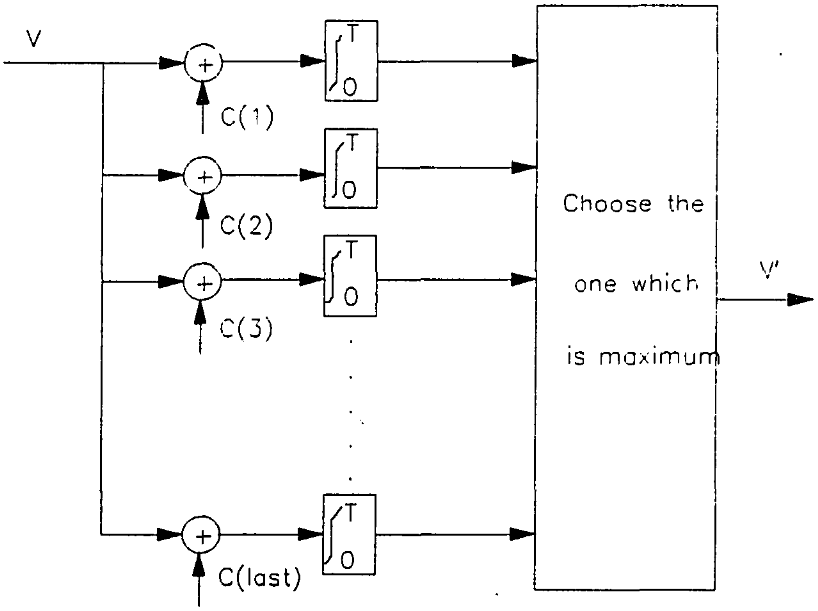


A P2 Cylindrical Encoder

Fig.6

A threshold decoding algorithm has been presented by Professor David Manela.[1]

In this dissertation, the potential performance of the Projection code will be studied theoretically and therefore, an optimum decoder will be used. And a new simulation technique will be given to simulate the projection code when the input error rate is small. In Fig.7, C(1), C(2), ..., C(last) are all of real codewords which can be obtained from the corresponding encoder. However, at the present time, it is impossible to realize the optimum decoder because of the tremendous storage and time requirement. Thus much attention will be given to the suboptimal decoder, which uses only the parity check equations to construct the decoder and does not distinguish between codewords and pseudocodewords which can not be obtained from the corresponding encoder but satisfy the parity check equations. Later except when necessary, we will not distinguish them. We will call the code including the codewords and the pseudocodewords as the extended code.



The Decoder for Projection Code
Fig.7

Another reason to use the suboptimum decoder is that: the larger the blocklength and the smaller the input error rate are, the smaller the difference between the suboptimum decoding and optimum decoding is.

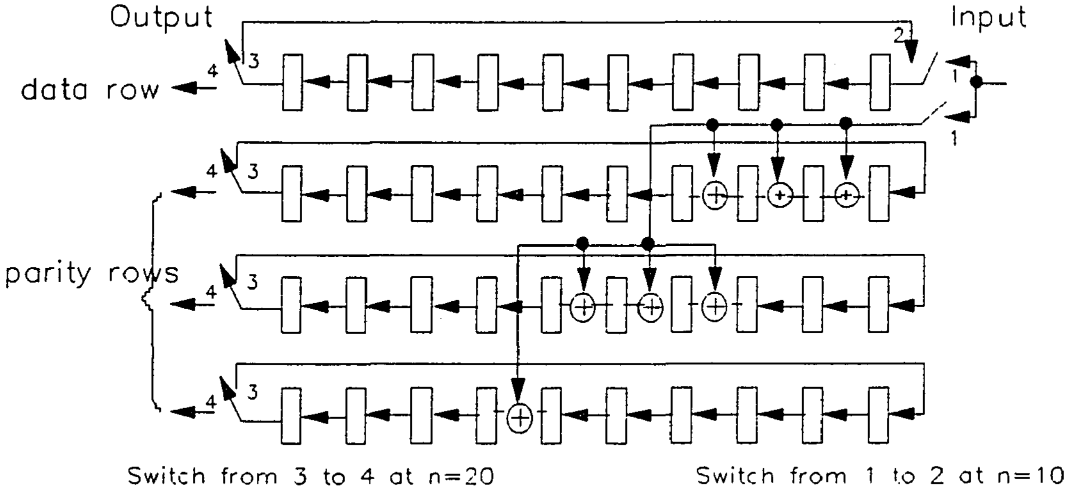
In addition, it is easier to obtain the weight distribution of the extended code than that of the real code.

The same structure as shown in Fig.7 can be used to realize the suboptimum decoder if C(1), C(2), ..., C(last) are used to stand for not only all of the real codewords but also all of the pseudocodewords. And no other decoder can give a better performance than the suboptimum decoder if only the parity check equations are used.

Although we can not realize the structure due to the tremendous requirement of memory, we can use it to analyze the best performance if only parity check equations are used.

2.1.3 The Encoder and Decoder of the P3 Cyclindrical Projection Code

The P2 coding scheme does not give each parity bit the same "attention". The parity bits on one parity check line only check other parity bits of the same line which are over this check bit. In other words, these parity bits over this parity bit do not check this parity bit. The third coding scheme, called P3 Projection code , will let each parity have the same role and therefore the structure become more robust. The encoder is given in Fig.8.



A P3 Cylindrical Encoder

Fig.8

As we did in analyzing the P2 codes, we only consider suboptimal decoding which makes use of the parity check equations. The decoder looks like the decoder for P2 except changing the codewords of P2 to the codewords of P3.

3 The Methods for Upperbounding and Lowerbounding the Projection Code

If the weight distribution of a binary cylindrical Projection code is given, upper and lower bounds for the output bit error rate, assuming maximum likelihood decoding, can be calculated.

Since the Projection code is a linear code, it is sufficient to consider the case when the all-zero codeword is transmitted. The following definitions are needed: Denote C_w as a codeword with weight of w. Let M_w be the number of codewords with weight w. Let $C(j)$ be the j^{th} codeword in the code. Then $|C(j)|$ is the weight of codeword $C(j)$. Let V be the received word. Denote p as the intrinsic bit error rate in a BSC channel and P_b as the decoded bit error rate. Finally, we let M be the total number of codewords and N be the total number of bits in the code.

3.1 The Derivation of the Upperbound I

Assume the all-zero codeword is transmitted. Then, when the received word, V, is closer to $C(j)$ than to any other codeword, the maximum likelihood decoder decides that $C(j)$ is transmitted. If V is at the same distance from two other codewords, $C(j)$ and $C(l)$, and is further from any other codewords, then the decoder decides that either $C(j)$ or $C(l)$ is transmitted with probability of 1/2. The probability of an error event, denoted by P_e, is then:

$$P\{|C(j)-V|<|C(i)-V|, \forall i \neq j\}$$
$$+\frac{1}{2}P\{|C(j)-V|=|C(i)-V| \text{ and } |C(j)-V|<|C(l)-V|, \text{ for some } i\neq j, \forall l\neq j \text{ and } l\neq i\}.$$

Since the weight of $C(j)$ is $|C(j)|$, the average number of errors in this event is $|C(j)|P_e$. Consider M codewords in the code. The average number of errors is

$$\sum_{j=1}^{M-1}|C(j)|P\{|C(j)-V|<|C(i)-V|, \forall i\neq j\}$$
$$+\frac{1}{2}\sum_{j=1}^{M-1}|C(j)|\{P\{|C(j)-V|=|C(i)-V| \text{ and }$$
$$|C(j)-V|<|C(l)-V|, \text{ for some } i\neq j, \ \forall l\neq j \text{ and } l\neq i\}\} \quad (3.1)$$

The bit error rate, P_b, is the average number of errors divided by N, thus

$$P_b = \frac{1}{N}\sum_{j=1}^{M-1} |C(j)| P\{|C(j)-V| < |C(i)-V|, \forall i \neq j\}$$
$$+ \frac{1}{2N}\sum_{j=1}^{M-1} \{|C(j)| P\{|C(j)-V| = |C(i)-V| \text{ and}$$
$$|C(j)-V| < |C(l)-V|\}, \text{ some } i \neq j, \quad \forall l \neq j \text{ and } l \neq i\} \quad (3.2)$$

The probability that V is closer to $C(j)$ than to any other codeword is less than the probability that V is closer to $C(j)$ than to all-zero codeword. P_b Can then be upper bounded if the all-zero codeword is substituted for $C(i)$ in Eq.(3.2). Hence,

$$P_b < \frac{1}{N}\sum_{j=1}^{M-1} |C(j)| P\{|C(j)-V| < |V|\}$$
$$+ \frac{1}{2N}\sum_{j=1}^{M-1} |C(j)| P\{|C(j)-V| = |V|\} \quad (3.3)$$

Equation (3.3) does not depend on particular codewords, it depends only on the weights of the codewords, therefore

$$P_b < \frac{1}{N}\sum_{w=1}^{N} M_w w P\{|C_w - V| < |V|\} + \frac{1}{2N}\sum_{w=1}^{N} M_w w P\{|C_w - V| = |V|\} \quad (3.4)$$

3.2 The Derivation of the Upperbound II

Let $C^{(i)}$ stand for the ith codeword starting from column 0 and let $w^{(i)}$ be its weight. Let h be the length of a projection code. Suppose the all zero codeword is transmitted. Similar to Viterbi's method, we have:

The average errors on the Cylindrical Projection Codes

$= \sum_{j=0}^{h-1}$ The average errors due to the errors starting from No. j column

$= h \cdot$ The average errors due to the errors starting from column 0

and

The average errors due to the errors starting from column 0

$$< \sum_{\forall C^{(i)}} w^{(i)} \left[\sum_{j=w^{(i)}/2+1}^{w^{(i)}} \binom{w^{(i)}}{j} p_b^j (1-p_b)^{w^{(i)}-j} + 1/2 \binom{w^{(i)}}{\frac{w^{(i)}}{2}} p_b^{\frac{w^{(i)}}{2}} (1-p_b)^{\frac{w^{(i)}}{2}} \right] \tag{3.5}$$

Therefore,

$P_b < \frac{1}{(k+r)h} \cdot h \cdot$ The average errors due to the errors starting from column 0

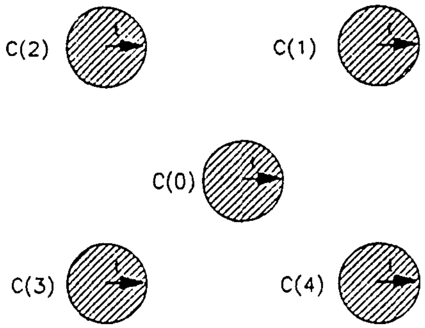
$$= \frac{1}{k+r} \sum_{\forall C^{(i)}} \left[\sum_{j=\frac{w^{(i)}}{2}+1}^{w^{(i)}} \binom{w^{(i)}}{j} p_b^j (1-p_b)^{w^{(i)}-j} + 1/2 \binom{w^{(i)}}{w^{(i)}/2} p_b^{\frac{w^{(i)}}{2}} (1-p_b)^{\frac{w^{(i)}}{2}} \right] \tag{3.6}$$

This bound is a little tighter than the previous one. If two codewords have no common bits which are one, the exclusive or the two codewords will produce a new codeword with the weight equaling to the summation of the weights of the two codewords. Unlike to previous method, here the new codeword will not be given further consideration.

The method to obtain the weight distribution of a projection code can been changed a little to obtain $w^{(i)}$. When obtaining a codeword, we record its weight and delete it to prevent from producing new codewords based on it.

3.3 The Derivation of the Lowerbound

Again, assume that the all-zero codeword is transmitted. Let us draw a circle with radius t around every codeword as shown in Figure 9. In the figure, four codewords with weight of w surrounds the all-zero codeword.



The Weight of C(1),C(2),C(3) and C(4) is w

Fig. 9

A received word falling into the circle of a codeword will be decoded as the codeword. If the weight of this codeword is w, then w errors are made. Denote $P(w)$ as the probability of a received word falling into the circle of a codeword with weight w. To obtain a lower bound on the average number of errors, assume every received word falls into a circle. Then, the average number of errors in the code is lower bounded by $\sum_{w=1}^{N} M_w w P(w)$. The bit error rate is therefore lower bounded by

$$P_b > \frac{1}{N} \sum_{w=1}^{N} M_w w P(w) \tag{3.7}$$

To calculate $P(w)$, let $N(w,l;s)$ be the number of error patterns of weight l which are at the distance s from a codeword of weight w. Let $d_{\min}$ be the minimum weight of the code. Then, $t = \frac{d_{\min}}{2} + 1$. For linear codes, $N(w,l;s)$ is the same for every codeword of weight w. It can be shown that in a binary code [5],

$$N(w,l;s) = \sum_{\substack{j=s-k\geq 0 \\ k=l+j-w\geq 0}} \binom{N-w}{k}\binom{w}{j} \tag{3.8}$$

Thus,

$$\begin{aligned} P(w) = \sum_{s=0}^{t}\sum_{l=0}^{N} p^{l}(1-p)^{N-l} N(w,l;s) \\ + \frac{1}{2}\sum_{l=0}^{N} p^{l}(1-p)^{N-l} N(w,l;t+1) \end{aligned} \tag{3.9}$$

Therefore, the lower bound on the bit error rate is

$$\begin{aligned} P_b > \frac{1}{N}\sum_{w=1}^{N}\sum_{l=0}^{N}\sum_{s=0}^{t} wM_w p^{l}(1-p)^{N-l} N(w,l;s) \\ + \frac{1}{2N}\sum_{w=0}^{N}\sum_{l=0}^{N} wM_w p^{l}(1-p)^{N-l} N(w,l;t+1) \end{aligned} \tag{3.10}$$

4 The Generating Function

The generating function technique is a convenient tool for handling selection and arrangement problems [6], [7]. Seeking codewords which satisfy the parity check equations of the Projection Code is a type of selection process. Therefore, it is useful to develop the combinatorial generating function technique and use it to obtain the weight distribution of the code.

4.1 The Concept of the Generating Function

Suppose a_r is the number of ways to select r objects in a certain procedure. If $g(x)=a_0+a_1x+a_2x^2+\ldots+a_rx^r+\ldots$, then g(x) is called the generating function of a_r. For example, $g(x)=(1+x)^n$ is the generating function of $\binom{n}{r}$, the ways to select r elements from n elements.

Let us consider the formal multiplication of $(a+x)^3$.

$$(a+x)(a+x)(a+x)=aaa+aax+axa+axx+xaa+xax+xxa+xxx \tag{4.1}$$

Let a=1,

$$(1+x)(1+x)(1+x)=111+11x+1x1+1xx+x11 \\ +x1x+xx1+xxx \qquad (4.2)$$

Such a formal expansion lists all ways of multiplying a term in the first factor times a term in the second factor times a term in the third factor. The problem of determining the coefficient of x^r in $(1+x)^3$, and more generally in $(1+x)^n$, reduces to the problem of counting the number of different formal products with exactly r x's and n-r 1's. So the coefficient of x^r in $(1+x)^3$ is $\binom{3}{r}$, and in $(1+x)^n$ is $\binom{n}{r}$.

It is worthwhile to view the multiplication of several polynomial factors as generating the collection of all formal products obtained by multiplying a term from each polynomial factor together. If the ith polynomial factor contains r_i different terms and there are n factors, then there will be $r_1 \cdot r_2 \cdot r_3 \cdot \ldots \cdot r_n$ different formal products. For example, there will be 2^n formal products in the expansion of $(1+x)^n$. In the expansion of $(1+x)^3 \cdot (1+x+x^2)^2$, the set of all formal products will be sequences of the form

$$\begin{Bmatrix} 1 \\ x \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x \\ x^2 \end{Bmatrix} \cdot \begin{Bmatrix} 1 \\ x \\ x^2 \end{Bmatrix}$$

That is, a 1 or an x in each of the first three brackets can be chosen and then a 1 or an x or an x^2 in each of the last two brackets can be chosen. For example, $1x1x^2x$ means taking 1 from the first bracket, taking x from the second bracket, taking 1 from the third bracket, taking x^2 from the forth bracket and taking x from the last bracket.

Example 1

Find a generating function for a_r, the number of ways to select r balls from a pile of three green, three white, three blue and three gold balls.

This selection can be modeled as the number of integer solutions to

$$x_1+x_2+x_3+x_4=n, \quad 0\leq x_i\leq 3 \tag{4.3}$$

Here x_1 presents the number of green balls chosen, x_2 the number of white, x_3 blue, x_4 gold. A multiplication of polynomial factors can be constructed such that the formal products will contain four terms each with an exponent between 0 and 3.

$$\left\{\begin{matrix}x^0\\x^1\\x^2\\x^3\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\\x^3\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\\x^3\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\\x^3\end{matrix}\right\}$$

Then the number of these formal products with the sum of the four exponents equaling to n, will be the number of integer solutions to the above equation. The desired generating function is thus $(x^1+x^1+x^2+x^3)^4=(1+x+x^2+x^3)^4$

Example 2

Another example is to find the weight distribution of a N=5 block code. Each bit in this code is represented by a binary variable x_i . Let us assume that the coding rule is described by the following system of equations:

$$\begin{aligned}
&x_1+x_2+x_3+x_4=0 \quad \text{mod } 2\\
&x_2+x_3+x_4+x_5=0 \quad \text{mod } 2\\
&x_3+x_4+x_5+x_1=0 \quad \text{mod } 2\\
&x_4+x_5+x_1+x_2=0 \quad \text{mod } 2\\
&\quad x_i=0 \text{ or } 1\ ,i=1,2,\ldots,5 \quad (4.4)
\end{aligned}$$

We will find the number of solutions with various weights. Let x_i^j denote weight j for variable x_i, w_i^j denote weight j for the sum of weights in the ith equation and

w^j denote weight j for the sum of weights in the system of equations. The generating factor of x_1 in the third equation is denoted by $x_1^0 + x_1^1 w_3^1$. The operational rules for the generating function are

$$x_i^j \cdot x_k^l = \begin{Bmatrix} x_i^j w^{-j}, i = k \text{ and } j = l \\ x_i^j \cdot x_k^l, \quad i \neq k \\ 0, \quad i = k, \quad j \neq l \end{Bmatrix}$$

$$0 \cdot x_i^j = 0 \qquad (4.5)$$

$$w^j \cdot w^l = w^{j+l}$$

$$w_i^j \cdot w_k^l = \begin{Bmatrix} w_i^{j+l}, \quad i = k \\ w_i^j \cdot w_k^l, \quad i \neq k \end{Bmatrix}$$

The rules are based on the following:

i) The weight of any variable can be counted only
ii) once;
iii) Any variable can take only one value at a time;
The weight of the summation of two variables is equal to the summation of the weight of each variable.

The generating function of the first equation, g_1 , is the product of all the generating factors in the first equation. That is

$$
\begin{aligned}
g_1 &= (x_1^0+x_1^1 w_1)(x_2^0+x_2^1 w_1)(x_3^0+x_3^1 w_1)(x_4^0+x_4^1 w_1) \\
&= x_1^0 x_2^0 x_3^0 x_4^0 + x_1 x_2 x_3^0 x_4^0 w_1^2 + x_1 x_2^0 x_3 x_4^0 w_1^2 \\
&\quad + x_1 x_2^0 x_3^0 x_4 w_1^2 + x_1^0 x_2 x_3 x_4^0 w_1^2 + x_1^0 x_2 x_3^0 x_4 w_1^2 \\
&\quad + x_1^0 x_2^0 x_3 x_4 w_1^2 + x_1 x_2 x_3 x_4 w_1^4. \qquad (4.6)
\end{aligned}
$$

The generating function of the second equation, g_2 , is

$$
\begin{aligned}
g_2 &= x_2^0 x_3^0 x_4^0 x_5^0 + x_2 x_3 x_4^0 x_5^0 w_2^2 + x_2 x_3^0 x_4 x_5^0 w_2^2 \\
&\quad + x_2 x_3^0 x_4^0 x_5 w_2^2 + x_2^0 x_3 x_4 x_5^0 w_2^2 + x_2^0 x_3 x_4^0 x_5 w_2^2 \\
&\quad + x_2^0 x_3^0 x_4 x_5 w_2^2 + x_2 x_3 x_4 x_5 w_2^4. \qquad (4.7)
\end{aligned}
$$

The generating function of the third equation, g_3 , is

$$
\begin{aligned}
g_3 &= x_1^0 x_3^0 x_4^0 x_5^0 + x_1 x_3 x_4^0 x_5^0 w_3^2 + x_1 x_3^0 x_4 x_5^0 w_3^2 \\
&\quad + x_1 x_3^0 x_4^0 x_5 w_3^2 + x_1^0 x_3 x_4 x_5^0 w_3^2 + x_1^0 x_3 x_4^0 x_5 w_3^2 \\
&\quad + x_1^0 x_3^0 x_4 x_5 w_3^2 + x_1 x_3 x_4 x_5 w_3^4. \qquad (4.8)
\end{aligned}
$$

The generating function of the forth equation, g_4 , is

$$g_4 = x_1^0 x_2^0 x_4^0 x_5^0 + x_1 x_2 x_4^0 x_5^0 w_4^2 + x_1 x_2^0 x_4 x_5^0 w_4^2$$
$$+ x_1 x_2^0 x_4^0 x_5 w_4^2 + x_1^0 x_2 x_4 x_5^0 w_4^2 + x_1^0 x_2 x_4^0 x_5 w_4^2$$
$$+ x_1^0 x_2^0 x_4 x_5 w_4^2 + x_1 x_2 x_4 x_5 w_4^4. \tag{4.9}$$

Only the terms with even powers of w_i in the above generating functions remain because the odd powers of w do not satisfy the system of equations. The total generating function of the system, g, is $g = g_1 \cdot g_2 \cdot g_3 \cdot g_4$. In g, there is no need to distinguish w_1, w_2, w_3 and w_4. Thus, setting $w = w_1 = w_2 = w_3 = w_4$ and simplifying g, we obtain

$$g = x_1^0 x_2^0 x_3^0 x_4^0 x_5^0 w^0 + x_1 x_2 x_3 x_4 x_5 w^5 \tag{4.10}$$

Having interest in the number of solutions of various weights, we replace x_i^j with 1. Then, we obtain $g = 1 + w^5$. This shows that in the system of equations there is only one solution with a weight of 0 and only one solution with a weight of 5. This result can be easily verified by noticing the symmetry of the system. Thus, for our

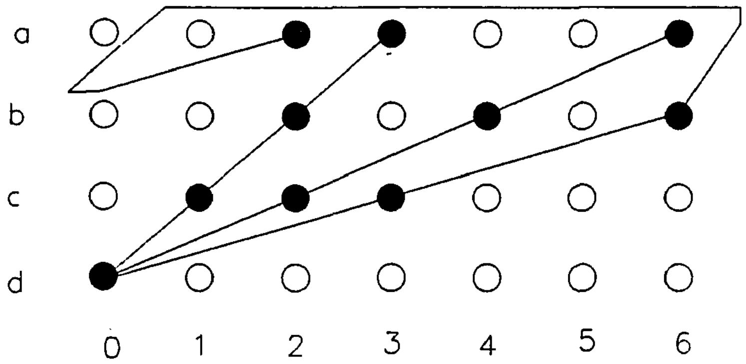
N=5 block code, there are only two codewords. One codeword is the all-zero codeword and the other is a codeword with Hamming weight of 5, i.e., each bit is a 1.

In this example, it seems unnecessary to distinguish various w_l if g_1 is calculated first, then g_2, g_3, g_4 and lastly, g. However, distinguishing w_l allows one to simplify the total generating function in each step by multiplying some of the factors of g_1 with some of the factors of g_2 without any confusion. In some very structured equations, this may save a lot of memory space. We will illustrate this argument in the following section.

4.2 Applying the Generating Function to Projection Codes

Part I

Consider a binary, rate-1/4, P3, cylindrical Projection code with slopes 1, 1/2 and 1/3, one data row (row a), three parity check rows (rows b, c and d) and a blocklength of 7. The code block is shown in Figure 10.



Every bit in the block is checked exactly three times for three parity check slopes in the code. There are only three parity check lines shown here for the sake of simplicity. Denote the parity check line with slope 1/3 passing through the point (b, 6) as B6, the parity check line with slope 1/2 passing through the point (C, 2) as C2 and the parity check line with slope 1 passing through the point (d, 0) as D0. Let b_6 represent the bit variable at the position (b, 6), c_2 represent the bit variable at the position (c, 2), d_0 represent the bit variable at the position (d, 0) and so on.

For line D0, the parity check equation is

$$a_3+b_2+c_1+d_0=0 \bmod 2 \tag{4.11}$$

and its generating function is

$$g_1 = (a_3^0 + a_3 \cdot w_{D0})(b_2^0 + b_2 \cdot w_{D0})(c_1^0 + c_1 \cdot w_{D0})(d_0^0 + d_0 \cdot w_{D0}). \tag{4.12}$$

The parity check equation for line C2 is

$$a_6 + b_4 + c_2 + d_0 = 0 \mod 2 \tag{4.13}$$

and its generating function is

$$g_2 = (a_6^0 + a_6 \cdot w_{C2})(b_4^0 + b_4 \cdot w_{C2})(c_2^0 + c_2 \cdot w_{C2})(d_0^0 + d_0 \cdot w_{C2}). \tag{4.14}$$

The parity check equation for line B6 is

$$a_2 + b_6 + c_3 + d_0 = 0 \mod 2 \tag{4.15}$$

and its generating function is

$$g_3 = (a_2^0 + a_2 \cdot w_{B6})(b_6^0 + b_6 \cdot w_{B6})(c_3^0 + c_3 w \cdot_{B6})(d_0^0 + d_0 \cdot w_{B6}). \tag{4.16}$$

The generating factor of d_0 denoted by f_{d_0} is the product of all the factors in Eq.(4.12), Eq.(4.14) and Eq.(4.16) which contain the symbol d_0, then

$$\begin{aligned} f_{d_0} &= (d_0^0 + d_0 \cdot w_{D0})(d_0^0 + d_0 \cdot w_{C2})(d_0^0 + d_0 \cdot w_{B6}) \\ &= (d_0^0 + d_0 \cdot w_{D0} \cdot w_{C2} \cdot w^{-1})(d_0^0 + d_0 \cdot w_{B6}) \\ &= (d_0^0 + d_0 \cdot w_{D0} \cdot w_{C2} \cdot w_{B6} \cdot w^{-2}). \end{aligned} \tag{4.17}$$

Equation (4.17) can be simplified by setting $d_0 = 1$ and $d_0^0 = 1$ since we are only interested in the weight of the generating factor. Further, $w_{D0} \cdot w_{C2} \cdot w_{B6} \cdot w^{-2}$ can be replaced by $B6C2D0w$ Thus, we obtain the generating factor

$$f_{d_0} = (1 + B6C2D0w) \tag{4.18}$$

The generating factor f_{d_0} can be explained intuitively:

Variable d_0 has two possible values, either 1 or 0. If it is 1, then on line B6, there must be at least one point with bit value 1 in order to satisfy the parity check equation. Similarly, there must be at least one point with bit value 1 on line C2 and at least one point with bit value 1 on line D0. So there must be such a factor as $1 + B6C2D0w$ for variable d_0.

The generating factors along a parity check line are multiplied to obtain the product of the factors of this line. One can delete any term which contains the line after all the products of the factors with the same slope have been obtained, because that line will not appear in any other product of the factors. For example, after obtaining the product of the factors of line $a_3b_2c_1d_0$, one can delete any term which contains the line D0. Similarly, one can delete any term which contain the line D1 after obtaining the product of the factors of line $a_4b_3c_2d_1$; delete any term which contain the line D2 in the product of the factors of line $a_5b_4c_3d_2$; and delete any term which contain the line D3 after obtaining the product of the factors of line $a_6b_5c_4d_3$. Afterward, multiple these four products together and delete any term which contains the line B4.

The total generating function of the code is then the product of all the factors. From the total generating function, the weight distribution is calculated and is shown in the table:

The Weight Distribution of A Rate 1/4, P3, Cylindrical Code with Slope 1, 1/2 and 1/3

Weight	Number
0	1
6	7
8	14
10	28
12	49
14	314
16	49
18	28
20	14
22	7
28	1

4.2.1 Examples of Using the Generating Function

Example 1

A P3 Projection code with block length 10, one data row, three parity check rows and slopes 1, 1/2, 1/4.

Weight	Number
0	1
8	10
10	4
12	150
14	180
16	1205

The output bit error rate is

Input Error	Lower Bound	Upper Bound
0.1	0.257×10^{-3}	0.421×10^{-1}
0.05	0.807×10^{-4}	0.834×10^{-3}
0.01	0.497×10^{-6}	0.717×10^{-6}
0.001	0.676×10^{-10}	0.700×10^{-10}
0.0001	0.698×10^{-14}	0.700×10^{-14}

Example 2

This is a P2 Projection code with one data line, three parity check lines, block length 10, slopes 1, 1/2, 1/4.

Weight	Number
0	1
8	10
10	10
12	20
14	50
16	130

The output bit error rate is

Input Error	Lower Bound	Upper Bound
0.1	0.259×10^{-3}	0.129×10^{-1}
0.05	0.811×10^{-4}	0.532×10^{-3}
0.01	0.497×10^{-6}	0.717×10^{-6}
0.001	0.676×10^{-10}	0.701×10^{-10}
0.0001	0.698×10^{-14}	0.700×10^{-14}

Example 3

This is a P3 Projection code with three data lines, three parity check lines, block length 32, slopes 1, 1/2, 1/7. The codewords starting from column 0 have following weight distribution:

Weight	Number
0	1
8	15
10	0
12	186
14	578
16	998

The output bit error rate is

Input Error	Lower Bound	Upper Bound
0.005	2.436×10^{-9}	$4.351 x 10^{-7}$
0.001	8.285×10^{-12}	6.985×10^{-10}
0.0005	5.689×10^{-13}	4.370×10^{-11}
0.0001	9.814×10^{-16}	6.998×10^{-14}
0.00005	6.192×10^{-17}	4.374×10^{-15}
0.00001	9.981×10^{-20}	7.000×10^{-18}

We notice that some two rows of the Projection code P2 and P3 can be inverted without violating any parity check equations. So in the extended code, there is at least one codeword having weight $2l$. Therefore, when we try to increase the minimum distance by increasing r, we have to increase the block length exponentially also.

4.3 Applying the Generating Function to the Projection Code

Part II

Another method to use generating function to the analysis of Projection code is based on the following thesis [15].

$$\sum_{j=1}^{n} a_{ij}x_j = b_i; \quad i = 1,2,\ldots,m$$

$$0 \le x_j \le u_j,\ j = 1,2,\ldots n \qquad (4.19)$$

where x_j, a_{ij} and b_i are nonnegative integers.

Define

$$x = (x_1,\ldots,x_n), \quad b = (b_1,\ldots,b_m)$$

$$t = (t_1,\ldots,t_m), \quad t^b = \prod_{i=1}^{m} t_i^{b_i} \qquad (4.20)$$

The notation $x \ge 0$ or $b \ge 0$ means that the respective components are nonnegative (and in this case, also integral).

For $x \ge 0$ and $b \ge 0$, let N(b) be the number of solutions of above equation. Then

$$\sum_{b\ge 0} N(b)t^b = \prod_{j=1}^{n}\left(1 + B_j^1 + \ldots + B_j^{u_j}\right) \qquad (4.21)\}$$

where

$$B_j = t_1^{a_{1j}} t_2^{a_{2j}} \cdots t_m^{a_{mj}}$$

Proof:

$$\sum_{b \geq 0} N(b) t^b = \sum_{x \geq 0} t_1^{\sum_j a_{1j} x_j} \cdots t_m^{\sum_j a_{mj} x_j}$$

$$= \prod_{j=1}^{n} \left(\sum_{x_j = 0}^{u_j} t_1^{a_{1j} x_j} \cdots t_m^{a_{mj} x_j} \right)$$

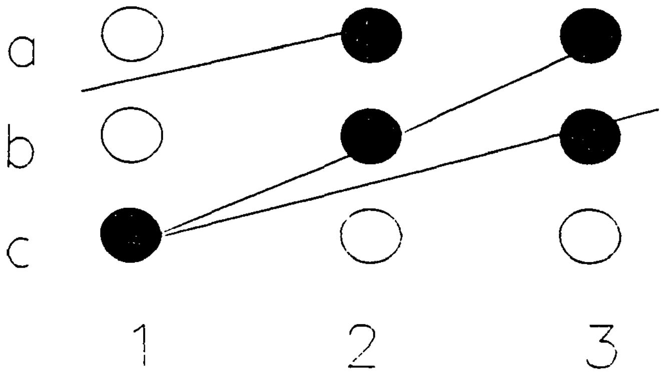
$$= \prod_{j=1}^{n} \sum_{x_j = 0}^{u_j} \left(t_1^{a_{1j}} t_2^{a_{2j}} \cdots t_m^{a_{mj}} \right)^{x_j}$$

$$= \prod_{j=1}^{n} \left(1 + t_1^{a_{1j}} t_2^{a_{2j}} \cdots t_m^{a_{mj}} + \cdots + \left(t_1^{a_{1j}} t_2^{a_{2j}} \cdots t_m^{a_{mj}} \right)^{u_j} \right) \tag{4.22}$$

Here we assume all a_{ij} are nonnegative to insure at most a finite number of solutions to above equations.

Example

To obtain the number of codewords with weight n of the P3 code shown in Fig.11. This code has one data row, two parity check rows, slopes 1 and 1/2, block length 3.



A P3 Cylindrical Projection Code

Fig 10

For the line B0, the summation of all the variables on that line should equal to an even number. Hence

$$a_2+b_0+c_1=2y_1 \tag{4.23}$$

Similarly, for the line B1, the equation is

$$a_0+b_1+c_2=2y_2 \tag{4.24}$$

For B2,

$$a_1+b_2+c_0=2y_3 \tag{4.25}$$

For C0,

$$a_2+b_1+c_0=2y_4 \tag{4.26}$$

For C1

$$a_0 + b_2 + c_1 = 2y_5 \tag{4.27}$$

For C2

$$a_1 + b_0 + c_2 = 2y_6 \tag{4.28}$$

The summation of all the variables should be equal to n, that is

$$\sum_{i=0}^{2} (a_i + b_i + c_i) = n \tag{4.29}$$

In this example,

$$\sum_{i=1}^{6} 2y_i = 2n \tag{4.30}$$

Rewrite these equations,

$$a_2+b_0+c_1+\sum_{\substack{i=1\\i\neq 1}}^{6}2y_i=2n$$

$$a_0+b_1+c_2+\sum_{\substack{i=1\\i\neq 2}}^{6}2y_i=2n$$

$$a_1+b_2+c_0+\sum_{\substack{i=1\\i\neq 3}}^{6}2y_i=2n$$

$$a_2+b_1+c_0+\sum_{\substack{i=1\\i\neq 4}}^{6}2y_i=2n \tag{4.31}$$

$$a_0+b_2+c_1+\sum_{\substack{i=1\\i\neq 5}}^{6}2y_i=2n$$

$$a_1+b_0+c_2+\sum_{\substack{i=1\\i\neq 6}}^{6}2y_i=2n$$

$$\sum_{i=1}^{6}y_i=n$$

Using the theorem, we obtain the result

The number of solutions with weight equaling to 0 is 1;

The number of solutions with weight equaling to 4 is 9;

The number of solutions with weight equaling to 6 is 6;

The number of solutions with other weight is 0;

5 Improved Importance Sampling Technique

5.1 The Importance Sampling Technique

It is well known that in a linear block code, the probability bit error rate of a code over symmetric memoryless channel is,

$$P_b = \frac{1}{N}\sum_{j=1}^{M-1} |C(j)| P(C(j)/C(0)) \quad (5.1)$$

where $P(C(j)/C(0))$ is the probability of making specific decoding error that a codeword $C(j) \neq C(0)$ is decoded when $C(0)$ is transmitted.

When codeword $C(0)$ is transmitted, the true joint density of the channel data output is $f(V/C(0))$ where V is a received word. To apply importance sampling, we use a different joint density function, denoted as $f^*(V/C(0))$, which is called the simulation density. To restore the true output error rate, the simulated output error rate will be multiplied by the simulation weight $f(V/C(0))/f^*(V/C(0))$.

This importance sampling technique [17] is designed for complete decoder. However, in many applications, the incomplete decoder is the only practical choice. For example, the practical Projection decoder is incomplete decoder. Therefore, we have no way to use the available importance sampling technique directly. Another problem of the importance sampling technique in [17] is that it only checks whether the output of decoder is the biasing codeword or not. If it is the biasing codeword, it declares errors happened. If it is not the biasing codeword, however, it declares no errors happened. Hence it can not distinguish the better decoder with worse one. In fact, it just reverses the batter one with the worse one.

To overcome the problem, a modified importance sampling simulation technique will be put forward, which can simulate a incomplete decoder as well as a complete decoder and tell the better decoder from the worse one.

5.2 The Improved Importance Sampling Technique

Suppose all zero codeword is transmitted. We bias the all-zero codeword, $C(0)$, as follows:

$V_k(j)=1$ with probability 0.5, if $C_k(j)=1$, biased part

$V_k(j)=1$ with probability p, if $C_k(j)=0$, unbiased part

where $V_k(j)$ is the kth component of the biased word $V(j)$ according to codeword $C(j)$.

We bias every codeword $C(j)$ for L times. Each time the biased word may be different. Denote, in the ith simulation, the number of 1 in the biased part as $x^i(j)$ and the number of 1 in the unbiased part as $u^i(j)$. Let $e_i(j)$ stand for the decoded errors of the $V(j)$ at the ith run. We use $Pn^i(j)$ to stand for the pattern of $V(j)$ at the ith run. Without biasing, the pattern $Pn^i(j)$, will occur with probability of $p^{x^i(j)+u^i(j)}(1-p)^{N-x^i(j)-u^i(j)}$. With biasing, this pattern will occur with probability of $p^{u^i(j)}(1-p)^{N-|C(j)|-u^i(j)}(0.5)^{|C(j)|}$. The simulation weight is $f^i(j)=(2p)^{x^i(j)}(2(1-p))^{|c^{(j)}|-x^i(j)}$. The enlarged factor in simulation at ith run, $f^i(j)$, is the inverse of simulation weight. $f^i(j)=(2p)^{-x^i(j)}(2(1-p))^{x^i(j)-|c^{(j)}|}$. The true average error over L runs will be estimated by:

$$\hat{E}(j)=\frac{1}{L}\sum_{i=1}^{L}e^i(j)f^i(j)^{-1}$$

$$=\frac{1}{L}\sum_{i=1}^{L}e^i(j)(2p)^{x(j)}(2(1-p))^{|C(j)|-x(j)} \quad (5.2)$$

We bias $C(0)$ for every codeword $C(j)$ and simulate it for L times. Denote the largest enlarged factor as $f(m)$ in all the simulation runs. Note in $f(m)$ simulation runs without biasing, the pattern $P^i(j)$ will occur $\frac{f(m)}{f(j)}$ times, the pattern $P^i(n)$ will occur $\frac{f(m)}{f(n)}$, and etc.. Therefore, the estimated average bit errors is,

$$\begin{aligned}\hat{E} &= \frac{1}{f(m)}\sum_{j=1}^{M-1}\sum_{i=1}^{L}\frac{f(m)}{f(j)}e^i(j)\\ &= \sum_{j=1}^{M-1}\hat{E}(j)\\ &= \frac{1}{L}\sum_{j=1}^{M-1}\sum_{i=1}^{L}e^i(j)(2p)^{x(j)}(2(1-p))^{|C(j)|-x(j)} \qquad (5.3)\end{aligned}$$

The estimated output BER, $\hat{P}_b$, is the estimated average bit errors divided by N.

$$\hat{P}_b = \frac{\hat{E}}{N} \qquad (5.4)$$

5.3 An Example

Using the third example in Section 4.2.1, we set L=1000, use David Manela's decoding method and stop decoding after 20 times failture in decoding a word. The simulation result by using importance sampling technique is:

Input Error Rate	Simulation Output Error Rate
0.005	4.6×10^{-8}
0.001	5.7×10^{-11}
0.0005	1.0×10^{-11}
0.0001	9.5×10^{-15}
0.00005	5.7×10^{-17}
0.00001	9.5×10^{-20}

6 Discussion

Due to the fact that not all the solutions of the equations based on the structure of a cylindrical Projection code correspond to codewords of the cylindrical Projection code, this situation exists: for some solutions, the parity check lines are satisfied but the corresponding codes are not codewords. For example, in the situation of P3, suppose two data rows are all 1 and all other rows are all 0. The parity check lines are satisfied but this pattern is not a codeword for P3. However, if the blocklength of a Projection code is long enough, its real weight distribution can be identified with the solution of the equations when only small weights are taken into consideration.

The lower bound should be independent or has little relation with the block length when the block length is long enough as in the upper bound. Further study will be necessary.

7 Conclusions

Two methods to obtain the generating functions of the Projection codes have been defined and analyzed. Using generating functions, we obtained the weight distribution of P2 and P3 Projection codes and then obtained bounds on the error performance of P2 and P3 codes.

Optimum decoders were obtained and by carefully choosing the slopes, for the same number of data rows , for the same number of parity check rows and for the same long block length, we have shown that P2 and P3 codes have almost the same error performance.

Both the first and the second methods for generating functions have very strong structure. Therefore, we hope, some much simpler calculation means can be found.

The Importance Sampling method was a tool to simulate the Projection Code when the input error rate is very small and other ways failed to provide appropriate results. Its use in simulating other forward error correction codes should be explored in the future.

8 References

[1] David Manela, "A New Class of Forward Error Correcting Codes for Burst and Random Errors," Ph.D. Dissertation of CUNY, 1987.

[2] Gary R. Lomp, "A New Burst & Random Error Correcting Code: Nonbinary Projection Code," Final Report, NSF Award Number ISI-8860421.

[3] H. Taub and D. L. Schilling, "Principles of Communication Systems," second edition, McGraw-Hill, New York, 1986.

[4] C. E. Shannon, "A Mathematical Theory of Communication," Bell Syst. Tech. J., vol. 27, pp 379-423 (Part I), pp.623-656 (Part II), July 1948.

[5] A. J. Viterbi and J. K. Omura, "Principles of Digital Communication," McGraw_Hill, New York, 1979.

[6] J. G. Proakis, "Digital Communications," McGraw_Hill, New York, 1983.

[7] R. E. Blahut, "Theory and Practice of Error Control Codes," Addison Wesley, Reading, Massachusetts, 1983.

[8] A. J. Viterbi, "Error Bounds for Convolutional Codes and Asymptotically Optimum Decoding Algorithm," IEEE Trans. Inf. Theory, IT-13 pp.260-269, April 1967.

[9] D. J. Costello, Jr., "Free Distance Bounds for Convolutional Codes," IEEE Trans. Inf. Theory, IT-20,pp.356-365, May 1974.

[10] E. R. Berlekamp, "Algebraic Coding Theory," McGraw-Hill, New York, 1968.

[11] W. W. Peterson and E. J. Weldon, "Error-correcting Codes," second edition, MIT, Press, Cambridge, Mass., 1972.

[12] S. Lin and D. J. Costello, Jr., "Error Control Coding: Fundamentals and Applications," Prentice-Hall, New Jersey, 1983.

[13] F. J. MacWilliams and N. J. A. Sloane, "The Theory

of Error-correcting Codes," North-Holland, Amsterdam, Netherlands, 1978.

[14] Alan Tucker, "Applied Combinatorics," Addison_Wesley, Reading, Mass., 1979.

[15] Gregory M. Constantine, "Combinatorial Theory and Statistical Design," John Wiley & Sons, New York, 1971.

[16] Dong Li, "Bounds for Comvolutional Projection Code," Ph.D dissertation of CUNY, 1993.

[17] Johns S. Sadowsky, "A New Method for Viterbi Decoder Simulation Using Importance Sampling," IEEE Trans. Commun., vol. 38, pp. 1341-1351, September 1990.

[18] K. S. Shanmugan and P. Balaban, " A Modified Monte Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems," IEEE Trans. Commun., vol.28, pp. 1916-1924, November 1980.

[19] M. C. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication

Systems," IEEE J. Select. Area Commun., vol.SAC-2, pp. 153-170, January 1984.

[20] B. R. Davis, "An Improved Importance Sampling Method for Digital Communication System Simulations," IEEE Trans. Commun., vol.34, pp.715-719, July 1986.

[21] P. M. Hahn and M. C. Jeruchim, "Developments in the Theory and Application of Importance Sampling Technique for Efficient Simulation of Digital Communications Systems," IEEE Trans. Commun., vol.35, pp.706-714, July 1987.

[22] D. Lu and K. Yao, "Improved Importance Sampling Technique for Efficient Simulation of Digital Communications Systems," IEEE J. Select. Area Commun., vol.6, pp.67-75, January 1988.

[23] M. C. Jeruchim, P. M. Hahn, K. P. Smyntek, and R. T. Ray, "An Experimental Investigation of Conventional and Efficient Importance Sampling," IEEE Trans. Commun., vol.37, pp.578-587, June 1989.

[24] M. A. Herro and J. M. Nowack, "Simulated Viterbi Decoding Using Importance Sampling," Proc. IEE, vol.135, part F, pp.133-142, April 1988.

[25] G. Orsak and B. Aazhang, "On the Theory of Importance Sampling Applied to the Analysis of Detection Systems," IEEE Trans. Commun., vol.37, pp.332-339, April 1989.