A HIGH RATE CODING SCHEME FOR BYTE-ORIENTED INFORMATION SYSTEMS

Hari Krishna

A Master Thesis

in

The Department

of

Electrical Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering at
Concordia University
Montreal, Quebec, Canada

December 1982

ABSTRACT

A HIGH RATE CODING SCHEME FOR BYTE-ORIENTED INFORMATION SYSTEMS

Hari Krishna

In this thesis, the mapping of non-binary codes defined over GF(2^m) into binary codes is studied and a binary coding scheme is derived that can be used to provide additional error-protection from random and burst errors for data consisting of bytes that have even parity. The emphasis here is on high rate codes.

A closed form decoding algorithm is given for the cases when two or three parity bytes are added to every block of k information bytes. An erasure is defined in the context of such systems. The decoding algorithm makes use of the parity bit that is present in every received byte.

Finally, the statistical performance of the coding scheme is analysed for the cases of interest on the non-binary and the binary symmetric channel models. For the non-binary symmetric channel, the Hamming weight distribution of the code is used to compute the probability of various post decoding events. The complete weight enumerator of the dual code and MacWilliams theorem are utilized for evaluating the performance on the binary symmetric channel.

To my parents

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. Vijay K. Bhargava and Dr. Gérald E. Séguin for their constant encouragement, help and guidance throughout the course of this research work. I owe much of my understanding of algebraic coding theory to them.

I would also like to thank Miss Anita Claassen for her excellent typing of the manuscript.

Finally, I thank all my friends who have their office in the Electrical Engineering Graduate Research Laboratory for their cooperation and help from time to time.

This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant G 0649.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF IMPORTANT SYMBOLS

(n,k)	Linear code of length n and k information symbols.
q	Number of distinct input symbols.
$g(X)$	Generator polynomial of a cyclic code.
$GF(p^m)$	Galois Field of p^m elements.
RS code	Reed-Solomon code.
d	Minimum distance of a (n,k) code.
α	Primitive element of $GF(p^m)$.
G	Generator matrix.
$c(X)$	Code polynomial.
$e(X)$	Error polynomial.
S_i	ith syndrome.
P_{ud}	The probability of undetected error.
P_{CD}	The probability of correct decoding.
P_{ICD}	The probability of incorrect decoding.
P_F	The probability of decoding failure.
P_{SE}	Post decoder symbol error rate.
ϵ	Input symbol error rate.
P	Input bit error rate.
$A(h)$	Number of code words of Hamming weight h in a code.
$\mathcal{C}$	Linear (n,k) code.
$\mathcal{C}^{\perp}$	Dual of a linear (n,k) code.
χ_β	A character of $GF(q)$.

CHAPTER 1

Introduction

In this thesis, a coding scheme is presented for application to even parity byte-oriented information systems (for example, data in ASCII format, Telidon Broadcast System). In such systems, the information consists of k information bytes $\underline{b}_1$, $\underline{b}_2, \ldots, \underline{b}_k$, where each $\underline{b}_i$ consists of eight bits of even parity (although a similar analysis can be performed for odd parity). The constraints in encoding such information are

(1) The parity bit in each information byte cannot be altered, i.e., the encoding must be systematic,

(2) The number of overall parity bits used must be a multiple of 8 and

(3) The rate of the code should be high.

It should be emphasized that if the parity bits in the information bytes are allowed to be recomputed in a manner up to the system designer then the coding problem reduces to the standard coding problem and the designer can use any one of the available error correcting codes such as the BCH codes [1]. Since the parity bit is not to be changed, the overall performance of the coding scheme will depend on its use of these added parity bits in the error control process.

Readers not familiar with the theory of error correcting codes may find references [2], [3] and [4] useful. A comprehensive coverage of algebraic coding theory and of the relative interfaces between algebraic coding theory and surrounding areas is contained in reference [5]. Also the theory of rings and finite fields is not discussed here and the reader is referred to [3] for it.

In this chapter, linear block codes and other relevant topics are discussed briefly and the plan of the thesis is described.

1.1 Linear Block Codes

Let q denote the number of distinct symbols used on the channel. A block code is a set of M sequences of channel symbols of length n. These q-ary n-tuples are called the code words of the code. The number of code words is taken to be a power of q, i.e. $M = q^k$.

The set of all n-tuples with entries chosen from the field of q elements is a vector space. A set of these vectors of length n is called a linear block code if and only if it is a subspace of the vector space of n-tuples. If the dimension of the subspace is k, then such a code is called an (n,k) code.

The Hamming distance between two vectors v_1 and v_2 is defined to be the number of positions in which the two vectors differ. The Hamming weight of a vector v, denoted by $w(v)$, is defined to be the number of non-zero components of v. Thus Hamming distance between two vectors v_1 and v_2 is $w(v_1 - v_2)$.

If c_1 and c_2 are both code words of a linear block code, then $c_1 - c_2$ must also be a code word, since the set of all code words is a vector space. Therefore, the distance between any two code words equals the weight of some other code word and the minimum distance d for a linear code equals the minimum weight of its non-zero vectors.

1.2 Cyclic Codes

A subspace V of n-tuples is called a cyclic subspace or a cyclic code if for each vector $c = (c_0, c_1, \ldots, c_{n-1})$ in V, the vector

$c' = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$ obtained by shifting the components of c cyclically one unit to the right is also in V. We will represent the components of a code vector as coefficients of a polynomial as follows

$$c = (c_0, c_1, \ldots, c_{n-1}) \Longleftrightarrow \qquad (1.2.1)$$
$$c(X) = c_0 + c_1 X + \cdots + c_{n-1} X^{n-1}$$

1.2.1 Generator Polynomial

Let $g(X) = g_0 + g_1 X + \cdots + g_{r-1} X^{r-1} + X^r$ be a non-zero code polynomial of minimum degree in the (n,k) cyclic code. It can be shown that $r = n-k$ and a polynomial $c(X)$ of degree $n-1$ or less is a code polynomial if and only if $c(X)$ is a multiple of $g(X)$.

The polynomial $g(X)$ is called the generator polynomial of the cyclic code. Thus every code polynomial $c(X)$ in an (n,k) cyclic code can be expressed as

$$\begin{aligned} c(X) &= m(X)\, g(X) \\ &= (m_0 + m_1 X + \cdots + m_{k-1} X^{k-1})\, g(X) \end{aligned} \qquad (1.2.2)$$

where $m(X)$ is the polynomial corresponding to the k information digits $(m_0, m_1, \ldots, m_{k-1})$.

1.2.2 Minimum Polynomial

Let α be an arbitrary element of the Galois field $GF(2^m)$. The monic[1] polynomial $m(X)$ of smallest degree with binary coefficients such that $m(\alpha) = 0$, is called the minimum polynomial of α. The minimum polynomial of α is irreducible.

[1] A polynomial is called monic if the coefficient of the highest power of X is 1.

1.2.3 BCH Codes

Let α be an element of $GF(p^m)$. For any specified m_0 and d_0, the code generated by $g(X)$ is a BCH code if and only if $g(X)$ is the polynomial of lowest degree over $GF(p)$ having $\alpha^{m_0}, \alpha^{m_0+1}, \ldots, \alpha^{m_0+d_0-2}$ as its roots. The length of the code is the least common multiple of the orders of the roots. The minimum distance of the code is at least d_0 and d_0 is called the designed distance [3]. The most important BCH codes are the binary codes obtained by letting α be a primitive element of $GF(2^m)$ and letting $m_0 = 1$ and $d_0 = 2t_0+1$. The generator polynomial of the t_0 error correcting code has $\alpha, \alpha^2, \ldots, \alpha^{2t_0}$ as its roots and is given by

$$g(X) = LCM(m_1(X), m_2(X), \ldots, m_{2t_0}(X)) \tag{1.2.3}$$

However, every even power of α is a root of the same minimum function as some previous odd power of α. Thus the generator polynomial of the code is

$$g(X) = LCM(m_1(X), m_3(X), \ldots, m_{2t_0-1}(X)) \tag{1.2.4}$$

As the degree of each minimum polynomial is m or less, the degree of $g(X)$ is at most mt_0 and the code has at most mt_0 parity checks. Hence such a code has the following parameters.

$$\begin{array}{ll} \text{Block length:} & n = 2^m-1 \\ \text{Number of parity check bits:} & n-k \leq mt_0 \\ \text{Minimum distance:} & d \geq 2t_0+1. \end{array} \tag{1.2.5}$$

This code is capable of correcting any combination of t_0 or fewer errors in a block of $n = 2^m-1$ bits.

1.2.4 Reed-Solomon Codes [6]

A Reed-Solomon code is a BCH code of length $n = q-1$ over GF(q), where $q = p^m$ and p is a prime number. Of course, q is never equal to 2. Thus the length of the code is the number of non-zero elements in the ground field.

The generator polynomial for such a code has $\alpha, \alpha^2, \ldots, \alpha^{\delta-1}$ as its roots for the minimum distance to be δ. Since the minimal polynomial of α^i is $m^{(i)}(X) = (X - \alpha^i)$, a RS code of length q-1 and designed distance δ has generator polynomial

$$g(X) = (X - \alpha)(X - \alpha^2) \ldots (X - \alpha^{\delta-1}). \tag{1.2.6}$$

As the degree of g(X) is δ-1, the RS code generated by g(X) has the following parameters

$$\begin{aligned} &\text{Block length:} && n = q-1 \text{ symbols} \\ &\text{Number of parity symbols:} && n-k = \delta-1 \\ &\text{Minimum distance:} && d = \delta \end{aligned} \tag{1.2.7}$$

Since $d = n-k+1$ for RS codes, these codes are called "maximum distance separable".

1.3 Shortened Cyclic Codes

In system design, if a code of suitable block length n or suitable dimension k cannot be found, it is natural to look for linear codes that, though actually not cyclic, share the mathematical structure and ease of implementation of cyclic codes. A technique for shortening a cyclic code is described in the following.

Given an (n,k) linear code, it is always possible to form an (n-i,k-i) linear code by making the i leading information symbols identically 0 and omit them from all code vectors. This corresponds to omitting the first i rows and columns of the generator matrix or the first i columns from the parity check matrix. The resulting code is called a shortened code and in general, is not cyclic.

A shortened code has at least the same error-correcting capability as the code from which it is derived.

1.4 Plan Of The Thesis

The thesis is divided into six chapters and a very brief description of these chapters is as follows.

In chapter 2, the encoding scheme is described for the byte oriented information systems. The generator matrix is obtained in systematic form for the values of d equal to 3 and 4. The encoding procedure is analysed for system implementation.

In chapter 3, the decoding algorithm is discussed. A simplified closed form decoding algorithm is given by Séguin [7] for d equal to 3. It has been further extended to the case of d equal to 4. The decoding algorithm makes use of the parity bit that is present in every information and parity bytes.

In chapter 4, expressions are derived for the statistical performance of the coding scheme over the q-ary symmetric channel for d equal to 3 and 4. These expressions are evaluated using the computer for different values of channel symbol error rate.

In chapter 5, the performance of the code is analysed on a binary symmetric channel model for d equal to 3. The expressions for the probability of various post decoder events, are derived in terms of the complete weight enumerator of the dual code which is generated on the computer.

Chapter 6 is the summary and conclusion.

CHAPTER 2

A Coding Scheme for Byte-Oriented Information Systems

This chapter describes a coding scheme that can be used very effectively to correct both random as well as burst errors. All the bytes in a code word computed on the basis of the information bytes, have even parity. The procedure for obtaining binary generator matrices in systematic form is explained and a method to implement the encoding is discussed briefly.

2.1 Mapping $GF(2^m)$ Codes into Binary Codes

We know that $GF(p^m)$ is a vector space of dimension m over $GF(p)$. Therefore, any set of m linearly independent elements can be used as a basis for this vector space [1].

Let $\xi_1, \ldots, \xi_m$ be a basis for $GF(p^m)$ over $GF(p)$. Then if $\beta = \sum_{i=1}^{m} b_i \xi_i$ is any element of $GF(p^m)$, $b_i \in GF(p)$, we map β into $(b_1, b_2, \ldots, b_m)$. This mapping sends linear codes into linear codes (but cyclic codes may not map into cyclic codes).

Usually $1, \alpha, \ldots, \alpha^{m-1}$ is chosen to be the basis with α^i being represented as an m-tuple with only its $(i+1)$th or $(m-i)$th element as 1 and all other elements to be 0. With this basis, the elements of $GF(p^m)$ can be represented as m-tuples of elements from $GF(p)$. Hence an (n,k,d) RS code defined over $GF(p^m)$ becomes an $(n_b = mn,\ k_b = mk,\ d_b \geq d)$ code over $GF(p)$. If $p = 2$, we get binary codes from this mapping.

Let $c = (c_0, c_1, \ldots, c_{n-1})$ belong to an (n,k,d) RS code over $GF(2^m)$. If each of c_i is replaced by a binary m-tuple according to the mapping given above and an overall parity check is added on each m-tuple, then the resulting binary code has the following parameters,

$$n_b = (m+1)(2^m - 1)$$
$$k_b = mk \tag{2.1.1}$$
$$d_b \geq 2(2^m - k)$$

for any $k = 1, \ldots, 2^m - 2$.

2.2 Encoding Scheme

For the type of information systems being considered here, the information is byte structured with each byte having an overall parity bit. Therefore, there are 128 ($=2^7$) different values that any information byte can take.

It can be readily observed that all the 8-tuples having even number of ones form a vector space of dimension 7. Again taking $1, \alpha, \ldots, \alpha^6$ as the basis and representing them as,

$$\begin{aligned}
1 &= (1,0,0,0,0,0,0,1) \\
\alpha &= (1,0,0,0,0,0,1,0) \\
\alpha^2 &= (1,0,0,0,0,1,0,0) \\
\alpha^3 &= (1,0,0,0,1,0,0,0) \\
\alpha^4 &= (1,0,0,1,0,0,0,0) \\
\alpha^5 &= (1,0,1,0,0,0,0,0) \\
\alpha^6 &= (1,1,0,0,0,0,0,0)
\end{aligned} \tag{2.2.1}$$

it can be shown that all the even parity 8-tuples can be represented as elements of $GF(2^7)$.

Hence let us define a RS code over $GF(2^7)$ for such an information system. This code has the following parameters

$$
\begin{aligned}
n &= 127 \\
k &= n - \delta + 1 \qquad\qquad (2.2.2) \\
d &= \delta .
\end{aligned}
$$

Each symbol c_i in a code word obtained as a result of this encoding is written as $c_i = b_0\alpha^0 + b_1\alpha + \cdots + b_6\alpha^6$, $b_j \in GF(2)$ and the corresponding even parity 8-tuple is obtained by replacing α^j by its 8-tuple representation defined by Equation (2.2.1).

Thus we have been able to introduce redundancy into each of the bytes that are transmitted. The parameters of the resulting binary code are

$$
\begin{aligned}
n_b &= 8 \cdot 127 \\
k_b &= 7 \cdot k \qquad\qquad (2.2.3) \\
d_b &\geq 2(128 - k).
\end{aligned}
$$

The code length and its dimensions can be shortened and the encoder can be put in the systematic form so as to match with the overall system requirements. A generator matrix can be obtained in the systematic form for the above RS code [3] and it is of the form

$$G = [P, I_k]$$

where I_k is the $k \times k$ identity matrix and P is $k \times (n-k)$ matrix. If $\underline{m}$ represents the information vector, then the encoded vector is of the form

$$\underline{c} = [\underline{m}P, \underline{m}].$$

The field $GF(2^7)$ may be generated by the recursion

$$\alpha^7 = 1 + \alpha^3 .$$

where α is a primitive element of $GF(2^7)$. Using this recursion and the basis defined by Equation (2.2.1), the complete table of $GF(2^7)$ is generated and is given in appendix A.

The complete encoding procedure can be described in the following three steps.

1. Represent each of the even parity information bytes as elements of $GF(2^7)$.
2. Take k of these symbols and encode it using the RS code defined over $GF(2^7)$ to get a code vector of length n.
3. Replace each one of the n symbols in the code word by its corresponding 8-tuple binary representation as explained above. Note that all the 8-tuples have even parity.

A binary generator matrix of size $k_b \times n_b$ can be obtained in systematic form for the above described encoding procedure. This is further illustrated for case of d equal to 3 and 4.

Since in many system applications, only codes of high rate can be considered, the two cases of d equal to 3 and 4 are of greater practical interest and, therefore, we will analyse these in detail.

2.3 Coding for d Equal to 3

The generator polynomial of a RS code defined over $GF(2^7)$ and having a minimum distance $d = 3$, is

$$g(X) = (X+\alpha)(X+\alpha^2)$$
$$= X^2 + \alpha^{32}X + \alpha^3.$$

This polynomial generates a RS code of length 127. Let us assume that the code is to be shortened to k equal to 25. We do this in order to make the analysis more relevant for the signal formats described in [7] for Canadian Telidon System. Hence the shortened RS code has the following parameters

$$n = 27 \qquad k = 25 \qquad d = 3 \tag{2.3.1}$$

The generator matrix G in systematic form for this code is as follows.

$$G = \begin{bmatrix}
\alpha^{3} & \alpha^{32} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\alpha^{35} & \alpha^{105} & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\
\alpha^{108} & \alpha^{96} & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\
\alpha^{99} & \alpha^{55} & & & & & & & \\
\alpha^{58} & \alpha^{115} & & & & & & & \\
\alpha^{118} & \alpha^{105} & & & & & & & \\
\alpha^{108} & \alpha^{97} & & & & & & & \\
\alpha^{100} & \alpha^{21} & & & & & & & \\
\alpha^{24} & \alpha^{15} & & & & & & & \\
\alpha^{18} & \alpha^{13} & & & & & & & \\
\alpha^{16} & \alpha^{8} & & & & & & & \\
\alpha^{11} & \alpha^{72} & & & & & & & \\
\alpha^{75} & \alpha^{115} & & & & & & & \\
\alpha^{116} & \alpha^{46} & & & & & & & \\
\alpha^{49} & \alpha^{99} & & & & & & & \\
\alpha^{102} & \alpha^{54} & & & & & & & \\
\alpha^{57} & \alpha^{74} & & & & & & & \\
\alpha^{77} & \alpha^{93} & & & & & & & \\
\alpha^{96} & \alpha^{62} & & & & & & & \\
\alpha^{65} & \alpha^{29} & & & & & & & \\
\alpha^{32} & \alpha^{58} & & & & & & & \\
\alpha^{61} & \alpha^{107} & & & & & & & \\
\alpha^{110} & \alpha^{48} & & & & & & & \\
\alpha^{51} & \alpha^{79} & & & & & \cdots & 1 & 0 \\
\alpha^{82} & \alpha^{49} & 0 & 0 & & & \cdots & 0 & 1
\end{bmatrix} \tag{2.3.2}$$

After mapping the above code into the binary code according to the encoding procedure described in section (2.2), we get a binary code having the following parameters

$$n_b = 216$$
$$k_b = 175 \qquad (2.3.3)$$
$$d_b \geq 6.$$

In order for this mapping of the RS code into the binary code to be complete, we should obtain the corresponding generator matrix having only binary elements.

It can be shown that the multiplication between α_1 and α_2, $\alpha_1, \alpha_2 \in GF(p^m)$ can be performed as a matrix multiplication of the form

$$\underline{a}_1 \times B = \underline{a}_2$$

where

$\underline{a}_1$ represents α_1 as m'-tuple of elements from $GF(p)$, $m' \geq m$,

B is $m' \times m'$ matrix of elements from $GF(p)$ determined uniquely from α_2 and

$\underline{a}_2$ represents the product $\alpha_1\alpha_2$ as an m'-tuple of elements from $GF(p)$.

Example (2.3.1)

In our case

$$p = 2 \ , \ m = 7 \ , \ m' = 8$$

and for the basis defined by Equation (2.2.1), the multiplication by $\alpha = (1\ 0\ 0\ 0\ 0\ 0\ 1\ 0) \in GF(2^7)$ can be performed by taking B as

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.3.4)$$

Generalizing the multiplication procedure, it can be shown that the binary matrix that corresponds to a multiplication by an element α^i, is given by

$$B = \begin{bmatrix} \underline{0} \\ \underline{b}_{i+6} \\ \underline{b}_{i+5} \\ \underline{b}_{i+4} \\ \underline{b}_{i+3} \\ \underline{b}_{i+2} \\ \underline{b}_{i+1} \\ \underline{b}_{i} \end{bmatrix} \qquad (2.3.5)$$

where each of $\underline{b}_i, \ldots, \underline{b}_{i+6}$ is a binary 8-tuple corresponding to $\alpha^i, \ldots, \alpha^{i+6}$ respectively and $\underline{0}$ is the all 0 8-tuple.

Hence the binary generator matrix can be obtained by replacing each element of the generator matrix G given in Equation (2.3.2) by its corresponding 8×8 binary matrix as described above.

2.4 Coding for d Equal to 4

The generator polynomial of the RS code for d equal to 4 is given by

$$g(X) = (X+\alpha)(X+\alpha^2)(X+\alpha^3)$$
$$= X^3+\alpha^{104}X^2+\alpha^{106}X+\alpha^6.$$

This polynomial generates a RS code of length 127. As was explained in section (2.3), let the code be shortened to $k = 25$. Hence the shortened code has the following parameters

$$\begin{aligned} n &= 28 \\ k &= 25 \\ d &= 4 \end{aligned} \tag{2.4.1}$$

The generator matrix G in systematic form for this code is as follows

$$G = \begin{bmatrix}
\alpha^{6} & \alpha^{106} & \alpha^{104} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\
\alpha^{110} & \alpha & \alpha^{40} & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\
\alpha^{46} & \alpha^{32} & \alpha^{116} & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\
\alpha^{122} & \alpha^{82} & \alpha^{7} & & & & & & & \\
\alpha^{13} & \alpha^{30} & \alpha^{56} & & & & & & & \\
\alpha^{62} & \alpha^{81} & \alpha^{37} & & & & & & & \\
\alpha^{43} & \alpha^{21} & \alpha^{79} & & & & & & & \\
\alpha^{85} & \alpha^{106} & \alpha^{123} & & & & & & & \\
\alpha^{2} & \alpha^{27} & \alpha^{114} & & & & & & & \\
\alpha^{120} & \alpha^{15} & \alpha^{106} & & & & & & & \\
\alpha^{112} & \alpha^{76} & \alpha^{37} & & & & & & & \\
\alpha^{43} & \alpha^{113} & \alpha^{16} & & & & & & & \\
\alpha^{22} & \alpha^{107} & \alpha^{116} & & & & & & & \\
\alpha^{122} & \alpha^{75} & \alpha^{99} & & & & & & & \\
\alpha^{105} & \alpha^{87} & \alpha^{106} & & & & & & & \\
\alpha^{112} & \alpha^{32} & \alpha^{80} & & & & & & & \\
\alpha^{86} & \alpha^{5} & \alpha^{118} & & & & & & & \\
\alpha^{124} & \alpha^{120} & \alpha^{105} & & & & & & & \\
\alpha^{111} & \alpha^{105} & \alpha^{40} & & & & & & & \\
\alpha^{46} & \alpha^{102} & \alpha^{35} & & & & & & & \\
\alpha^{41} & \alpha^{114} & \alpha^{112} & & & & & & & \\
\alpha^{118} & \alpha^{16} & \alpha^{101} & & & & & & & \\
\alpha^{107} & \alpha^{38} & \alpha^{72} & & & & & & & \\
\alpha^{78} & \alpha^{75} & \alpha^{72} & & & & & \cdots & 1 & 0 \\
\alpha^{78} & \alpha^{41} & \alpha^{104} & 0 & 0 & & & \cdots & 0 & 1
\end{bmatrix} \tag{2.4.2}$$

After mapping the above code into a binary code according to the encoding procedure described in section (2.2), we get a binary code having the following parameters

$$n_b = 224$$

$$k_b = 175$$

$$d_b \geq 8.$$

The binary generator matrix corresponding to the generator matrix given above can be calculated according to the procedure described in section (2.3).

2.5 Encoding

There are two methods for encoding linear cyclic codes - the serial shift register method and the parallel matrix method.

Let $m(X)$ be a message polynomial with k symbols encoded into a code polynomial $c(X)$ with n symbols. In the serial shift register method, encoding in systematic form is done by dividing $X^{n-k}m(X)$ by $g(X)$ and appending the remainder $r(X)$ to $X^{n-k}m(X)$. Thus

$$c(X) = r(X) + X^{n-k}m(X) = q(X)\,g(X) \tag{2.5.1}$$

where $q(X)$ is the quotient. It indicates that $[r(X) + X^{n-k}m(X)]$ is a multiple of $g(X)$, and, therefore, is a code polynomial generated by $g(X)$. The code word generated is given by

$$(\underbrace{r_0, r_1, \ldots, r_{n-k-1}}_{\text{parity check symbols}}, \underbrace{m_0, m_1, \ldots, m_{k-1}}_{\text{message symbols}})$$

and the most significant symbol of the message, m_{k-1} is sent first.

Equation (2.5.1) can be implemented by a dividing circuit, which is (n-k)-stage shift register with feedback connections according to the generator polynomial. The feedback multipliers $g_0, g_1, \ldots, g_{n-k-1}$ are coefficients of the generator polynomial

$$g(X) = (X - \alpha)(X - \alpha^2) \cdots (X - \alpha^{n-k}).$$

An encoding circuit with an (n-k)-stage shift register is shown in Figure (2.1). In our case, each r_i register stage is a 8-tuple shift register. The encoding is accomplished as follows. With gate turned on, k information symbols are shifted into the encoder and simultaneously sent into the communication channel. Then the gate is turned off and the contents of the shift register are shifted out to the channel.

Figure (2.2) shows the (n-k)-stage shift register encoding circuit for the minimum distance d of the encoding scheme to be 3 and 4.

The parallel matrix method is more complex as compared to the serial method and is not described here. A binary generator matrix can be obtained for the coding scheme using the procedure given in section (2.3) and then encoding can be performed using this method. The reader is referred to [8] where the parallel matrix method is used to design and construct a (75,50) forward error correcting codec.

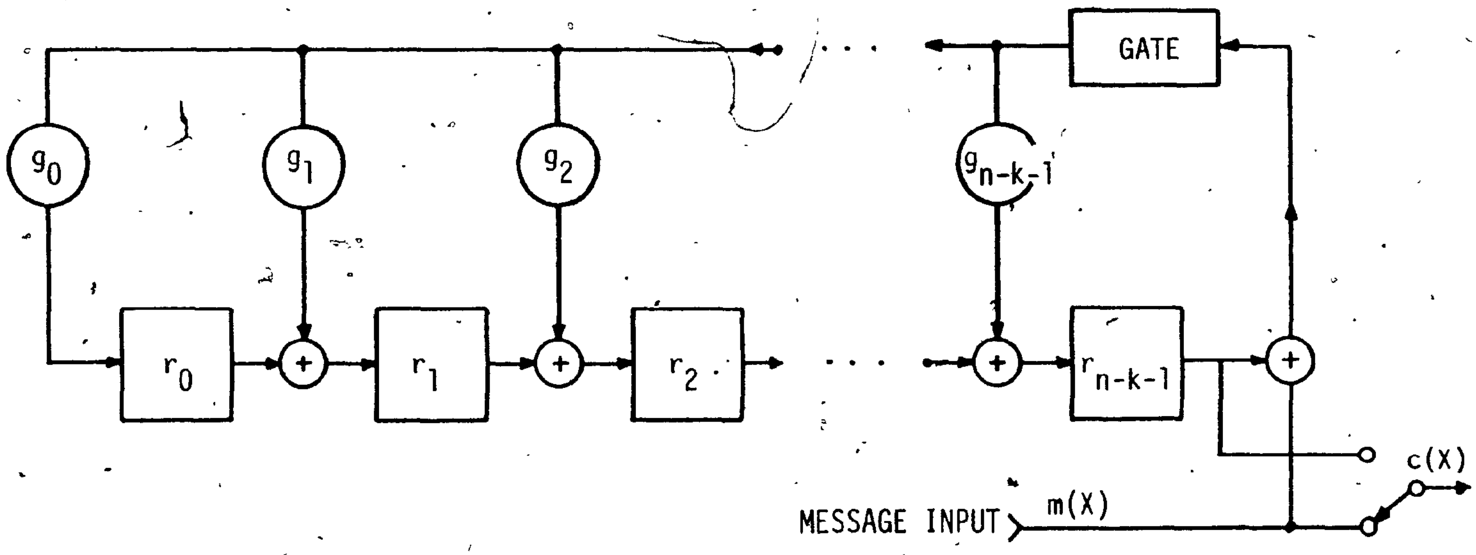

FIG. 2.1. ENCODER FOR (n,k) CODE: g_i IS AN ELEMENT OF GF(2^7) AND r_i IS AN 8-TUPLE SHIFT REGISTER STAGE.

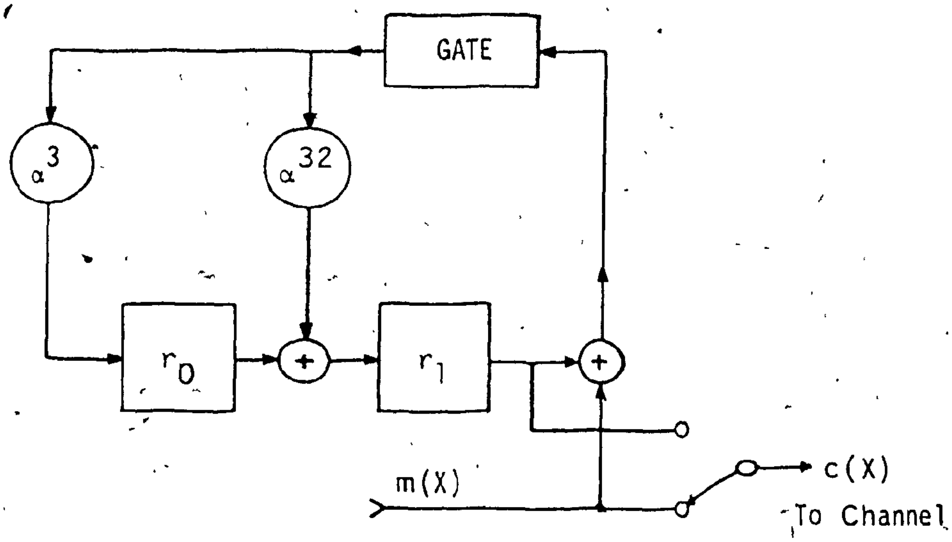

FIG. 2.2(a). ENCODER FOR (27,25) CODE

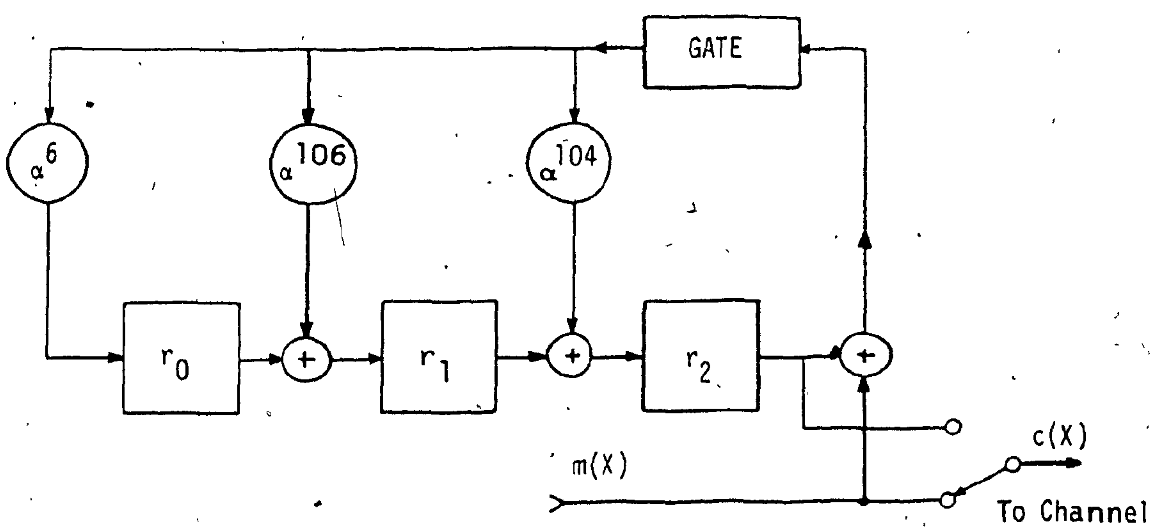

FIG. 2.2(b). ENCODER FOR (28,25) CODE

CHAPTER 3

The Decoding Algorithm

For the coding scheme described in chapter 2, all the bytes of any given code vector have even parity. To make use of this inherent redundancy in the code vector bytes, the decoder first verifies the parity in the received bytes $r_0, r_1, \ldots, r_{n-1}$. If any of them do not check out then there is a detectable error (though the value of the error is not known) in that position. Such a position is termed as an "erasure". The advantage of this becomes readily apparent by noting that a RS code with minimum distance d can correct t errors and s erasures provided

$$2t + s < d.$$

Hence the amount of redundancy required to correct an erasure is only half of that required to correct an error.

The Berlekamp-Massey decoding algorithm described in [5] can be used to correct the simultaneous occurrence of errors and erasures. The frequency domain decoding algorithm and the computational complexity associated with them has been discussed in [7] and it is shown that it is more promising to implement the transform technique for decoding erasures and errors of RS codes as compared to the time domain technique. The transform computations are independent of code rate and, therefore, the transform decoder is most efficient for low rate codes.

As the main emphasis in this thesis work is on high rate codes, a closed form decoding algorithm is described below for the minimum distance d of the code equal to 3 and 4. The decoding algorithm for d equal to 3 is given by Séguin [7] and here, it is extended to the case of d equal to 4. This algorithm makes use of the parity bit present in every received byte.

3.1 A Closed Form Decoding Algorithm for d Equal to 3

For the encoding scheme given in section (2.3), the minimum distance of the code is 3 and, therefore, it can correct t errors and s erasures if

$$2t + s < 3 \tag{3.1.1}$$

It is clear from Equation (3.1.1) that the received vector is correctly decoded to the transmitted code word, iff

(a) $s \leq 2$, $t = 0$, i.e. a maximum of two erasures and no error takes place.

(b) $t \leq 1$, $s = 0$, i.e. a maximum of one error and no erasure takes place.

Thus, the following decoding procedure is viable for such an encoding scheme.

(i) Two Detectable Errors in the Received Bytes

Let the error polynomial be

$$e(X) = e_i X^i + e_j X^j \tag{3.1.2}$$

where $e_i, e_j \in GF(2^7)$ and $1 \leq i < j$ and let $r(X)$ be the received polynomial, i.e.

$$r(X) = c(X) + e(X) \tag{3.1.3}$$

where $c(X)$ is the transmitted code vector. Then the two syndromes S_1 and S_2 are given by

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i + e_j\alpha^j \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} + e_j\alpha^{2j} \end{aligned} \tag{3.1.4}$$

and the values of i and j are known from the parity bits associated with the bytes in positions i+1 and j+1. We then have

$$(e_i \quad e_j)\begin{bmatrix} \alpha^i & \alpha^{2i} \\ \alpha^j & \alpha^{2j} \end{bmatrix} = (S_1 \quad S_2) \tag{3.1.5}$$

The latter matrix has an inverse which is

$$\frac{1}{\Delta}\begin{bmatrix} \alpha^{2j} & \alpha^{2i} \\ \alpha^j & \alpha^i \end{bmatrix} \tag{3.1.6}$$

where $\Delta = \alpha^{i+2j} + \alpha^{2i+j}$.

Multiplying both sides of Equation (3.1.5) by (3.1.6), we obtain

$$e_i = \frac{S_1\alpha^{2j} + S_2\alpha^j}{\Delta} \tag{3.1.7}$$

$$e_j = \frac{S_1\alpha^{2i} + S_2\alpha^i}{\Delta} \tag{3.1.8}$$

Hence the errors in positions i and j are determined by Equations (3.1.7) and (3.1.8) respectively.

(ii) A Single Detectable Error in the Received Bytes

Suppose there is a detectable error in position i, then the error polynomial is given by

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$ and the syndromes S_1 and S_2 are

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} \end{aligned} \tag{3.1.9}$$

The value of i is known. Thus,

$$e_i = \alpha^{-i} S_1. \qquad (3.1.10)$$

Also it can be observed that

$$S_2 = \alpha^i S_1. \qquad (3.1.11)$$

The Equation (3.1.10) can be used to calculate the value of the error at position i.

(iii) A Single Byte in Error

If there is a single byte in error, then the error polynomial is given by

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$ and i is not known.

The two syndromes S_1 and S_2 are calculated as

$$\begin{aligned} S_1 &= r(\alpha) = e_i \alpha^i \\ S_2 &= r(\alpha^2) = e_i \alpha^{2i} \end{aligned} \qquad (3.1.12)$$

Solving Equation (3.1.12) for the value of e_i and i, we get

$$\begin{aligned} e_i &= S_1^2 S_2^{-1} \\ \alpha^i &= S_2 S_1^{-1}. \end{aligned} \qquad (3.1.13)$$

A look up table of (i, α^i) can be used to determine i.

Hence, the decoding algorithm for the coding scheme presented in section (2.3) can be described as follows

Step I : Check the parity of each of the received bytes.

If they all check, go to II

If exactly one does not check, go to III

If exactly two do not check, go to IV

If more than two do not check, declare a decoding failure.

Proceed to next frame.

Step II : Represent all the bytes as elements of $GF(2^7)$ and compute the syndromes $S_1 = r(\alpha)$ and $S_2 = r(\alpha^2)$.

If both are 0, assume r(X) is error free.

If both are nonzero, assume a single symbol in error and decode it using the procedure (iii) described above.

If exactly one is 0, declare a decoding failure. Proceed to next frame.

Step III: Compute the syndromes. Check for Equation (3.1.11). If it is satisfied, decode it using procedure (ii) given above.

If Equation (3.1.11) is not satisfied, declare a decoding failure. Proceed to next frame.

Step IV : Assume exactly two symbols in error and correct them using procedure (i) described above. Proceed to next frame.

The above decoding algorithm ensures that a decoded code word has bytes that have even parity only. The complete decoding algorithm is given by the flowchart in Figure (3.1).

3.2 A Closed Form Decoding Algorithm for d Equal to 4

The encoding scheme and the generator matrix for d equal to 4 are given in section (3.4). It can correct t errors and s erasures if

$$2t + s < 4 \qquad (3.2.1)$$

Thus a received vector is correctly decoded to the transmitted code word if one of the following combinations of errors and erasures takes place

(a) No erasure, no error $(2t+s=0)$

(b) One erasure, no error $(2t+s=1)$

(c) Two erasures, no error $(2t+s=2)$

(d) Three erasures, no error $(2t+s=3)$

(e) One error, no erasure $(2t+s=2)$

(f) One error, one erasure $(2t+s=3)$.

A simplified decoding procedure described below is viable for such a coding scheme

(i) A Single Detectable Error in the Received Bytes

Let there be a detectable error in position i, then the error polynomial is

$$e(X) = e_i X^i$$

where $e_i \in GF(2^7)$. The syndromes S_1 and S_2 and S_3 are given by

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i \\ S_2 &= r(\alpha^2) = e_i\alpha^{2i} \\ S_3 &= r(\alpha^3) = e_i\alpha^{3i} \end{aligned} \tag{3.2.2}$$

Since the value of i is known, solving Equation (3.2.2) for e_i, we get

$$e_i = \alpha^{-i} S_1. \tag{3.2.3}$$

Also we have to perform a check that there is no error in the received bytes. This can be done by noting that

$$S_2 = e_i\alpha^{2i} = \alpha^{-i} S_1 \alpha^{2i} = S_1\alpha^i \tag{3.2.4}$$

and

$$S_3 = e_i\alpha^{3i} = \alpha^{-i} S_1 \alpha^{3i} = S_1\alpha^{2i}. \tag{3.2.5}$$

Combining Equations (3.2.4) and (3.2.5), we get

$$S_3S_1 = S_2^2 \tag{3.2.6}$$

Hence if there is only one detectable error, then

$$e_i = \alpha^{-i}S_1$$

and

$$S_3S_1 = S_2^2 .$$

(ii) Two Detectable Errors in the Received Bytes

Let there be detectable errors in positions i and j. The procedure for finding the error values at these positions is given in section (3.1). Additionally, we have to check if there are any other errors in the received bytes. It can be done by observing that

$$S_3 = e_i\alpha^{3i} + e_j\alpha^{3j} \tag{3.2.7}$$

Substituting for e_i and e_j from Equations (3.1.7) and (3.1.8) respectively, we get

$$S_3 = \alpha^{(i+j)}S_1 + (\alpha^i + \alpha^j)S_2$$

Thus if there are only two detectable errors, then

$$e_i = \frac{S_1\alpha^{2j} + S_2\alpha^j}{\Delta}$$

$$e_j = \frac{S_1\alpha^{2i} + S_2\alpha^{i}}{\Delta} \tag{3.2.8}$$

where $\Delta = \alpha^{i+2j} + \alpha^{j+2i}$,

and $S_3 = \alpha^{i+j}\, S_1 + (\alpha^{i} + \alpha^{j})S_2$

(iii) Three Detectable Errors in the Received Bytes

If there are detectable errors in positions i, j and k, then the error polynomial is

$$e(X) = e_iX^i + e_jX^j + e_kX^k$$

where e_i, e_j, $e_k \in GF(2^7)$ and $1 \leq i < j < k$.

The syndromes S_1, S_2 and S_3 calculated on the basis of the received vector are

$$\begin{aligned} S_1 &= e_i\alpha^{i} + e_j\alpha^{i} + e_k\alpha^{k} \\ S_2 &= e_i\alpha^{2i} + e_j\alpha^{2j} + e_k\alpha^{2k} \\ S_3 &= e_i\alpha^{3i} + e_j\alpha^{3j} + e_k\alpha^{3k}. \end{aligned} \tag{3.2.9}$$

Equation (3.2.9) can be rewritten in the matrix form as

$$\begin{bmatrix} \alpha^{i} & \alpha^{j} & \alpha^{k} \\ \alpha^{2i} & \alpha^{2j} & \alpha^{2k} \\ \alpha^{3i} & \alpha^{3j} & \alpha^{3k} \end{bmatrix} \begin{bmatrix} e_i \\ e_j \\ e_k \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} \tag{3.2.10}$$

or

$$A\,\underline{e} = \underline{S}.$$

The matrix A has an inverse which is given by

$$A^{-1} = \frac{1}{\Delta}\begin{bmatrix} \alpha^{2j+3k} + \alpha^{3j+2k} & \alpha^{j+3k} + \alpha^{3j+k} & \alpha^{j+2k} + \alpha^{2j+k} \\ \alpha^{2i+3k} + \alpha^{3i+2k} & \alpha^{i+3k} + \alpha^{3i+k} & \alpha^{i+2k} + \alpha^{2i+k} \\ \alpha^{2i+3j} + \alpha^{3i+2j} & \alpha^{i+3j} + \alpha^{3i+j} & \alpha^{i+2j} + \alpha^{2i+j} \end{bmatrix} \tag{3.2.11}$$

where

$$\Delta = \alpha^{i+j+k}[\alpha^{2i}(\alpha^{j} + \alpha^{k}) + \alpha^{2j}(\alpha^{i} + \alpha^{k}) + \alpha^{2k}(\alpha^{i} + \alpha^{j})].$$

Multiply both sides of Equation (3.2.10) by A^{-1}, we get ,

$$e_i = \frac{S_1(\alpha^{2j+3k} + \alpha^{3j+2k}) + S_2(\alpha^{j+3k} + \alpha^{3j+k}) + S_3(\alpha^{j+2k} + \alpha^{2j+k})}{\Delta}$$

$$e_j = \frac{S_1(\alpha^{2i+3k} + \alpha^{3i+2k}) + S_2(\alpha^{i+3k} + \alpha^{3i+k}) + S_3(\alpha^{i+2k} + \alpha^{2i+k})}{\Delta}$$

$$e_k = \frac{S_1(\alpha^{2i+3j} + \alpha^{3i+2j}) + S_2(\alpha^{i+3j} + \alpha^{3i+j}) + S_3(\alpha^{i+2j} + \alpha^{2i+j})}{\Delta} \tag{3.2.12}$$

Hence the erasure values are determined from the Equation (3.2.12).

(iv) One Error

The procedure for finding the error magnitude and its position is given in section (3.1). Also, we have to check if there is only one error in the received bytes. This can be done by observing that

$$S_3 = r(\alpha^3) = e_i\alpha^{3i}$$

from which it follows that

$$S_1S_3 = S_2^2. \tag{3.2.13}$$

(v) One Detectable and One Undetectable Error in the Received Bytes

Let there be a detectable error in position i and an undetectable error in position j. The error polynomial, therefore, is given by

$$e(X) = e_i X^i + e_j X^j$$

where $e_i, e_j \in GF(2^7)$ and i is known.

The syndromes S_1, S_2 and S_3 are then calculated as

$$\begin{aligned} S_1 &= r(\alpha) = e_i\alpha^i + e_j\alpha^j \\ S_2 &= r(\alpha^2) = e_i\alpha^{2j} + e_j\alpha^{2j} \\ S_3 &= r(\alpha^3) = e_i\alpha^{3i} + e_j\alpha^{3j} \end{aligned} \quad (3.2.14)$$

The value of j is calculated from the above equation and is given by

$$\alpha^j = (S_3 + \alpha^i S_2)(S_2 + \alpha^i S_1)^{-1}. \quad (3.2.15)$$

Once the location of error j is determined, the magnitudes e_i and e_j can be calculated as in part (ii).

Hence the decoding algorithm for the coding scheme described in chapter 2 for minimum distance d equal to 4 can be stated as follows.

Step I Check the parity of each of the received bytes.

If all parity bits check, go to II.

If exactly one does not check, go to III.

If exactly two do not check, go to IV.

If exactly three do not check, go to V.

If more than three do not check, declare a decoding failure. Proceed to next frame.

Step II Compute the syndromes

$S_1 = r(\alpha)$, $S_2 = r(\alpha^2)$ and $S_3 = r(\alpha^3)$.

If all three of S_1, S_2 and S_3 are zero, declare r(X) as error free.

If all of S_1, S_2 and S_3 are nonzero, assume a single error and decode it using the procedure (iv) given above. Also check if $S_3 S_1 = S_2^2$. If not, declare a decoding failure.

If some of S_1, S_2 and S_3 are zero, declare a decoding failure.

Go to step VI.

Step III Check if $S_3 S_1 = S_2^2$. If yes, then go to III(a), else go to III(b).

III(a) Assume a single detectable error and correct it using procedure (i) given above.

Go to step VI.

III(b) Assume that an undetectable and a detectable error have occurred and decode it using procedure (v).

Go to step VI.

Step IV Check if $S_3 = \alpha^{i+j} S_1 + (\alpha^i + \alpha^j) S_2$.

If yes, then assume that exactly two detectable errors have occurred and use procedure (ii) to decode them.

Go to step VI.

If not, declare a decoding failure.

Go to step VI.

Step V Assume exactly three bytes in error and correct them using procedure (iii) given above.

Go to step VI.

Step VI Go to next frame.

Again, this decoding algorithm ensures that a decoded code word has bytes that have even parity only. The complete decoding algorithm is given by the flowchart in Figure (3.2).

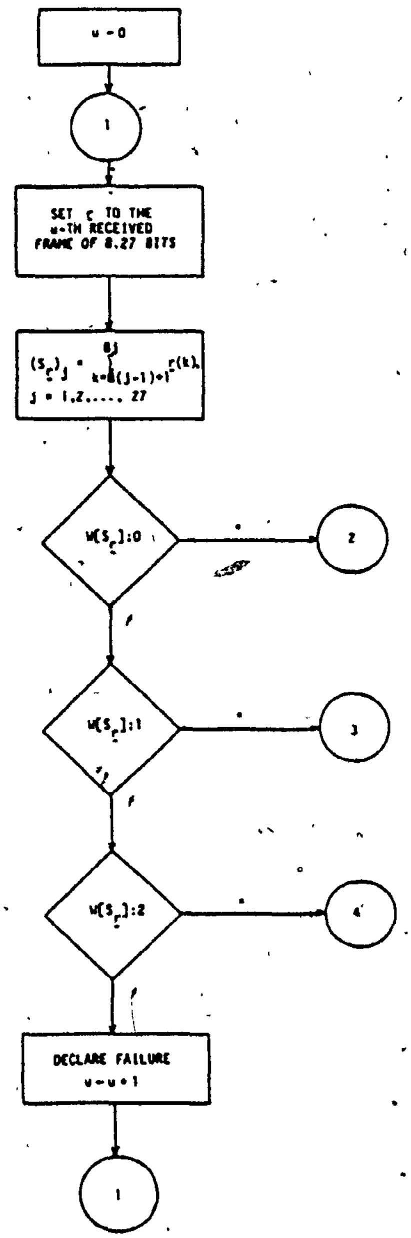

FIG. 3.1. FLOWCHART FOR DECODER OF (27,25) CODE

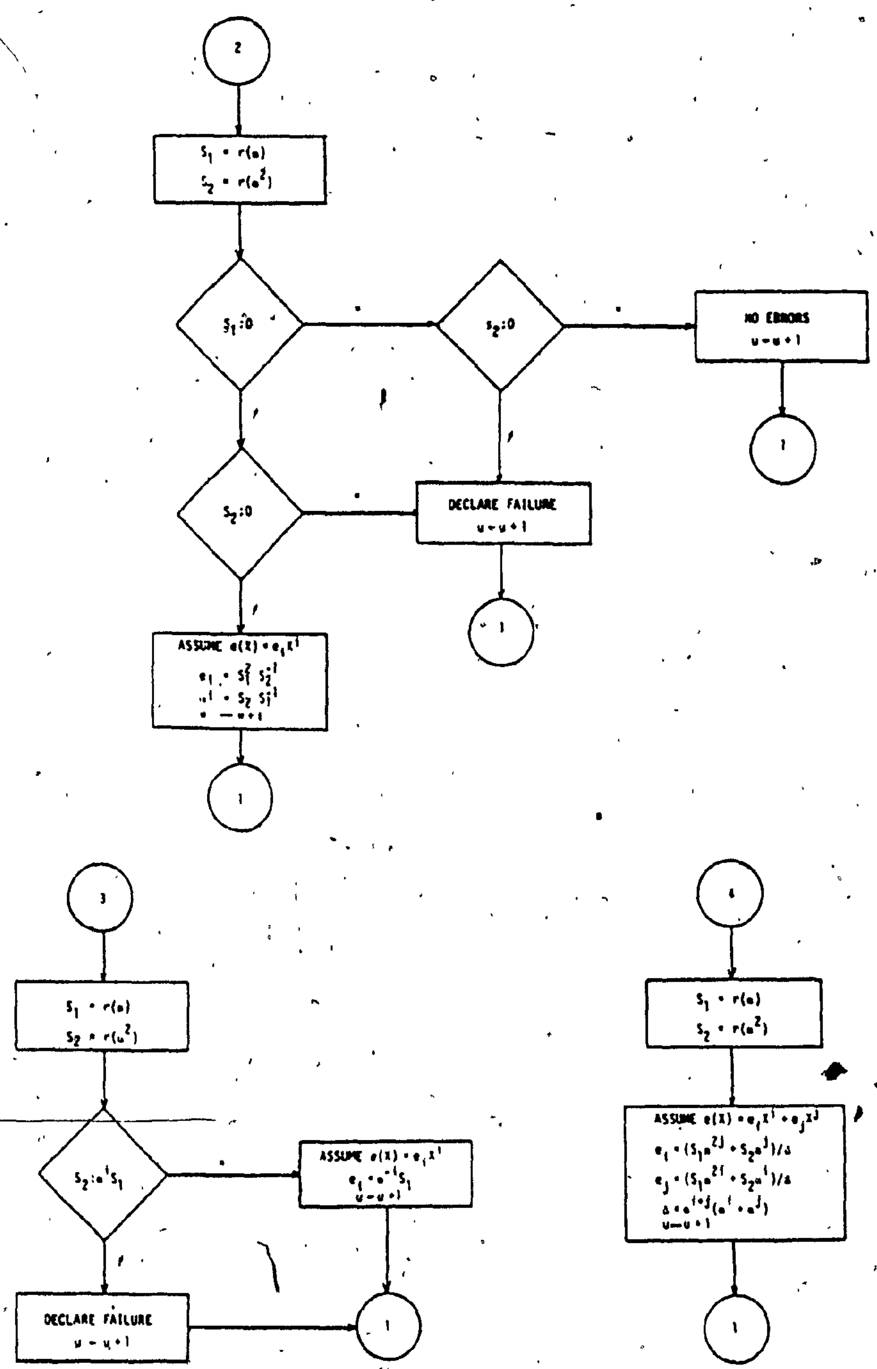

FIG. 3.1. CONTINUED

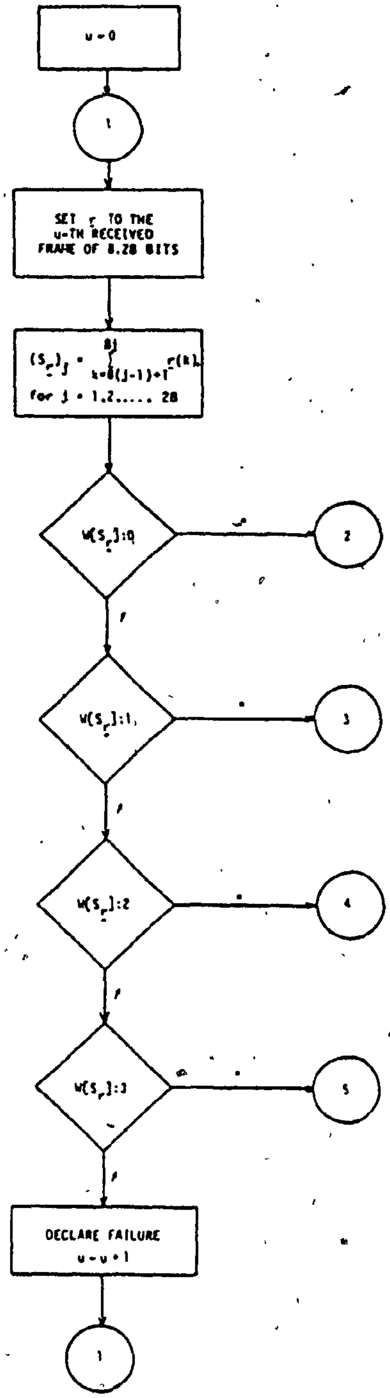

FIG. 3.2. FLOWCHART FOR DECODER OF (28,25) CODE

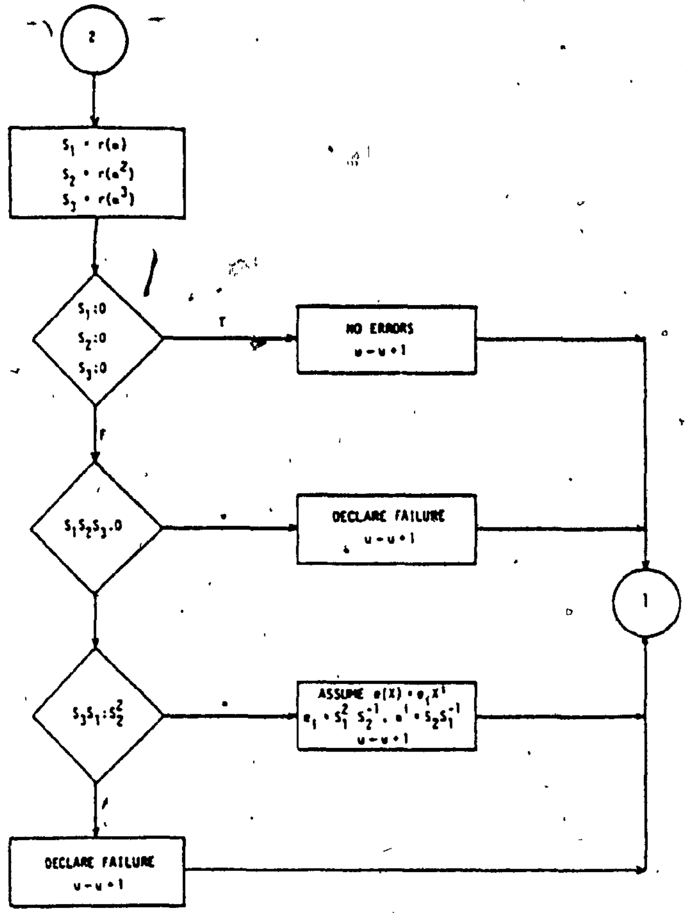

FIG. 3.2. CONTINUED

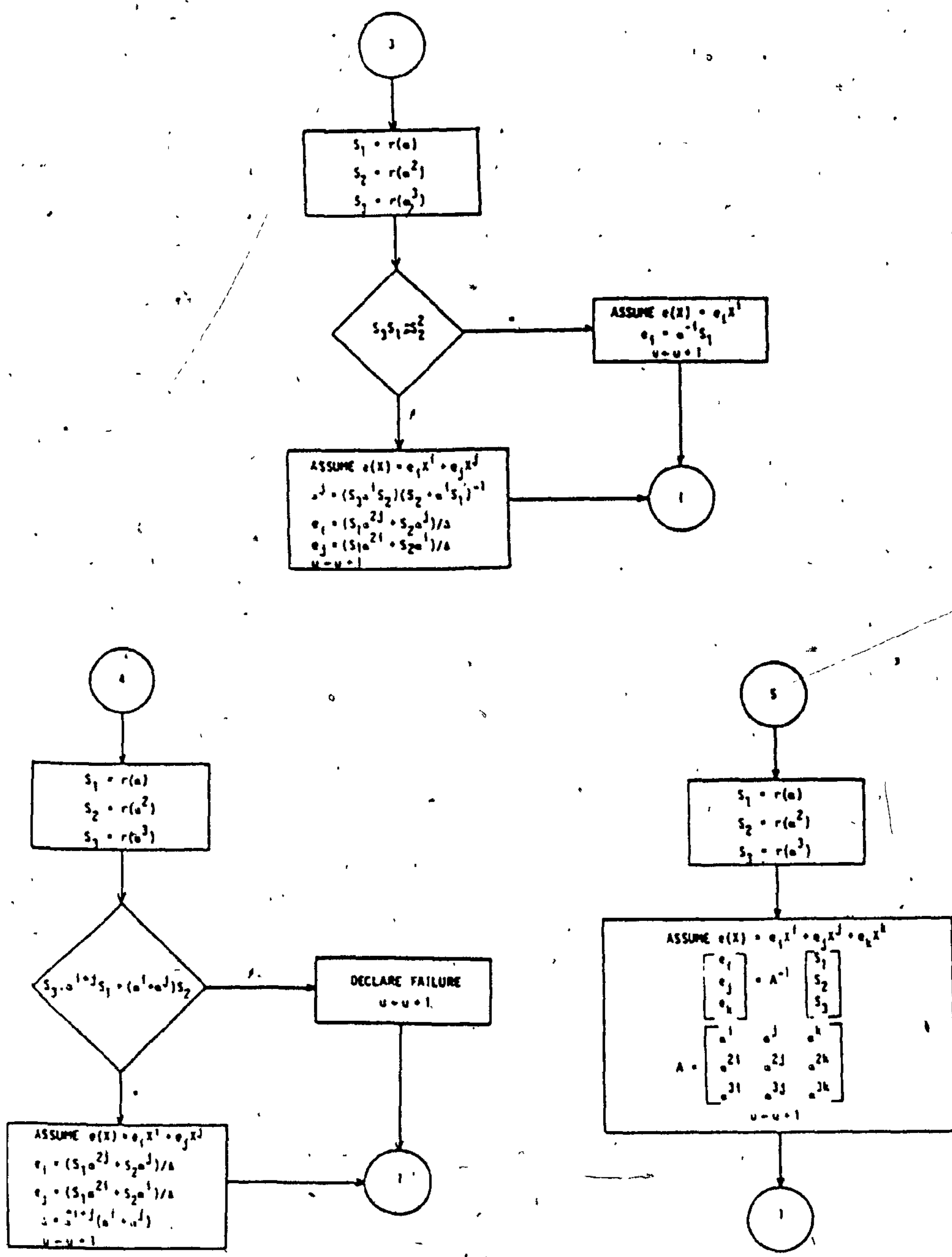

FIG. 3.2. CONTINUED

CHAPTER 4

Performance Evaluation-I

In the selection of error-control coding technique, alternative coding schemes are compared on the basis of various probabilistic measures of performance and system configuration.

One basis of comparison can be the probability of undetected error P_{ud}, if the code is used for error detection only. The receiver in such a system, makes no attempt to correct errors but just checks if the received vector is a code word or not. However, errors may occur in a way that one transmitted code word is received as another code word and the probability of such an event is called the probability of undetectable error. This probability is calculated using the weight distribution of the code for the case of d equal to 3 and 4.

Another basis for comparison is the probability of correct decoding P_{CD}, a quantity that can be calculated when the decoding algorithm is known and a memoryless channel can be assumed. Also, the probability of incorrect decoding P_{ICD}, the probability of decoding failure P_F and the post decoder symbol error rate P_{SE}, can be useful in the evaluation of a coding scheme. Expressions for P_{ICD}, P_{SE} are available in the literature for q-ary block codes with known weight distributions. The reader is referred to [9], [10] and [11] for these general expressions. Here, modified expressions are presented for the coding scheme described in chapter 2 and the decoding algorithm given in chapter 3 for the case of d equal to 3 and 4. Plots of P_{CD}, P_{ICD}, P_F and P_{SE} versus the input symbol error rate $\in$, are given for the two cases.

In the following performance analysis of the coding scheme, it has been assumed that the all zero code word is transmitted. However, the same analysis holds for the transmission of an arbitrary code word since the coding scheme being analysed is linear.

In this chapter, the performance of the coding scheme is analysed on a non-binary symmetric memoryless channel[1]. The performance of the coding scheme for d equal to 3 is evaluated on a binary symmetric channel[1] in chapter 5 using the complete enumerator of the dual code and MacWilliams theorem for complete weight enumerators.

4.1 Probabilistic Model of the Channel

The randomness associated with the transmission process has to be defined in order to be able to compute the probability of the various events of interest. For the code described in chapter 2, all the even parity 8-tuples were represented as elements of $GF(2^7)$. However, in general, the set of all possible binary 8-tuples forms a vector space of dimension 8 and these 8-tuples can be used to represent elements of $GF(2^8)$. Therefore, we will assume that any symbol that is transmitted has probability $(1-\epsilon)$ of being received correctly and a probability of $\epsilon/(q'-1)$ of being transformed into each of the $(q'-1)$ other symbols, where $q'=2^8=256$. This is based on the assumption that a received symbol can have either even or odd parity. Note that there are $q'/2=128$ elements of $GF(2^8)$ that have odd number of ones in their binary 8-tuple representation and if any one of these symbols is received, it is termed as an erasure.

---

[1] To be defined later

We also assume that successive symbols incur errors independently. Hence the probability that the received word differs from the transmitted word in exactly i positions is given by

$$\binom{n}{i} (q'-1)^i \left(\frac{\epsilon}{q'-1}\right)^i (1-\epsilon)^{n-i} = \binom{n}{i} \epsilon^i (1-\epsilon)^{n-i}$$

Note that if $\epsilon = (q'-1)/q'$, each of the q' symbols from the alphabet occurs at the receiver with equal probability. Therefore, we consider the case when $\epsilon \leq (q'-1)/q'$. One example of such a channel model is q-ary FSK modulation in additive white gaussian noise. Hence the probability that an error pattern having exactly i non-zero symbols will occur at the receiver is

$$P(i) = \left(\frac{\epsilon}{q'-1}\right)^i (1-\epsilon)^{n-i} \tag{4.1.1}$$

4.2 On the Probability of Undetected Error (P_{ud})

An error pattern will be accepted as a code word and lead to an undetected error if and only if it is the same as a non-zero code word [12].

Let A(h) denote the number of code words having exactly h non-zero symbols. Then the probability of undetected error P_{ud} is given by

$$P_{ud} = \sum_{h=1}^{n} A(h)\ P(h) \tag{4.2.1}$$

where P(i) is defined by Equation (4.1.1).

The coding scheme described in chapter 2 is essentially a RS code defined over $GF(2^7)$ and for such a code, the number of code words having exactly h non-zero symbols is given by

$$A(h) = \binom{n}{h} (q - 1) \sum_{i=0}^{h-d} (-1)^i \binom{h-1}{i} q^{h-d-i} \qquad (4.2.2)$$

for $d \leq h \leq n$ and $q = 2^7 = 128$.

Thus for a given block length n and a minimum distance d, the weight distribution A(h) can be calculated using Equation (4.2.2) and then P_{ud} is computed using Equation (4.2.1).

The above expressions have been evaluated for dimension of the code k equal to 25 and minimum distance d equal to 3 and 4. The weight distribution of (27,25,3) and (28,25,4) RS codes is given in appendix B. Plots of P_{ud} versus ϵ are given in Figure 4.1 for the above codes.

4.3 Post Decoder Error Distribution and Symbol Error Rate for (27,25,3) Code

4.3.1 The Probability of Correct Decoding, P_{CD}

It was shown in section (3.1) that this code decodes correctly the patterns that correspond to the following events

1. No erasure, no error
2. One erasure, no error
3. Two erasures, no error
4. One error, no erasures.

The probability of each of these events is calculated as below

$$P_{event\ 1} = P(0)$$

$$P_{event\ 2} = \binom{n}{1} (q'/2)\ P(1)$$

$$P_{event\ 3} = \binom{n}{2} (q'/2)^2\, P(2)$$

$$P_{event\ 4} = \binom{n}{1} (q'/2 - 1)\, P(1)$$

Hence the probability of correct decoding P_{CD} is

$$P_{CD} = P(0) + \binom{n}{1}(q'/2)P(1) + \binom{n}{2}(q'/2)^2P(2) + \binom{n}{1}(q'/2-1)P(1) \qquad (4.3.1)$$

where n = 27, q' = 256.

4.3.2 The Probability of Incorrect Decoding, P_{ICD}

An incorrect decoding takes place if the received word is decoded to a code word other than the all 0 code word. It occurs if one of the four events described in (4.3.1) takes place with respect to a non-zero code word. Thus, if $P_{ICD}(h)$ is the probability of incorrect decoding to a code word of weight h, the probability of incorrect decoding P_{ICD} is

$$P_{ICD} = \sum_{h=d}^{n} P_{ICD}(h) \,. \qquad (4.3.2)$$

For a code word of weight h, the probability of various events described in (4.3.1) is given by

$$P_{event\ 1} = P(h)$$

$$P_{event\ 2} = \binom{h}{1} (q'/2)\, P(h) + \binom{n-h}{1} (q'/2)\, P(h+1)$$

$$P_{event\ 3} = \binom{h}{2} (q'/2)^2\, P(h) + \binom{n-h}{2} (q'/2)^2\, P(h+2)$$

$$+ \binom{h}{1} \binom{n-h}{1} (q'/2)^2\, P(h+1)$$

$$P_{event\ 4} = \binom{h}{1} (q'/2-2)P(h) + \binom{h}{1} P(h-1) + \binom{n-h}{1} (q'/2-1)P(h+1)$$

and

$$P_{ICD}(h) = A(h)(P_{event\ 1} + P_{event\ 2} + P_{event\ 3} + P_{event\ 4}) \tag{4.3.3}$$

where $A(h)$ denotes the number of code words having exactly h non-zero symbols and is calculated using Equation (4.2.2). $P_{ICD}(h)$ is calculated for the values of h going from d to n and these values are substituted in Equation (4.3.2) to calculate P_{ICD}. Also if P_F is the probability of decoding failure, then

$$P_{CD} + P_{ICD} + P_F = 1$$

and, therefore,

$$P_F = 1 - P_{CD} - P_{ICD} \tag{4.3.4}$$

4.3.3 Post Decoding Symbol Error Rate, P_{SE}

The post decoding symbol error rate P_{SE} is defined as the expected number of errors in a code word following decoding. Hence

$$P_{SE} = \frac{1}{n} \sum_{h=d}^{n} h\, P_{ICD}(h) \tag{4.3.5}$$

and can be calculated easily once $P_{ICD}(h)$, $d \leq h \leq n$ is known.

Plots of P_{CD}, P_{ICD}, P_F and P_{SE} versus ϵ are given in Figures (4.2), (4.3), (4.4) and (4.5) respectively.

4.4 Post Decoder Error Distribution and Symbol Error Rate for (28,25,4) Code

4.4.1 The Probability of Correct Decoding, P_{CD}

This code can decode correctly if the received patterns correspond to one of the following events

1. No erasure, no error
2. One erasure, no error
3. Two erasures, no error
4. Three erasures, no error
5. One error, no erasure
6. One error, one erasure.

The expression for the probability of events (1), (2), (3) and (5) is given in section (4.3.1) and

$$P_{\text{event } 4} = \binom{n}{3} (q'/2)^3 \, P(3)$$

$$P_{\text{event } 6} = 2\binom{n}{2} (q'/2)(q'/2 - 1) \, P(2).$$

Hence

$$P_{CD} = P(0) + \binom{n}{1}(q'/2)P(1) + \binom{n}{2}(q'/2)^2 P(2) + \binom{n}{3}(q'/2)^3 P(3)$$
$$+ \binom{n}{1}(q'/2-1)P(1) + 2\binom{n}{2}(q'/2)(q'/2-1)P(2) \qquad (4.4.1)$$

where $n = 28$, $q' = 256$.

4.4.2 The Probability of Incorrect Decoding

An incorrect decoding takes place if one of the 6 events given in section (4.4.1) takes place with respect to a non-zero code word. The events (1), (2), (3) and (5) correspond to events (1), (2), (3) and (4) of section (4.3.1) respectively and the expressions for the probability are given in section (4.3.2). For a code word of weight h, the probability of events (4) and (6) is given by

$$P_{\text{event 4}} = \binom{h}{3}(q'/2)^3 P(h) + \binom{n-h}{3}(q'/2)^3 P(h+3)$$

$$+ \binom{h}{2}\binom{n-h}{1}(q'/2)^3 P(h+1) + \binom{h}{1}\binom{n-h}{2}(q'/2)^3 P(h+2)$$

$$P_{\text{event 6}} = 2\binom{h}{2}(q'/2)(q'/2-2)\,P(h) + 2\binom{h}{2}(q'/2)\,P(h-1)$$

$$+ 2\binom{n-h}{2}(q'/2)(q'/2-1)P(h+2) + \binom{h}{1}\binom{n-h}{1}(q'/2-2)(q'/2)P(h+1)$$

$$+ \binom{h}{1}\binom{n-h}{1}(q'/2)P(h) + \binom{h}{1}\binom{n-h}{1}(q'/2)(q'/2-1)P(h+1)$$

and

$$P_{ICD}(h) = A(h)\left(\sum_{i=1}^{6} P_{\text{event } i}\right) \qquad (4.4.2)$$

Hence P_{ICD} can be calculated using Equation (4.3.2). Also Equations (4.3.4) and (4.3.5) can be used to calculate P_F and P_{SE} respectively. Plots of P_{CD}, P_{ICD}, P_F and P_{SE} versus $\in$ are given in Figures (4.2), (4.3), (4.4) and (4.5) respectively.

4.5 Discussion of Results

As it is clear from Figure (4.1), these codes can be used extremely effectively for error control in systems, where the codes are employed for the purpose of error detection only. For noisy channels ($\epsilon \sim 10^{-2}$), the (27,25,3) code has P_{ud} of the order of 10^{-8} while (28,25,4) code has P_{ud} of the order of 10^{-11}. However, the receiver has to check for one more syndrome to be zero in latter code, which may not be a very high price to pay for the lower value of P_{ud} in many systems where high reliability on the received information is required.

One interesting result obtained from the analysis performed in this chapter is that the probability of incorrect decoding is much lower as compared to the probability of decoding failure. Thus, most of the time the decoder either decodes the received message block correctly or it declares a decoding failure. It can particularly be helpful in situations where failure to decode can be tolerated but the penalty that one pays for decoding incorrectly is high.

For ϵ of the order of 10^{-2}, the post decoder symbol error rate, P_{SE} is of the order of 10^{-5} for the (28,25,4) code while it is of the order of 10^{-4} for the (27,25,3) code. Thus the additional complexity of the decoding algorithm in case of (28,25,4) code leads to significant reduction in the post decoder symbol error rate. These values could be quite acceptable for most digital communication systems. If still lower value of P_{SE} is required, then a code of higher redundancy may be designed. But it must be pointed out that though the additional complexity in the encoder would be marginal, the decoding algorithm will no longer be in closed form and a more general, Berlekamp-Massey algorithm would be needed to perform the decoding.

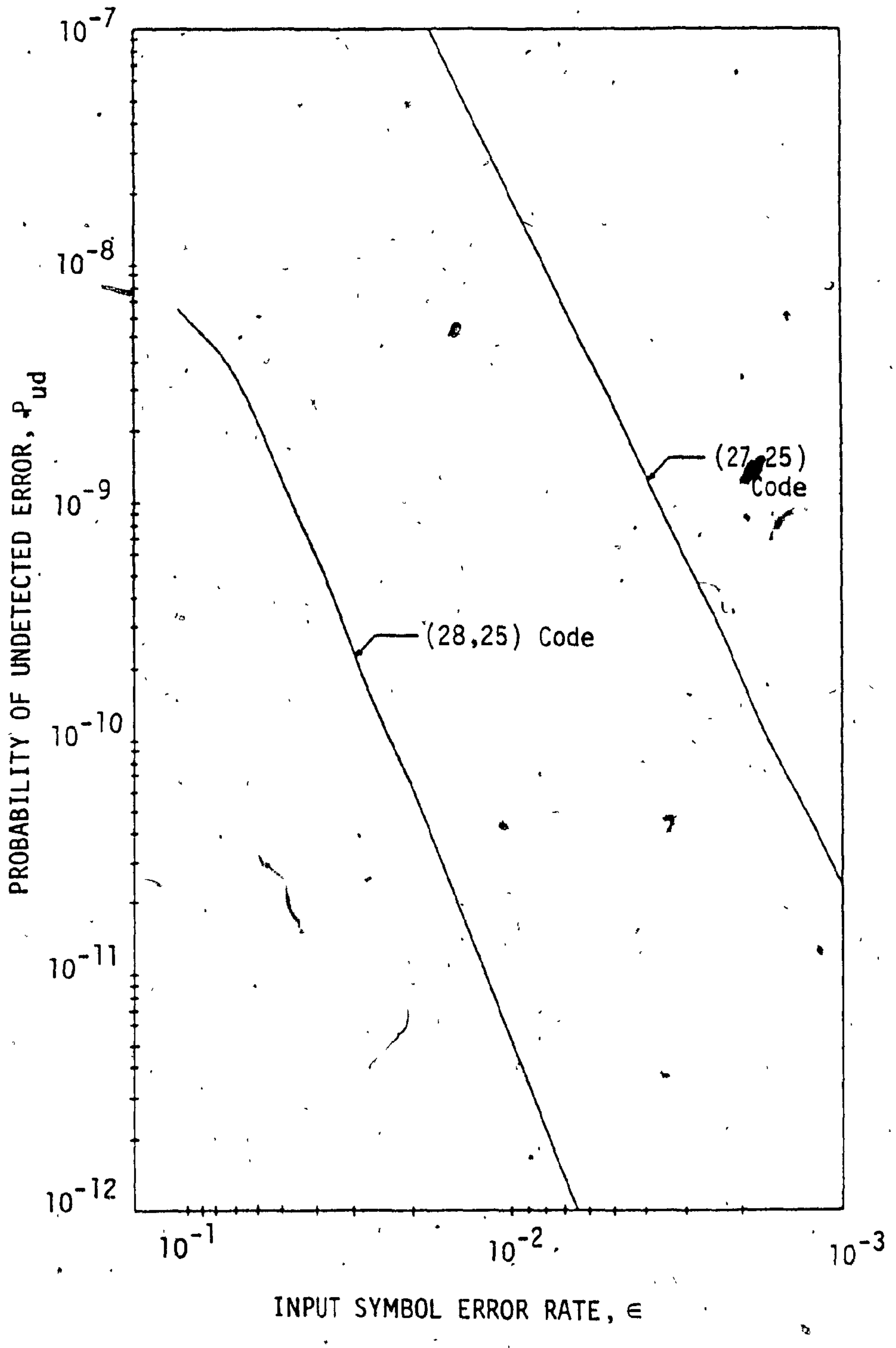

FIG. 4.1. PROBABILITY OF UNDETECTED ERROR Vs. INPUT SYMBOL ERROR RATE

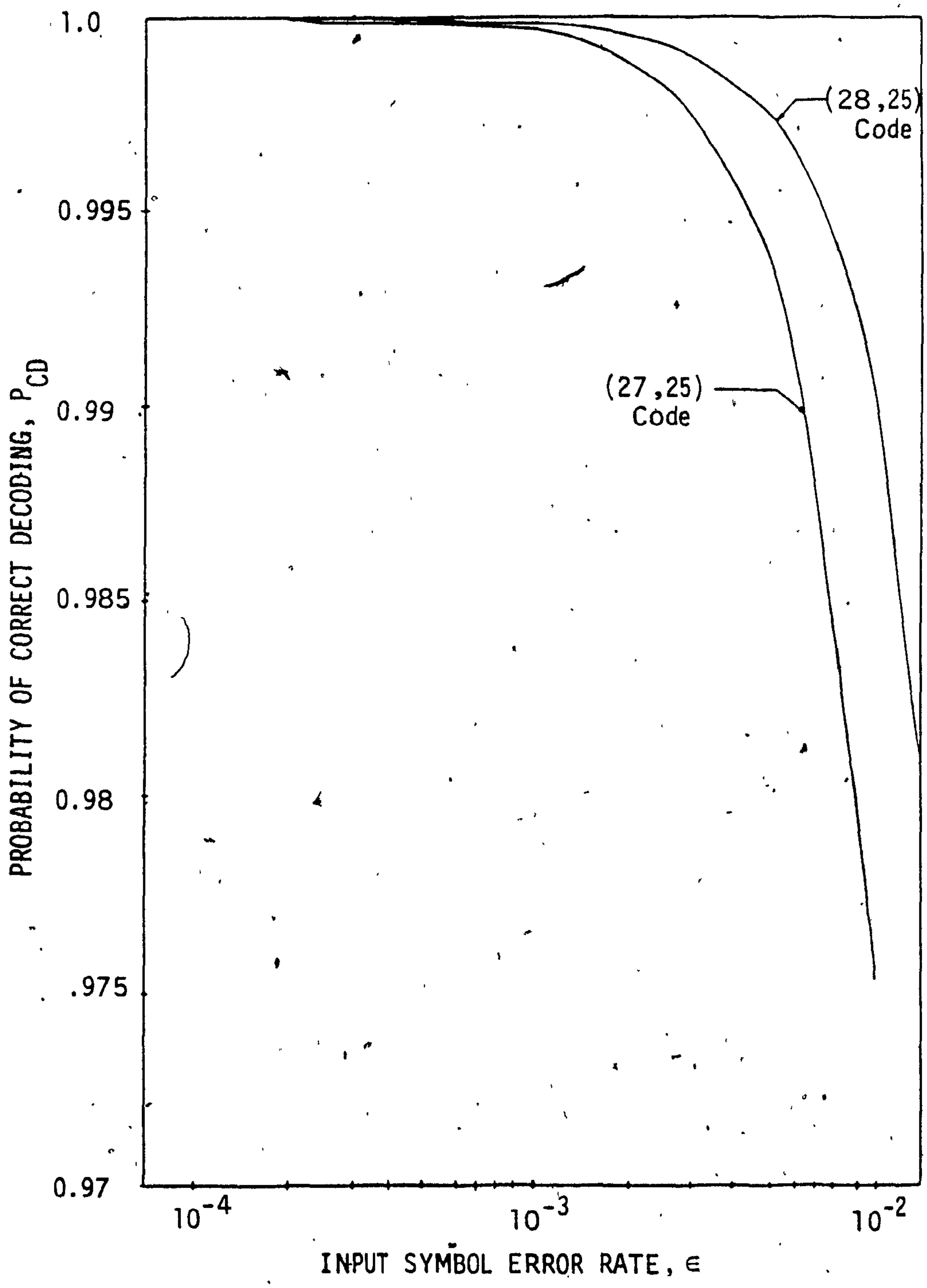

Fig. 4.2 PROBABILITY OF CORRECT DECODING Vs. INPUT SYMBOL ERROR RATE

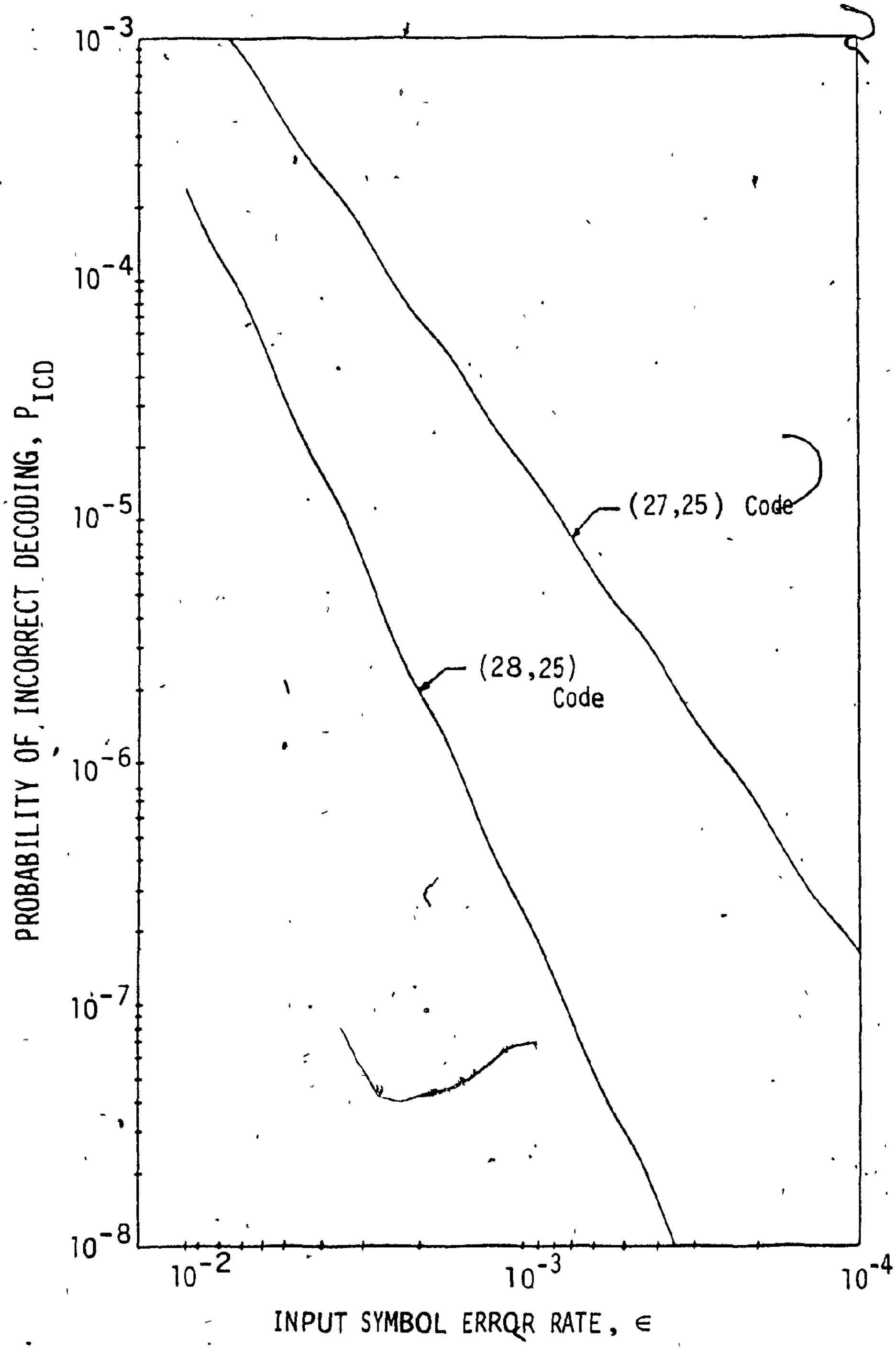

FIG. 4.3. PROBABILITY OF INCORRECT DECODING Vs. INPUT SYMBOL ERROR RATE

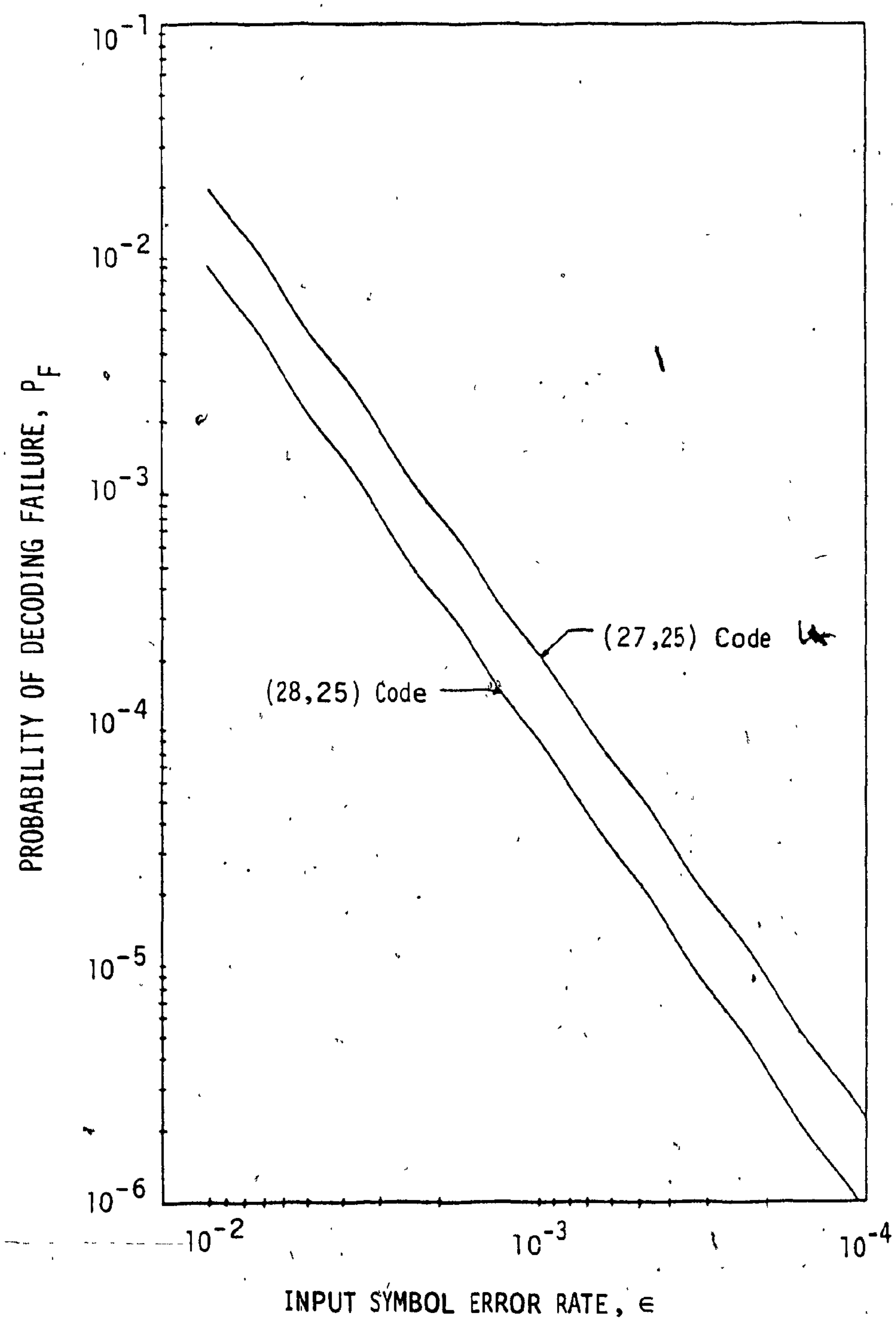

FIG. 4.4. PROBABILITY OF DECODING FAILURE Vs. INPUT SYMBOL ERROR RATE

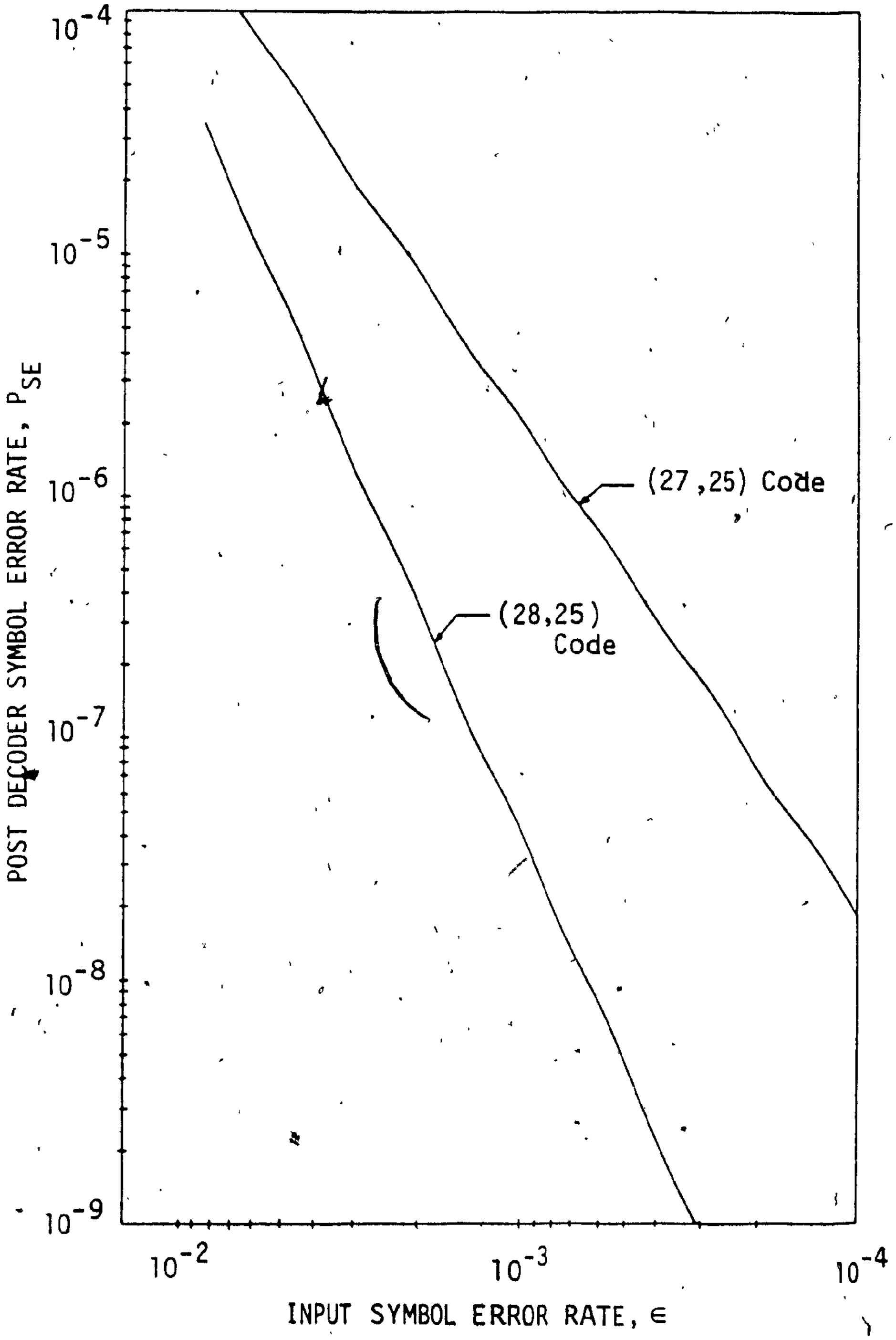

FIG. 4.5. POST DECODER SYMBOL ERROR RAT[illegible]s.
INPUT SYMBOL ERROR RATE

CHAPTER 5

Performance Evaluation-II

In chapter 4, the statistical performance of the coding scheme was analysed on a q-ary symmetric memoryless channel. The RS code defined on $GF(2^7)$ is mapped into a binary code as described in chapter 2, and therefore, it is interesting to evaluate the performance of the binary code obtained as a result of this mapping, on a binary channel. Though most communication channels are not accurately represented by the binary symmetric channel (BSC), shown in Figure (5.1), it has been studied extensively. For the binary symmetric channel, the probability is Q that the same symbol will be received as transmitted. It is assumed that $Q > P$ and that each symbol is independent of all others. The example of such a channel model can be PSK, FSK and QPSK modulation in additive white gaussian noise with hard decision decoding.

In this chapter, expressions for P_{CD}, P_{ICD}, P_{SE} are derived for the coding scheme and the decoding algorithm described in earlier chapters. However, since the analysis is very complicated, it has been carried out for d equal to 3 only. For the case of d equal to 4, only the expression for P_{CD} is presented and evaluated. It will be seen shortly that the analysis of a code can be based on the structure of the dual code. The dual of a linear code with generator matrix G is defined as the linear code whose parity check matrix is the transpose of G.

As was also stated earlier, we evaluate the performance on the assumption that the all zero code vector is transmitted. But since the code is linear, the same analysis holds for the transmission of any arbitrary code word.

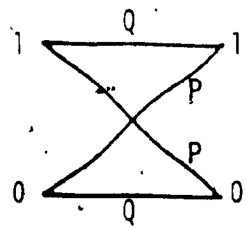

FIG. 5.1. THE BINARY SYMMETRIC CHANNEL

Based on the assumption that the channel is a binary symmetric memoryless channel, the probability that an all zero 8-tuple is received as an 8-tuple having exactly i ones is P^iQ^{8-i}. Thus the probability that one 8-tuple gets converted to another 8-tuple when transmitted over this channel is P^iQ^{8-i}, where i is the number of positions the two 8-tuples differ. Hence the model of binary symmetric channel does not extend to a q-ary symmetric channel model.

5.1 Definitions, Notations

Let GF(q) be a Galois field of q elements, $q = 2^m$. $GF^n(q)$ is the set of all possible row vectors of length n, in which each coordinate is an element of GF(q). Addition of two vectors is defined coordinate by coordinate, under the rules prevailing in GF(q). $GF^n(q)$ is a vector space of dimension n over GF(q). Choose a basis consisting of n vectors

$$e_1 = (1\ 0\ 0\ \ldots\ 0)$$

$$e_2 = (0\ 1\ 0\ \ldots\ 0)$$

$$\cdots\cdots$$

$$e_n = (0\ 0\ 0\ \ldots\ 1)$$

An element u of $GF^n(q)$ can be expressed uniquely as

$$u = \sum_{i=1}^{n} u_i e_i \ , \quad u_i \in GF(q) \qquad (5.1.1)$$

We can write $u = (u_1, u_2, \ldots, u_n)$.

The Hamming weight of u is defined as the number of non-zero coordinates in u.

5.1.1 Complete Weight Enumerator

Let the elements of GF(q) be denoted by $\omega_0 = 0, \omega_1, \ldots, \omega_{q-1}$ in some fixed order. Complete weight enumerator classifies code words c in $GF^n(q)$ according to the number of times each field element ω_i appears in c [1].

The composition of $c = (c_0, c_1, \ldots, c_{n-1})$ denoted by comp(c) is $(s_0, s_1, \ldots, s_{q-1})$, where $s_i = s_i(c)$ is the number of components c_j equal to ω_i. Thus

$$\sum_{i=0}^{q-1} s_i = n.$$

Let $\mathcal{E}$ be a linear code over GF(q) and let A(t) be the number of code words $c \in \mathcal{E}$ with $\mathrm{comp}(c) = t = (t_0, \ldots, t_{q-1})$. Then the complete weight enumerator of $\mathcal{E}$ is

$$W_{\mathcal{E}}(z_0, \ldots, z_{q-1}) = \sum_{t} A(t) z_0^{t_0} \ldots z_{q-1}^{t_{q-1}}$$
$$= \sum_{c \in \mathcal{E}} z_0^{s_0} \ldots z_{q-1}^{s_{q-1}}.$$

where $z_0, z_1, \ldots, z_{q-1}$ is a set of q commutating indeterminants and the indeterminant z_i corresponds to the element ω_i.

5.2 On the Probability of Post Decoder Events for (216,175) Binary Code

For the decoding algorithm described in chapter 3 the various parameters of interest are

1. The probability of correct decoding, P_{CD}
2. The probability of incorrect decoding, P_{ICD}
3. The probability of decoding failure, P_F
4. The output symbol error rate, P_{SE} and
5. The output bit error rate, BER.

It was shown in chapter 2 that all the 8-tuples in the binary code obtained as a result of the mapping, are of even weight and a received 8-tuple having odd weight is termed as an erasure. Let the 8-tuples having odd numbers of 1's be represented by indeterminants $z_1^*, z_2^*, \ldots, z_{128}^*$.

It can be readily observed that on a BSC, the probability that the all 0 8-tuple will get converted to any one of these is given by

$$P_E = \sum_{i=0}^{3} \binom{8}{2i+1} P^{2i+1} Q^{8-(2i+1)} \tag{5.2.1}$$

P being the cross over probability.

In the following analysis, the patterns corresponding to various events are derived and the probability of occurrence of each pattern can be calculated by replacing each of the z_i by $P^{\ell_i} Q^{8-\ell_i}$, where ℓ_i is the number of 1's in the binary 8-tuple representation of z_i. Also note that q = 128.

5.2.1 The Probability of Correct Decoding, P_{CD}

The received vector is decoded correctly if

(a) It is same as the all 0 code word.

(b) It has only one non-zero symbol and the symbol has even weight (one error

(c) It has only one non-zero symbol and that symbol has odd weight (one erasure)

(d) It has two non-zero symbols and both have odd weight (two erasures).

Pattern corresponding to event (a) is given by z_0^n. Similarly, patterns corresponding to events (b), (c) and (d) are given by

$$n(z_1 + z_2 + \cdots + z_{127})z_0^{n-1},$$

$$n(z_1^* + z_2^* + \cdots + z_{128}^*)z_0^{n-1} \quad \text{and}$$

$$\binom{n}{2} (z_1^* + z_2^* + \cdots + z_{128}^*)^2 \, z_0^{n-2} .$$

Hence, all the received patterns that are decoded correctly are

$$z_0^n + n(z_1 + \cdots + z_{127})z_0^{n-1} + n(z_1^* + \cdots + z_{128}^*)z_0^{n-1} + \binom{n}{2} (z_1^* + \cdots + z_{128}^*)^2 z_0^{n-2}, \tag{5.2.2}$$

where n is the block length of the code.

5.2.2 The Probability of Incorrect Decoding, P_{ICD}

An incorrect decoding takes place if the received vector leads to one of the following events

(a) It is same as a non-zero code word

(b) It differs from a non-zero code word in one position and the symbol received in that position has even weight.

(c) It differs from a non-zero code word in one position and the symbol received in that position has odd weight

(d) It differs from a non-zero code word in two positions and the symbols received in these positions have odd weight.

Let a code word be represented as

$$c = z_0^{s_0} z_1^{s_1} \cdots z_{127}^{s_{127}} \tag{5.2.3}$$

and let the patterns corresponding to the received vectors that will be decoded to this code word be represented by e. For each of the above given error events, the patterns are obtained as follows

Event[1] (a). It is same as the code word and, therefore, is given by

$$e = z_0^{s_0} z_1^{s_1} \cdots z_{127}^{s_{127}}$$

It is clear that all the 0 received word is decoded correctly and hence all the patterns that lead to this error event are

$$e = \sum_{c \in \mathcal{E}} c - z_0^n \tag{5.2.4}$$

Event (b). All the patterns that differ from the code word in a position that corresponds to z_0 in a code word say, can be represented as

$$z_0^{s_0-1} z_1^{s_1+1} \cdots z_{127}^{s_{127}}$$
$$+ z_0^{s_0-1} z_1^{s_1} z_2^{s_2+1} \cdots z_{127}^{s_{127}}$$
$$+ \cdots$$
$$+ z_0^{s_0-1} z_1^{s_1} \cdots z_{127}^{s_{127}+1}$$

---

1 This event will also lead to an undetectable error, if this coding scheme is used for the purpose of error detection only.

$$= (z_1 + z_2 + \cdots + z_{127})\, c/z_0$$

and there are s_0 of such patterns.

Hence, it can be shown that all the patterns that differ from a code word c in one place are

$$\begin{aligned} e &= s_0(z_1 + z_2 + \cdots + z_{127})\, c/z_0 \\ &\quad + s_1(z_0 + z_2 + \cdots + z_{127})\, c/z_1 \\ &\quad + \cdots \\ &\quad + s_{127}(z_0 + z_1 + \cdots + z_{126})\, c/z_{127} \\ &= \sum_{i=0}^{127} s_i(z_0 + \cdots + z_{i-1} + z_{i+1} + \cdots + z_{127})\, c/z_i \\ &= \sum_{i=0}^{127} s_i(z_0 + z_1 + \cdots + z_{127})\, c/z_i - \sum_{i=0}^{127} s_i c \end{aligned} \tag{5.2.5}$$

Clearly

$$\sum_{i=0}^{127} s_i = n,$$

and, therefore, Equation (5.2.5) can be written as

$$e = (z_0 + z_1 + \cdots + z_{127}) \sum_{i=0}^{127} s_i c/z_i - n\, c\,.$$

Since the received word that has only one non-zero symbol is decoded correctly, all the patterns that lead to this error event are

$$\begin{aligned} e = \sum_{c \in \mathcal{E}} & (z_0 + z_1 + \cdots + z_{127}) \sum_{i=0}^{127} s_i c/z_i - n\, c \\ & - n(z_1 + z_2 + \cdots + z_{127}) z_0^{n-1} \end{aligned} \tag{5.2.6}$$

Event (c). The received vector can differ from a code word at any one of the positions corresponding to $z_0, z_1, \ldots, z_{127}$ in the code word and the symbol received at that position can be any one of $z_1^*, z_2^*, \ldots, z_{128}^*$. A similar analysis can be performed as above and it can be shown that all such patterns that lead to this error event are

$$e = \sum_{c \in \mathcal{C}} (z_1^* + z_2^* + \cdots + z_{128}^*) \sum_{i=0}^{127} s_i c/z_i - n(z_1^* + z_2^* + \cdots + z_{128}^*) z_0^{n-1} \qquad (5.2.7)$$

Event (d). For a code word represented by Equation (5.2.3), the patterns that differ from the code word in two positions and have symbols of odd weight in those positions can be represented as

$$(z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{i=0}^{127} \tfrac{1}{2} s_i(s_i - 1)\, z_0^{s_0} z_1^{s_1} \cdots z_i^{s_i-2} \cdots z_{127}^{s_{127}}$$

$$+ (z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{j=i+1}^{127} \sum_{i=0}^{126} s_j s_i z_0^{s_0} z_1^{s_1} \cdots z_i^{s_i-1} \cdots z_j^{s_j-1} \cdots z_{127}^{s_{127}}$$

$$= (z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{j=0}^{127} \sum_{i=0}^{127} \tfrac{1}{2} s_j s_i^* z_0^{s_0} z_1^{s_1} \cdots z_i^{s_i-1} \cdots z_j^{s_j-1} \cdots z_{127}^{s_{127}}$$

where $s_i^* = s_i - \delta(i-j)$, $\delta(x)$ being the delta function.

From Equation (5.2.3), it can be seen that

$$z_0^{s_0} z_1^{s_1} \cdots z_i^{s_i-1} \cdots z_j^{s_j-1} \cdots z_{127}^{s_{127}} = c/(z_j z_i).$$

Therefore, all the patterns that lead to this error event are

$$e = (z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{c \in \mathcal{C}} \sum_{j=0}^{127} \sum_{i=0}^{127} \tfrac{1}{2} s_j s_i^* c/(z_j z_i) - \frac{n(n-1)}{2} (z_1^* + z_2^* + \cdots + z_{128}^*)^2 z_0^{n-2} \qquad (5.2.8)$$

Hence, all the received words that would lead to incorrect decoding can be represented by the sum of expressions (5.2.4), (5.2.6), (5.2.7) and (5.2.8).

5.2.3 The Probability of Decoding Failure, P_F

This is an event when the decoder detects an error but it is beyond its correcting capability. The probability of correct decoding, incorrect decoding and decoding failure are related as

$$P_F + P_{ICD} + P_{CD} = 1$$

or

$$P_F = 1 - P_{ICD} - P_{CD} \tag{5.2.9}$$

5.2.4 Post Decoder Symbol Error Rate, P_{SE}

The number of non-zero symbols for a code word represented by Equation (5.2.3) is $(s_1 + s_2 + \cdots + s_{127})$. Also

$$s_0 + s_1 + \cdots + s_{127} = n$$

or

$$s_1 + s_2 + \cdots + s_{127} = n - s_0$$

If e represents all the error patterns that are incorrectly decoded to a code word c, then the post decoder symbol error rate is given by.

$$P_{SE} = \frac{1}{n} \sum_{c \in \&} (s_1 + s_2 + \cdots + s_{127})e$$

$$= \frac{1}{n} \sum_{c \in \&} (n - s_0)e$$

$$= \sum_{c \in \&} e - \frac{1}{n} \sum_{c \in \&} s_0 e$$

There are four error events as described above, that lead to incorrect decoding and, therefore, would also contribute to the symbol error rate. Thus, the post decoder symbol error rate is the sum of the symbol error rates due to each of the four error events. Let these be represented by $P_{SE}(a)$, $P_{SE}(b)$, $P_{SE}(c)$, $P_{SE}(d)$ respectively.

Event (a)

The pattern e for any code word c is the same as the code word itself. $P_{SE}(a)$ is, then, given by

$$P_{SE}(a) = \sum_{c\in \&} c - \frac{1}{n} \sum_{c\in \&} s_0\, c \tag{5.2.10}$$

Event (b)

The error patterns that lead to this event are given by Equation (5.2.6) and it can be shown that $P_{SE}(b)$ is

$$\begin{aligned} P_{SE}(b) = \sum_{c\in \&} [\, &(z_0 + z_1 + \cdots + z_{127}) \sum_{i=0}^{127} s_i c/z_i - n\,c \\ &- \frac{1}{n}\{(z_0 + z_1 + \cdots + z_{127}) \sum_{i=0}^{127} s_0 s_i\, c/z_i - n\, s_0\, c\}] \end{aligned} \tag{5.2.11}$$

Event (c)

The error patterns that lead to this event are given by Equation (5.2.7) and $P_{SE}(c)$ can be expressed as

$$\begin{aligned} P_{SE}(c) = \sum_{c\in \&} [\, &(z_1^* + z_2^* + \cdots + z_{127}^*) \sum_{i=0}^{127} s_i c/z_i \\ &- \frac{1}{n}(z_1^* + z_2^* + \cdots + z_{127}^*) \sum_{i=0}^{127} s_0 s_i\, c/z_i] \end{aligned} \tag{5.2.12}$$

Event (d)

Again, the error patterns that lead to this event are given by Equation (5.2.8) and $P_{SE}(d)$ can be written as

$$P_{SE}(d) \quad \sum_{c \in \&} \Big[(z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{j=0}^{127} \sum_{i=0}^{127} \tfrac{1}{2} s_j s_i^* \, c/(z_j z_i) - (z_1^* + z_2^* + \cdots + z_{128}^*)^2 \sum_{j=0}^{127} \sum_{i=0}^{127} \frac{1}{2n} s_0 s_j s_i^* \, c/(z_j z_i) \Big] \tag{5.2.13}$$

and finally

$$P_{SE} = P_{SE}(a) + P_{SE}(b) + P_{SE}(c) + P_{SE}(d)$$

5.2.4 Post Decoder Bit Error Rate, BER

For every received vector decoded to a code word represented by Equation (5.2.3), the number of bits in error is given by

$$s_1 \ell_1 + s_2 \ell_2 + \cdots + s_{127} \ell_{127},$$

where ℓ_i is the number of 1's in the binary 8-tuple representation of the element ω_i.

Thus, if e represents all the patterns that are incorrectly decoded to this code word, then BER is

$$BER = \frac{1}{8.n} \sum_{c \in \&} (s_1 \ell_1 + s_2 \ell_2 + \cdots + s_{127} \ell_{127}) e$$

It is an extremely difficult task to evaluate such an expression. However, a very close bound may be obtained as

$$BER = P_{SE}.$$

This bound is obtained by noting that the maximum value that any ℓ_i takes is 8.

5.2.5 Note

The code to be analysed is a (27,25) RS code defined over GF(2^7), where the binary 8-tuples forming a vector space of dimension 7 are used to represent elements of GF(2^7). There are no closed form expressions for complete weight enumerator of such a code. Also, as the total number of code words in this code is 4.789×10^{52}, it is not possible to generate the complete weight enumerator of this code on the computer and then evaluate the probability of all the error patterns for each of the code words.

It is further noted that if the probability of 0 symbol being received as a non-zero symbol was the same for all the non-zero symbols, then the Hamming weight enumerator of the code was sufficient for the statistical performance analysis of the code. The Hamming weight enumerator of this code is given by the Equation (4.2.2).

Hence we have to look for alternative ways of computing the expressions derived here. One such method is to use MacWilliams theorem for complete weight enumerator. The dual code of (27,25) code is (27,2) code and has only 16384 code words. Therefore, it is possible to generate the complete weight distribution of the dual code on the computer.

In the following section, MacWilliams theorem for complete weight enumerator and the related theory is covered in brief.

5.3 MacWilliams Theorem for Complete Weight Enumerator

5.3.1 Characters of GF(q)

Any element β of GF(q), $q=2^m$ can be written in the form

$$\beta = \beta_0 + \beta_1\alpha + \cdots + \beta_{m-1}\alpha^{m-1}$$

or equivalently as an m-tuple

$$\beta = (\beta_0, \beta_1, \ldots, \beta_{m-1}),$$

where α is a primitive element of GF(q) and $0 \le \beta_i \le 1$. Let ξ be a complex number $e^{2\pi i/2}$. This is a primitive 2^{nd} root of unity, i.e. $\xi^2 = e^{2\pi i} = 1$, while $\xi^\ell \ne 1$ for $0 < \ell < 2$. It implies $\xi = -1$.

Definition. For each $\beta = (\beta_0, \beta_1, \ldots, \beta_{m-1})$ of GF(q), define χ_β to be the complex valued mapping defined on GF(q) by

$$\chi_\beta(\nu) = \xi^{\beta_0\nu_0 + \cdots + \beta_{m-1}\nu_{m-1}}$$

for $\nu = (\nu_0, \ldots, \nu_{m-1}) \in GF(q)$. χ_β is called a character of GF(q). It can be easily shown that

(i) $\chi_\beta(\nu) = \chi_\nu(\beta)$ for all $\beta, \nu \in GF(q)$

(ii) $\chi_\beta(\nu + \nu') = \chi_\beta(\nu) \cdot \chi_\beta(\nu')$ for all $\beta, \nu, \nu' \in GF(q)$

Thus, χ_β is a homomorphism from the additive group of GF(q) into the multiplicative group of complex numbers of magnitude 1.

(iii) $\chi_{\beta+\beta'}(\nu) = \chi_\beta(\nu) \cdot \chi_{\beta'}(\nu)$ for all $\beta, \beta', \nu \in GF(q)$

Thus the set of all q characters χ_β form a group which is isomorphic to the additive group of GF(q).

To state MacWilliams theorem [1], any one of the characters χ_β with $\beta \neq 0$ is selected, say $\beta = 1$, i.e. the character χ_1 defined by

$$\chi_1(\nu) = \xi^{\nu_0} \quad \text{for} \quad \nu = (\nu_0, \ldots, \nu_{m-1}) \in GF(q)$$

5.3.2 MacWilliams Theorem for Complete Weight Enumerator [13]

If $\mathcal{C}$ is a linear (n,k) code over $GF(q)$ with complete weight enumerator $W_{\mathcal{C}}$, the complete weight enumerator of the dual code $W_{\mathcal{C}^\perp}$ is

$$W_{\mathcal{C}^\perp}(z_0, \ldots, z_{q-1}) = \frac{1}{|\mathcal{C}|} W_{\mathcal{C}}\left(\sum_{i=0}^{q-1} \chi_1(\omega_0\omega_i)z_i, \ldots, \sum_{i=0}^{q-1} \chi_1(\omega_r\omega_i)z_i, \ldots\right) \tag{5.3.1}$$

where $|\mathcal{C}| = q^k$,

or alternatively

$$W_{\mathcal{C}}(z_0, \ldots, z_{q-1}) = \frac{1}{|\mathcal{C}^\perp|} W_{\mathcal{C}^\perp}\left(\sum_{s=0}^{q-1} \chi_1(\omega_0\omega_i)z_i, \ldots, \sum_{i=0}^{q-1} \chi_1(\omega_r\omega_i)z_i, \ldots\right) \tag{5.3.2}$$

where $|\mathcal{C}| = q^{n-k}$.

Let

$$y_0 = \sum_{i=0}^{q-1} \chi_1(\omega_0\omega_i)z_i$$

$$\cdots\cdots \tag{5.3.3}$$

$$y_{q-1} = \sum_{i=0}^{q-1} \chi_1(\omega_{q-1}\omega_i)z_i$$

and we can write Equation (5.3.2) as

$$W_{\mathcal{C}}(z_0, \ldots, z_{q-1}) = \frac{1}{|\mathcal{C}^\perp|} W_{\mathcal{C}^\perp}(y_0, \ldots, y_{q-1}) \tag{5.3.4}$$

In our case

$$q = 128,$$
$$n-k = 2.$$

Hence, the weight enumerator of the original code is related to the weight enumerator of the dual code as

$$W_{\&}(z_0,\ldots, z_{127}) = \frac{1}{128^2} W_{\&^{\perp}} (y_0,\ldots, y_{127}).$$

Thus if $c^{\perp}$ is an arbitrary code word in the dual code with comp($c^{\perp}$) = $(t_0,\ldots, t_{127})$, then the weight distribution of the original code is

$$\sum_{c\in\&} z_0^{s_0} z_1^{s_1} \cdots z_{127}^{s_{127}} = \frac{1}{128^2} \sum_{c^{\perp}\in\&^{\perp}} y_0^{t_0} y_1^{t_1} \cdots y_{127}^{t_{127}}. \quad (5.3.5)$$

5.4 Further Analysis

Though the complete weight enumerator of the dual code can be generated on the computer, it is to be observed that it is not possible to find the composition of each and every code word in the original code by using Equation (5.3.5). Also note that expressions derived for various events in section (5.2) are in terms of the composition variables $s_0,s_1,\ldots, s_{127}$. Hence, we should find expressions for these events that are functions of the variables $z_0,z_1,\ldots, z_{127}$ only and where the composition variables $s_0,s_1,\ldots, s_{127}$ which define the composition of each of the code words, do not appear explicitly. Now, using Equation (5.2.3), we can show that

$$s_i c/z_i = \frac{\partial c}{\partial z_i}$$

and (5.4.1)

$$s_j s_i^* c/(z_j z_i) = \frac{\partial^2 c}{\partial z_j \partial z_i}.$$

Using above equation, Equations (5.2.6), (5.2.7), (5.2.8), (5.2.10), (5.2.11), (5.2.12) and (5.2.13) can be rewritten as

$$e = \sum_{c \in \&} (z_0 + \cdots + z_{127}) \sum_{i=0}^{127} \frac{\partial c}{\partial z_i} - n\, c - n(z_1 + \cdots + z_{127}) z_0^{n-1} \tag{5.4.2}$$

$$e = \sum_{c \in \&} (z_1^* + \cdots + z_{128}^*) \sum_{i=0}^{127} \frac{\partial c}{\partial z_i} - n(z_1^* + \cdots + z_{128}^*) z_0^{n-1} \tag{5.4.3}$$

$$e = (z_1^* + \cdots + z_{128}^*)^2 \sum_{c \in \&} \sum_{j=0}^{127} \sum_{i=0}^{127} \tfrac{1}{2} \frac{\partial^2 c}{\partial z_j \partial z_i} - \frac{n(n-1)}{2} (z_1^* + \cdots + z_{128}^*)^2\, z_0^{n-2} \tag{5.4.4}$$

$$P_{SE}(a) = \sum_{c \in \&} c - \frac{1}{n} \sum_{c \in \&} z_0 \frac{\partial c}{\partial z_0} \tag{5.4.5}$$

$$P_{SE}(b) = \sum_{c \in \&} \Big[(z_0 + \cdots + z_{127}) \sum_{i=0}^{127} \frac{\partial c}{\partial z_i} - n\, c - \frac{1}{n} \Big\{ (z_0 + \cdots + z_{127}) \Big(\frac{\partial c}{\partial z_0} + z_0 \frac{\partial}{\partial z_0} \Big(\sum_{i=0}^{127} \frac{\partial c}{\partial z_i} \Big) \Big) - n\, z_0 \frac{\partial c}{\partial z_0} \Big\} \Big] \tag{5.4.6}$$

$$P_{SE}(c) = (z_1^* + \cdots + z_{128}^*) \sum_{c \in \&} \Big[\sum_{i=0}^{127} \frac{\partial c}{\partial z_i} - \frac{1}{n} \Big(\frac{\partial c}{\partial z_0} + z_0 \frac{\partial}{\partial z_0} \Big(\sum_{i=0}^{127} \frac{\partial c}{\partial z_i} \Big) \Big) \Big] \tag{5.4.7}$$

$$P_{SE}(d) = \tfrac{1}{2} (z_1^* + \cdots + z_{128}^*)^2 \sum_{c \in \&} \Big[\sum_{j=0}^{127} \sum_{i=0}^{127} \frac{\partial^2 c}{\partial z_j \partial z_i} - \frac{1}{n} \Big\{ z_0 \frac{\partial}{\partial z_0} \Big(\sum_{j=0}^{127} \sum_{i=0}^{127} \frac{\partial^2 c}{\partial z_j \partial z_i} \Big) + 2 \frac{\partial}{\partial z_0} \Big(\sum_{i=0}^{127} \frac{\partial c}{\partial z_i} \Big) \Big\} \Big] \tag{5.4.8}$$

A close examination of the above equations reveals that, we have to evaluate expressions of the type

(i) $\sum\limits_{c \in \&} \sum\limits_{i=0}^{127} \frac{\partial c}{\partial z_i}$

(ii) $\sum\limits_{c \in \&} \sum\limits_{j=0}^{127} \sum\limits_{i=0}^{127} \frac{\partial^2 c}{\partial z_j \partial z_i}$.

The corresponding expressions are obtained in terms of the weight distribution of the dual code by using Equation (5.3.5). This is done as follows.

(i) $\sum\limits_{c \in \&} \sum\limits_{i=0}^{127} \frac{\partial c}{\partial z_i}$.

$$\sum_{c \in \&} \sum_{i=0}^{127} \frac{\partial c}{\partial z_i} = \sum_{i=0}^{127} \frac{\partial}{\partial z_i} \left(\sum_{c \in \&} c \right)$$

Using Equation (5.3.5), we get

$$\sum_{i=0}^{127} \frac{\partial}{\partial z_i} \left(\sum_{c \in \&} c \right) = \sum_{i=0}^{127} \frac{\partial}{\partial z_i} \left(\frac{1}{128^2} \sum_{c^{\perp} \in \&^{\perp}} y_0^{t_0} \cdots y_{127}^{t_{127}} \right)$$

$$= \frac{1}{128^2} \sum_{c^{\perp} \in \&^{\perp}} \sum_{i=0}^{127} \frac{\partial}{\partial z_i} (y_0^{t_0} \cdots y_{127}^{t_{127}})$$

$$= \frac{1}{128^2} \sum_{c^{\perp} \in \&^{\perp}} \sum_{i=0}^{127} \sum_{j=0}^{127} t_j \, y_0^{t_0} \cdots y_j^{t_j - 1} \cdots y_{127}^{t_{127}} \; \frac{\partial y_j}{\partial z_i}$$

Using Equation (5.3.3), we get

$$\frac{\partial y_j}{\partial z_i} = \chi_1(\omega_j \omega_i).$$

Substituting $\frac{\partial y_j}{\partial z_i}$ from the above equation, we can write

$$\sum_{c\in\&}\sum_{i=0}^{127}\frac{\partial c}{\partial z_i} = \frac{1}{128^2}\sum_{c^\perp\in\&^\perp}\sum_{i=0}^{127}\sum_{j=0}^{127} t_j\, y_0^{t_0}\cdots y_j^{t_j-1}\cdots y_{127}^{t_{127}}\,\chi_1(\omega_j\omega_i)$$

$$= \frac{1}{128^2}\sum_{c^\perp\in\&^\perp}\sum_{j=0}^{127} t_j\, y_0^{t_0}\cdots y_j^{t_j-1}\cdots y_{127}^{t_{127}}\sum_{i=0}^{127}\chi_1(\omega_j\omega_i) \tag{5.4.9}$$

From the theory of group characters [13]

$$\sum_{i=0}^{127}\chi_1(\omega_j\omega_i) = 128 \text{ if } j = 0$$

$$= 0 \quad \text{otherwise}$$

Using the above result, expression (5.4.9) can be reduced to

$$\sum_{c\in\&}\sum_{i=0}^{127}\frac{\partial c}{\partial z_i} = \frac{1}{128^2}\sum_{c^\perp\in\&^\perp} 128\, t_0\, c^\perp/y_0, \tag{5.4.10}$$

where $c^\perp = y_0^{t_0}\, y_1^{t_1}\cdots y_{127}^{t_{127}}$.

(ii) $\sum_{c\in\&}\sum_{j=0}^{127}\sum_{i=0}^{127}\frac{\partial^2 c}{\partial z_j\partial z_i}$

$$\sum_{c\in\&}\sum_{j=0}^{127}\sum_{i=0}^{127}\frac{\partial^2 c}{\partial z_j\partial z_i} = \sum_{j=0}^{127}\sum_{i=0}^{127}\frac{\partial^2}{\partial z_j\partial z_i}\sum_{c\in\&^\perp} c$$

$$= \sum_{j=0}^{127}\sum_{i=0}^{127}\frac{\partial^2}{\partial z_j\partial z_i}\left(\frac{1}{128^2}\sum_{c^\perp\in\&^\perp} y_0^{t_0}\cdots y_{127}^{t_{127}}\right)$$

$$= \frac{1}{128^2}\sum_{c^\perp\in\&^\perp}\sum_{j=0}^{127}\frac{\partial}{\partial z_j}\left(\sum_{i=0}^{127}\frac{\partial}{\partial z_i}(y_0^{t_0}\cdots y_{127}^{t_{127}})\right) \tag{5.4.11}$$

It was shown in part (i) that

$$\sum_{i=0}^{127}\frac{\partial}{\partial z_i}(y_0^{t_0}\cdots y_{127}^{t_{127}}) = 128\, t_0\, y_0^{t_0-1}\, y_1^{t_1}\cdots y_{127}^{t_{127}},$$

and, therefore, Equation (5.4.11) can be simplified to

$$\frac{1}{128^2} \sum_{c^\perp \in \mathscr{C}^\perp} \sum_{j=0}^{127} \frac{\partial}{\partial z_j} (128\ t_0\ y_0^{t_0-1}\ y_1^{t_1} \cdots y_{127}^{t_{127}})$$

$$= \frac{1}{128^2} \cdot \sum_{c^\perp \in \mathscr{C}^\perp} 128 \cdot 128 \cdot t_0(t_0 - 1) y_0^{t_0-2}\ y_1^{t_1} \cdots y_{127}^{t_{127}} \tag{5.4.12}$$

Since the dual of a maximum distance separable code is also a maximum distance separable code, the dual code of (27,25) RS code which is (27,2) code has the following Hamming weight distribution

1 code word has all n symbols as zeros (all 0 code word), $t_0 = n$

3429 code words have only one 0 symbol, $t_0 = 1$

12984 code words have no 0 symbols, $t_0 = 0$

Note that t_0 is the number of zero symbols in code word of the dual code and, therefore, the term $t_0(t_0 - 1)$ is non-zero only for the all zero code word, i.e. for $t_0 = n$. Consequently,

$$\sum_{c \in \mathscr{C}} \sum_{j=0}^{127} \sum_{i=0}^{127} \frac{\partial^2 c}{\partial z_j \partial z_i} = n(n-1)\, y_0^{n-2} \tag{5.4.13}$$

Equations (5.4.10) and (5.4.13) are used to substitute for the corresponding summation in the expressions for P_{ICD} and P_{SE}. Finally, all the patterns that correspond to each of the events of interest for the (27,25) code are given as below

(1) $P_{CD} = z_0^{27} + \binom{27}{1} (z_1 + z_2 + \cdots + z_{127}) z_0^{26} + \binom{27}{1} (z_1^* + \cdots + z_{128}^*) z_0^{26}$

$+ \binom{27}{2} (z_1^* + \cdots + z_{128}^*)^2\ z_0^{25}$.

(2) $$P_{ICD} = \frac{1}{128^2} \sum_{c^\iota \in \&^\iota} [128\, t_0\, c^\iota - 26c^\iota + 128(z_1^* + \cdots + z_{128}^*)\, t_0\, c^\iota / y_0]$$

$$+ \binom{27}{2} (z_1^* + \cdots + z_{128}^*)^2\, y_0^{25} - P_{CD}$$

(3) $$P_F = 1 - P_{CD} - P_{ICD}$$

(4) $$P_{SE} = P_{ICD} + P_{CD} + \frac{1}{27.128^2} \sum_{c^\iota \in \&^\iota} \Big[(25z_0 - z_1 - \cdots - z_{127}) \Big(\sum_{i=0}^{127} \frac{t_i}{y_i} \Big) c^\iota$$

$$- 128.z_0\, t_0 \Big(\frac{t_0 - 1}{y_0} + \frac{t_1}{y_1} + \cdots + \frac{t_{127}}{y_{127}} \Big) c^\iota - (z_1^* + \cdots + z_{128}^*) \Big(\sum_{i=0}^{127} \frac{t_i}{y_i} \Big) c^\iota$$

$$- 128.z_0 (z_1^* + \cdots + z_{128}^*)\, t_0 \Big(\frac{t_0 - 1}{y_0} + \frac{t_1}{y_1} + \cdots + \frac{t_{127}}{y_{127}} \Big) c^\iota / y_0$$

$$- 128(z_1^* + \cdots + z_{128}^*)^2\, t_0 \Big(\frac{t_0 - 1}{y_0} + \frac{t_1}{y_1} + \cdots + \frac{t_{127}}{y_{127}} \Big) c^\iota / y_0 \Big]$$

$$- \frac{26.25}{2} (z_1^* + \cdots + z_{128}^*)^2\, z_0\, y_0^{24}$$

These expressions are computed for different values of the input bit error rate P. Plots of P_{ud}, P_{CD}, P_{ICD}, P_f and P_{SE} versus P are given in Figures (5.2), (5.3), (5.4), (5.5) and (5.6) respectively.

5.5 A Note on the Performance of (224,175) Binary Code

At this moment complete performance evaluation of (224,175) binary code obtained from (28,25) RS code does not seem to be computationally feasible on a binary symmetric channel. However, the probability of correct decoding P_{CD} for this code can be calculated as follows.

This code can decode correctly all patterns of received vector patterns given in section (5.2.1). It can also decode correctly those received vectors that

- have three non-zero symbols and all three symbols have odd weight (three erasures)
- have two non-zero symbols with one of the two having odd weight and the other one having even weight (one erasure, one error).

These vectors can be represented by the patterns

$$\binom{n}{3}(z_1^* + \cdots + z_{128}^*)^3\ z_0^{n-3}$$

$$+\ n(n-1)(z_1 + \cdots + z_{127})(z_1^* + \cdots + z_{128}^*)z_0^{n-2}$$

Hence for (224,175) binary code, all the received patterns that are decoded correctly are

$$z_0^{28} + 28(z_1 + \cdots + z_{127})z_0^{27} + 28(z_1^* + \cdots + z_{128}^*)z_0^{27}$$

$$+ \binom{28}{2}(z_1^* + \cdots + z_{128}^*)^2\ z_0^{26} + \binom{28}{3}(z_1^* + \cdots + z_{128}^*)^3\ z_0^{25}$$

$$+\ 28.27\ (z_1 + \cdots + z_{127})(z_1^* + \cdots + z_{128}^*)z_0^{26}.$$

A plot of P_{CD} versus the input bit error rate P is given in Figure (5.3) for this code.

5.6 Discussion of Results

The binary code obtained as a result of mapping (27,25) RS code defined over $GF(2^7)$ into a binary code, is a (216,175) code and has a minimum distance d_b equal to 6. Therefore, it should correct all the possible single and double errors in a received word. This, indeed, is the case for the decoding algorithm described in chapter 3. The code can also decode correctly certain patterns of errors in more than two bits. Such a scheme can find a very wide application in digital communication system where the errors occur randomly or cluster in bursts.

Although the exact performance analysis of the (224,175) binary code obtained as a result of mapping (28,25) RS code defined as $GF(2^7)$, is not feasible, yet we are assured of a better performance as compared to the (216,175) binary code since the (224,175) binary code has a minimum distance d_b of 8 and the decoding algorithm described in chapter 3 corrects any three or fewer bits in error as well as some error patterns of more than three bits.

Both of the high rate codes analysed here compare well with the codes described in [14] and chapters 2 and 3 of [7] for byte oriented information systems. It must be pointed that the codes detailed in [14] and [7] are strictly random error correcting while the codes given here can correct both random as well as bursts of error. However, the latter codes require one more byte of redundancy as compared to the former code. For example, the (216,175) code based on the scheme given in [7] can correct all single errors, double errors and any triple error pattern occuring in three distinct information bytes[1], whereas, the (224,175) binary code described in this thesis can decode correctly all the possible error patterns of single, double and triple errors and certain restricted error patterns of more than three bits.

If further improvement in performance is required for a system, then a code of higher minimum distance can be derived from the coding scheme described in chapter 2. The formulation of the code given in

---

[1] It is 70 percent of all the possible triple error patterns.

[7] needs to be explored further for the case of more than two bytes of redundancy and the properties of the scheme are not known completely.

Hence if a high rate coding scheme is required for a byte oriented information system and the channel introduces only random errors during the transmission, then the scheme given in [7] may be desirable for the purpose of error control but if the channel characteristics are not determined completely or if it introduces both random and burst errors, then the coding scheme analysed in this thesis, may be preferred.

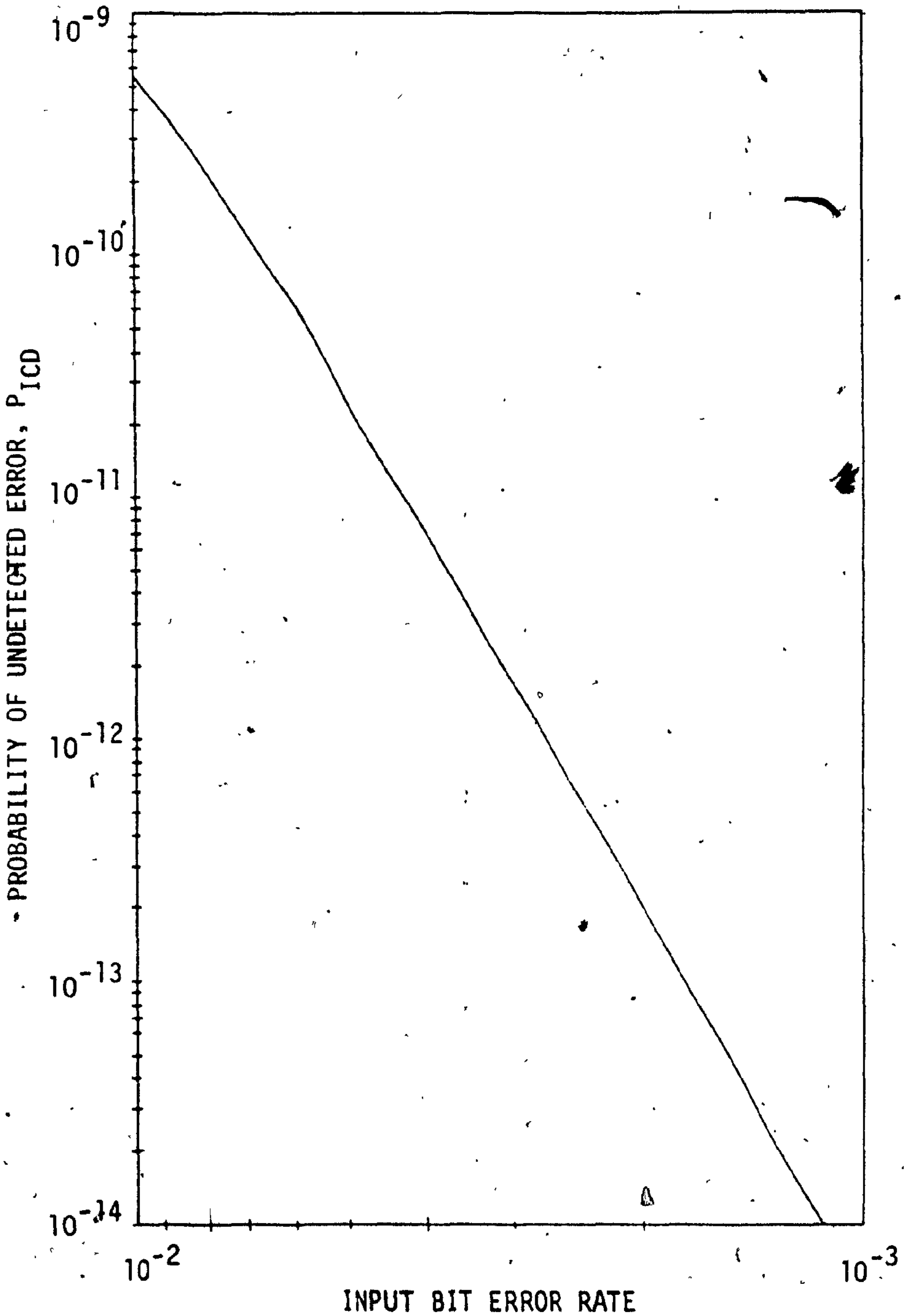

FIG. 5.2. PROBABILITY OF UNDETECTED ERROR Vs. INPUT BIT ERROR RATE FOR (216,175) BINARY CODE

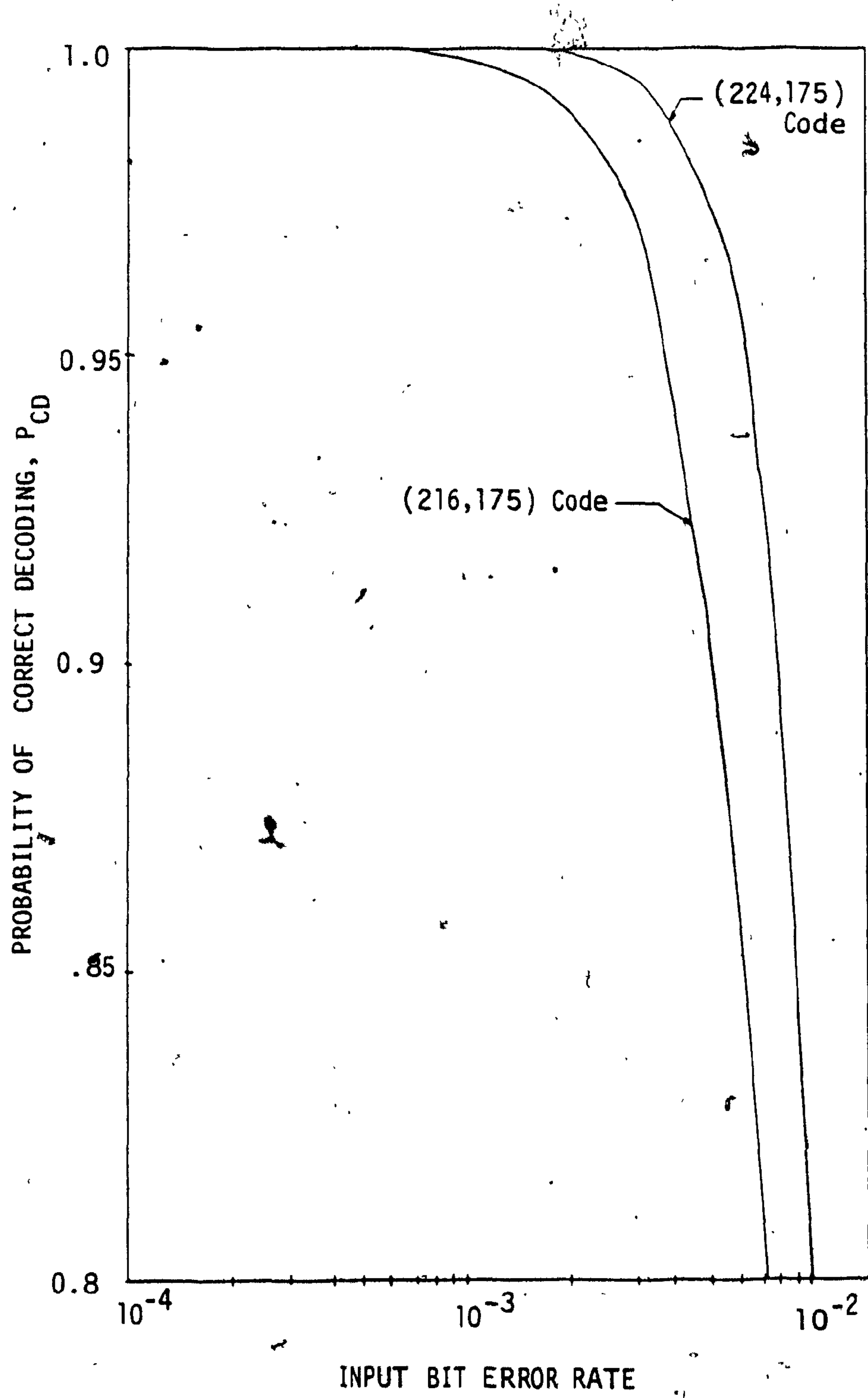

FIG. 5.3. PROBABILITY OF CORRECT DECODING Vs. INPUT BIT ERROR RATE FOR BINARY CODES

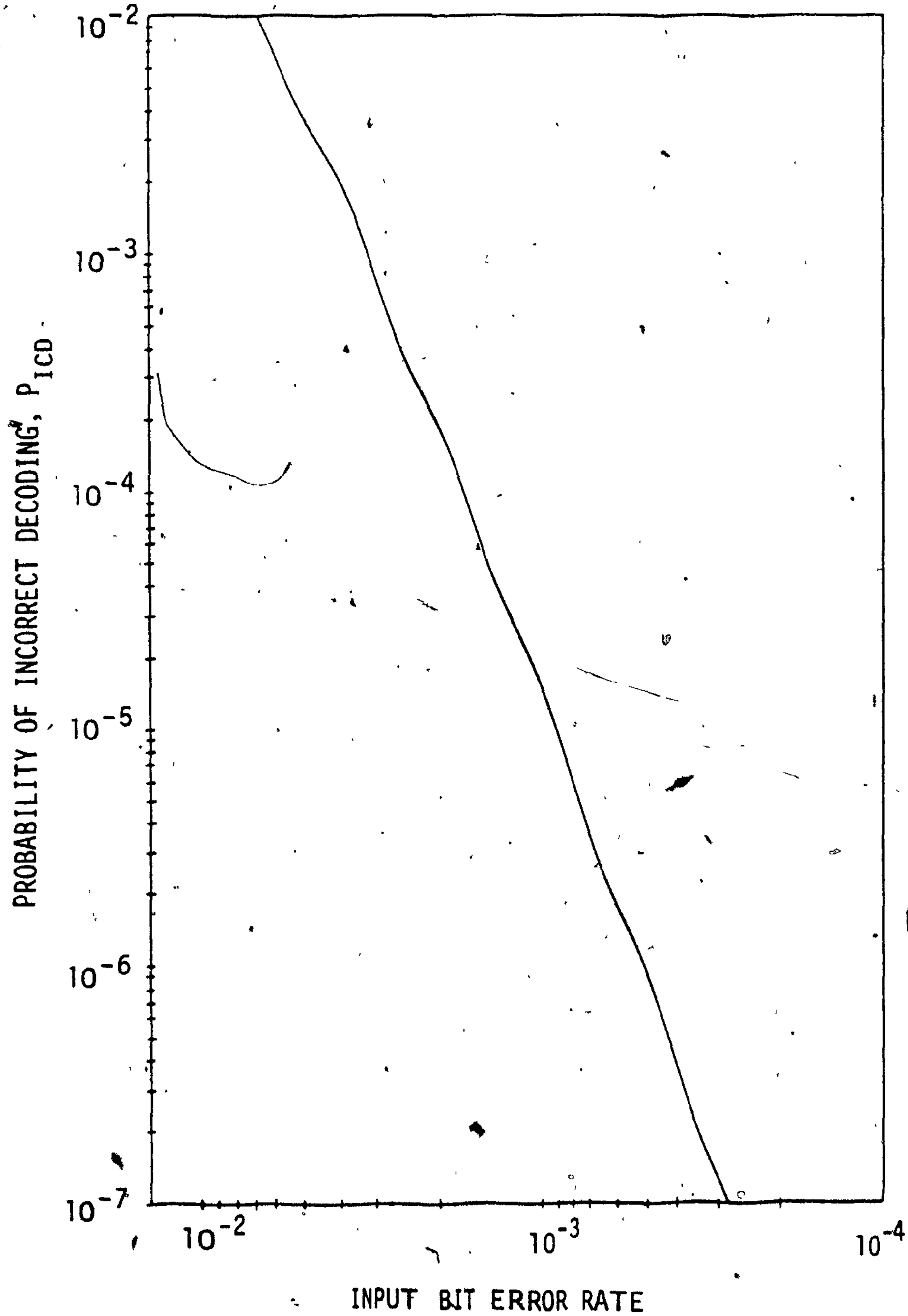

FIG. 5.4. PROBABILITY OF INCORRECT DECODING Vs. INPUT BIT ERROR RATE FOR (216,175) BINARY CODE

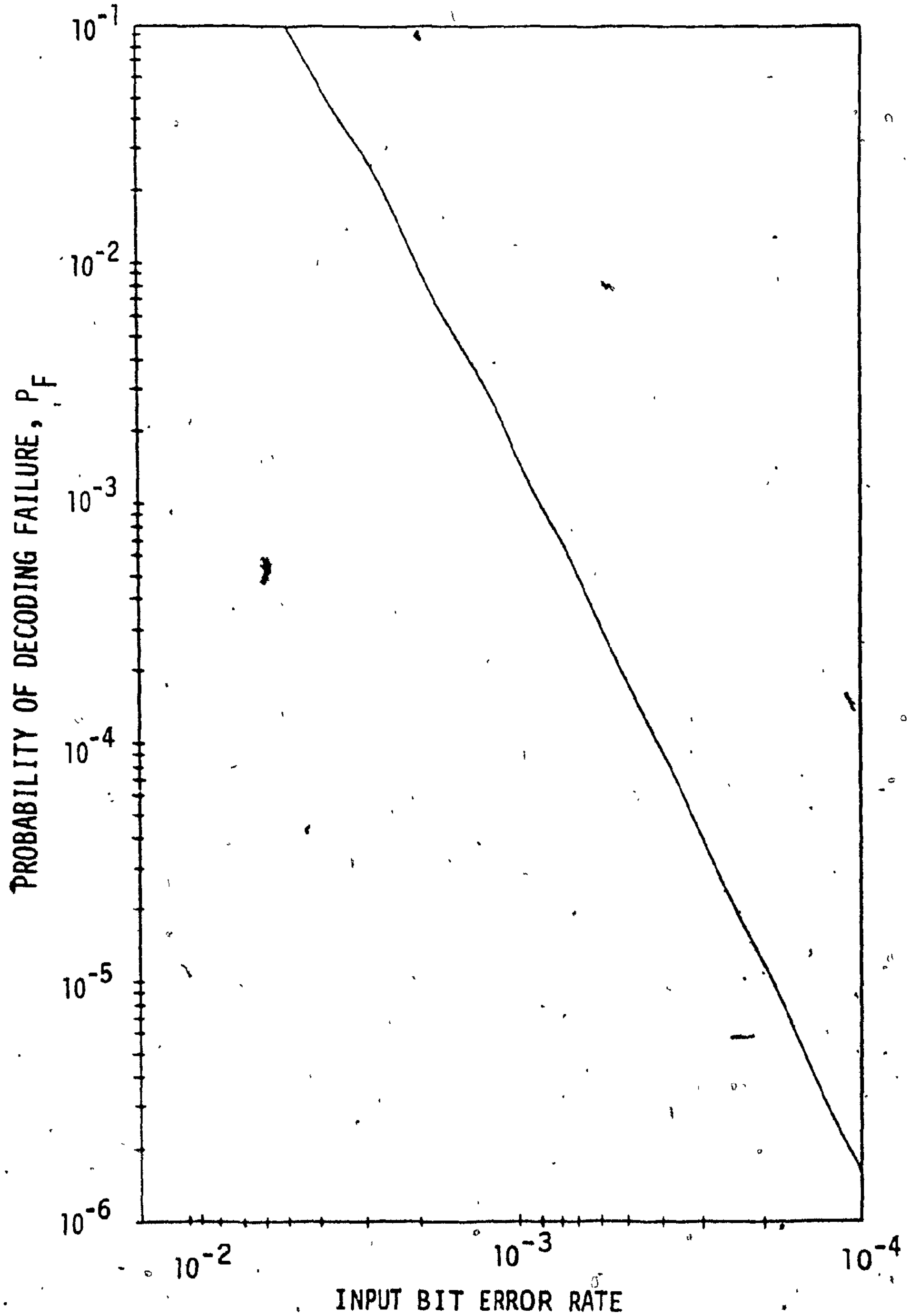

FIG. 5.5. PROBABILITY OF DECODING FAILURE Vs. INPUT BIT ERROR RATE FOR (216,175) BINARY CODE

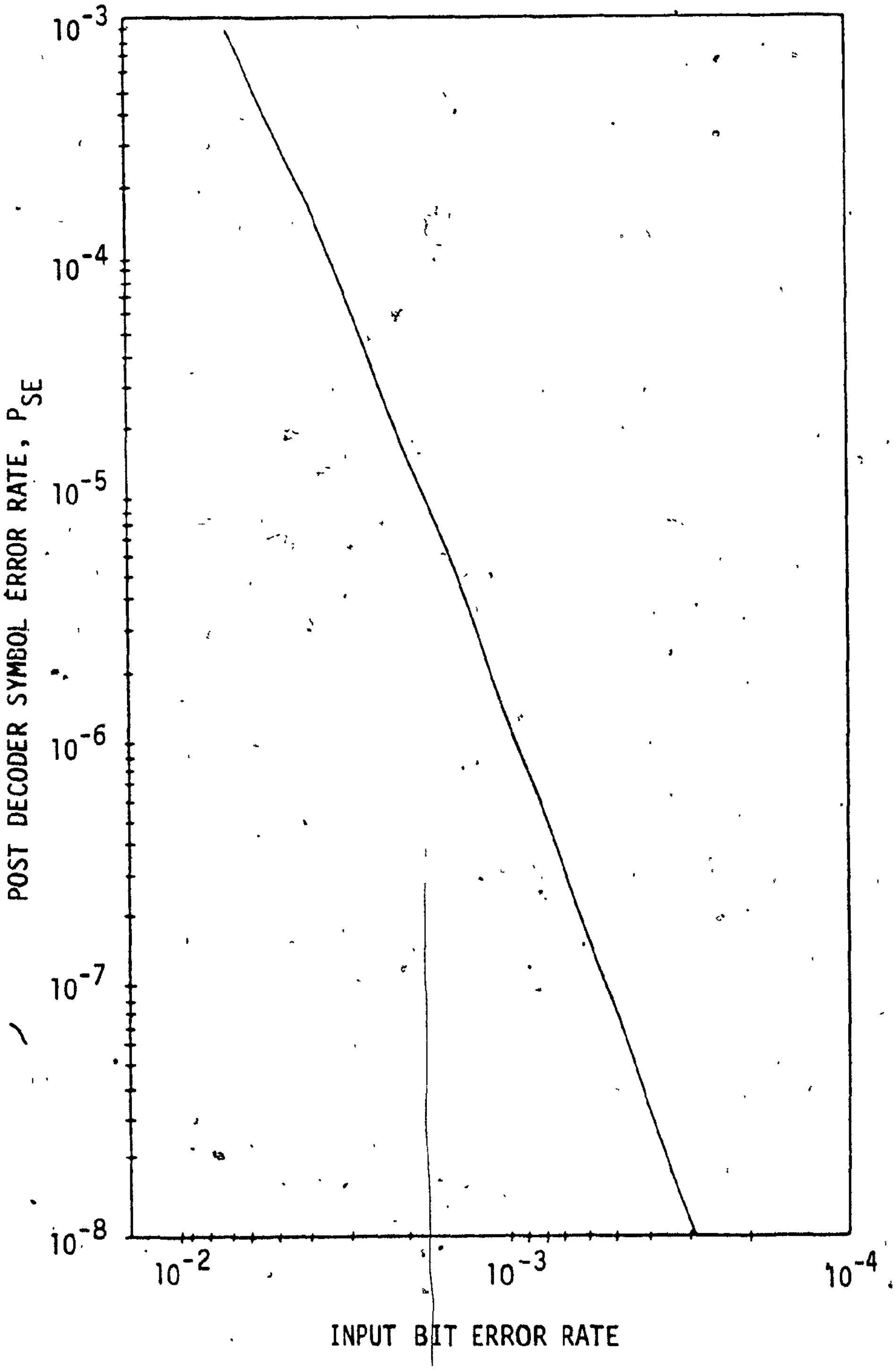

FIG. 5.6. POST DECODER SYMBOL ERROR RATE Vs. INPUT BIT ERROR RATE FOR (216,175) BINARY CODE

CHAPTER 6

Conclusions

The main objective of this thesis has been to design a coding scheme for even parity byte information systems where the number of parity bits added to any block of information bytes are a multiple of 8 and the parity bit present in each of the information bytes is not allowed to be altered. Although codes of any rate can be obtained by employing the coding scheme presented here, the emphasis has been on the development and analysis of high rate codes.

In chapter 2, the mapping of codes defined over $GF(2^m)$ into binary codes has been studied and it is shown that the set of all possible even parity 8-tuples forms a vector space of dimension 7. A RS code defined over $GF(2^7)$ was considered and the procedure to get the generator matrix in the systematic form for the binary codes obtained by mapping the RS codes was outlined. The parity bytes added to the information bytes, are shown to have even parity. A shift register implementation for the coding scheme was illustrated.

Since the emphasis is on high rate codes, a decoding algorithm was presented only for the cases when the RS code, used to derive the binary code, had a minimum distance of 3 and 4. The decoding algorithm makes use of the parity bit present in each of the received bytes. As it is in closed form, the decoding can be implemented by digital hardware that is capable of operating on very high speed of data transmission.

A channel can be modelled as either a q-ary symmetric channel $(q \neq 2)$ or as a binary symmetric channel, depending on the modulation scheme that is used to transmit the data. Both models were considered for the statistical performance analysis of the binary code. However, it was observed that the analysis of the code on the binary symmetric channel is very complicated and, therefore, it was performed only for the (216,175) binary code. The analysis of the (224,175) binary code was restricted to the evaluation of the probability of correct decoding.

The coding scheme derived in this thesis has the potential of combating random and burst errors that normally occur in digital communication systems. It is clear from the plots for the probability of various post decoder error events given in chapters 4 and 5 that using these codes can lead to significant improvement in the overall system performance.

BIBLIOGRAPHY

1. F.J. MacWilliams and N.J.A. Sloane, The Theory of Error-Correcting Codes, North-Holland Pub. Co., 1977.

2. V.K. Bhargava, David Haccoun, Robert Matyas and Peter P. Nuspl, Digital Communications by Satellite, John Wiley and Sons, 1981.

3. W.W. Peterson and E.J. Weldon, Jr., Error-Correcting Codes, MIT Press, 1972.

4. Shu Lin, An Introduction to Error-Correcting Codes, Prentice Hall, 1970.

5. E.R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, 1968.

6. I.S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields", J. Soc. Indust. Appl. Math., 8, pp. 300-304, June 1960.

7. P.E. Allard, V.K. Bhargava and G.E. Séguin, "Realization, Economic and Performance Analysis of Error-Correcting Codes and ARQ Systems for Broadcast Telidon and Other Videotex Systems", Research Report, Department of Electrical Engineering, Concordia University, Montreal, June 1981. (Prepared for the Canadian Dept. of Communications under DSS contract No. OSU80-00133.)

8. Jeremy Clark, "Design and Construction of a (75,50) Forward Error Correcting CODEC for High Speed Digital Communications", A major technical report in the Faculty of Engineering and Computer Science, Concordia University, Montreal, 1982.

9. G.D. Forney, Concatenated Codes, Research Monograph 37, MIT Press, 1966.

10. P.J. Trafton, "Performance of Reed-Solomon Codes on a Symmetric Erasure Channel", Proceedings IEEE Int. Conf. Comm., pp. 18-1 - 18-5, 1970.

11. Zelma McC. Huntoon and Arnold M. Michelson, "On the Computation of the Probability of Post-Decoding Error Events for Block Codes", IEEE Trans. on Info. Theory, Vol. IT-23, pp. 399-403, May 1977.

12. Jack K. Wolf, Arnold M. Michelson and Allen H. Levesque, "On the Probability of Undetected Error For Linear Block Codes", IEEE Trans. on Comm., Vol. COM-30, No. 2, pp. 317-324, Feb. 1980.

13. F.J. MacWilliams, "A Theorem on the Distribution of Weights in a Systematic Code", Bell System Technical Journal, Vol. 12, pp. 79-94, Jan. 1963.

14. G.E. Séguin, P.E. Allard and V.K. Bhargava, "A Class of High Rate Codes for Byte-Oriented Information Systems", To appear in IEEE Trans. on Comm.

APPENDIX A

Table of Elements of $GF(2^7)$

In this appendix, a table of 'i' with the binary 8-tuple representation of α^i for the basis chosen in Equation (2.2.1) is given, where α is a primitive element of $GF(2^7)$. We have used the recursion $\alpha^7 = 1 + \alpha^3$.

i	α^i
0	10000001
1	10000010
2	10000100
3	10001000
4	10010000
5	10100000
6	11000000
7	00001001
8	00010010
9	00100100
10	01001000
11	10011001
12	10110010
13	11100100
14	01000001
15	10001011
16	10010110
17	10101100
18	11011000
19	00111001
20	01110010
21	11101101
22	01010011
23	10101111
24	11011110
25	00110101
26	01101010
27	11011101
28	00110011
29	01100110
30	11000101
31	00000011
32	00000110
33	00001100
34	00011000
35	00110000
36	01100000
37	11001001
38	00011011
39	00110110
40	01101100
41	11010001
42	00101011
43	01010110
44	10100101
45	11001010
46	00011101
47	00111010
48	01110100
49	11100001
50	01001011
51	10011111
52	10111110
53	11111100
54	01110001
55	11101011
56	01011111
57	10110111
58	11101110
59	01010101
60	10100011
61	11000110
62	00000101
63	00001010
64	00010100
65	00101000
66	01010000
67	10101001
68	11010010
69	00101101
70	01011010
71	10111101
72	11111010
73	01111101
74	11110011
75	01101111
76	11010111
77	00100111
78	01001110
79	10010101

80	10101010
81	11010100
82	00100001
83	01000010
84	10001101
85	10011010
86	10110100
87	11101000
88	01011001
89	10111011
90	11110110
91	01100101
92	11000011
93	00001111
94	00011110
95	00111100
96	01111000
97	11111001
98	01111011
99	11111111
100	01110111
101	11100111
102	01000111
103	10000111
104	10001110
105	10011100
106	10111000
107	11110000
108	01101001
109	11011011
110	00111111
111	01111110
112	11110101
113	01100011
114	11001111
115	00010111
116	00101110
117	01011100
118	10110001
119	11100010
120	01001101
121	10010011
122	10100110
123	11001100
124	00010001
125	00100010
126	01000100

It can be observed that all the binary 8-tuples listed above have even weight and along with the all 0 8-tuple, these 8-tuples form a vector space of dimension 7.

APPENDIX B

Weight Distribution of (27,25) and (28,25) RS Codes

The number of code words having exactly h non-zero symbols for a maximum distance separable code is given by Equation (4.2.2). If $A_1(h)$ and $A_2(h)$ represent the Hamming weight distribution of (27,25,3) and (28,25,4) RS codes defined over $GF(2^7)$ respectively, then Equation (4.2.2) can be used to obtain the following table.

h	$A_1(h)$	$A_2(h)$
0	1	1
1	0	0
2	0	0
3	371475	0
4	2.786 E8	2.600 E06
5	1.628 E11	1.548 E09
6	7.581 E13	7.538 E11
7	2.888 E16	3.001 E14
8	9.170 E18	1.003 E17
9	2.459 E21	2.831 E19
10	5.620 E23	6.830 E22
11	1.103 E26	1.419 E24
12	1.868 E28	2.554 E26
13	2.737 E30	3.992 E28
14	3.476 E32	5.432 E30
15	3.826 E34	6.439 E32
16	3.645 E36	6.644 E34
17	2.995 E38	5.956 E36
18	2.113 E40	4.622 E37
19	1.271 E42	3.090 E40
20	6.458 E43	1.766 E42
21	2.734 E45	8.543 E43
22	9.469 E46	3.452 E45
23	2.614 E48	1.144 E47
24	5.533 E49	3.026 E48
25	8.433 E50	6.149 E49
26	8.238 E51	9.011 E50
27	3.875 E52	8.477 E51
28	-	3.845 E52

where a E b means $a \times 10^b$.

Note that $\sum_{h=0}^{27} A_1(h) = \sum_{h=0}^{28} A_2(h) = 128^{25}$ as the code is defined on $GF(2^7)$ and the number of information symbols in each code is 25.