On linear unequal error protection codes

Eindhoven
University of Technology
the Netherlands

Department of
Mathematics and
Computing Science

On Linear Unequal Error Protection Codes

by W.J. van Gils

EUT-Report 82-WSK-02
June 1982

MASTER'S THESIS

On Linear Unequal Error Protection Codes

by

W.J. van Gils

Supervisor: Prof. dr. J.H. van Lint

Adviser : Dr. ir. H.C.A. van Tilborg

July 1982

Eindhoven University of Technology

Department of Mathematics and Computing Science

ABSTRACT

It is possible for a linear block code to provide more protection against errors for selected message positions than is guaranteed by the minimum distance of the code. Codes having this property are called Linear Unequal Error Protection (LUEP) codes. In this report the optimal encoding of LUEP codes is discussed and bounds on the length of a code that ensures a given unequal error protection are derived. A number of constructions of LUEP codes are given. Cyclic UEP codes together with Majority Logic Decoding of certain classes of these are treated. A list of LUEP codes of minimal length and a list of cyclic UEP codes are included.

AMS Subject Classification: 94B05, 94B60.

PREFACE

In data transmission and processing a desired level of error control is guaranteed by using error-correcting codes. Most block codes considered in the literature have the property that their correcting capabilities are described in terms of the correct reception of the entire message. These codes can successfully be applied in those cases where all positions in a message word require equal protection against errors.
However, many applications exist in which some message positions are more important than other ones. For example in transmitting numerical data, errors in the sign or in the high-order digits are more serious than are errors in the low-order digits. As another example consider the transmission of message words from different sources simultaneously in only one codeword, while the different sources have mutually different demands concerning the protection against errors.
Accordingly there is an interest in codes which protect some positions in a message word against a larger number of errors than other ones. Such codes are called Unequal Error Protection codes (abbreviated: UEP codes). Masnick and Wolf (1967) introduced the concept of Unequal Error Protection. But, in contrast with what one would expect, they considered error protection of each single position in a codeword. In this report we consider error protection of single positions in the message words, following the formal definitions of Dunning and Robbins (1978).
In Chapter 1 we introduce the concept of Linear Unequal Error Protection codes (LUEP codes) and define a vector, the so-called separation vector, by which the error-correcting capability of a LUEP code is measured. In Section 1.2 we consider a special form of a generator matrix for a LUEP code, the so-called canonical form, introduced by Boyarinov and Katsman (1981).
The error-correcting capability of a LUEP code, measured by the separation vector, depends upon the choice of a generator matrix which is used for the encoding of the message set. But fortunately every code has a so-called optimal generator matrix, whose separation vector is componentwise larger than or equal to the separation vector of any other generator matrix of the code. Chapter 2 provides a necessary and sufficient condition for a generator matrix to be optimal. It is also shown that a generator matrix of a code which has the smallest number of nonzero entries is optimal. The results in Chapter 2 are from Dunning and Robbins (1978).

An interesting and basic problem is to find a LUEP code with a given dimension and separation vector such that its length is minimal and hence its information rate is maximal. In Chapter 3 we derive a number of bounds on the length of LUEP codes. For the special case where all message positions are equally protected, some of our bounds reduce to the well-known Singleton, Plotkin, and Griesmer Bounds. Some earlier work on bounds was done by Katsman (1980); he derived Corollary (3.3.14) for the binary case. Our bounds give better results than the bound of Katsman (1980) does (cf. Section 3.4). The Theorems (3.3.2), (3.3.6), the binary version of (3.3.12), the Corollaries (3.3.7), (3.3.8), and formula (35) for linear UEP codes were already reported in van Gils (1981). Appendix A provides a table of all binary LUEP codes with maximal separation vector and length less than or equal to 15.

In Chapter 4 we construct a number of LUEP codes. Section 4.1 provides some infinite families of LUEP codes which have minimal length and maximal separation vector. Section 4.2 contains a number of constructions which build LUEP codes from (LUEP) codes of smaller length, such as the direct sum and direct product construction, the $|u|u+v|$ construction, and concatenation.

Chapter 5 deals with cyclic UEP codes. In Section 5.1 we give an optimal generator matrix for a cyclic UEP code and observe how its error-correcting capability depends on the weight distribution of its cyclic subcodes. In Section 5.2 we consider classes of cyclic UEP codes which can be decoded by Majority Logic Decoding Methods. Earlier results (Theorem (5.2.1) and (5.2.8)) on cyclic UEP codes were obtained by Dyn'kin and Togonidze (1976). Appendix B provides a table of all binary cyclic UEP codes of length less than or equal to 39.

CONTENTS

LIST OF SYMBOLS

$\mathbf{F}_q$, GF(q)	: the finite field (Galois field) of q elements.
$\mathbf{F}_q[x]$	: the ring of polynomials in x over $\mathbf{F}_q$.
$\mathbf{F}_q[x]/(x^n-1)$	: the residue class ring $\mathbf{F}_q[x]$ modulo (x^n-1).
n	: the wordlength of a code.
k	: the dimension of a code.
d	: the minimum distance of a code.
$\underline{s}$	: the separation vector of a code.
$wt(\underline{c})$	: the Hamming weight of the vector $\underline{c}$.
$wt[C]$	: $\min \{ wt(\underline{c}) \mid \underline{c} \in C \}$.
$WT(C)$	: $\{ wt(\underline{c}) \mid \underline{c} \in C \}$.
$C(\rho)$	: $\{ \underline{c} \in C \mid wt(\underline{c}) \leq \rho \}$.
G	: the generator matrix of a code.
G_{i*}	: the i^{th} row of the matrix G.
G_{*j}	: the j^{th} column of the matrix G.
$R(G)$	: the set of rows of the matrix G.
$R(G)(\rho)$	: $\cap \{ X \subset R(G) \mid C(\rho) \subset \langle X \rangle \}$.
$\underline{s}(G)$	: the separation vector of the matrix G.
$\langle X \rangle$	: the linear span of the set X.
[n,k,d]	: a linear code of wordlength n, dimension k, and minimum distance d.
$[n,k,\underline{s}]$	: a linear code of wordlength n, dimension k , and separation vector $\underline{s}$.
$n_q(\underline{s})$	: cf. page 10.
$n_q^{ex}(\underline{s})$	: cf. page 10.
M_i	: a minimal ideal in $\mathbf{F}_q[x]/(x^n-1)$.
M_i	: a generator matrix of M_i.
C_i	: the cyclotomic coset modulo n containing i.
$\lfloor x \rfloor$	: the largest integer less than or equal to x.
$\lceil x \rceil$	: the smallest integer larger than or equal to x.
$a \mid b$	: a is a divisor of b.
I_k	: the k by k unit matrix.

1. INTRODUCTION

In this chapter we give an introduction to the concept of Linear Unequal Error Protection Codes. The reader is assumed to be familiar with the basic principles of linear algebra, finite fields, and error-correcting codes. For an extensive treatment we refer to MacWilliams and Sloane (1978) and van Lint (1982). In Section 1.1 we define "Unequal Error Protection" and in Section 1.2 we derive a special form of the generator matrix for a linear UEP code, the so-called canonical form.

1.1 Definition of Linear Unequal Error Protection Codes

Let q be a prime power and let $\mathbb{F}_q = GF(q)$ be the Galois field of order q. A linear [n,k] code C of length n and dimension k over $\mathbb{F}_q$ is a k-dimensional linear subspace of $\mathbb{F}_q^n$. A generator matrix G of this code is a k by n matrix whose rows form a basis of C. The bijection from $\mathbb{F}_q^k$ onto C which maps any element $\underline{m} \in \mathbb{F}_q^k$ of the message set onto a codeword $\underline{c} = \underline{m}G$ is called an encoding of C by means of the generator matrix G. For $\underline{x} \in \mathbb{F}_q^n$, $wt(\underline{x})$ denotes the (Hamming) weight of $\underline{x}$, i.e. the number of nonzero components in $\underline{x}$.
Dunning and Robbins (1978) have introduced the following formal definition.

(1.1.1) Definition: For a linear [n,k] code C over the alphabet $\mathbb{F}_q$ the separation vector $\underline{s}(G) = (s(G)_1,\ldots,s(G)_k)$ of length k, with respect to a generator matrix G of C, is defined by

$$s(G)_i := \min \{ wt(\underline{m}G) \mid \underline{m} \in \mathbb{F}_q^k , m_i \neq 0 \} \qquad (1)$$

$(i=1,\ldots,k)$.

This means that for any $\alpha,\beta \in \mathbb{F}_q$, $\alpha \neq \beta$, the sets $\{ \underline{m}G \mid \underline{m} \in \mathbb{F}_q^k , m_i = \alpha \}$ and $\{ \underline{m}G \mid \underline{m} \in \mathbb{F}_q^k , m_i = \beta \}$ are at distance $s(G)_i$ apart $(i=1,\ldots,k)$. This observation implies the following error-correcting capability of a code when we use it on a q-ary symmetric channel.

(1.1.2) Theorem: For a linear [n,k] code C over $\mathbf{F}_q$, which uses the matrix G for its encoding, we can guarantee the correct reception of the i^{th} digit of the message word if the error pattern has a Hamming weight less than or equal to $\lfloor (s(G)_i - 1)/2 \rfloor$ by using maximum likelihood decoding.

From Definition (1.1.1) it is immediately clear that the minimum distance of the code equals

$$d = \min \{ s(G)_i \mid i = 1,\ldots,k \}. \qquad (2)$$

Hence by Theorem (1.1.2) we can guarantee correct reception of the complete message if the error pattern has a weight less than or equal to $\lfloor (d-1)/2 \rfloor$.
The following definition is an immediate consequence of Theorem (1.1.2).

(1.1.3) Definition: If a linear code C has a generator matrix G such that the components of the separation vector $\underline{s}(G)$ are not mutually equal, then the code C is called a Linear Unequal Error Protection Code (LUEP code).

One can easily decode LUEP codes by using Syndrome Decoding (cf. MacWilliams and Sloane (1978)). This decoding method reaches the correction capability given by Theorem (1.1.2), because of the following fact. For a fixed coset R of a linear code C, encoded by means of a generator matrix G, let U be the set of coset leaders of R. For any $\underline{r} \in R$, $\underline{r}+U$ contains all codewords which are closest to $\underline{r}$, i.e. at a distance $d(\underline{r},C)$, the distance between $\underline{r}$ and C, from $\underline{r}$. If $i \in \{ 1,\ldots,k \}$ is such that the weight of the elements of U is less than of equal to $\lfloor (s(G)_i - 1)/2 \rfloor$, then the i^{th} digits of the messages corresponding to the elements of $\underline{r}+U$ are mutually equal. Hence if $\underline{r}$ is the received word, Syndrome Decoding correctly reproduces the i^{th} digit of the message sent.
In Section 5.2 we treat a Majority Logic Decoding method for certain classes of cyclic UEP codes.

1.2 The canonical form of a generator matrix

By simultaneously permuting the message positions in the message words and the rows of a generator matris G, we may obtain a generator matrix $\overline{G}$ for the code such that $\underline{s}(\overline{G})$ is nonincreasing, i.e. $\underline{s}(\overline{G})_i \geq \underline{s}(\overline{G})_{i+1}$ for $i = 1,\ldots,k-1$. From now on we assume that the message positions and the rows in generator matrices are ordered such that the corresponding separation vectors are nonincreasing.
Boyarinov and Katsman (1981) have introduced a special form of a generator matrix, called a canonical form.

(1.2.1) Definition: A generator matrix G of a linear [n,k] code, whose nonincreasing separation vector $\underline{s}(G)$ has z distinct components $s_{i_1} > s_{i_2} > \ldots\ldots > s_{i_z}$ with multiplicities resp. $k_1,k_2,\ldots\ldots,k_z$, is called canonical if G contains a lower triangular partitioned matrix of order k by k having z unit matrices of order $k_1 \times k_1$, $k_2 \times k_2$,....., $k_z \times k_z$ on its diagonal. That is, after a proper permutation of the columns of G we get a matrix of the following form.

$$\left[\begin{array}{c|c|c|c|c} I_{k_1} & 0 & \cdots & 0 & 0 \\ \hline G_{2,1} & I_{k_2} & & 0 & 0 \\ \hline \vdots & \vdots & & \vdots & \vdots \\ \hline G_{z-1,1} & G_{z-1,2} & & I_{k_{z-1}} & 0 \\ \hline G_{z,1} & G_{z,2} & & G_{z,z-1} & I_{k_z} \end{array}\right] P \qquad (3)$$

For $k \in \mathbb{N}$ we define a partial order in $\mathbb{R}^k$ by

$$\underline{x} \leq \underline{y} \;:\Leftrightarrow\; x_i \leq y_i \text{ for } i = 1,\ldots,k,$$

where $\underline{x},\underline{y} \in \mathbb{R}^k$. We say that $\hat{\underline{x}}$ is a maximum of the set $A \subset \mathbb{R}^k$ if for all $\underline{x} \in A$, $\underline{x} \leq \hat{\underline{x}}$.
Any generator matrix G of a code can be transformed into a canonical generator matrix G_{can} of the code such that $\underline{s}(G_{can}) \geq \underline{s}(G)$ by a number

of elementary transformations on the rows of G, i.e. permutation and addition of rows and multiplication of rows by scalars. This is a consequence of the following theorem.

(1.2.2) Theorem: For $k,n \in \mathbb{N}$, $i,j \in \{1,\ldots,k\}$, $i \neq j$, $\alpha \in \mathbb{F}_q \setminus \{0\}$ and a k by n matrix G over $\mathbb{F}_q$ let G' be a k by n matrix obtained by replacing the i^{th} row of G by the sum of the i^{th} row and α times the j^{th} row of G. Then the separation vector $\underline{s}(G')$ satisfies

$$s(G')_v = s(G)_v \quad \text{for } v \neq j \tag{4}$$

$$s(G')_j \begin{cases} = s(G)_j & \text{if } s(G)_j < s(G)_i \quad (5) \\ \geq s(G)_j & \text{if } s(G)_j = s(G)_i \quad (6) \\ = s(G)_i & \text{if } s(G)_j > s(G)_i. \quad (7) \end{cases}$$

Proof: For a set $X \in \mathbb{F}_q^n$, $\langle X \rangle$ denotes the linear span of X and wt[X] is the minimum weight in X, i.e. $\min \{ wt(\underline{x}) \mid \underline{x} \in X \}$. For a matrix A, R(A) denotes the set of rows of A and A_{i*} denotes the i^{th} row of A.

For $v \neq i,j$ we have $s(G')_v := wt[G'_{v*} + \langle R(G') \setminus \{G'_{v*}\} \rangle] =$

$= wt[G_{v*} + \{\beta(G_{i*} + \alpha G_{j*}) \mid \beta \in \mathbb{F}_q\} + \langle R(G) \setminus \{G_{i*}, G_{v*}\} \rangle] =$

$= wt[G_{v*} + \{\beta G_{i*} \mid \beta \in \mathbb{F}_q\} + \langle R(G) \setminus \{G_{i*}, G_{v*}\} \rangle] = wt[G_{v*} + \langle R(G) \setminus \{G_{v*}\} \rangle] =$

$= s(G)_v$.

$s(G')_i := wt[G'_{i*} + \langle R(G') \setminus \{G'_{i*}\} \rangle] = wt[G_{i*} + \alpha G_{j*} + \langle R(G) \setminus \{G_{i*}\} \rangle] =$

$= wt[G_{i*} + \langle R(G) \setminus \{G_{i*}\} \rangle] = s(G)_i$.

$s(G')_j := wt[G'_{j*} + \langle R(G') \setminus \{G'_{j*}\} \rangle] =$

$= wt[G_{j*} + \{\beta(G_{i*} + \alpha G_{j*}) \mid \beta \in \mathbb{F}_q\} + \langle R(G) \setminus \{G_{i*}, G_{j*}\} \rangle]$.

For $s(G)_j < s(G)_i$ we have that $wt[G_{j*} + \langle R(G) \setminus \{G_{i*}, G_{j*}\} \rangle] = s(G)_j$,

$wt[G_{j*} + \{\beta(G_{i*} + \alpha G_{j*}) \mid \beta \in \mathbb{F}_q \setminus \{0\}\} + \langle R(G) \setminus \{G_{i*}, G_{j*}\} \rangle] \geq s(G)_i$

and hence $s(G')_j = s(G)_j$, i.e. formula (5).

In a similar way we obtain formula (6) and (7).

□

From this theorem it is immediately clear that we can transform an arbitrary generator matrix G of a code into a canonical generator matrix G_{can} such that $\underline{s}(G_{can}) \geq \underline{s}(G)$ by applying a sequence of elementary transformations on G. This theorem also shows (by formula (7)) that if we want to transform a generator matrix G into a systematic generator matrix G_{syst}, we cannot guarantee that $\underline{s}(G_{syst}) \geq \underline{s}(G)$.

(1.2.3) Example: For q = 2,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \qquad (8)$$

has separation vector $\underline{s}(G) = (5,4,4,4,4)$. It is impossible to transform G into a systematic generator matrix G_{syst} such that $\underline{s}(G_{syst}) \geq (5,4,4,4,4)$. Actually, a 5 x 10 binary systematic generator matrix with a separation vector of at least (5,4,4,4,4) does not exist (cf. van Gils (1981)).

1.3 Notes

One can generalize Definition (1.1.1) for nonlinear codes. Consider a code C over the alphabet $\mathbb{F}_q$ containing q^k codewords. Let the message set $\mathbb{F}_q^k$ be encoded according to the bijection η , mapping $\mathbb{F}_q^k$ onto C. The separation vector $\underline{s}(\eta)$ of C with respect to the encoding function η is defined by

$$s(\eta)_i := \min \{ wt(\eta(m)-\eta(m')) \mid m,m' \in \mathbb{F}_q^k , m_i \neq m_i' \} \qquad (9)$$

for i = 1,...,k. Of course Theorem (1.1.2) also holds for nonlinear UEP codes.

Dunning and Robbins (1978) have also considered the Lee Metric.

Boyarinov and Katsman (1981) have introduced the canonical form of a generator matrix.

2. OPTIMAL ENCODING OF LINEAR UNEQUAL ERROR PROTECTION CODES

The separation vector defined by formula (1) depends upon the choice of a generator matrix for the code. In this chapter we show that a linear code C has an optimal generator matrix G^*, i.e. $\underline{s}(G^*) \geq \underline{s}(G)$ for all generator matrices G of C. In Section 2.1 we give a necessary and sufficient condition for a generator matrix to be optimal. In Section 2.2 we show that for a linear code a generator matrix with the smallest number of nonzero entries is optimal. The results in this chapter are from Dunning and Robbins (1978).

2.1 A necessary and sufficient condition for a generator matrix to be optimal

(2.1.1) Definition: For a linear code C a generator matrix G is called optimal, whenever $\underline{s}(G)$ is the maximum of the set of nonincreasing separation vectors $\underline{s}(A)$, where A is a generator matrix of C.

For a linear [n,k] code we define $WT(C) := \{ wt(\underline{c}) \mid \underline{c} \in C \}$, i.e. the set of all possible weights of codewords in C. For $\rho \in WT(C)$, $C(\rho) := \{ \underline{c} \in C \mid wt(\underline{c}) \leq \rho \}$ is the set of codewords in C having a weight of at most ρ. For a generator matris G of C let $R(G) := \{G_{1*},\ldots,G_{k*}\}$ denote the set of rows of G and let $R(G)(\rho) := \cap \{ X \subset R(G) \mid C(\rho) \subset \langle X \rangle \}$ be the smallest subset of R(G) such that $C(\rho)$ is contained in its linear span. The relation between $\underline{s}(G)$ and $R(G)(\rho)$ is given in the following lemma.

(2.1.2) Lemma: A generator matrix G of a linear [n,k] code C satisfies

$$s(G)_i \leq \rho \Leftrightarrow G_{i*} \in R(G)(\rho) \qquad (10)$$

for each $i \in \{1,\ldots,k\}$ and $\rho \in WT(C)$.

Proof: Let $i \in \{1,\ldots,k\}$ and $\rho \in WT(C)$. If $G_{i*} \in R(G)(\rho)$ then $C(\rho) \not\subset \langle R(G)\backslash\{G_{i*}\}\rangle$ and hence $C(\rho) \cap C\backslash\langle R(G)\backslash\{G_{i*}\}\rangle \neq \emptyset$, which implies that

$s(G)_{i*} := wt[C\backslash\langle R(G)\backslash\{G_{i*}\}\rangle] \leq \rho$. On the other hand, if $G_{i*} \notin R(G)(\rho)$ then $C(\rho) \subset \langle R(G)\backslash\{G_{i*}\}\rangle$ and hence $s(G)_i := wt[C\backslash\langle R(G)\backslash\{G_{i*}\}\rangle] \geq$ $\geq wt[C\backslash C(\rho)] > \rho$.

□

The following theorem provides a necessary and sufficient condition for a generator matrix to be optimal for its rowspace.

(2.1.3) Theorem: A generator matrix G of a linear [n,k] code C is optimal if and only if for any $\rho \in WT(C)$ a subset $X \subset R(G)$ of the rows of G exists such that $\langle C(\rho)\rangle = \langle X\rangle$.

Proof:
Sufficiency: Suppose a generator matrix G of C satisfies the condition in the theorem. Assume that G is not optimal, i.e. a generator matrix A exists such that $\underline{s}(G) \neq \underline{s}(A)$. Let i be minimal such that $s(G)_i < s(A)_i$ and set $\rho := s(A)_i - 1$. Since $s(A)_1 \geq \ldots \geq s(A)_i > \rho$, we have $C(\rho) \subset \langle A_{(i+1)*},\ldots,A_{k*}\rangle$ and thus also $\langle C(\rho)\rangle \subset \langle A_{(i+1)*},\ldots,A_{k*}\rangle$. On the other hand we have that $\rho \geq s(G)_i \geq \ldots \geq s(G)_k$, which by Lemma (2.1.2) implies that $G_{i*},\ldots,G_{k*} \in R(G)(\rho)$. Combining these observations with the fact that $\langle C(\rho)\rangle = \langle R(G)(\rho)\rangle$ we obtain $\langle G_{i*},\ldots,G_{k*}\rangle \subset \langle R(G)(\rho)\rangle = \langle C(\rho)\rangle \subset \langle A_{(i+1)*},\ldots,A_{k*}\rangle$, which is a contradiction. Hence our assumption was wrong and so G is optimal.
Necessity: Suppose G is an optimal generator matrix for the code C. Let $\rho \in WT(C)$ and let A be a generator matrix of C such that $\langle C(\rho)\rangle = \langle A_{(k-p+1)*},\ldots,A_{k*}\rangle$, where $p := \dim\langle C(\rho)\rangle$. By Definition (1.1.1) we have that $s(A)_1,\ldots,s(A)_{k-p} > \rho$ and hence $s(G)_1 \geq \ldots \geq s(G)_{k-p} > \rho$, since G is optimal. Again applying Lemma (2.1.2) yields that $R(G)(\rho) \subset \{G_{(k-p+1)*},\ldots,G_{k*}\}$ and hence $\langle C(\rho)\rangle = \langle G_{(k-p+1)*},\ldots,G_{k*}\rangle$, since $\langle C(\rho)\rangle \subset \langle R(G)(\rho)\rangle$ and $\dim\langle G_{(k-p+1)*},\ldots,G_{k*}\rangle = p$.

□

(2.1.4) Corollary: Any linear code has an optimal generator matrix.

Hence the following definition makes sense.

(2.1.5) Definition: The separation vector of a linear code is defined as the separation vector of an optimal generator matrix of the code.

We shall use the notation [n,k,$\underline{s}$] for a linear code of length n, dimension k, and separation vector $\underline{s}$. For $i = 1,\ldots k$, $\lfloor (s_i - 1)/2 \rfloor$ is called

the protection level of the i^{th} message position.

2.2 Minimal weight generator matrices

Optimal generator matrices which are easy to compute, given the rowspace, are the so-called minimal weight generator matrices.

(2.2.1) Definition: For a linear [n,k] code C over $\mathbf{F}_q$ a generator matrix G is called a minimal weight generator matrix if $\sum_{i=1}^{k} wt(G_{i*})$ is a minimum of the set

$$\{ \sum_{i=1}^{k} wt(A_{i*}) \mid A \text{ is a generator matrix of } C \}.$$

We shall show that a minimal weight generator matrix is optimal. First we show that it is easy to compute the separation vector of these matrices.

(2.2.2) Lemma: If G is a minimal weight generator matrix of a k-dimensional code, then

$$wt(G_{i*}) = s(G)_i \tag{11}$$

for $i = 1,..,k$.

Proof: Let G be a generator matrix of a k-dimensional code such that $wt(G_{i*}) \neq s(G)_i$ for some $i \in \{1,\ldots,k\}$. Since $s(G)_i \leq wt(G_{i*})$ we have the strict inequality $s(G)_i < wt(G_{i*})$.
Let $\underline{v} \in C\backslash\langle R(G)\backslash\{G_{i*}\}\rangle$ be such that $wt(\underline{v}) = s(G)_i$. Then we have that

$$\sum_{j=1}^{k} wt(G_{j*}) > \sum_{j=1, j\neq i}^{k} wt(G_{j*}) + wt(\underline{v}).$$

$G_{1*},\ldots,G_{(i-1)*},\underline{v},G_{(i+1)*},\ldots,G_{k*}$ are linearly independent and so they form the rows of a generator matrix for C. Hence G is not a minimal weight generator matrix. This proves the lemma.

□

(2.2.3) Lemma: A minimal weight generator matrix G of a k-dimensional linear code C satisfies

$$\langle C(\rho)\rangle = \langle \{ G_{i*} \mid i \in \{1,\ldots,k\} , wt(G_{i*}) \leq \rho \} \rangle \qquad (12)$$

for any $\rho \in WT(C)$.

Proof: That LHS $\supset$ RHS is trivial. On the other hand let $\underline{c} \in C(\rho)$. The message $\underline{m}$ such that $\underline{c} = \underline{m}G$ satisfies $m_j = 0$ for all j satisfying $s(G)_j > \rho$, which by Lemma (2.2.2) is equivalent to $wt(G_{j*}) > \rho$. Hence $\underline{c} \in$ RHS. The RHS of formula (12) is a linear space, so $\langle C(\rho)\rangle \subset$ RHS. □

(2.2.4) Theorem: A minimal weight generator matrix is optimal.

Proof: Combine Theorem (2.1.3) and the Lemmas (2.2.2) and (2.2.3). □

Besides a proper permutation of the rows any minimal weight generator matrix of a k-dimensional linear code C can be constructed by the following algorithm.

1. Set i := k.
2. Choose $\underline{v} \in C\backslash\langle G_{(i+1)*},\ldots,G_{k*}\rangle$ such that $wt(\underline{v}) = wt[C\backslash\langle G_{(i+1)*},\ldots,G_{k*}\rangle]$.
3. Set $G_{i*} := \underline{v}$.
4. If i > 1 then decrease i by 1 and go to step 2, otherwise stop.

2.3 Notes

The results of this chapter are from Dunning and Robbins (1978). They also show that a linear code has an optimal encoding (linear or nonlinear) and that no nonlinear encoding is better, i.e. has a larger separation vector, than an optimal linear encoding. They also give an example of a nonlinear code which has no optimal encoding.
If we replace the Hamming metric by the Lee metric, all lemmas and theorems in this chapter remain valid (cf. Dunning and Robbins (1978)).

3. BOUNDS ON THE LENGTH OF LINEAR UNEQUAL ERROR PROTECTION CODES

A basic problem is to find linear UEP codes with a given dimension and separation vector such that their length is minimal and hence their information rate is maximal. In Section 3.1 we give two formal definitions of functions we want to consider together with their properties. In Section 3.2 resp. 3.3 we give upper resp. lower bounds for these functions. Appendix A gives function values for binary LUEP codes of length less than or equal to 15.

3.1 Definitions and properties

(3.1.1) Definition: For any $k \in \mathbf{N}$, $\underline{s} \in \mathbf{N}^k$ and prime power q we define

$n_q(\underline{s})$ as the length of the shortest linear code over $\mathbf{F}_q$ of dimension k with a separation vector of at least $\underline{s}$.

and

$n_q^{ex}(\underline{s})$ as the length of the shortest linear code over $\mathbf{F}_q$ of dimension k with separation vector (exactly) $\underline{s}$.

An $[n_q(\underline{s}),k,\underline{s}]$ code is called length-optimal. It is called optimal, if an $[n_q(\underline{s}),k,\underline{t}]$ with $\underline{t} \geq \underline{s}$, $\underline{t} \neq \underline{s}$ does not exist.

(3.1.2) Properties: For any $k \in \mathbf{N}$, $\underline{s},\underline{t} \in \mathbf{N}^k$ and prime power q the functions $n_q(.)$ and $n_q^{ex}(.)$ have the following properties.

$$n_q(\underline{s}) \leq n_q^{ex}(\underline{s}), \tag{13}$$

$$\underline{s} \leq \underline{t} \Rightarrow n_q(\underline{s}) \leq n_q(\underline{t}), \tag{14}$$

$$\underline{s} \leq \underline{t} \not\Rightarrow n_q^{ex}(\underline{s}) \leq n_q^{ex}(\underline{t}). \tag{15}$$

To illustrate (15), observe that $n_2^{ex}(5,4,4) = 8$ (cf. Appendix A) and $n_2^{ex}(5,4,3) = 9$, which can be seen by easy verification.

3.2 Upper bounds

The following theorem provides a trivial upper bound for $n_q(.)$ and $n_q^{ex}(.)$ and an easy way to construct linear UEP codes.

(3.2.1) Theorem: For any prime power q, $k \in \mathbb{N}$, $v \in \mathbb{N}$, $\underline{s} \in \mathbb{N}^k$ and $k_0 = 0 < k_1 < k_2 < \ldots < k_v = k$ we have

$$n_q^{ex}(s_1,\ldots,s_k) \le \sum_{u=0}^{v-1} n_q^{ex}(s_{k_u+1}, s_{k_u+2}, \ldots, s_{k_{u+1}}). \tag{16}$$

The same inequality holds for $n_q(.)$ (Replace $n_q^{ex}(.)$ in (16) by $n_q(.)$).

Proof: For $u = 0,1,\ldots,v-1$ let G_u be a generator matrix of a $[n_q^{ex}(s_{k_u+1},\ldots,s_{k_{u+1}}), k_{u+1}-k_u]$ code over $\mathbb{F}_q$ with separation vector $(s_{k_u+1},\ldots,s_{k_{u+1}})$. Then

$$G := \left[\begin{array}{c|c|c} G_0 & 0 & \\ \hline 0 & G_1 & \\ \hline & & \\ 0 & & \end{array}\right.\quad \left.\begin{array}{c} 0 \\ \\ \\ \\ 0 \\ \hline 0 \quad G_{v-1} \end{array}\right] \tag{17}$$

is the generator matrix of a $[\sum_{u=0}^{v-1} n_q^{ex}(s_{k_u+1},\ldots,s_{k_{u+1}}), k]$ code with separation vector $\underline{s}$.

□

(3.2.3) Corollary: For any prime power q, $k \in \mathbb{N}$ and $\underline{s} \in \mathbb{N}^k$ we have

$$n_q^{ex}(\underline{s}) \le \sum_{i=1}^{k} s_i. \tag{18}$$

Proof: Apply Theorem (3.2.1) with $v = k$ and for $i = 1,\ldots k$, $k_i = i$ and G_i the $1 \times s_i$ all-one matrix.

□

Hence for any $\underline{s} \in \mathbb{N}^k$ it is possible to construct a k-dimensional code over $\mathbb{F}_q$ with separation vector $\underline{s}$.

3.3 Lower bounds

We start with a trivial lower bound on $n_q(.)$.

(3.3.1) Theorem: For any $k \in \mathbb{N}$, prime power q and nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$, $n_q(\underline{s})$ satisfies the inequality

$$n_q(\underline{s}) \geq 1 + n_q(s_1-1, s_2-1, \ldots, s_k-1). \tag{19}$$

Proof: By deleting a column from a k by $n_q(\underline{s})$ matrix G with separation vector $\underline{s}(G) \geq (s_1,\ldots,s_k)$ we obtain a k by $n_q(\underline{s})-1$ matrix G' with separation vector $\underline{s}(G') \geq (s_1-1, s_2-1, \ldots, s_k-1)$. □

(3.3.2) Theorem: For $q = 2$ and any $k \in \mathbb{N}$, $(s_1,\ldots,s_k) \in \mathbb{N}^k$ we have

$$n_2(s_1,s_2,\ldots,s_k) \geq n_2\left(2\left\lfloor\frac{s_1+1}{2}\right\rfloor, 2\left\lfloor\frac{s_2+1}{2}\right\rfloor, \ldots, 2\left\lfloor\frac{s_k+1}{2}\right\rfloor\right) - 1. \tag{20}$$

The same inequality holds when we replace $n_2(.)$ by $n_2^{ex}(.)$.

Proof: By adding an overall parity-check to a binary $[n = n_2(s_1,\ldots,s_k), k]$ code with a separation vector of at least $(s_1,\ldots,s_k)$, we obtain an $[n+1,k]$ code with a separation vector of at least $(2\lfloor(s_1+1)/2\rfloor, 2\lfloor(s_2+1)/2\rfloor, \ldots, 2\lfloor(s_k+1)/2\rfloor)$. □

(3.3.3) Example: $n_2(5,4,3) = 8$, $n_2(6,4,4) = 9$ (cf. Appendix A).

(3.3.4) Theorem: For a linear $[n,k]$ code over $\mathbb{F}_q$ with nonincreasing separation vector $\underline{s}$ the weight distribution $(A_i)_{i=0}^n$ must satisfy the inequality

$$\sum_{i \geq s_j} A_i \geq q^k - q^{k-j} \tag{21}$$

for all $j = 1,\ldots,k$.

Proof: For any $j \in \{1,\ldots,k\}$ a codeword corresponding to a message $\underline{m} \in \mathbb{F}_q^k$ such that $m_i \neq 0$ for some $i \in \{1,\ldots,j\}$ has a weight of at least s_j. □

The weight distribution $(A_i)_{i=0}^{n}$ of a linear $[n,k,\underline{s}]$ code also has to satisfy the following conditions.

$$\text{(i): } A_0 = 1;\ A_i = 0 \text{ for } i = 1,\ldots,s_k-1;\ \textstyle\sum_{i=0}^{n} A_i = q^k$$

and

$$\text{(ii): } \textstyle\sum_{i=0}^{n} p_m(i;n)A_i \geq 0 \text{ for } m = 0,1,\ldots,n,$$

where $p_m(x;n)$ are the so-called Krawtchouk polynomials defined by

$$p_m(x;n) := \textstyle\sum_{j=0}^{m} (-1)^j (q-1)^{k-j} \binom{x}{j} \binom{n-x}{m-j} \tag{22}$$

for $m = 0,1,\ldots,n$ and $x \in \mathbb{R}$ (cf. MacWilliams and Sloane (1978), Ch. 5 Theorem 6).
Combining the conditions (i) and (ii) with formula (21) we obtain a set of inequalities for the weight distribution of a code and hence a necessary condition on the existence of certain linear UEP codes. In many cases we can even add more conditions on $(A_i)_{i=0}^{n}$.

(3.3.5) Example: The weight distribution $(A_i)_{i=0}^{9}$ of a $[9,5,(4,4,4,3,3)]$ binary code has to satisfy

$$A_0 = 1,\ A_1 = A_2 = 0,\ A_3 = 2,\ A_i \in \mathbb{N} \text{ for } i = 4,\ldots,9,$$
$$A_4+A_6+A_8 = 15,\ A_5+A_7+A_9 = 14,$$

and formula (22) for $m = 1,2,6,7$, i.e.

$$\begin{aligned}
A_4 - A_5 - 3A_6 - 5A_7 - 7A_8 - 9A_9 &\geq -15\\
-A_4 - A_5 + 2A_7 + 10A_8 + 9A_9 &\geq -9\\
-A_4 - A_5 + 2A_6 - 7A_8 + 21A_9 &\geq -25\\
-A_4 + A_5 - 2A_7 + 10A_8 - 9A_9 &\geq -9.
\end{aligned}$$

It is easy to verify that this has no solution and hence a $[9,5,(4,4,4,3,3)]$ binary code does not exist.

(3.3.6) Theorem: For any $k \in \mathbb{N}$, any prime power q and any nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ we have

$$n_q(s_1,s_2,\ldots,s_k) \geq 1 + n_q(s_1,s_2,\ldots,s_{k-1}). \tag{23}$$

Proof: By deleting the column $\underline{e}_k := (0,0,\ldots,0,1)^T$ and the k^{th} row of an optimal canonical generator matrix of a linear $[n=n_q(s_1,\ldots,s_k),k]$ code over $\mathbb{F}_q$ with a separation vector of at least $\underline{s}$, we obtain a generator matrix of an $[n-1,k-1]$ code with a separation vector of at least $(s_1,s_2,\ldots,s_{k-1})$. □

(3.3.7) Corollary: For any $k,j \in \mathbb{N}$, $1 \le j \le k$, prime power q and nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ we have

$$n_q(s_1,s_2,\ldots,s_k) \ge j + n_q(s_1,s_2,\ldots,s_{k-j}). \tag{24}$$

(3.3.8) Corollary: For any $k \in \mathbb{N}$, prime power q and nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ we have

$$n_q(s_1,s_2,\ldots,s_k) \ge s_1 + k - 1. \tag{25}$$

For $s_1 = s_2 = \ldots = s_k$ Corollary (3.3.8) reduces to the Singleton Bound (cf. MacWilliams and Sloane (1978), Ch. 1 Theorem 11). By this last corollary we see that in a Maximum-Distance-Separable Code all information positions have the maximal protection level which is possible, i.e. $\lfloor (n-k+1)/2 \rfloor$.

(3.3.9) Example:
$n_2(6,6,4,4,4) \ge n_2(6,6,4,4) + 1 = 12.$
$n_2(6,6,4,4,4) \ge n_2(6,6) + 3 = 12.$
Actually $n_2(6,6,4,4,4) = 12$ (cf. Appendix A).

(3.3.10) Theorem: For $k,v \in \mathbb{N}$ and a nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ such that $s_{v-1} > s_v$ and $\sum_{i=v}^{k} s_i \le n_q^{ex}(\underline{s}) - 1$ we must have

$$n_q^{ex}(s_1,\ldots,s_k) \ge n_q(s_1-1,\ldots,s_{v-1}-1,s_v,\ldots,s_k) + 1. \tag{26}$$

Proof: Let $k,v \in \mathbb{N}$ and $\underline{s} \in \mathbb{N}^k$ be such that $s_{v-1} > s_v$ and $\sum_{i=v}^{k} s_i \le n_q^{ex}(\underline{s}) - 1$ and let G be a minimal weight generator matrix of an $[n=n_q^{ex}(\underline{s}),k,\underline{s}]$ code over $\mathbb{F}_q$. Since $\sum_{i=v}^{k} s_i \le n - 1$, G has a column containing zero elements in the last $k - v + 1$ positions. Deleting this column from G we obtain an $(n-1)$ by k matrix G', whose separation vector satisfies $\underline{s}(G') \ge$ $\ge (s_1-1,\ldots,s_{v-1}-1,s_v,\ldots s_k)$, since $s_{v-1} > s_v$. □

(3.3.11) Example: A binary linear code with separation vector (6,4,4,3,3,3,3) has a length of at least 13. Hence by Theorem (3.3.10), $n_2^{ex}(6,4,4,3,3,3,3) \geq$ $\geq n_2(5,3,3,3,3,3,3) + 1 \geq 14$ (cf. Appendix A).

(3.3.12) Theorem: For any $k \in \mathbb{N}$, prime power q and any nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ we have

$$n_q^{ex}(s_1,\ldots,s_k) \geq s_i + n_q(\hat{s}_1,\ldots,\hat{s}_{i-1},\hat{s}_{i+1},\ldots,\hat{s}_k) \tag{27}$$

for any $i \in \{1,\ldots,k\}$, where

$$\hat{s}_j := \begin{cases} s_j - \left\lfloor (q-1)s_i/q \right\rfloor & \text{for } j < i \\ \left\lceil s_j/q \right\rceil & \text{for } j > i. \end{cases} \tag{28}$$

Proof: Let C be a linear $[n=n_q^{ex}(\underline{s}),k,\underline{s}]$ code over $\mathbb{F}_q$ and let G be a minimal weight generator matrix for C. By Lemma(2.2.2), $wt(G_{i*}) = s_i$ for all $i = 1,\ldots,k$.
Fix $i \in \{1,\ldots,k\}$. Without loss of generality the first s_i columns of G have a 1 in the i^{th} row. Deleting these first s_i columns and the i^{th} row from G, we obtain a $(k-1)$ by $(n-s_i)$ matrix, $\hat{G}$. Clearly $\hat{G}$ has rank $(k-1)$, otherwise there would be a nonzero linear combination of rows of $\hat{G}$ which equals $\underline{0}$, and hence the corresponding linear combination of rows of G would have a distance less than s_i to αG_{i*} for some $\alpha \in \mathbb{F}_q \setminus \{0\}$, a contradiction. Hence $\hat{G}$ is a generator matrix of an $[n-s_i,k-1]$ code with a separation vector $\underline{\hat{s}} := \underline{s}(\hat{G}) = (\hat{s}_1,\ldots,\hat{s}_{i-1},\hat{s}_{i+1},\ldots,\hat{s}_k)$.
Let $j \in \{1,\ldots,k\}$, $j \neq i$ and let $\underline{m} \in \mathbb{F}_q^k$ be such that $m_i = 0$, $m_j \neq 0$ and $\underline{c} := \underline{m}G = (\underline{c}_1|\underline{c}_2)$, where $\underline{c}_1$ has length s_i, satisfies $wt(\underline{c}_2) = \hat{s}_j$. Since $m_j \neq 0$, we have that

$$wt(\underline{c}_1) + \hat{s}_j \geq s_j. \tag{29}$$

Furthermore, for some $\alpha \in \mathbb{F}_q \setminus \{0\}$ at least $\left\lceil wt(\underline{c}_1)/(q-1) \right\rceil$ components of $\alpha\underline{c}_1$ equal 1, and hence

$$wt(G_{i*}-\alpha\underline{c}) \leq s_i - \left\lceil wt(\underline{c}_1)/(q-1) \right\rceil + \hat{s}_j. \tag{30}$$

On the other hand we have that

$$wt(G_{i*}-\alpha\underline{c}) \geq \max\{s_i, s_j\}. \tag{31}$$

Combining (29), (30) and (31) gives formula (28), and hence (27) holds. □

(3.3.13) Lemma: For any $k \in \mathbb{N}$, prime power q and any nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ a linear $[n_q(\underline{s}),k]$ code with a nonincreasing separation vector $\underline{s}^*$ such that $\underline{s} \leq \underline{s}^* \leq s_1\underline{1}$ exists. ($\underline{1}$ is the all-one vector of length k).

Proof: Let G be a minimal weight generator matrix of an $[n=n_q(\underline{s}),k]$ code. If $s(G)_1 > s_1$ then replace a nonzero element in the first row of G by zero, to obtain a matrix G' whose separation vector satisfies $\underline{s}(G') \geq \underline{s}$ and $s(G')_1 = s_1 - 1$. We can repeat this procedure until we obtain an k x n matrix G^* with $\underline{s} \leq \underline{s}(G^*) \leq s_1\underline{1}$. □

Combining (3.3.12) and (3.3.13) gives the following corollary.

(3.3.14) Corollary: For any $k \in \mathbb{N}$, prime power q and nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$, $n_q(\underline{s})$ satisfies the inequalities

$$n_q(s_1,\ldots,s_k) \geq s_1 + n_q(\lceil s_2/q\rceil,\ldots,\lceil s_k/q\rceil), \tag{32}$$

$$n_q(s_1,\ldots,s_k) \geq \sum_{i=1}^{k} \lceil s_i/q^{i-1}\rceil. \tag{33}$$

Proof: According to Lemma (3.3.13), $n_q(\underline{s}) = n_q^{ex}(\underline{s}')$ for some $\underline{s} \leq \underline{s}' \leq s_1\underline{1}$ and hence by Theorem (3.3.12) we have that $n_q(\underline{s}) = n_q^{ex}(\underline{s}') \geq s_1' + n_q(\lceil s_2'/q\rceil,\ldots,\lceil s_k'/q\rceil) \geq s_1 + n_q(\lceil s_2/q\rceil,\ldots,\lceil s_k/q\rceil)$. Repeating this gives formula (33). □

For $s_1 = s_2 = \ldots = s_k$ Corollary (3.3.14) reduces to the Griesmer Bound (cf. MacWilliams and Sloane (1978), Ch. 17 Theorem 24). Deleting the $\lceil\ \rceil$ brackets in formula (33) we obtain an analog of the Plotkin Bound (cf. MacWilliams and Sloane (1978), Ch. 2 Theorem 1) for linear UEP codes.

Lemma (3.3.13) also implies the following corollary.

(3.3.15) Corollary: For any $k \in \mathbb{N}$, prime power q and any nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ we have

$$n_q(\underline{s}) = \min \{ n_q^{ex}(\underline{s}') \mid \underline{s} \leq \underline{s}' \leq s_{l\underline{l}} \}. \qquad (34)$$

This corollary allows us to use the bounds on $n_q^{ex}(.)$ to obtain bounds on $n_q(.)$.

(3.3.16) Examples:

(i): What is the minimum length of a binary linear code with a separation vector of at least (5,4,3,3,3,3)?

By Theorem (3.3.15) we have

$n_2(5,4,3,3,3,3) = \min \{ n_2^{ex}(\underline{s}) \mid (5,4,3,3,3,3) \leq \underline{s} \leq (5,5,5,5,5,5) \}$.

By (3.3.12), $n_2^{ex}(\underline{s}) \geq 3 + n_2(4,3,2,2,2) = 12$ for $(5,4,3,3,3,3) \leq \underline{s} \leq (5,4,4,4,4,3)$.

By (3.3.1), $n_2^{ex}(5,4,4,4,4,4) \geq 1 + n_2(4,3,3,3,3,3) = 12$.

By (3.3.12), $n_2^{ex}(\underline{s}) \geq 5 + n_2(3,2,2,2,2) = 12$ for $(5,5,3,3,3,3) \leq \underline{s} \leq (5,5,5,5,5,5)$.

(For values of $n_2(.)$ see Appendix A)

Hence $n_2(5,4,3,3,3,3) \geq 12$. A [12,6,(5,5,4,4,4,4)] code exists, so $n_2(5,4,3,3,3,3) = 12$.

(ii):What is the minimum length of a binary linear code with a separation vector of at least (6,6,6,5,5)?

By Theorem (3.3.15) we have

$n_2(6,6,6,5,5) = \min \{ n_2^{ex}(\underline{s}) \mid (6,6,6,5,5) \leq \underline{s} \leq (6,6,6,6,6) \}$.

By (3.3.10), $n_2^{ex}(6,6,6,5,5) \geq 1 + n_2(5,5,5,5,5) = 14$.

By (3.3.10), $n_2^{ex}(6,6,6,6,5) \geq 1 + n_2(5,5,5,5,5) = 14$.

$n_2^{ex}(6,6,6,6,6) = 14$ (cf. Helgert and Stinaff (1973)).

Hence $n_2(6,6,6,5,5) = 14$.

The separation vectors of all optimal binary linear UEP codes of length less than or equal to 15 are listed in Table A.1 of Appendix A.

3.4 Notes

Katsman (1980) has shown Corollary (3.3.14) for q = 2. In many cases a combination of Corollary (3.3.15) and the bounds on $n_q^{ex}(.)$ give better results than Corollary (3.3.14). For instance, compare the results of Corollary (3.3.14) for $n_2(5,4,3,3,3,3)$ and $n_2(6,6,6,5,5)$ with those obtained in Example (3.3.16):

(3.3.14): $n_2(5,4,3,3,3,3) \geq 5 + n_2(2,2,2,2,2) = 11,$
$n_2(6,6,6,5,5) \geq 6 + n_2(3,3,3,3) = 13.$

(3.3.15): $n_2(5,4,3,3,3,3) \geq 12,$
$n_2(6,6,6,5,5) \geq 14.$

Another interesting fact is to observe that Theorem (3.3.12) gives better results than the bound of Katsman (cf. Katsman (1980)), i.e. Theorem (3.3.12) for i = 1 and q = 2. For example, Theorem (3.3.12) gives

$$n_2^{ex}(6,6,3,3,3,3,3) \geq 6 + n_2(3,2,2,2,2,2) = 14 \text{ for } i = 1$$

and

$$n_2^{ex}(6,6,3,3,3,3,3) \geq 3 + n_2(5,5,2,2,2,2) = 15 \text{ for } i = 7.$$

A nonlinear $(n,q^k,\underline{s})$ UEP code also satisfies

$$n \geq \sum_{i=1}^{k} s_i/q^{i-1}. \tag{35}$$

This can be proven by generalizing the proof of the Plotkin bound for nonlinear codes (cf. MacWilliams and Sloane (1978), Ch. 2 Theorem 1).

The Theorems (3.3.2), (3.3.6), the binary version of (3.3.12), the Corollaries (3.3.7), (3.3.8), and formula (35) for linear UEP codes were already reported in van Gils (1981).

4. CONSTRUCTIONS OF LINEAR UNEQUAL ERROR PROTECTION CODES

In this chapter we give some constructions of LUEP codes. In Section 4.1 we construct certain families of (length-) optimal LUEP codes and in Section 4.2 we describe methods for combining codes to obtain LUEP codes of larger length.

4.1 Certain families of codes

By trying to construct LUEP codes with the parameters given in Table A.1 (cf. Appendix A), that are binary optimal LUEP codes of small length, the following classes of binary codes came up (the empty entries should be read as zeros).

(4.1.1) Construction: For $k \in \mathbb{N}$,

$$\begin{bmatrix} \overbrace{1 1 \ldots . 1}^{k+3} 1 1 1 & & \\ & 1 1 1 1 1 1 & \\ 1 & 1 \quad 1 \quad 1 & \end{bmatrix} \tag{36}$$

is a generator matrix of an optimal binary $[k+10,3,(k+6,6,4)]$ code.

Proof: It is easy to check that the code has separation vector $(k+6,6,4)$. Furthermore by formula (32) and Table A.1 we have $n_2(k+6,6,4) \geq k+6+n_2(3,2) = k+10$ and $n_2(\underline{s}) > k+10$ for $\underline{s} \geq (k+6,6,4)$, $\underline{s} \neq (k+6,6,4)$ (by $\underline{s} \geq \underline{t}$ we mean, as before, $s_i \geq t_i$ for all i). □

(4.1.2) Construction: For $k \in \mathbb{N}$,

$$\begin{bmatrix} \overbrace{1 1 \ldots . 1}^{k} 1 1 1 1 1 1 1 1 1 & & \\ & 1 1 1 1 \quad 1 1 1 1 & \\ & 1 \quad 1 \quad 1 \quad 1 & \end{bmatrix} \tag{37}$$

is a generator matrix of an optimal binary $[k+13,3,(k+8,8,4)]$ code.

Proof: It is easy to check that the code has separation vector (k+8,8,4). Furthermore by formula (32) and Table A.1 we have $n_2(k+8,8,4) \geq k+8+n_2(4,2)$ $= k+13$ and $n_2(\underline{s}) > k+13$ for $\underline{s} \geq (k+8,8,4)$, $\underline{s} \neq (k+8,8,4)$.

(4.1.3) Construction: For $k \in \mathbb{N}$,

$$\begin{bmatrix} \overset{k}{\overbrace{11\ldots1}}1111111 & 11 & \\ & 1111 & 11 & 11 \\ & 11 & 1111 & 11 \end{bmatrix} \tag{38}$$

is a generator matrix of an optimal binary $[k+14,3,(k+8,8,8)]$ code.

Proof: It is easy to check the parameters of the code. Furthermore by formula (32) and Table A.1 we have $n_2(k+8,8,8) \geq k+8+n_2(4,4) = k+14$ and $n_2(\underline{s}) > k+14$ for $\underline{s} \geq (k+8,8,8)$, $\underline{s} \neq (k+8,8,8)$.

(4.1.4) Construction: For $n,k \in \mathbb{N}$, $n \geq k+1$, the k by n matrix

$$\left[\begin{array}{c|c} I_k & \begin{array}{c} 111\ldots1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{array} \end{array}\right] \tag{39}$$

is a generator matrix of an optimal binary $[n,k,(n-k+1,2,2,\ldots,2)]$ code.

Proof: It is easy to check that the parameters of the code are correct. Furthermore by formula (32) we have that the length of a k-dimensional binary code with separation vector $(n-k+1,2,2,\ldots,2)$ is at least n, and with a separation vector larger than $(n-k+1,2,2,\ldots,2)$ is at least $n+1$ (by $\underline{s} > \underline{t}$ we mean $\underline{s} \geq \underline{t}$ and $\underline{s} \neq \underline{t}$).

(4.1.5) Construction: For $n,k \in \mathbb{N}$, $n \geq 2k+1$, the k by n matrix

$$\left[\begin{array}{c|c|c|c} & 111\ldots1 & 1111\ldots\ldots1 & 1 \\ \hline I_{k-1} & I_{k-1} & & \begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \end{array} \end{array}\right] \tag{40}$$

is a generator matrix of an binary optimal $[n,k,(n-k,4,4,\ldots,4)]$ code.

Proof: It is easy to check that the parameters of the code are correct. By formula (32) we have that the length of a k-dimensional binary code with separation vector (n-k,4,4,....,4) is at least n, and with a separation vector larger than (n-k,4,4,....,4) is at least n+1.

□

(4.1.6) Construction: for $k \in \mathbb{N}$,

$$\left[\begin{array}{c|c|c} & 11.....1 & \\ \hline I_{k-1} & I_{k-1} & \begin{matrix}1\\ \vdots\\ 1\end{matrix} \end{array}\right] \qquad (41)$$

is a generator matrix of an optimal binary $[2k-1,k,(k-1,3,3,....,3)]$ code.

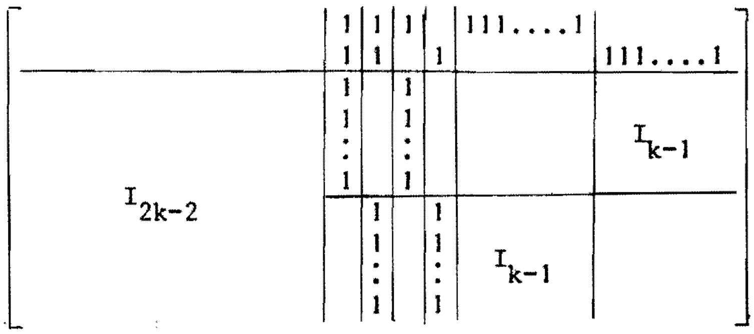
Proof: It is easy to verify that the parameters of the code are correct. By formula (32) we have that the length of a k-dimensional binary code with separation vector (k-1,3,3,....,3) is at least 2k-1.
Applying formula (32) for a k-vector $\underline{s}$ such that $s_1 \geq k$ and $s_i \geq 3$ for $i = 2,...,k$ shows that $n_2(\underline{s}) \geq 2k$.
Applying Theorem (3.3.12) and formula (32) for a k-vector $\underline{s}$ such that $s_1 = k-1$, $s_2 \geq 4$, $s_i \geq 3$ for $i = 3,...,k-1$, and $s_k = 3$ shows that

$$\begin{aligned} n_2^{ex}(\underline{s}) &\geq 3 + n_2(s_1-1,....,s_{k-1}-1) \\ &\geq 3 + s_1-1 + n_2(\lceil (s_1-1)/2 \rceil,...., \lceil (s_{k-1}-1)/2 \rceil) \\ &\geq 3 + k-2 + n_2(\underbrace{2,1,1,....,1}_{k-2}) \\ &\geq 3 + k-2 + k-1 = 2k. \end{aligned}$$

Furthermore it is easy to check that a binary $[2k-1,k,(k-1,4,4,....,4)]$ code does not exist.
Finally, the length of a k-dimensional binary code with a separation vector of at least (k-1,5,4,....,4) is at least 2k.
These observations show that the code in (4.1.6) is optimal.

□

(4.1.7) <u>Construction</u>: For $k \in \mathbb{N}$,

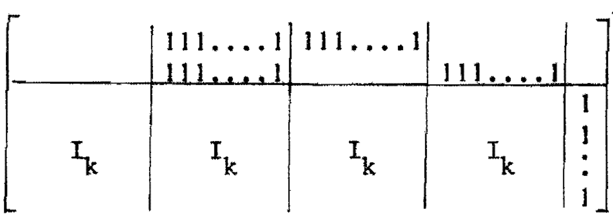


(42)

is a generator matrix of a binary $[4k,2k,(k+2,k+2,4,4,\ldots.,4)]$ code.

For $k = 2,3$ the codes in (4.1.7) are optimal (cf. Table A.1), but in general they are not.

(4.1.8) <u>Construction</u>: For $k \in \mathbb{N}, k \geq 3$,



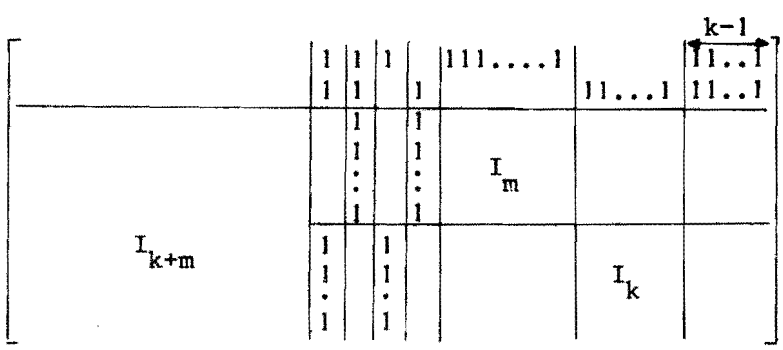
(43)

is a generator matrix of a length-optimal binary $[4k+1,k+2,(2k,2k,5,5,\ldots.,5)]$ code.

<u>Proof</u>: It is easy to check that the code has the given parameters. By formula (33) the length of a code with dimension $k+2$ and separation vector $(2k,2k,5,5,\ldots.,5)$ is at least $4k+1$.

□

(4.1.9) <u>Construction</u>: For $k,m \in \mathbb{N}$,



(44)

is a generator matrix of an optimal binary $[3k+2m+3,k+m+2,(k+m+2,2k+2,4,4,..,4)]$ code.

Proof: It is easy to verify that the code has the given parameters. Furthermore by formula (33) we have that the length of a $(k+m+2)$-dimensional binary code with separation vector $(k+m+2,2k+2,4,4,..,4)$ is at least $3k+2m+3$, and with a separation vector larger than $(k+m+2,2k+2,4,4,..,4)$ is at least $3k+2m+4$.

□

4.2 Combining codes

In this section we consider constructions which combine (LUEP) codes to obtain LUEP codes of larger length, such as the direct sum and direct product construction, the $|u|u+v|$ construction, and concatenation.

(4.2.1) Construction: For $k,m,n \in \mathbb{N}$, and a nonincreasing k-vector $\underline{s} \in \mathbb{N}^k$ such that $s_1 \leq n/2$ let G_1 be an optimal generator matrix of a binary $[n,k,\underline{s}]$ code C_1, and for $i = 0,1,\ldots,2^m-1$ let A_i be an m by n matrix whose columns are all equal to the binary representation of i, i.e. $\sum_{u=1}^{m} (A_i)_{uv} 2^{u-1} = i$ for all $v = 1,\ldots,n$. Then the $(m+k)$ by $n2^m$ matrix

$$G_2 := \left[\begin{array}{c|c|c|c|c} A_0 & A_1 & A_2 & \cdots\cdots & A_{2^m-1} \\ \hline G_1 & G_1 & G_1 & & G_1 \end{array}\right] \qquad (45)$$

is a generator matrix of a binary $[n2^m,m+k,(n2^{m-1}\underline{1}|2^m\underline{s})]$ code C_2. If C_1 is optimal, so is C_2.

Proof: It is easy to check that the parameters of the code C_2 are correct. Suppose that C_1 is optimal. Then by formula (32) the length of a $(k+m)$-dimensional binary code with separation vector $(n2^{m-1}|2^m\underline{s})$ is at least $n(2^m-1)+n_2(s_1,\ldots,s_k) = n2^m$ and with a separation vector larger than $(n2^{m-1}|2^m\underline{s})$ is at least $n2^m+1$. □

(4.2.2) Examples:

(i): If in (4.2.1) we take $m = 1$ and for G_1 a generator matrix of a binary $[2^t-1,2^t-t-1,(3,3,\ldots,3)]$ Hamming code, then G_2 is a generator matrix

of an optimal $[2^{t+1}-2, 2^t-t, (2^t-1,6,6,\ldots,6)]$ binary code.

(ii): If in (4.2.1) we take $m = 1$ and for G_1 an optimal generator matrix of an optimal binary $[7,5,(3,2,2,2,2)]$ code, then we obtain an optimal $[14,6,(7,6,4,4,4,4)]$ code.

(4.2.3) The direct sum construction: If for $i = 1,2$, C_i is an $[n_i,k_i,\underline{s}^{(i)}]$ linear code, then the direct sum $\{ (\underline{c}_1|\underline{c}_2) \mid \underline{c}_1 \in C_1, \underline{c}_2 \in C_2 \}$ is an $[n_1+n_2, k_1+k_2, (\underline{s}^{(1)}|\underline{s}^{(2)})]$ linear code.

(4.2.4) The $|u|u+v|$ construction: If for $i = 1,2$, C_i is an $[n,k_i,\underline{s}^{(i)}]$ linear code with an optimal generator matris G_i, then

$$G := \left[\begin{array}{c|c} G_1 & G_1 \\ \hline & G_2 \end{array}\right] \tag{46}$$

is a generator matrix of a $[2n,k_1+k_2,\underline{s}]$ code C, where

$$\begin{aligned} &s_i \geq \min \{ 2s_i^{(1)}, \max \{ s_i^{(1)}, s_{k_2}^{(2)} \} \} \text{ for } i = 1,\ldots,k_1, \\ &\text{and} \\ &s_{k_1+i} \geq s_i^{(2)} \qquad \text{for } i = 1,\ldots,k_2 \end{aligned} \tag{47}$$

($\underline{s}$ is not necessarily nonincreasing in this case).

Proof: For $\underline{m}^{(1)} \in \mathbb{F}_q^{k_1}$, $\underline{m}^{(2)} \in \mathbb{F}_q^{k_2}$, $\underline{m} = (\underline{m}^{(1)}|\underline{m}^{(2)})$ we have for $i = 1,\ldots,k_1$ that

$$\begin{aligned} &wt(\underline{m}G) \geq 2s_i^{(1)} && \text{if } m_i^{(1)} \neq 0, \ \underline{m}^{(2)} = \underline{0}, \\ &wt(\underline{m}G) \geq \max \{ s_i^{(1)}, s_{k_2}^{(2)} \} && \text{if } m_i^{(1)} \neq 0, \ \underline{m}^{(2)} \neq \underline{0}, \end{aligned}$$

and for $j = 1,\ldots,k_2$ that

$$\begin{aligned} &wt(\underline{m}G) \geq s_j^{(2)} && \text{if } \underline{m}^{(1)} = \underline{0}, \ m_j^{(2)} \neq 0, \\ &wt(\underline{m}G) \geq \max \{ s_j^{(2)}, s_{k_1}^{(1)} \} && \text{if } \underline{m}^{(1)} \neq \underline{0}, \ m_j^{(2)} \neq 0. \end{aligned}$$

This proves formula (47).

□

(4.2.5) Example: If in (4.2.5) C_1 is the binary [13,12] even-weight code and C_2 is a binary [13,6,(5,5,5,5,4,4)] code, then $\underline{s}(G)_i \geq 4$ for $i = 1,\ldots,12$ and $i = 17,18$, and $\underline{s}(G)_i \geq 5$ for $i = 13,14,15,16$. Now C is a [26,18,(5,5,5,5,4,...,4,4)] code. The length of an 18-dimensional binary code with a separation vector of at least (5,5,5,5,4,4,....,4) is at least 25 (by Corollary (3.3.14)).

(4.2.6) Construction: If for $i = 1,2$, X_i is a $k_i \times n_1$-matrix over $\mathbb{F}_q$ and $(\underline{u}|\underline{v})$, where $\underline{u}$ is a k_1-vector, is the separation vector of the matrix $(X_1^T|X_2^T)^T$, and Y is a $k_1 \times n_2$-matrix over $\mathbb{F}_q$ with separation vector $\underline{w}$, then

$$\left[\begin{array}{c|c} X_1 & Y \\ \hline X_2 & \end{array}\right] \tag{48}$$

is a generator matrix of an $[n_1+n_2, k_1+k_2, (\underline{s}^{(1)}|\underline{s}^{(2)})]$ code, where

$$\underline{s}^{(1)} \geq \underline{u} + \underline{w} \quad \text{and} \quad \underline{s}^{(2)} \geq \underline{v}. \tag{49}$$

Proof: Trivial. □

(4.2.7) Example: If G_1 is a 5 x 12 binary matrix such that $\underline{s}(G_1) = (5,5,5,5,4)$ and G_2 is a 2 x 3 binary matrix such that $\underline{s}(G_2) = (2,2)$, then

$$\left[\begin{array}{c|c} G_1 & \begin{array}{c} G_2 \\ \hline 0 \end{array} \end{array}\right]$$

is a generator matrix of a length-optimal (cf. Table A.1) binary [15,5,(7,7,5,5,4)] code.

(4.2.8) The direct product construction: By taking the direct product (cf. MacWilliams and Sloane (1978), Ch. 18 Section 2) of an $[n_1,k_1,\underline{s}^{(1)}]$ code and an $[n_2,k_2,\underline{s}^{(2)}]$ code, both over the same field, we obtain an $[n_1n_2,k_1k_2,\underline{s}^{(1)} \otimes_{\mathbb{R}} \underline{s}^{(2)}]$ code, where $\otimes_{\mathbb{R}}$ denotes the Kronecker product over $\mathbb{R}$. This is shown by the following two theorems.

(4.2.9) Theorem: For the matrices A and B over a common finite field $\mathbb{F}$ the separation vector of the Kronecker product of A and B, $A \otimes_{\mathbb{F}} B$, equals the Kronecker product of the separation vectors of A and B, i.e.

$$\underline{s}(A \otimes_{\mathbb{F}} B) = \underline{s}(A) \otimes_{\mathbb{R}} \underline{s}(B). \tag{50}$$

Proof: Let A be a k_1 by n_1 matrix and let B be a k_2 by n_2 matrix. Let $(i,j) \in \{1,\ldots,k_1\} \times \{1,\ldots k_2\}$. For any k_1 by k_2 matrix M over $\mathbb{F}$ such that $M_{ij} \neq 0$ we have $wt((MB)_{i*}) \geq s(B)_j$, and hence

$$\sum_{u=1}^{n_1} wt((A^T MB)_{u*}) \geq s(A)_i s(B)_j.$$

For $\underline{m}_1 \in \mathbb{F}^{k_1}$, $\underline{m}_2 \in \mathbb{F}^{k_2}$ such that $m_{1i} \neq 0$, $m_{2j} \neq 0$ and $wt(\underline{m}_1 A) = s(A)_i$, $wt(\underline{m}_2 B) = s(B)_j$ we have that $(\underline{m}_1^T \underline{m}_2)_{ij} \neq 0$ and

$$\sum_{u=1}^{n_1} wt((A^T \underline{m}_1^T \underline{m}_2 B)_{u*}) = s(A)_i s(B)_j.$$

From these observations it follows that $\underline{s}(A \otimes_{\mathbb{F}} B) = \underline{s}(A) \otimes_{\mathbb{R}} \underline{s}(B)$. □

(4.2.10) Theorem: For generator matrices A and B over a common finite field $\mathbb{F}$, $A \otimes_{\mathbb{F}} B$ is an optimal generator matrix for its rowspace if and only if A and B are both optimal generator matrices for their rowspaces.

Proof: Suppose A and B are optimal generator matrices for their rowspaces. Let $\hat{A}$ and $\hat{B}$ be minimal weight generator matrices for the rowspaces of A and B. Hence $\underline{s}(A) = \underline{s}(\hat{A}) = (wt(\hat{A}_{1*}),\ldots,wt(\hat{A}_{k_1*}))$ and $\underline{s}(B) = \underline{s}(\hat{B}) = \underline{s}(B) = \underline{s}(B) = (wt(\hat{B}_{1*}),\ldots,wt(\hat{B}_{k_2*}))$. Furthermore we have that

$$\sum_{i=1}^{k_1 k_2} wt((\hat{A} \otimes_{\mathbb{F}} \hat{B})_{i*}) = (\sum_{i=1}^{k_1} wt(\hat{A}_{i*}))(\sum_{i=1}^{k_2} wt(\hat{B}_{i*})).$$

From this it follows that $\hat{A} \otimes_{\mathbb{F}} \hat{B}$ is a minimal weight generator matrix for its rowspace and so by Theorem (2.2.4) it is optimal. Since $\underline{s}(A \otimes_{\mathbb{F}} B) = \underline{s}(\hat{A} \otimes_{\mathbb{F}} \hat{B})$, $A \otimes_{\mathbb{F}} B$ is also optimal for its rowspace.
On the other hand suppose that A is not optimal. Then for an optimal generator matrix A' of the rowspace of A we have that $\underline{s}(A') \geq \underline{s}(A)$ and $\underline{s}(A') \neq \underline{s}(A)$. This implies that $\underline{s}(A' \otimes_{\mathbb{F}} B) \geq \underline{s}(A \otimes_{\mathbb{F}} B)$ and $\underline{s}(A' \otimes_{\mathbb{F}} B) \neq \underline{s}(A \otimes_{\mathbb{F}} B)$, i.e. $A \otimes_{\mathbb{F}} B$ is not optimal for its rowspace. □

(4.2.11) Concatenation: Let C be an $[N,K,\underline{S}=(S_1,\ldots,S_K)]$ linear code over $GF(q^k)$ with an optimal generator matrix G_C and let D be an $[n,k,d]$ linear code over GF(q) with generator matrix G_D.
The encoding procedure of the concatenation of these codes is as follows.

Let $\underline{m} = (m_1^{(1)},\ldots,m_k^{(1)}|\ldots\ldots\ldots|m_1^{(K)},\ldots,m_k^{(K)})$ be a Kk-tuple over GF(q). This Kk-tuple is equivalent with a K-tuple $\underline{M} = (M^{(1)},\ldots\ldots,M^{(K)})$ over $GF(q^k)$, which is encoded into $(A^{(1)},\ldots\ldots\ldots,A^{(N)}) := (M^{(1)},\ldots\ldots,M^{(K)})G_C$. Now we regard $A^{(i)}$ as a k-tuple $(a_1^{(i)},\ldots,a_k^{(i)})$ over GF(q) and encode it into $(c_1^{(i)},\ldots\ldots,c_n^{(i)}) := (a_1^{(i)},\ldots,a_k^{(i)})G_D$ $(i = 1,\ldots,N)$.
If $\underline{m}$ is a q-ary Kk-tuple such that $m_i^{(j)} \neq 0$, then $M_j \neq 0$ and hence $\underline{A} := \underline{M}G_C$ satisfies $wt(\underline{A}) \geq S_j$, which in turn implies that $wt(\underline{c}) \geq S_j d$. Hence we have shown the following theorem.

(4.2.12) Theorem: The concatenation of an $[N,K,\underline{S}=(S_1,\ldots,S_K)]$ outer code over $GF(q^k)$ and an [n,k,d] inner code over GF(q) is an $[Nn,Kk,\underline{s}]$ linear code over GF(q), where

$$s_{(j-1)k+i} \geq dS_j \qquad (51)$$

for $i = 1,\ldots,k$ and $j = 1,\ldots,K$.

(4.2.13) Examples: Let α be a primitive element of GF(4) and D be the binary [3,2,2] even-weight code.

(i): For the optimal [7,3,(5,4,4)] code C over GF(4) with generator matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 0 & 1 & \alpha^2 & \alpha \end{bmatrix}$$

the concatenated code of C and D is a [21,6,(10,10,8,8,8,8)] binary code. The maximal minimum distance of a binary [21,6] code equals 8 (cf. Helgert and Stinaff (1973)).

(ii): For the optimal [8,4,(5,4,4,4)] code C over GF(4) with generator matrix

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & \alpha & \alpha^2 & 0 & 0 \\ 0 & 1 & 0 & 1 & \alpha & 0 & \alpha^2 & 0 \\ 0 & 0 & 1 & 1 & \alpha & 0 & 0 & \alpha^2 \end{bmatrix}$$

the concatenated code of C and D is a [24,8,(10,10,8,8,8,8,8,8)] binary code.

(4.2.14) Theorem: For $k,K,n,N,d \in \mathbb{N}$, $N \geq 2^k + 1$ the concatenation of the $[N,2,(n-1,2^k)]$ outer code over $GF(2^k)$ with generator matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & . & . & 1 & 1 & . & 1 & 0 \\ 1 & \alpha & \alpha^2 & \alpha^3 & . & . & \alpha^{2^k-2} & 0 & . & 0 & 1 \end{bmatrix}, \tag{52}$$

where α is a primitive element of $GF(2^k)$, and an $[n,k,d]$ inner code over $GF(2)$ gives an $[Nn,2k,\underline{s}]$ binary code, where

$$\begin{aligned} &s_i \geq (N-1)d \quad \text{for } i = 1,\ldots,k, \\ &\text{and} \\ &s_i \geq 2^k d \quad \text{for } i = k+1,\ldots,2k. \end{aligned} \tag{53}$$

If $n = (2^k-1)d/2^{k-1}$ (i.e. equality in the Plotkin Bound), then equality holds in (53) and the concatenated code is optimal.

Proof: We find formula (53) by applying Theorem (4.2.12). By Corollary (3.3.14) a 2k-dimensional binary code with a separation vector of at least $\underline{s}$, where $s_i = (N-1)d$ for $i = 1,\ldots,k$ and $s_i = 2^k d$ for $i = k+1,\ldots,2k$, has a length of at least

$$\sum_{i=0}^{k-1} \left\lceil (N-1)d/2^i \right\rceil + \sum_{i=0}^{k-1} \left\lceil d/2^i \right\rceil \geq Nd(2^k-1)/2^{k-1}. \tag{54}$$

If $n = (2^k-1)d/2^{k-1}$ then equality must hold in formula (54) and hence in (53). This also shows that the concatenated code is optimal.

(4.2.15) Example: Take the $[11,2,(10,8)]$ code over $GF(8)$ as the outer code and the $[7,3,4]$ simplex code over $GF(2)$ as the inner code to obtain an optimal $[77,6,(40,40,40,32,32,32)]$ binary code. In general all concatenated codes constructed in Theorem (4.2.14) with the inner code being a simplex code are optimal LUEP codes, because simplex codes satisfy the Plotkin Bound with equality.

4.3 Notes

The Theorems (4.2.9) and (4.2.10) are from Dunning and Robbins (1978). Other methods for combining codes can be found in Zinov'ev and Zyablov (1979), and Boyarinov and Katsman (1981).

5. CYCLIC UNEQUAL ERROR PROTECTION CODES

In this chapter we consider cyclic UEP codes and try to find the separation vector of these codes. In Section 5.1 we give an optimal generator matrix for a cyclic UEP code and we show how the separation vector can be determined from the weight distributions of the cyclic subcodes. Section 5.2 shows that certain classes of cyclic UEP codes can easily be decoded by using Majority Logic Decoding Methods.

5.1 The separation vector of a cyclic UEP code

A cyclic $[n,k]$ code over $\mathbb{F}_q$ is the direct sum of the minimal ideals in $\mathbb{F}_q[x]/(x^n-1)$ contained in it (cf. MacWilliams and Sloane (1978), Ch. 7 and 8).

(5.1.1) Theorem: For a cyclic code C which is the direct sum of the minimal ideals with generator matrices resp. $M_1, M_2, \ldots, M_v$,

$$G := \begin{bmatrix} M_1 \\ \hline M_2 \\ \hline \vdots \\ \hline M_v \end{bmatrix} \tag{55}$$

is an optimal generator matrix.

Proof: For $\rho \in WT(C)$, $\langle C(\rho) \rangle$ is a cyclic code. Hence $\langle C(\rho) \rangle$ is the sum of a number of minimal ideals of $\mathbb{F}_q[x]/(x^n-1)$. By applying Theorem (2.1.3) we get the theorem.

□

The following corollaries are immediate consequences of Lemma (2.1.2) and the above proof.

(5.1.2) Corollary: For a minimal ideal in $\mathbb{F}_q[x]/(x^n-1)$ all components of the separation vector are mutually equal.

(5.1.3) Corollary: For a cyclic code C with an optimal generator matrix G defined by formula (55) the i^{th} and j^{th} component of the separation vector $\underline{s} = \underline{s}(G)$ are equal if the i^{th} and j^{th} row of G are in the same minimal ideal of $\mathbf{F}_q[x]/(x^n-1)$.

If the generator polynomial of a cyclic code C has minimal weight, i.e. its weight equals the minimum distance d of the code, then all components of the separation vector are mutually equal, because C = <C(d)> (cf. Theorem (2.1.3)). If this is not the case, we can compute the separation vector of a cyclic code by comparing the weight distributions of its cyclic subcodes.

(5.1.4) Theorem: For i = 1,2 let M_i be a minimal ideal in $\mathbf{F}_q[x]/(x^n-1)$ with minimum distance d_i and weight distribution $(A_j^{(i)})_{j=0}^n$ such that $M_1 \neq M_2$ and $d_1 \geq d_2$; let $(A_j)_{j=0}^n$ be the weight distribution of their direct sum $M_1 \oplus M_2$. Then the components of the separation vector of $M_1 \oplus M_2$ are all equal to the minimum distance d of $M_1 \oplus M_2$ if $d < d_2$ or if $d = d_2$ and $A_d^{(2)} < A_d$; they take two different values if $d = d_2$ and $A_d^{(2)} = A_d$, namely d_2 and $\min \{ j \mid A_j^{(2)} < A_j \}$.

Proof: If $d < d_2$ or if $d = d_2$ and $A_d^{(2)} < A_d$ then a sum of an element in $M_1\backslash\{\underline{0}\}$ and one in $M_2\backslash\{\underline{0}\}$ exists such that its weight equals d. For $d = d_2$ and $A_d^{(2)} = A_d$, if $A_j^{(2)} < A_j$ then a sum of an element in $M_1\backslash\{\underline{0}\}$ and one in $M_2\backslash\{\underline{0}\}$ exists such that its weight equals j; if $A_j^{(2)} = A_j$ it does not. Combining these observations with Theorem (5.1.1) and Corollary (5.1.3) proves the theorem. □

(5.1.5) Examples:

(i): Let $\alpha \in GF(2^{10})$ be a primitive 33^{rd} root of unity and let C be the binary cylic [33,23] code with nonzeros $\{ \alpha^i \mid i \in C_{11} \cup C_1 \cup C_0 \cup C_3 \}$, where C_i denotes the cyclotomic coset modulo 33 over GF(2) containing i. Let M_i denote the minimal ideal in $\mathbf{F}_2[x]/(x^{33}-1)$ having nonzeros $\{ \alpha^j \mid j \in C_i \}$. Then $C = M_{11} \oplus M_1 \oplus M_0 \oplus M_3$ and $G := [M_{11}^T | M_1^T | M_0^T | M_3^T]^T$ is an optimal generator matrix of C, where M_i denotes a generator matrix of M_i (i = 0,1,3,11). $\underline{s} = \underline{s}(G)$.
Table 5.1 provides the minimum distances of all cyclic subcodes of C (taken from Peterson and Weldon (1972), Appendix D).

nonzeros 0 1 3 11	min. dist.	nonzeros 0 1 3 11	min. dist.
x	33	x x	10
x	12	x x	6
x	6	x x x	3
x	22	x x x	10
x x	11	x x x	3
x x	3	x x x	6
x x	11	x x x x	3
x x	6		

Table 5.1: The minimum distances of the cyclic subcodes of C.

The code C has minimum distance 3 and C and $M_0 \oplus M_3$ both contain 11 codewords of weight 3. Combining this with Theorem (5.1.4) and Table 5.1 we find that $s_{13} = s_{14} = \ldots = s_{23} = 3$ and $s_1, s_2, \ldots, s_{12} > 3$. C contains no codewords of weight 4 and 165 codewords of weight 5. From Table 5.1 we see that the codewords of weight 5 in C can only occur in the cyclic subcodes (1): $M_0 \oplus M_3$, (2): $M_0 \oplus M_1 \oplus M_3$, (3): $M_0 \oplus M_3 \oplus M_{11}$, and (4): $M_0 \oplus M_1 \oplus M_3 \oplus M_{11}$. However, (1),(2), and (3) contain no codewords of weight 5, as one can easily check. Hence a codeword $\underline{c}$ in C of weight 5 is the sum $\underline{c} = \underline{c}_0 + \underline{c}_1 + \underline{c}_3 + \underline{c}_{11}$ of nonzero elements $\underline{c}_i \in M_i$ $(i = 0,1,3,11)$. This shows that $s_1 = s_2 = \ldots = s_{12} = 5$ (by Corollary (5.1.3) and Theorem (5.1.4)).

So the code C provides a protection level 2 to twelve message positions and a protection level 1 to the remaining eleven positions.

(ii): Let $\alpha \in GF(2^{12})$ be a primitive 35^{th} root of unity and let C be the binary cyclic [35,22] code with nonzeros $\{ \alpha^i \mid i \in C_5 \cup C_7 \cup C_1 \cup C_{15} \}$. Then $C = M_5 \oplus M_7 \oplus M_1 \oplus M_{15}$ and $G := [M_5^T | M_7^T | M_1^T | M_{15}^T]^T$ is an optimal generator matrix of C. $\underline{s} = \underline{s}(G)$. The minimum distances of the cyclic subcodes of C are listed in Table 5.2 (cf. Peterson and Weldon (1972), Appendix D). The cyclic subcodes M_1, M_{15} and $M_1 \oplus M_{15}$ have minimum distances resp. 8,20, and 4. Hence by Theorem (5.1.4) we have that $s_8 = s_9 = \ldots = s_{22} = 4$. The number of codewords of weight 4 in C resp. $M_1 \oplus M_{15}$ both equal 35 and all weights in C are even, hence $s_1, s_2, \ldots, s_7 \geq 6$. The minimum distance of $M_1 \oplus M_7$ equals 6, hence $s_4 = s_5 = s_6 = s_7 = 6$.

nonzeros				min.	nonzeros				min.
1	5	7	15	dist.	1	5	7	15	dist.
x				8		x		x	10
	x			20			x	x	14
		x		14	x	x	x		6
			x	20	x	x		x	4
x	x			8	x		x	x	4
x		x		6		x	x	x	10
x			x	4	x	x	x	x	4
	x	x		14					

Table 5.2: The minimum distances of the cyclic subcodes of C.

The number of codewords of weight 6 in $M_1 \oplus M_7 \oplus M_{15}$ equals 490, while C contains 595 codewords of weight 6, hence $s_1 = s_2 = s_3 = 6$. So seven components of the separation vector of C equal 6, fifteen of them equal 4.

5.2 Majority Logic Decoding of cyclic UEP codes

In this section we discuss certain classes of cyclic UEP codes which can be decoded by Majority Logic Decoding. It is easy to implement this method and it is very useful whenever the number of orthogonal checks on a message digit equals (or is not much less than) the separation component corresponding to that message position. We restrict ourselves to binary codes.

Fix $n \in \mathbb{N}$, n odd, and let $T_0, T \in \mathbb{N}$ be such that $T_0 < T < n$ and $T_0 | T | n$. Let $p_i(x)$, $i = 0,1,\ldots,w$ be irreducible polynomials in $\mathbb{F}_2[x]$ such that $p_0(x)$ has exponent T_0 (i.e. the minimal j such that $p_0(x) | x^j+1$) and

$$(x^n+1) = \prod_{i=0}^{w} p_i(x).$$

Let $v \leq w$ be such that none of the polynomials $p_i(x)$, $i = 1,2,\ldots,v-1$ has an exponent which divides T and let C be the code with check polynomial

$$h(x) = \prod_{i=0}^{v-1} p_i(x).$$

Then by Theorem (5.1.1) $G := (M_0^T|M_1^T|.....|M_{v-1}^T)^T$, where for $i = 0,1,...,v-1$ the rows of M_i are the cyclic shifts of $(x^n+1)/p_i(x)$, is an optimal generator matrix for C.

Let k_i denote the degree of $p_i(x)$ for $i = 0,1,...,v-1$. The $[n,k_0]$ cyclic subcode with check polynomial $p_0(x)$ consists of n/T_0 repetitions of the $[T_0,k_0]$ code with generator polynomial $(x^{T_0}+1)/p_0(x)$. Let this $[T_0,k_0]$ code have an orthogonal check set of size δ on any code position (cf. Cameron and van Lint (1980), Ch. 11). For $k_0 = 1$ one has that $T_0 = 1$ and $\delta = 0$. Then the following theorem provides a lower bound on the k_0 separation components corresponding to M_0 in terms of T,T_0, and δ.

(5.2.1) Theorem: The first k_0 components of the separation vector of the code defined above are larger than or equal to $T(\delta+1)/T_0$.

Proof: With a message $\underline{m} = (m_0,m_1,...,m_{k-1}) \in \mathbb{F}_2^k$ resp. a codeword $\underline{c} = (c_0,c_1,.....,c_{n-1}) \in \mathbb{F}_2^n$ we associate the polynomials $m(x) := \sum_{i=0}^{k-1} m_i x^i$ resp. $c(x) := \sum_{i=0}^{n-1} c_i x^i$. By $C_{x^i}\{q(x)\}$ we denote the coefficient of $x^{i \bmod n}$ in the polynomial $q(x) \bmod (x^n+1)$.

Note the following:

(i): For $j \in \{0,1,...,T-1\}$ and $q(x) \in \mathbb{F}_2[x]/(x^n+1)$ we have that

$$\sum_{i=0}^{(n/T)-1} C_{x^{j+iT}}\{q(x)\} = C_{x^j}\{(x^n+1)q(x)/(x^T+1)\}. \tag{56}$$

(ii): For $i = 1,2,...,v-1$ and $u,j \in \mathbb{N}$ we have that

$$C_{x^{j+uT_0}}\{((x^n+1)/(x^T+1))((x^n+1)/p_i(x))\} = 0, \tag{57}$$

since $p_i(x)(x^T+1) \mid (x^n+1)$.

(iii): For $\lambda(x) = \sum_{i=0}^{T_0-1} \lambda_i x^i := (x^{T_0}+1)/p_0(x)$, $j \in \{0,1,...T_0-1\}$ and $u \in \{0,1,...,(T/T_0)-1\}$ we have that

$$C_{x^{j+uT_0}}\{((x^n+1)/(x^T+1))((x^n+1)/p_0(x))\} = \lambda_j, \tag{58}$$

since n/T is odd.

Combining (i),(ii), and (iii) with the fact that $\lambda_{T_0-k_0+1} = ... = \lambda_{T_0-1} = 0$ we get for $j = 0,1,...,T_0-1$ and $u = 0,1,...,(T/T_0)-1$ that

$$\sum_{i=0}^{(n/T)-1} c_{iT+uT_0+j}$$

$$= \sum_{i=0}^{(n/T)-1} C_{x^{iT+uT_0+j}} \{\sum_{s=0}^{v-1} \sum_{t=0}^{k_s-1} m_{k_{s-1}+t}\, x^t (x^n+1)/p_s(x)\}$$

$$= \sum_{s=0}^{v-1} \sum_{t=0}^{k_s-1} m_{k_{s-1}+t} \sum_{i=0}^{(n/T)-1} C_{x^{iT+uT_0+j}} \{x^t (x^n+1)/p_s(x)\}$$

$$= \sum_{t=0}^{k_0-1} m_t\, C_{x^{j+uT_0}} \{x^t ((x^n+1)/(x^T+1))((x^n+1)/p_0(x))\}$$

$$= \sum_{t=0}^{k_0-1} m_t\, C_{x^{j-t+uT_0}} \{((x^n+1)/(x^T+1))((x^n+1)/p_0(x))\}$$

$$= \sum_{t=0}^{j} m_t \lambda_{j-t} + \sum_{t=j+1}^{k_0-1} m_t \lambda_{j-t+T_0} = \sum_{t=0}^{j} m_t \lambda_{j-t},$$

where $k_{-1} := 0$.
Hence we have that

$$\sum_{i=0}^{(n/T)-1} c_{iT+uT_0+j} = \sum_{t=0}^{j} \lambda_t m_{j-t} \tag{59}$$

holds for $j = 0,1,\ldots,T_0-1$ and $u = 0,1,\ldots,(T/T_0)-1$.
For $j \in \{0,1,\ldots,T_0-1\}$ the $[T_0,k_0]$ code, Λ, with generator polynomial $\lambda(x)$ has an orthogonal check set of size δ on position j. Unlike the usual definition we define these checks to be subsets $A_1^{(j)}, A_2^{(j)}, \ldots, A_\delta^{(j)}$ of $\{0,1,\ldots,T_0-1\}$ which satisfy the following three conditions.

$$A_r^{(j)} \cap A_s^{(j)} = \emptyset \quad \text{for } r \neq s, \tag{60}$$

$$\bigcup_{r=1}^{\delta} A_r^{(j)} \subset \{0,1,\ldots,T_0-1\}\backslash\{j\}, \tag{61}$$

$$y_j = \sum_{p \in A_r^{(j)}} y_p \quad \text{for all } \underline{y} \in \Lambda \text{ and all } r \in \{1,\ldots,\delta\}. \tag{62}$$

We define the weight of a check $A_r^{(j)}$ as the number of elements in $A_r^{(j)}$.
In an analog way as we have shown formula (59), we can show that the equality

$$\sum_{i=0}^{(n/T)-1} \sum_{p \in A_r^{(j)}} c_{iT+uT_0+p} = \sum_{t=0}^{j} \lambda_t m_{j-t} \qquad (63)$$

holds for $j = 0,1,\ldots,T_0-1$, $u = 0,1,\ldots,(T/T_0)-1$, and $r = 1,\ldots,\delta$, by combining the formulas (56),(57),(59), and (62).
Because $A_1^{(j)},\ldots,A_\delta^{(j)}$ satisfy (60) and (61), formulas (59) and (63) provide $T(\delta+1)/T_0$ orthogonal checks on $\sum_{t=0}^{j} \lambda_t m_{j-t}$.
The k_0 linear functions $\sum_{t=0}^{j} \lambda_t m_{j-t}$, $j = 0,1,\ldots,k_0-1$ are linearly independent, since $\lambda_0 = 1$. Hence the k_0 message positions corresponding to M_0 have a separation component of at least $T(\delta+1)/T_0$.

□

(5.2.2) Example: $n = 63$, $T = 21$, $T_0 = 7$, $p_0(x) = x^3+x+1$.

$$h(x) := \prod_{i=0}^{6} p_i(x),$$

where $p_i(x)$, $i = 1,\ldots,6$ are the six irreducible factors of $x^{63}+1$ with exponents equal to 63. We have that $\lambda(x) = 1+x+x^2+x^4$, which is the generator polynomial of a [7,3] code with three orthogonal checks {1,5}, {2,3}, and {4,6} on code position 0. By Theorem (5.2.1) the [63,39] code with check polynomial h(x) has a separation vector with three components larger than or equal to 12. This code has minimum distance 4. A [63,39] BCH code has minimum distance 9.

Now consider two irreducible polynomials p(x) and q(x) in $\mathbb{F}_2[x]/(x^n+1)$ with degrees resp. k_p and k_q and exponents resp. T_p and T_q such that T_p and T_q are relatively prime. Let C_p and C_q be resp. $[T_p,k_p]$ and $[T_q,k_q]$ codes with check polynomials resp. p(x) and q(x). Furthermore let C_p have an orthogonal check set of size δ_p on any code position such that any check has the same even weight, i.e. any code symbol equals the sum of 2w other ones for some fixed integer w.
Define C to be the binary cyclic $[T_pT_q,k_p+k_q]$ code with check polynomial p(x)q(x) and C^* the binary cyclic $[T_pT_q,k_p+k_q+1]$ code with check polynomial (x+1)p(x)q(x). The following theorem provides a lower bound on the separation vector $\underline{s}$ and $\underline{s}^*$ of resp. C and C^* if their encoding is defined by the generator matrices resp. $G = (M_p^T|M_q^T)^T$ and $G^* = (M_p^T|M_q^T|M_0^T)^T$, where M_p,M_q, and M_0 are generator matrices of the minimal ideals in $\mathbb{F}_2[x]/(x^n+1)$ with check polynomials resp. p(x),q(x), and (x+1).

(5.2.3) Theorem:

(i): The separation vector $\underline{s}$ of the code C defined above satisfies

$$s_i \geq T_q \delta_p + 1 \quad \text{if } C_q \text{ is an even-weight code,}$$
$$s_i \geq T_q \delta_p \quad \text{otherwise,} \tag{64}$$

for $i = 1,\ldots,k_p$.

(ii): The separation vector $\underline{s}^*$ of the code C^* defined above satisfies

$$s_i^* \geq T_q \delta_p \tag{65}$$

for $i = 1,\ldots k_p$.

Proof:

(i): Without loss of generality we consider the first message digit m_0. Let G_p and $G_q := [\underline{y}_1|\underline{y}_2|\ldots|\underline{y}_{T_q}]$ be systematic generator matrices of C_p resp. C_q. Without loss of generality the first column of G_p equals $\underline{e}_0 := (1,0,0,\ldots,0)^T$.
Since C_p has an orthogonal check set of size δ_p on any code position such that any check has the same even weight, say 2w, we have $2w\delta_p$ mutually different columns $\underline{a}_i^{(1)}, \underline{a}_i^{(2)}, \ldots, \underline{a}_i^{(2w)}$, $i = 1,\ldots,\delta_p$ of G_p such that

$$\underline{a}_i^{(1)} + \underline{a}_i^{(2)} + \ldots\ldots + \underline{a}_i^{(2w)} = \underline{e}_0 \tag{66}$$

for $i = 1,\ldots,\delta_p$. The matrix

$$G := \left[\begin{array}{c|c|c|c|c} G_p & G_p & \multicolumn{2}{c|}{\cdots\cdots\cdots} & G_p \\ \hline G_q & G_q & G_q & \cdots\cdots\cdots\cdots & G_q \end{array}\right] \tag{67}$$

is an optimal generator matrix of C. Because $\text{g.c.d.}(T_p,T_q) = 1$, any pair $\begin{pmatrix}\underline{x}\\ \underline{y}\end{pmatrix}$, where $\underline{x}$ and $\underline{y}$ are columns of resp. G_p and G_q, occurs exactly once as a column of G. By combining this fact with formula (66), we get that for any $i \in \{1,\ldots,\delta_p\}$ and any $j \in \{1,\ldots T_q\}$ the equality

$$\begin{bmatrix} \underline{a}_i^{(1)} \\ \underline{y}_j \end{bmatrix} + \begin{bmatrix} \underline{a}_i^{(2)} \\ \underline{y}_j \end{bmatrix} + \cdots\cdots\cdots + \begin{bmatrix} \underline{a}_i^{(2w)} \\ \underline{y}_j \end{bmatrix} = \begin{bmatrix} \underline{e}_0 \\ \underline{0} \end{bmatrix} \tag{68}$$

holds.

Formula (68) implies $T_q\delta_p$ orthogonal checks on the message digit m_0. If C_q is an even-weight code,

$$m_0 = c_0 + c_{T_p} + c_{2T_p} + \cdots\cdots + c_{(T_q-1)T_p}$$

is an additional check for m_0, orthogonal to the $T_q\delta_p$ previous ones.

(ii): Immediate consequence of (i). □

If in addition C_q also has an orthogonal check set of size δ_q on any code position such that any check has the same even weight, then we have the following lower bound for $\underline{s}$.

(5.2.4) Theorem:

(i): If $wt((x^{T_q}+1)/q(x))$ is even and $T_q\delta_p + 1 \geq T_p(\delta_q + 1)$, then the separation vector $\underline{s}$ of the $[T_pT_q, k_p+k_q]$ code with check polynomial $p(x)q(x)$ satisfies

$$\begin{aligned} s_i &\geq T_q\delta_p + 1 \qquad \text{for } i = 1,\ldots,k_p, \\ s_i &\geq T_p(\delta_q + 1) \qquad \text{for } i = k_p+1,\ldots k_p+k_q. \end{aligned} \tag{69}$$

(ii): If $wt((x^{T_q}+1)/q(x))$ is odd and $T_q\delta_p \geq T_p(\delta_q + 1)$, then the separation vector $\underline{s}$ of the $[T_pT_q, k_p+k_q]$ code with check polynomial $p(x)q(x)$ satisfies

$$\begin{aligned} s_i &\geq T_q\delta_p \qquad \text{for } i = 1,\ldots,k_p, \\ s_i &\geq T_p(\delta_q + 1) \qquad \text{for } i = k_p+1,\ldots,k_p+k_q. \end{aligned} \tag{70}$$

Proof:

(i): For $i = 1,\ldots,k_p$ formula (69) was shown in Theorem (5.2.3). Without loss of generality we consider the message digit m_{k_p}. For $j = 0,1,\ldots,T_p-1$, m_{k_p} equals

$$m_{k_p} = c_{jT_q} + \sum_{i=0}^{k_p-1} m_i G_{i,jT_q}, \tag{71}$$

where G is the matrix of (67). If an error of weight less than or equal to $\lfloor (T_p(\delta_q+1)-1)/2 \rfloor$ occurs, then the message digits $m_0,\ldots,m_{k_p-1}$ are correctly decodable, since $T_q\delta_p + 1 \geq T_p(\delta_q + 1)$. If we fill in these values of $m_0,\ldots,m_{k_p-1}$ in formula (71), then the $T_p(\delta_q + 1)$ checks on m_{k_p} obtained from the formulas (68) and (71) are mutually orthogonal. Hence $s_{k_p+1} \geq T_p(\delta_q + 1)$.

(ii): Analoguos to (i).

□

(5.2.5) Example: Take $p(x) := x^3+x+1$, $q(x) := x^4+x^3+x^2+x+1$. $T_p = 7$, $T_q = 5$. The [7,3] code C_p with check polynomial p(x) has an orthogonal check set of size $\delta_p = 3$ on any code position, where all checks have weight 2 (for example, for the 0-position we have the checks {1,5}, {2,3}, and {4,6}). The [5,4] code C_q with check polynomial q(x) has an orthogonal check set of size $\delta_q = 1$ on any code position, where the check has weight 4 (for example, for the 0-position we have the check {1,2,3,4}). By Theorem (5.2.4) the [35,7] code C with check polynomial p(x)q(x) has a separation vector $\underline{s}$ which satisfies $\underline{s} \geq (16,16,16,14,14,14,14)$.

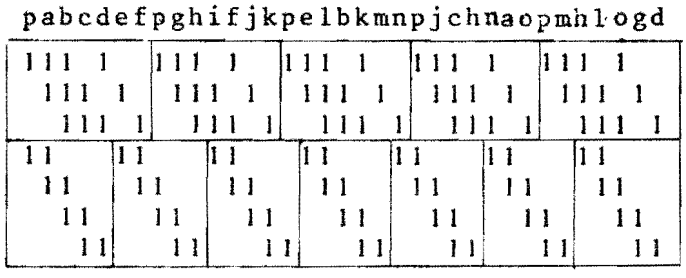


Fig. 5.1: Generator matrix of the [35,7] code with check polynomial $(x^3+x+1)(x^4+x^3+x^2+x+1)$.

Fig. 5.1 shows an optimal generator matrix for C, together with the checks on m_0 and m_3. The message bit m_0 is equal to the following sixteen orthogonal checks.

$$\begin{array}{lll} a: c_1 + c_{26} & f: c_6 + c_{11} & k: c_{13} + c_{18} \\ b: c_2 + c_{17} & g: c_8 + c_{33} & l: c_{16} + c_{31} \\ c: c_3 + c_{23} & h: c_9 + c_{24} & m: c_{19} + c_{29} \\ d: c_4 + c_{34} & i: c_{10} + c_{30} & n: c_{20} + c_{25} \\ e: c_5 + c_{15} & j: c_{12} + c_{22} & o: c_{27} + c_{32} \end{array} \tag{72}$$

$$p: c_0 + c_7 + c_{14} + c_{21} + c_{28}.$$

The message bit m_3 is equal to the following fourteen "orthogonal" checks.

$$\begin{array}{ll} A: c_1 + c_8 + c_{22} + c_{29} & c_0 + m_0 \\ B: c_2 + c_9 + c_{16} + c_{23} & c_5 + m_1 \\ C: c_3 + c_{17} + c_{24} + c_{31} & c_{10} + m_1 + m_2 \\ D: c_4 + c_{11} + c_{18} + c_{32} & c_{15} + m_0 + m_1 \\ E: c_6 + c_{13} + c_{27} + c_{34} & c_{20} + m_2 \\ F: c_7 + c_{14} + c_{21} + c_{28} & c_{25} + m_0 + m_2 \\ G: c_{12} + c_{19} + c_{26} + c_{33} & c_{30} + m_0 + m_1 + m_2. \end{array} \tag{73}$$

Actually the separation vector of C equals (16,16,16,14,14,14,14), as one can easily check.

The [35,8] code C^* with check polynomial $(x+1)p(x)q(x)$ has a separation vector equal to (15,15,15,7,7,7,7,7). For C^*, a,b,....,o are fifteen orthogonal checks on m_0; A,B,....,G are seven orthogonal checks on m_3. For the message bit m_7 we have the following seven checks.

$$\begin{array}{l} c_0 + c_7 + c_{14} + c_{21} + c_{28} + m_0 \\ c_1 + c_8 + c_{15} + c_{22} + c_{29} + m_0 + m_1 \\ c_2 + c_9 + c_{16} + c_{23} + c_{30} + m_0 + m_1 + m_2 \\ c_3 + c_{10} + c_{17} + c_{24} + c_{31} + m_1 + m_2 \\ c_4 + c_{11} + c_{18} + c_{25} + c_{32} + m_0 + m_2 \\ c_5 + c_{12} + c_{19} + c_{26} + c_{33} + m_1 \\ c_6 + c_{13} + c_{20} + c_{27} + c_{34} + m_2. \end{array} \tag{74}$$

We can extend Theorem (5.2.3) to codes with a check polynomial which is a product of more than two irreducible polynomials in $\mathbb{F}_2[x]$.

(5.2.6) Theorem: For $i = 1,\ldots,v$ let $p_i(x)$ be an irreducible polynomial in $\mathbb{F}_2[x]$ of degree k_i and exponent T_i such that $g.c.d.(T_i,T_j) = 1$ for all i,j, $i \neq j$, and let the $[T_i,k_i]$ binary code with check polynomial $p_i(x)$ have an orthogonal check set of size δ_i such that all checks have the same even weight. Then the code C of length $n = \prod_{i=1}^{v} T_i$ and dimension $\sum_{i=1}^{v} k_i$ with check polynomial $\prod_{i=1}^{v} p_i(x)$ has a separation vector $\underline{s}$ which satisfies

$$s_j \geq n\delta_i/T_i \tag{75}$$

for $i = 1,\ldots,v$ and $j = (\sum_{u=1}^{i-1} k_u)+1,\ldots,\sum_{u=1}^{i} k_u$.

Proof: Analogous to Theorem (5.2.3). □

In many cases we can do much better than formula (75) by adding other checks, e.g. checks like the ones in formula (71). This will be shown in the next example.

(5.2.7) Example: Take $p(x) := x^3+x+1$, $q(x) := x^2+x+1$, $r(x) := x^4+x^3+x^2+x+1$. $T_p = 7$, $T_q = 3$, $T_r = 5$.
Let C be the $[105,9]$ code with check polynomial $p(x)q(x)r(x)$. C has an optimal generator matrix $G := (M_p^T|M_q^T|M_r^T)^T$, where M_p, M_q, and M_r are repetitions of resp.

$$P = \begin{bmatrix} 1110100 \\ 0111010 \\ 0011101 \end{bmatrix}, \quad Q = \begin{bmatrix} 110 \\ 011 \end{bmatrix}, \quad R = \begin{bmatrix} 11000 \\ 01100 \\ 00110 \\ 00011 \end{bmatrix}.$$

The $[7,3]$ code with generator matrix P has three orthogonal checks $\{1,5\}$, $\{2,3\}$, and $\{4,6\}$ of weight 2 on code position 0. The $[3,2]$ code with generator matrix Q has one check $\{1,2\}$ of weight 2 on code position 0. The $[5,4]$ code with generator matrix R has one check $\{1,2,3,4\}$ of weight 4 on code position 0. Hence $\delta_p = 3, \delta_q = 1$, and $\delta_r = 1$.
Any vector $(\underline{x}^T,\underline{y}^T,\underline{z}^T)^T$, where $\underline{x},\underline{y}$, and $\underline{z}$ are columns of resp. P,Q, and R, occurs exactly once as a column of G. Now for any $i \in \{1,..,T_q\}$ and

and $j \in \{1,\ldots,T_r\}$ we have

$$\begin{bmatrix} P_{*1} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*5} \\ Q_{*i} \\ R_{*j} \end{bmatrix} = \begin{bmatrix} P_{*2} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*3} \\ Q_{*i} \\ R_{*j} \end{bmatrix} = \begin{bmatrix} P_{*4} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*6} \\ Q_{*i} \\ R_{*j} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (76)$$

This implies $T_q T_r \delta_p = 45$ orthogonal checks on m_0. These checks do not contain the fifteen code digits $\{ c_j \mid j \equiv 0 \bmod 7 \}$, which correspond to the columns of G given in Fig. 5.2.

```
0 21 42 63 84 35 56 77 98 14 70 91  7 28 49 ← column numbers
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  1  1  1  1  .  .  .  .  .  1  1  1  1  1
.  .  .  .  .  1  1  1  1  1  1  1  1  1  1
1  1  .  .  .  1  1  .  .  .  1  1  .  .  .
.  1  1  .  .  .  1  1  .  .  .  1  1  .  .
.  .  1  1  .  .  .  1  1  .  .  .  1  1  .
.  .  .  1  1  .  .  .  1  1  .  .  .  1  1
a  a  a  b  c  c  c  c  a  b  b  b  b  c  a
```

From now on points (.) in matrices should be read as zeros (0).

Fig. 5.2: Columns G_{*j} of G where $j \equiv 0 \bmod 7$.

From Fig. 5.2 it is easy to see that

$$\begin{aligned} &a: \; c_0 + c_{21} + c_{42} + c_{49} + c_{98} \\ &b: \; c_7 + c_{14} + c_{63} + c_{70} + c_{91} \\ &c: \; c_{28} + c_{35} + c_{56} + c_{77} + c_{84} \end{aligned} \qquad (77)$$

are three additional orthogonal checks on m_0, which are orthogonal to the 45 checks implied by formula (76). Hence we have 48 orthogonal checks on m_0. Analogously we can find 48 orthogonal checks on m_1 and m_2.
For any $i \in \{1,\ldots,T_p\}$ and any $j \in \{1,\ldots,T_r\}$ we have

$$\begin{bmatrix} P_{*i} \\ Q_{*1} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*2} \\ R_{*j} \end{bmatrix} = \underline{e}_3 := (0,0,0,1,0,0,0,0,0)^T. \tag{78}$$

This implies $T_p T_r \delta_q = 35$ orthogonal checks on m_3. These checks do not contain the code digits $\{ c_j \mid j \equiv 0 \bmod 3 \}$, which correspond to the columns of G given in Fig. 5.3.

```
0 21 42 63 84 15 36 57 78 99 30 51 72 93  9 45 66 87  3 24 60 81 102 18 39 75 96 12 33 54 90  6 27 48 69
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  .  .  .  .  .  1  1   1  1  1  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1   .  .  .  .  .  1  1  1  1  1  .  .  .  .
.  .  .  .  .  .  .  .  .  .  1  1  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  .  .  .  .  .  1  1  1  1  1
1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1   1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1  1   .  .  .  1  1  .  .  .  1  1  .  .  .  1  1
.  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1   1  .  .  .  1  1  .  .  .  1  1  .  .  .  1
.  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .   1  1  .  .  .  1  1  .  .  .  1  1  .  .  .
.  .  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .  .  1  1  .  .   .  .  1  1  .  .  .  1  1  .  .  .  1  1  .

a b c d d e a f g f c d b a a b c e f g g e d b b d f a e e f g g c c
```

Fig. 5.3: Columns G_{*j} of G where $j \equiv 0 \bmod 3$.

From Fig. 5.3 it is easy to see that

$$\begin{aligned}
&a: \; c_0 + c_9 + c_{12} + c_{36} + c_{93} \\
&b: \; c_{18} + c_{21} + c_{39} + c_{45} + c_{72} \\
&c: \; c_{30} + c_{42} + c_{48} + c_{66} + c_{69} \\
&d: \; c_{51} + c_{63} + c_{75} + c_{84} + c_{102} \\
&e: \; c_{15} + c_{33} + c_{54} + c_{81} + c_{87} \\
&f: \; c_3 + c_{57} + c_{90} + c_{96} + c_{99} \\
&g: \; c_6 + c_{24} + c_{27} + c_{60} + c_{78}
\end{aligned} \tag{79}$$

are seven additional orthogonal checks on m_3, which are orthogonal to the 35 checks implies by formula (78). Hence we have 42 orthogonal checks on m_3. Analogously we can find 42 checks on m_4.

For any $i \in \{1,\ldots,T_p\}$ and any $j \in \{1,\ldots,T_q\}$ we have

$$\begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*1} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*2} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*3} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*4} \end{bmatrix} = \underline{e}_5 = (0,0,0,0,0,1,0,0,0)^T. \quad (80)$$

This implies $T_p T_q \delta_r = 21$ orthogonal checks on m_5. These checks do not contain the 21 code digits $\{ c_j \mid j \equiv 0 \bmod 5 \}$. Since $s_1, s_2, \ldots, s_5 \geq 42$,

$$c_j + \sum_{i=0}^{4} m_i G_{ij} \quad (81)$$

for $j \in \{0,5,10,\ldots,100\}$ build 21 additional checks on m_5 (cf. the proof of Theorem (5.2.4)). Hence we have 42 "orthogonal" checks on m_5. Analogously we can find 42 checks on m_6, m_7, and m_8.
We have shown that the $[105,9]$ code with check polynomial $(x^3+x+1)(x^2+x+1)(x^4+x^3+x^2+x+1)$ has a separation vector of at least $(48,48,48,42,42,42,42,42,42)$. Actually equality holds, as one can easily check. So we have derived a Majority Logic Decoding Scheme for the code that reaches the actual separation vector.

For a binary cyclic code whose check polynomial is the product of $(x+1)$ and two primitive polynomials we have the following theorem.

(5.2.8) Theorem: For the primitive polynomials $p(x), q(x) \in \mathbb{F}_2[x]$ of degrees k_p resp. k_q such that $k_p > k_q$ and $g.c.d.(k_p,k_q) = 1$ the binary cyclic $[(2^{k_p}-1)(2^{k_q}-1), k_p+k_q+1]$ code with check polynomial $(x+1)p(x)q(x)$ has a separation vector $\underline{s}$ which satisfies

$$s_i = \begin{cases} (2^{k_q}-1)(2^{k_p-1}-1) & \text{for } i = 1,\ldots,k_p \\ (2^{k_p}-1)(2^{k_q-1}-1) & \text{for } i = k_p+1,\ldots,k_p+k_q+1. \end{cases} \quad (82)$$

Proof: The $[2^{k_p}-1, k_p]$ cyclic code C_p with primitive check polynomial $p(x)$ of degree k_p is a simplex code (i.e. all elements of $\mathbb{F}_2^{k_p} \setminus \{0\}$ occur as columns in a generator matrix of C_p). Hence we have an orthogonal check set of size $\delta_p := (2^{k_p-1}-1)$ on any code position, where all weights of the checks equal 2. The same holds for the $[2^{k_q}-1, k_q]$ code C_q with primitive check polynomial $q(x)$; $\delta_q := (2^{k_q-1}-1)$. Since $g.c.d.(k_p,k_q) = 1$, we may apply Theorem (5.2.3)(ii) to the first k_p message bits as well as to the message bits $m_{k_p}, \ldots, m_{k_p+k_q-1}$.

Furthermore

$$wt((x^n+1)/p(x) + \sum_{i=0}^{n-1} x^i) = (2^{k_q}-1)(2^{k_p}-1) \quad \text{and}$$

$$wt((x^n+1)/q(x) + \sum_{i=0}^{n-1} x^i) = (2^{k_p}-1)(2^{k_q}-1),$$

where $n := (2^{k_p}-1)(2^{k_q}-1)$.

These observations imply formula (82). □

(5.2.9) Examples:

(i): Take $k_p = 3$, $k_q = 2$, $p(x) = x^3+x+1$, $q(x) = x^2+x+1$.

$$G := \begin{bmatrix}
1&1&1&.&1&.&.&1&1&1&.&1&.&.&1&1&1&.&1&.&. \\
.&1&1&1&.&1&.&.&1&1&1&.&1&.&.&1&1&1&.&1&. \\
.&.&1&1&1&.&1&.&.&1&1&1&.&1&.&.&1&1&1&.&1 \\
1&1&.&1&1&.&1&1&.&1&1&.&1&1&.&1&1&.&1&1&. \\
.&1&1&.&1&1&.&1&1&.&1&1&.&1&1&.&1&1&.&1&1 \\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

is an optimal generator matrix for the [21,6] cyclic code C with check polynomial $(x+1)p(x)q(x)$. $\underline{s}(G) = (9,9,9,7,7,7)$.

m_0, m_3, and m_5 have the following checks.

$$\begin{aligned}
m_0 &= c_1 + c_{19} = c_2 + c_{17} = c_3 + c_9 = c_4 + c_{13} = c_5 + c_8 \\
&= c_6 + c_{18} = c_{10} + c_{16} = c_{11} + c_{20} = c_{12} + c_{15},
\end{aligned}$$

$$\begin{aligned}
m_3 &= c_1 + c_8 = c_2 + c_{16} = c_4 + c_{11} = c_5 + c_{19} \\
&= c_7 + c_{14} = c_{10} + c_{17} = c_{13} + c_{20},
\end{aligned}$$

$$\begin{aligned}
m_5 &= c_0 + c_1 + c_2 + m_0 + m_2 = c_3 + c_4 + c_5 + m_0 \\
&= c_6 + c_7 + c_8 + m_1 + m_2 = c_9 + c_{10} + c_{11} + m_2 \\
&= c_{12} + c_{13} + c_{14} + m_0 + m_1 + m_2 = c_{15} + c_{16} + c_{17} + m_1 \\
&= c_{18} + c_{19} + c_{20} + m_0 + m_1.
\end{aligned}$$

The generator matrix G' for this code, whose rows are the cyclic shifts of the generator polynomial, has a separation vector $\underline{s}(G') = (7,7,7,7,7,7)$.

A binary [21,6] code has a minimum distance of at most 8 (cf. Helgert and Stinaff (1973)).

(ii): Take $k_p = 4$, $k_q = 3$, $p(x) := (x^4+x+1)$, $q(x) := (x^3+x+1)$. The [105,8] code with check polynomial $(x+1)p(x)q(x)$ has separation vector (49,49,49,49,45,45,45,45).

We can also extend Theorem (5.2.9) to the following one.

(5.2.10) Theorem: For the primitive polynomials $p_i(x) \in \mathbb{F}_2[x]$, $i = 1,\ldots,v$ of degrees resp. k_i, $i = 1,\ldots,v$ such that $k_1 > k_2 > \ldots > k_v$ and g.c.d.$(k_i,k_j) = 1$ for all i,j, $i \neq j$, the binary cyclic $[\prod_{i=1}^{v} (2^{k_i}-1), 1+\sum_{i=1}^{v} k_i]$ code with check polynomial $(x+1)\prod_{i=1}^{v} p_i(x)$ has a separation vector $\underline{s}$ which satisfies

$$s_i = (2^{k_j-1}-1)\prod_{u=1}^{v} (2^{k_u}-1)/(2^{k_j}-1) \tag{83}$$

for $i = \sum_{u=1}^{j-1} k_u + 1,\ldots,\sum_{u=1}^{j} k_u$ and $j = 1,\ldots,v$.

Proof: Analogous to the proof of Theorem (5.2.8). □

Table B.1 in Appendix B contains the separation vectors of all binary cyclic UEP codes of length less than or equal to 39.

5.3 Notes

Theorem (5.1.1) is from Dunning and Robbins (1978). It is easy to prove the Corollaries (5.1.2) and (5.1.3) without the results of Chapter 2, by only using the special configuration of the generator matris G in formula (55). Theorem (5.2.1) is from Dyn'kin and Togonidze (1976). They also mention Theorem (5.2.8) without a proof.

APPENDIX A: BINARY OPTIMAL LINEAR UEP CODES OF LENGTH LESS THAN OR EQUAL TO 15

n	k	d(n,k)	separation vector
4	2	2	32
5	2	3	42
5	3	2	322
6	2	4	52
6	2	3	422
6	4	2	3222
7	2	4	62, 54
7	3	4	522
7	4	3	4222
7	5	2	32222
8	2	5	72, 64
8	3	4	622, 544
8	4	4	5222
8	5	2	42222, 33332
8	6	2	322222
9	2	6	82, 74
9	3	4	722, 644, 554
9	4	4	6222, 5444
9	5	3	52222, 44442, 43333
9	6	2	422222, 333322
9	7	2	3222222
10	2	6	92, 84, 76
10	3	5	822, 744, 664
10	4	4	7222, 6444, 5544
10	5	4	62222, 54444
10	6	3	522222, 444422, 433332
10	7	2	4222222, 3333222
10	8	2	32222222
11	2	7	10.2, 94, 86
11	3	6	922, 844, 764
11	4	5	8222, 7444, 6644
11	5	4	72222, 64444, 55444
11	6	4	622222, 544442, 533333
11	7	3	5222222, 4444222, 4333322
11	8	2	42222222, 33332222
11	9	2	322222222
12	2	8	11.2, 10.4, 96
12	3	6	10.22, 944, 864, 774, 766
12	4	6	9222, 8444, 7644
12	5	4	82222, 74444, 66444, 55554
12	6	4	722222, 644444, 554444
12	7	4	6222222, 5444422, 5333332
12	8	3	52222222, 44442222, 43333222
12	9	2	422222222, 333322222
12	10	2	3222222222

Table A.1: The separation vectors of the binary optimal LUEP codes of length less than or equal to 15 (Part I).

n	k	d(n,k)	separation vector
13	2	8	12.2, 11.4, 10.6, 98
13	3	7	11.22, 10.44, 964, 884, 866
13	4	6	10.22, 9444, 8644, 7744, 7666
13	5	5	92222, 84444, 76444, 66664, 66555
13	6	4	822222, 744444, 664444, 555544
13	7	4	7222222, 6444442, 6333333, 5544442, 5444444
13	8	4	62222222, 54444222, 53333322
13	9	3	522222222, 444422222, 433332222
13	10	2	4222222222, 3333222222
13	11	2	32222222222
14	2	9	13.2, 12.4, 11.6, 10.8
14	3	8	12.22, 11.44, 10.64, 984, 966
14	4	7	11.222, 10.444, 9644, 8844, 8666
14	5	6	10.2222, 94444, 86444, 77444, 76666 (a)
14	6	5	922222, 844444, 764444, 666644, 665552
14	7	4	8222222, 7444444 (b), 6644442, 6544444, 5555444 (c)
14	8	4	72222222, 64444422, 63333332, 55444422, 54444444
14	9	4	622222222, 544442222, 533333222
14	10	3	5222222222, 4444222222
14	11	2	42222222222, 33332222222
14	12	2	322222222222
15	2	10	14.2, 13.4, 12.6, 11.8
15	3	8	13.22, 12.44, 11.64, 10.84, 10.66, 988
15	4	8	12.222, 11.444, 10.644, 9844 (d), 9666
15	5	7	11.2222, 10.4444, 96444, 88444, 86666
15	6	6	10.22222, 944444, 864444, 774444, 766662, 766644, 765554 (e)
15	7	5	9222222, 8444444, 7644444 (f), 6666444, 6655522
15	8	4	82222222, 74444442, 66444422, 65444442, ... (1), 64444444, ... (2)
15	9	4	722222222, 644444222, 633333322, 554444222, 544444444
15	10	4	6222222222, 5444422222, 5333332222
15	11	3	52222222222, 44442222222
15	12	2	422222222222, 333322222222
15	13	2	3222222222222

Table A.1: The separation vectors of the binary optimal LUEP codes of length less than or equal to 15 (Part II).

Any separation vector of a binary linear [n,k] code ($4 \leq n \leq 15$, $2 \leq k \leq n-2$) is less than or equal to one of the separation vectors in the row of Table A.1 corresponding to n,k. If a component of a separation vector consists of two digits it is followed by a point. Let d(n,k) denote the maximal minimum distance of a binary linear code of length n and dimension k (cf. Helgert and Stinaff (1973)).
The nontrivial codes in Table A.1 are constructed in one of the following ways.

(i): Adding a parity check to another code in the table.

(ii): Shortening or puncturing other codes in the table.

(iii): Adding a column of weight one to a generator matrix of a LUEP code of length one less.

(iv): The following construction:

If the matrix G_1 has separation vector $\underline{s}(G_1)$, then

$$G_2 := \left[\begin{array}{c|c} \begin{matrix}0\\0\\\vdots\\0\end{matrix} & G_1 \\ \hline 1 & 1\ 0\ 0\ \ldots\ldots\ 0 \end{array}\right]$$

has separation vector $\underline{s}(G_2) = (\underline{s}(G_1),2)$.

(v): The constructions in Section 4.1.

(vi): (a). Construction (4.2.1) with $m = 1$ and G_1 a generator matrix of a [7,4,3] Hamming code.

(b). Construction (4.2.1) with $m = 1$ and G_1 a generator matrix of a [7,6,2] code.

(c).
```
[...111111......]
[...11...111...]
[...1.1..1..1.1]
[....1.1..1..11]
[1..1...11.....]
[.1..1.1...1...]
[..1........111]
```
is a generator matrix of a [14,7,(5,5,5,5,4,4,4)] code.

(d).
```
[111111111......]
[.1111....1111..]
[.1...1...1...1.]
[..1...1...1...1]
```
is a generator matris of a [15,4,(9,8,4,4)] code.

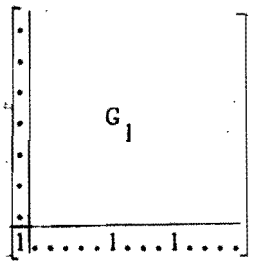
(e).
```
[....11111...111]
[.........111111]
[1...11...1..1..]
[.1..1.1...1..1.]
[..1.1..1...1..1]
[...1....1..1..1]
```
is a generator matrix of a [15,6,(7,6,5,5,5,4)] code.

(f).
```
[.....1111111...]
[.....111....111]
[1....1..1...1..]
[.1...1..1....1.]
[..1...1..1....1]
[...1..1...1...1]
[....1.1....1..1]
```
is a generator matrix of a [15,7,(7,6,4,4,4,4,4)] code.

Table A.1 has two open places.

(1): Does a [15,8,$\underline{s}$] code with $(6,5,3,3,3,3,3,3) \leq \underline{s} \leq (6,5,4,4,4,3,3,3)$ exist?

(2): Does a [15,8,(5,5,5,x,4,4,4,4)] code with $x \in \{4,5\}$ exist?



with G_1 the matrix in (vi)(c) has separation vector (5,5,5,5,4,4,4,3). By adding a column of weight one to an optimal generator matrix of a [14,8,(5,4,4,4,4,4,4,4)] code, we obtain a [15,8,(5,5,4,4,4,4,4,4)] code.

APPENDIX B: A TABLE OF ALL BINARY CYCLIC UEP CODES OF LENGTH LESS THAN OR EQUAL TO 39

Table B.1 contains the parameters of all binary cyclic UEP codes of length less than or equal to 39. In this table for each code of length n the exponents i,j,k,... of a primitive n^{th} root of unity α are given such that $\alpha^i,\alpha^j,\alpha^k,\ldots$ are nonzeros of the code. For example the first row of the table denotes a binary cyclic [15,7,(5,5,3,3,3,3,3)] code with nonzeros $\{ \alpha^i \mid i \in C_5 \cup C_0 \cup C_3 \}$, where C_i (i = 5,0,3) denotes the cyclotomic coset modulo 15 containing i. The order of the nonzeros corresponds to the order of the components in the separation vector. I.e. if the order of the nonzeros is i,j,k,.., then the separation vector equals $\underline{s}((M_i^T|M_j^T|M_k^T|\ldots))$, where M_x (x = i,j,k,...) denotes a generator matrix of the minimal ideal in $\mathbb{F}_2[x]/(x^n+1)$ with nonzeros $\{ \alpha^y \mid y \in C_x \}$. In the above example $\underline{s}((M_5^T|M_0^T|M_3^T)) = (5,5,3,3,3,3,3)$.
The last column of the table contains the minimum length or a bound on the minimum length of a binary linear code with a separation vector of at least the one of the corresponding cyclic code. The separation components (and the corresponding nonzeros) larger than the minimum distance of the code are underlined.

length	dim.	nonzeros	separation vector $\underline{s}$	$n(\underline{s})$
15	7	5,0,3	5,5,3,3,3,3,3	14
	9	1,0,3	4,4,4,4,3,3,3,3,3	14
	9	0,1,7	5,4,4,4,4,4,4,4,4	15
	11	0,1,5,7	5,2,2,2,2,2,2,2,2,2,2	15
	13	0,1,3,7	3,2,2,2,2,2,2,2,2,2,2,2,2	15
21	6	3,0,7	9,9,9,7,7,7	≥ 20
	7	0,1	9,8,8,8,8,8,8	21
	8	7,3,9	8,8,6,6,6,6,6,6	20
	9	7,0,3,9	7,7,3,3,3,3,3,3,3	18
	9	0,1,7	7,6,6,6,6,6,6,6,6	19
	10	0,1,9	9,4,4,4,4,4,4,4,4,4	20
	11	7,1,9	6,6,4,4,4,4,4,4,4,4,4	19
	12	3,1,9	6,6,6,4,4,4,4,4,4,4,4,4	≥ 20
	12	7,0,1,3	6,6,5,5,5,5,5,5,5,5,5,5	21
	13	0,1,5	7,4,4,4,4,4,4,4,4,4,4,4,4	21
	13	1,0,3,9	4,4,4,4,4,4,3,3,3,3,3,3,3	≥ 18
	15	0,1,5,7	7,2,2,2,2,2,2,2,2,2,2,2,2,2,2	21
	17	3,1,5,7	4,4,4,2,2,2,2,2,2,2,2,2,2,2,2,2	21
25	21	0,1	5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	15
27	7	0,3	9,6,6,6,6,6,6	19
	19	0,1	9,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	27
	20	9,1	6,6,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	27
	21	0,9,1	3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	24
	25	0,1,3	3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	27
31	11	0,1,15	11,10,10,10,10,10,10,10,10,10,10	≥ 30
	16	0,1,3,15	9,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6	≥ 29

Table B.1: All binary cyclic UEP codes of length less than or equal to 39 (Part I).

length	dim.	nonzeros	separation vector $\underline{s}$	$n(\underline{s})$
33	12	11,3	12,12,6,6,6,6,6,6,6,6,6,6	≥ 29
	13	0,1,11	11,10,10,10,10,10,10,10,10,10,10,10,10	≥ 32
	13	11,0,3	11,11,3,3,3,3,3,3,3,3,3,3,3	≥ 28
	21	0,1,5	11,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 32
	23	0,1,5,11	11,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	33
	23	1,11,0,3	5,5,5,5,5,5,5,5,5,5,5,5,3,3,3,3,3,3,3,3,3,3,3	≥ 32
	31	0,1,3,5	3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	33
35	7	5,7	16,16,16,14,14,14,14	≥ 34
	8	5,0,7	15,15,15,7,7,7,7,7	≥ 32
	11	7,0,5,15	7,7,7,7,5,5,5,5,5,5,5	22
	13	0,1	15,8,8,8,8,8,8,8,8,8,8,8,8	≥ 33
	15	5,1	12,12,12,8,8,8,8,8,8,8,8,8,8,8,8	≥ 33
	16	0,1,15	15,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 32
	17	0,1,7	7,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6	≥ 28
	18	5,1,15	8,8,8,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 29
	19	0,5,1,15	5,5,5,5,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 26
	19	5,1,7	8,8,8,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6	≥ 30
	19	7,1,15	6,6,6,6,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 27
	20	0,5,7,1	7,7,7,7,7,7,7,7,6,6,6,6,6,6,6,6,6,6,6,6	≥ 31
	22	5,7,1,15	6,6,6,6,6,6,6,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 31
	25	0,1,3	7,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 33
	28	0,1,3,5	5,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	35
	29	0,1,3,7	7,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	35

Table B.1: All binary cyclic UEP codes of length less than or equal to 39 (Part II).

length	dim.	nonzeros	separation vector $\underline{s}$	$n(\underline{s})$
35	31	0,1,3,5,15	5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	35
	31	5,1,3,7	4,4,4,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	35
	32	0,5,1,3,7	3,3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	35
39	13	0,1	15,12,12,12,12,12,12,12,12,12,12,12,12	$\geq$ 37
	14	13,3	14,14,6,6,6,6,6,6,6,6,6,6,6,6	$\geq$ 35
	15	0,1,13	13,10,10,10,10,10,10,10,10,10,10,10,10,10,10	$\geq$ 36
	15	13,0,3	13,13,3,3,3,3,3,3,3,3,3,3,3,3,3,3	$\geq$ 33
	25	1,0,3	6,6,6,6,6,6,6,6,6,6,6,6,3,3,3,3,3,3,3,3,3,3,3,3,3	$\geq$ 35
	25	0,1,7	13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	39
	27	1,13,0,3	6,6,6,6,6,6,6,6,6,6,6,6,6,6,3,3,3,3,3,3,3,3,3,3,3,3,3	$\geq$ 37
	27	0,1,7,13	12,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	39
	37	0,1,3,7	3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	39

Table B.1: All binary cyclic UEP codes of length less than or equal to 39 (Part III).

REFERENCES

Boyarinov, I.M., and Katsman, G.L. (1980), "Linear Unequal Error Protection Codes", IEEE Trans. on Information Theory 27, no. 2, pp. 168-175.

Cameron, P.J., and van Lint, J.H. (1980), "Graphs, Codes and Designs", London Mathematical Society Lecture Note Series 43.

Dunning, L.A., and Robbins, W.E. (1978), "Optimal Encodings of Linear Block Codes for Unequal Error Protection", Information and Control 37, pp. 150-177.

Dyn'kin, V.N., and Togonidze, V.A. (1976), "Cyclic Codes with Unequal Symbol Protection", Problems of Information Transmission 12, no. 1, pp.16-19 (translated from Problemy Peredachi Informatsii 12, no.1, pp.24-28).

van Gils, W.J. (1981), "On Linear Unequal Error Protection Codes", Report Philips' Research Laboratories Eindhoven, The Netherlands.

Helgert, H.J., and Stinaff, R.D. (1973), "Minimum-Distance Bounds for Binary Linear Codes", IEEE Trans. on Information Theory 19, no. 3, pp. 344-356.

Katsman, G.L. (1980), "Bounds on Volume of Linear Codes with Unequal Information-Symbol Protection", Problems of Information Transmission 16, no. 2, pp. 99-105 (translated from Problemy Peredachi Informatsii 16, no. 2, pp. 25-32).

van Lint, J.H. (1982), "Introduction to Coding Theory", Graduate Texts in Mathematics 86, Springer Verlag New York.

MacWilliams, F.J., and Sloane, N.J.A. (1978), "The Theory of Error-Correcting Codes", North-Holland Mathematical Library, Vol. 16, Amsterdam.

Masnick, B., and Wolf, J. (1967), " On Linear Unequal Error Protection Codes", IEEE Trans. on Information Theory 13, no. 4, pp. 600-607.

Peterson, W.W., and Weldon jr., E.J. (1972), "Error-Correcting Codes", 2nd edition, The MIT-Press.

Zinov'ev, V.A., and Zyablov, V.V. (1979), "Codes with Unequal Protection of Information Symbols", Problems of Information Transmission 15, no. 3, pp. 197-205 (translated from Problemy Peredachi Informatsii 15, no. 3, pp. 50-60).

INDEX