IEEE
Advancing Technology for Humanity

DECLARATION OF GERARD P. GRENIER

I, Gerard P. Grenier, am over twenty-one (21) years of age. I have never been convicted of a felony, and I am fully competent to make this declaration. I declare the following to be true to the best of my knowledge, information and belief:

1. I am Senior Director of Publishing Technologies of The Institute of Electrical and Electronics Engineers, Incorporated ("IEEE").

2. IEEE is a neutral third party in this dispute.

3. Neither I nor IEEE itself is being compensated for this declaration.

4. Among my responsibilities as Senior Director of Publishing Technologies, I act as a custodian of certain records for IEEE.

5. I make this declaration based on my personal knowledge and information contained in the business records of IEEE.

6. As part of its ordinary course of business, IEEE publishes and makes available technical articles and standards. These publications are made available for public download through the IEEE digital library, IEEE Xplore.

7. It is the regular practice of IEEE to publish articles and other writings including article abstracts and make them available to the public through IEEE Xplore. IEEE maintains copies of publications in the ordinary course of its regularly conducted activities.

8. The articles below have been attached as Exhibits A – B to this declaration:

A.	W. van Gils, "Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes" IEEE Transactions on Information Theory, Vol. 29, Issue 6, November 1983.
B.	U. Dettmar, et al. "Modified generalized concatenated codes and their application to the construction and decoding of LUEP codes" IEEE Transaction on Information Theory, Vol. 41, Issue 5, September 1995.

9. I obtained copies of Exhibits A – B through IEEE Xplore, where it is maintained in the ordinary course of IEEE's business. Exhibits A – B are true and correct copies of the Exhibits, as they existed on or about August 13, 2018.

10. The article abstracts from IEEE Xplore shows the date of publication. IEEE Xplore populates this information using the metadata associated with the publication.

11. W. van Gils, "Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes" was published as part of IEEE Transactions on Information Theory, Vol. 29, Issue 6. IEEE Transactions on Information Theory, Vol. 29, Issue 6 was published in November 1983. Copies of this publication were made available no later than the last day of the publication month. The article is currently available for public download from the IEEE digital library, IEEE Xplore.

12. U. Dettmar, et al. "Modified generalized concatenated codes and their application to the construction and decoding of LUEP codes" was published as part of IEEE Transaction on Information Theory, Vol. 41, Issue 5. IEEE Transaction on Information Theory, Vol. 41, Issue 5 was published in September 1995. Copies of this publication were made available no later than the last day of the publication month. The article is currently available for public download from the IEEE digital library, IEEE Xplore.

13. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

I declare under penalty of perjury that the foregoing statements are true and correct.

Executed on: 14 August 2018

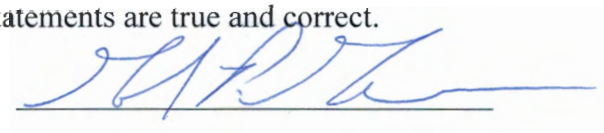

EXHIBIT A

Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes

>> Results

1 Author(s)

W. van Gils W. van Gils

View All Authors

Abstract | Authors | Figures | References | Citations | Keywords

Abstract: It is possible for a linear block code to provide more protection for selected positions in the input message words than is guaranteed by the minimum distance of the code. Linear codes having this property are called linear unequal error protection (LUEP) codes. Bounds on the length of a LUEP code that ensures a given unequal error protection are derived. A majority decoding method for certain classes of cyclic binary UEP codes is treated. A list of short (i.e., of length less than 16) binary LU... **View more**

Metadata

Abstract:
It is possible for a linear block code to provide more protection for selected positions in the input message words than is guaranteed by the minimum distance of the code. Linear codes having this property are called linear unequal error protection (LUEP) codes. Bounds on the length of a LUEP code that ensures a given unequal error protection are derived. A majority decoding method for certain classes of cyclic binary UEP codes is treated. A list of short (i.e., of length less than 16) binary LUEP codes of optimal (i.e., minimal) length and a list of all cyclic binary UEP codes of length less than 40 are included.

Published in: IEEE Transactions on Information Theory (Volume: 29, Issue: 6, Nov 1983)

Page(s): 866 - 876

DOI: 10.1109/TIT.1983.1056753

Date of Publication: Nov 1983

Publisher: IEEE

ISSN Information:

Sponsored by: IEEE Information Theory Society

Related Articles

Modified majority logic decoding of cyclic codes in hybrid-ARQ systems

The Weight Distributions of the Duals of Cyclic Codes With Two Zeros

The Weight Distributions of Cyclic Codes and Elliptic Curves

View All View All

Authors

References

Citations

Keywords

Related Articles

Back to Top

Download PDF

Download Citation

View References

Email

Print

Request Permissions

Export to Collabratec

Alerts

Download PDF

Download Citation

View References

Email

Print

Request Permissions

Export to Collabratec

Alerts

Authors

References

Citations

Keywords

Related Articles

Authors

References

Citations

Keywords

Related Articles

Back to Top

IEEE Account

Profile Information

Purchase Details

Need Help?

Other

US & Canada: +1 800 678 4333
Worldwide: +1 732 981 0060

Two Topics on Linear Unequal Error Protection Codes: Bounds on Their Length and Cyclic Code Classes

WIL J. VAN GILS, MEMBER, IEEE

Abstract—**It is possible for a linear block code to provide more protection for selected positions in the input message words than is guaranteed by the minimum distance of the code. Linear codes having this property are called linear unequal error protection (LUEP) codes. Bounds on the length of a LUEP code that ensures a given unequal error protection are derived. A majority decoding method for certain classes of cyclic binary UEP codes is treated. A list of short (i.e., of length less than 16) binary LUEP codes of optimal (i.e., minimal) length and a list of all cyclic binary UEP codes of length less than 40 are included.**

I. Introduction

MOST error-correcting 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 others. 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, where the different sources have different demands concerning the protection against errors.

Linear codes that protect some positions in a message word against a larger number of errors than other ones are called linear unequal error protection (LUEP) codes. Masnick and Wolf [8] introduced the concept of unequal error protection (UEP). But, in contrast with what one would expect, they considered error protection of single positions in codewords. In this paper we consider error protection of single positions in the input message words, following the formal definitions of Dunning and Robbins [2]. They introduced a so-called separation vector to measure the error-correcting capability of a LUEP code. Whenever a k-dimensional LUEP code over $\mathrm{GF}(q)$ with separation vector $\boldsymbol{s} = (s_1, s_2, \cdots, s_k)$ is used on q-ary symmetric channel, complete nearest neighbor decoding [7, p. 11] guarantees the correct interpretation of the ith input message digit if no more than $\lfloor (s_i - 1)/2 \rfloor$ errors have occurred in the transmitted codeword.

A 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 Section III 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 [6]; he derived Corollary 14 for the binary case. Our bounds give better results than the bound in [6]. Table I provides a table of binary LUEP codes with maximal separation vector and length less than or equal to 15.

In Section IV we consider classes of cyclic UEP codes that can be decoded by majority decoding methods. Earlier results on cyclic UEP codes were obtained by Dyn'kin and Togonidze [3]. Table II provides a table of all binary cyclic UEP codes of length less than or equal to 39.

II. Definitions and Preliminaries

A. The Separation Vector

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

Dunning and Robbins [2] have introduced the following formal definition.

Definition 1: For a linear $[n, k]$ code C over the alphabet $\mathrm{GF}(q)$ the *separation vector* $s(G) = (s(G)_1, \cdots, s(G)_k)$ of length k, with respect to a generator matrix G of C, is defined by

$$s(G)_i = \min \left\{ \mathrm{wt}(\boldsymbol{m}G) \mid \boldsymbol{m} \in \mathrm{GF}(q)^k, m_i \neq 0 \right\}, \quad i = 1, \cdots, k. \quad (1)$$

This means that for any $\alpha, \beta \in \mathrm{GF}(q)$, $\alpha \neq \beta$, the sets

Manuscript received June 25, 1982; revised December 23, 1982. This paper was partially presented at the IEEE International Symposium on Information Theory, Les Arcs, France, June 21–25, 1982.

The author was with the Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands. He is now with Philips Research Laboratories, 5600 MD Eindhoven, The Netherlands.

0018-9448/83/1100-0866$01.00

$\{\boldsymbol{mG}|\boldsymbol{m} \in \mathrm{GF}(q)^k,\ m_i = \alpha\}$ and $\{\boldsymbol{mG}|\boldsymbol{m} \in \mathrm{GF}(q)^k,\ m_i = \beta\}$ are at distance $S(G)_i$ apart ($i = 1,\cdots,k$). This observation implies the following error-correcting capability of a code when we use it on a q-ary symmetric channel.

Theorem 1: For a linear $[n, k]$ code C over $\mathrm{GF}(q)$, which uses a matrix G for its encoding, complete nearest neighbor decoding guarantees the correct interpretation of the ith message digit whenever the error pattern has a Hamming weight less than or equal to $\lfloor (s(G)_i - 1)/2 \rfloor$ ($\lfloor x \rfloor$ denotes the largest integer less than or equal to x).

From Definition 1 it is immediately clear that the minimum distance of the code equals $d = \min\{s(G)_i | i = 1,\cdots,k\}$. If a linear code C has a generator matrix G such that the components of the separation vector $\boldsymbol{s}(G)$ are not mutually equal, then the code C is called a *linear unequal error protection* (LUEP) code.

One can easily decode LUEP codes by applying a syndrome decoding method using a standard array (cf. [7, p. 15]). This decoding method reaches the correction capability given by Theorem 1, 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 all possible coset leaders of R. For any $\boldsymbol{r} \in R$, $\boldsymbol{r} + U$ contains all codewords that are closet to $\boldsymbol{r}$, i.e., at distance $d(\boldsymbol{r}, C) = \min\{\mathrm{wt}(\boldsymbol{r} - \boldsymbol{c}) | \boldsymbol{c} \in C\}$ from $\boldsymbol{r}$. If $i \in \{1,\cdots,k\}$ is such that the weight of the elements of U is less than or equal to $\lfloor (s(G)_i - 1)/2 \rfloor$, then the ith digits of the messages corresponding to the elements of $\boldsymbol{r} + U$ are easily seen to be mutually equal. Hence, if $\boldsymbol{c} = \boldsymbol{mG}$ is the transmitted codeword and $\boldsymbol{r}$ is the received word such that $\mathrm{wt}(\boldsymbol{r} - \boldsymbol{c}) \leqslant \lfloor (s(G)_i - 1)/2 \rfloor$ then syndrome decoding correctly reproduces the ith digit m_i of the message $\boldsymbol{m}$ sent.

For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{N}^k$ we define the ordering $\geqslant$ by

$$\boldsymbol{x} \geqslant \boldsymbol{y} \quad \text{if} \quad x_i \geqslant y_i, \qquad \text{for all } i = 1,\cdots,k,$$

where the ordering $\geqslant$ in $x_i \geqslant y_i$ denotes the natural ordering in the integers. We call a vector $\boldsymbol{x} \in \mathbb{N}^k$ nonincreasing if $x_i \geqslant x_{i+1}$ for $i = 1,\cdots,k-1$.

By simultaneously permuting the message positions in the message words and the rows of a generator matrix G, we may obtain a generator matrix $\bar{G}$ for the code such that $\boldsymbol{s}(\bar{G})$ is nonincreasing, i.e., $s(\bar{G})_i \geqslant s(\bar{G})_{i+1}$ for $i = 1,\cdots, k-1$. 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.

B. Optimal Encoding

The separation vector defined by (1) depends upon the choice of a generator matrix for the code. But, fortunately every code has a so-called *optimal generator matrix* G^*, whose separation vector $\boldsymbol{s}(G^*)$ is componentwise larger than or equal to the separation vector $\boldsymbol{s}(G)$ of any other generator matrix G of the code (notation: $\boldsymbol{s}(G^*) \geqslant \boldsymbol{s}(G)$). This was shown in [2]. From [2] we mention the following two results, which we will need later on. For a linear $[n, k]$ code C and $\rho \in \{0,\cdots,n\}$ let $C(\rho)$ denote the set of codewords in C of weight less than or equal to ρ, i.e., $C(\rho) = \{\boldsymbol{c} \in C | \mathrm{wt}(\boldsymbol{c}) \leqslant \rho\}$.

Theorem 2 [2, ths. 4 and 6]: a) A generator matrix G of a linear $[n, k]$ code C is optimal if and only if for any $\rho \in \{1,\cdots,n\}$ a subset X of the rows of G exists such that the linear span $\langle C(\rho)\rangle$ of $C(\rho)$ equals the linear span $\langle X\rangle$ of X. b) For $\rho \in \{1,\cdots,n\}$, $\dim\langle C(\rho)\rangle - \dim\langle C(\rho-1)\rangle$ components of the separation vector of an optimal generator matrix G of a linear $[n, k]$ code C are equal to ρ.

Theorem 3 [2, ths. 5 and 6]: For a linear $[n, k]$ code C a minimal weight generator matrix G, i.e., a generator matrix of C with the minimal number of nonzero entries, is optimal and satisfies $\mathrm{wt}(G_{i*}) = s(G)_i$ for $i = 1,\cdots,k$, where G_{i*} denotes the ith row of G.

Hence the following definition makes sense.

Definition 2: 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 $[n, k, \boldsymbol{s}]$ for a linear code of length n, dimension k, and (nonincreasing) separation vector $\boldsymbol{s}$.

C. The Canonical Generator Matrix

Boyarinov and Katsman [1] have introduced a special form of a generator matrix, called a canonical form.

Definition 3: A generator matrix G of a linear $[n, k]$ code, whose nonincreasing separation vector $\boldsymbol{s}(G)$ has z distinct components $t_1 > t_2 > \cdots > t_z$ with multiplicities resp. $k_1, k_2,\cdots,k_z$, is called *canonical* if the submatrix consisting of the k rows of G and the first k columns of G is a k-by-k lower triangular partitioned matrix having unit matrices of order resp. k_1-by-k_1, k_2-by-k_2,$\cdots$, k_z-by-k_z on its diagonal. That is, G has the following form:

$$\left[\begin{array}{ccccc|c} I_{k_1} & 0 & \cdots & 0 & 0 & \\ G_{2,1} & I_{k_2} & \cdots & 0 & 0 & \\ \vdots & & \ddots & & \vdots & P \\ G_{z-1,1} & G_{z-1,2} & \cdots & I_{k_{z-1}} & 0 & \\ G_{z,1} & G_{z,2} & \cdots & G_{z,z-1} & I_{k_z} & \end{array}\right]. \tag{3}$$

Any generator matrix G of a code can be transformed into a canonical generator matrix G_{can} of the code, such that $\boldsymbol{s}(G_{\mathrm{can}}) \geqslant \boldsymbol{s}(G)$, by a number of elementary transformations on the rows of G, that are permutation and addition of rows and multiplication of rows by scalars (cf. [1], [4]). If we want to transform a generator matrix G into a systematic generator matrix G_{syst}, we cannot guarantee that $\boldsymbol{s}(G_{\mathrm{syst}}) \geqslant \boldsymbol{s}(G)$. For example [4], for $q = 2$,

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

has separation vector $\boldsymbol{s}(G) = (5, 4, 4, 4, 4)$. It is easy to see that it is impossible to transform G into a systematic

generator matrix G_{syst} such that $s(G_{syst}) \geq (5,4,4,4,4)$. Actually, it can be easily verified that a 5×10 binary systematic generator matrix with a separation vector of at least $(5,4,4,4,4)$ does not exist.

III. Bounds on the Length of LUEP Codes

A basic problem is to find LUEP codes with a given dimension and separation vector such that their length is minimal and hence their information rate is maximal.

Definition 4: For any prime power q, $k \in \mathbb{N}$, and $s \in \mathbb{N}^k$ we define $n_q(s)$ as the length of the shortest linear code over GF(q) of dimension k with a separation vector of at least s and $n_q^{ex}(s)$ as the length of the shortest linear code over GF(q) of dimension k with separation vector (exactly) s.

An $[n_q(s), k, s]$ code is called *optimal* if an $[n_q(s), k, t]$ code with $t \geq s$, $t \neq s$, does not exist. For any prime power q, $k \in \mathbb{N}$ and $s, t \in \mathbb{N}^k$, the functions $n_q(\cdot)$ and $n_q^{ex}(\cdot)$ satisfy the following properties.

$$n_q(s) \leq n_q^{ex}(s), \tag{4}$$

$$s \leq t \Rightarrow n_q(s) \leq n_q(t), \tag{5}$$

$$s \leq t \not\Rightarrow n_q^{ex}(s) \leq n_q^{ex}(t). \tag{6}$$

To illustrate (6), observe that $n_2^{ex}(5,4,4) = 8$ (cf. Table I) and $n_2^{ex}(5,4,3) = 9$, which can be seen by easy verification.

We now derive upper and lower bounds for these functions.

A. Upper Bounds

The following theorem provides a trivial upper bound for $n_q(\cdot)$ and $n_q^{ex}(\cdot)$ and an easy way to construct LUEP codes. Let "$|$" denote concatenation.

Theorem 4: For any prime power q, $k \in \mathbb{N}$, $v \in \mathbb{N}$, and an arbitrarily partitioned vector $(s_1|s_2|\cdots|s_v) \in \mathbb{N}^k$ we have

$$n_q^{ex}(s_1|s_2|\cdots|s_v) \leq \sum_{i=1}^{v} n_q^{ex}(s_i). \tag{7}$$

The same inequality holds for $n_q(\cdot)$ (replace $n_q^{ex}(\cdot)$ in (7) by $n_q(\cdot)$).

Proof: For $u = 1,\cdots,v$ let G_u be a generator matrix of a code with length $n_q^{ex}(s_u)$ and separation vector $s(G_u) = s_u$. Then $G = \text{diag}(G_1, G_2, \cdots, G_v)$ has separation vector $s(G) = (s_1|s_2|\cdots|s_v)$.

Corollary 5: For any prime power q, $k \in \mathbb{N}$, and $s \in \mathbb{N}^k$ we have

$$n_q^{ex}(s) \leq \sum_{i=1}^{k} s_i. \tag{8}$$

Proof: Apply Theorem 4 with $v = k$, and $G_u = [111\cdots1]$, 1 by s_u, for all $u = 1,\cdots,k$.

Hence for any $s \in \mathbb{N}^k$ it is possible to construct a k-dimensional code over GF(q) with separation vector s.

TABLE I
Separation Vectors of Binary Optimal LUEP 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	3	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, 333222222
12	10	2	3222222222
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, 444222222, 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
14	6	5	922222, 844444, 764444, 666644, 665552
14	7	4	8222222, 7444444, 6644442, 6544444, 5555444
14	8	4	72222222, 64444422, 63333332, 55444422, 54444444
14	9	4	622222222, 544442222, 533333222
14	10	3	5222222222, 4444222222

TABLE I
(*continued*)

n	k	d(n,k)	separation vector
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, 9666
15	5	7	11.2222, 10.4444, 96444, 88444, 86666
15	6	6	10.22222, 944444, 864444, 774444, 766662, 766644, 765554
15	7	5	9222222, 8444444, 7644444, 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

B. Lower Bounds

We start with a trivial, but useful, lower bound on $n_q(\cdot)$.

Theorem 6: For any prime power q, $k \in \mathbb{N}$, and $\boldsymbol{s} \in \mathbb{N}^k$ we have

$$n_q(s_1, s_2, \cdots, s_k) \geq n_q(s_1 - 1, s_2 - 1, \cdots, s_k - 1) + 1. \tag{9}$$

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

Theorem 7: For $q = 2$, any $k \in \mathbb{N}$, and $(s_1, s_2, \cdots, s_k) \in \mathbb{N}^k$ we have

$$n_2(s_1, s_2, \cdots, s_k) \geq n_2\left(2\left\lfloor \frac{s_1 + 1}{2} \right\rfloor, 2\left\lfloor \frac{s_2 + 1}{2} \right\rfloor, \cdots, 2\left\lfloor \frac{s_k + 1}{2} \right\rfloor\right) - 1. \tag{10}$$

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

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

Theorem 8: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $\boldsymbol{s} \in \mathbb{N}^k$ we have

$$n_q(s_1, s_2, \cdots, s_k) \geq 1 + n_q(s_1, s_2, \cdots, s_{k-1}). \tag{11}$$

Proof: By deleting the column $\boldsymbol{e}_k := (0, 0, \cdots, 0, 1)^T$ and the kth row from an optimal canonical (cf. Definition 3) generator matrix of a linear $[n = n_q(\boldsymbol{s}), k]$ code over GF(q) with a separation vector of at least $\boldsymbol{s}$, we obtain a generator matrix of an $[n - 1, k - 1]$ code with a separation vector of at least $(s_1, s_2, \cdots, s_{k-1})$.

Corollary 9: For any prime power q, k, $j \in \mathbb{N}$, $1 \leq j \leq k$, and nonincreasing $\boldsymbol{s} \in \mathbb{N}^k$ we have

$$n_q(s_1, s_2, \cdots, s_k) \geq j + n_q(s_1, s_2, \cdots, s_{k-j}). \tag{12}$$

Corollary 10: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $\boldsymbol{s} \in \mathbb{N}^k$ we have

$$n_q(s_1, s_2, \cdots, s_k) \geq s_1 + k - 1. \tag{13}$$

For $s_1 = s_2 = \cdots = s_k$ Corollary 10 reduces to the Singleton bound (cf. [7, ch. 1, th. 11]).

Theorem 11: For any prime power q and $k \in \mathbb{N}$, and any $v \in \mathbb{N}$ and nonincreasing $\boldsymbol{s} \in \mathbb{N}^k$ such that s_{v-1} is strictly larger than s_v and

$$\sum_{i=v}^{k} s_i \leq n_q^{ex}(\boldsymbol{s}) - 1, \tag{14}$$

we must have

$$n_q^{ex}(s_1, \cdots, s_k) \geq 1 + n_q(s_1 - 1, \cdots, s_{v-1} - 1, s_v, \cdots, s_k). \tag{15}$$

Proof: Let $v \in \mathbb{N}$ and a nonincreasing vector $\boldsymbol{s} \in \mathbb{N}^k$ be such that $s_{v-1} > s_v$ and (14) holds. Let G be a minimal weight generator matrix of an $[n = n_q^{ex}(\boldsymbol{s}), k, \boldsymbol{s}]$ code over GF(q). Because of (14) and Theorem 3, G has a column containing zero elements in the last $k - v + 1$ positions. By deleting this column from G we obtain a k-by-$(n - 1)$ matrix G', whose separation vector satisfies $\boldsymbol{s}(G') \geq (s_1 - 1, \cdots, s_{v-1} - 1, s_v, \cdots, s_k)$, since $s_{v-1} > s_v$.

Theorem 12: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $\boldsymbol{s} \in \mathbb{N}^k$ we have that

$$n_q^{ex}(s_1, \cdots, s_k) \geq s_i + n_q(\hat{s}_1, \cdots, \hat{s}_{i-1}, \hat{s}_{i+1}, \cdots, \hat{s}_k) \tag{16}$$

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

$$\hat{s}_j := \begin{cases} s_j - \lfloor (q-1)s_i/q \rfloor, & \text{for } j < i; \\ \lceil s_j/q \rceil, & \text{for } j > i, \end{cases} \tag{17}$$

where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x.

Proof: Let C be a linear $[n = n_q^{ex}(\boldsymbol{s}), k, \boldsymbol{s}]$ code over GF(q) and let G be a minimal weight generator matrix for C. By Theorem 3, $\mathrm{wt}(G_{i*}) = s_i$ for all $i = 1, \cdots, k$.

Fix $i \in \{1, \cdots, k\}$. Without loss of generality the first s_i columns of G have a 1 in the ith row. Deleting these first s_i columns and the ith row from G, we obtain a $(k - 1)$ by $(n - s_i)$ matrix $\hat{G}$. Clearly $\hat{G}$ has rank $(k - 1)$, for otherwise there would be a nontrivial linear combination of rows of $\hat{G}$ that equals $\mathbf{0}$, and hence the corresponding linear combination of rows of G would have a distance less than s_i to αG_{i*} for some $\alpha \in \mathrm{GF}(q) \setminus \{0\}$, a contradiction. Hence $\hat{G}$ is a generator matrix of an $[n - s_i, k - 1]$ code with separation vector $\hat{\boldsymbol{s}} := \boldsymbol{s}(\hat{G}) = (\hat{s}_1, \cdots, \hat{s}_{i-1}, \hat{s}_{i+1}, \cdots, \hat{s}_k)$.

Let $j \in \{1, \cdots, k\}$, $j \neq i$, and let $\boldsymbol{m} \in \mathrm{GF}(q)^k$ be such that $m_i = 0$, $m_j \neq 0$, and $\boldsymbol{c} := \boldsymbol{m}G = (\boldsymbol{c}_1 | \boldsymbol{c}_2)$, where $\boldsymbol{c}_1$ has length s_i, satisfies $\mathrm{wt}(\boldsymbol{c}_2) = \hat{s}_j$. Since $m_j \neq 0$, we have that

$$\mathrm{wt}(\boldsymbol{c}_1) + \hat{s}_j \geq s_j. \tag{18}$$

Furthermore, for some $\alpha \in \mathrm{GF}(q) \setminus \{0\}$ at least $\lceil \mathrm{wt}(c_1)/(q-1) \rceil$ components of αc_1 equal 1, and hence

$$\mathrm{wt}(G_{i*} - \alpha c) \leqslant s_i - \lceil \mathrm{wt}(c_1)/(q-1) \rceil + \hat{s}_j. \quad (19)$$

On the other hand we have that

$$\mathrm{wt}(G_{i*} - \alpha c) \geqslant \max\{s_i, s_j\}. \quad (20)$$

The combination of (18), (19), and (20) yields (16) and (17).

Lemma 13: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $s \in \mathbb{N}^k$ a linear $[n_q(s), k]$ code over $\mathrm{GF}(q)$ with a nonincreasing separation vector s^* such that $s \leqslant s^* \leqslant s_1\mathbf{1}$ ($s_1\mathbf{1}$ denotes the k-vector with all components equal to s_1) exists, i.e., $n_q^{ex}(s^*) = n_q(s)$.

Proof: Let G be a minimal weight generator matrix of an $[n = n_q(s), k]$ code with separation vector $s(G) \geqslant s$. 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 $s(G') \geqslant s$ and $s(G')_1 = s(G)_1 - 1$. We may transform G' into a minimal weight generator matrix G'' spanning the same linear subspace.

Now, we repeat the above procedure until we obtain a k-by-n matrix G^* such that $s \leqslant s(G^*) \leqslant s_1\mathbf{1}$.

The combination of Theorem 12 and Lemma 13 gives the following corollary.

Corollary 14: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $s \in \mathbb{N}^k$, $n_q(s)$ satisfies the inequalities

$$n_q(s_1, \cdots, s_k) \geqslant s_1 + n_q(\lceil s_2/q \rceil, \cdots, \lceil s_k/q \rceil), \quad (21)$$

$$n_q(s_1, \cdots, s_k) \geqslant \sum_{i=1}^{k} \lceil s_i/q^{i-1} \rceil. \quad (22)$$

For $s_1 = s_2 = \cdots = s_k$, Corollary 14 reduces to the Griesmer bound (cf. [7, ch. 17, th. 24]). Deleting the $\lceil \cdot \rceil$ brackets in (22) we obtain an analog of the Plotkin bound (cf. [7, ch. 2, th. 1]) for LUEP codes.

Lemma 13 also implies the following corollary.

Corollary 15: For any prime power q, $k \in \mathbb{N}$, and nonincreasing $s \in \mathbb{N}^k$ we have

$$n_q(s) = \min\left\{ n_q^{ex}(s') \mid s \leqslant s' \leqslant s_1\mathbf{1} \right\}. \quad (23)$$

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

Katsman [6] has shown Corollary 14 for $q = 2$. In many cases a combination of Corollary 15 and the bounds on $n_q^{ex}(\cdot)$ give better results than Corollary 14. For example, Corollary 14 yields that $n_2(5,4,3,3,3,3) \geqslant 11$, while a combination of Corollary 15 and the Theorems 6 and 12 yield that $n_2(5,4,3,3,3,3) \geqslant 12$ (using the values of Table I for $n \leqslant 10$). Actually $n_2(5,4,3,3,3,3) = 12$ (cf. Table I). Another interesting fact is the observation that Theorem 12 gives better results than the bound in [6], i.e., Theorem 12 for $i = 1$ and $q = 2$. For example, Theorem 12 yields that

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

and

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

Table I provides the separation vectors of the binary optimal LUEP codes of length less than or equal to 15, except for two. n denotes the length of the code, k denotes the dimension, and $d(n, k)$ denotes the maximal minimum distance of a binary linear code of length n and dimension k. The brackets and commas commonly appearing in separation vectors have been deleted. Only in the cases where a component of a separation vector is larger than 9, it is followed by a point $(\cdot)$. The construction of the codes in Table I can be found in [4]. In Table I there are two open places. 1) Does a binary $[15, 8, s]$ code with $(6,5,3,3,3,3,3,3) \leqslant s \leqslant (6,5,4,4,4,3,3,3)$ exist? 2) Does a $[15,8,(5,5,5,x,4,4,4,4)]$ binary code with $x \in \{4,5\}$ exist? A $[15,8,(5,5,5,5,5,4,4,4,3)]$ binary code exists (cf. [4]).

In [4] also a number of constructions of optimal LUEP codes and methods for combining (LUEP) codes to obtain LUEP codes of larger length are given.

IV. Cyclic UEP Codes

A. The Separation Vector of a Cyclic UEP Code

A cyclic $[n, k]$ code over $\mathrm{GF}(q)$ is the direct sum of a number of minimal ideals in the residue class ring $\mathrm{GF}(q)[x]/(x^n - 1)$ of polynomials in x over $\mathrm{GF}(q)$ modulo $(x^n - 1)$ (cf. [7, ch. 8, sec. 3]).

Theorem 16, [2]: For a cyclic code C that is the direct sum of v minimal ideals, an ordering $M_1, M_2, \cdots, M_v$ of generator matrices of these minimal ideals exist such that

$$G = \begin{bmatrix} M_1 \\ \hline M_2 \\ \hline \vdots \\ \hline M_v \end{bmatrix} \quad (24)$$

is an optimal generator matrix.

Proof: For $\rho \in \{1, \cdots, n\}$, $\langle C(\rho) \rangle$ is a cyclic code. Hence $\langle C(\rho) \rangle$ is the direct sum of a number of minimal ideals of $\mathrm{GF}(q)[x]/(x^n - 1)$. By applying Theorem 2a) we get the theorem.

The following corollaries are immediate consequences of Theorem 2 and 16.

Corollary 17: For a minimal ideal in $\mathrm{GF}(q)[x]/(x^n - 1)$ all components of the separation vector are mutually equal.

Corollary 18: For a cyclic code C with an optimal generator matrix G defined by formula (24) the ith and jth component of the separation vector $s = s(G)$ are equal if the ith and jth row of G are in the same minimal ideal of $\mathrm{GF}(q)[x]/(x^n - 1)$.

TABLE II
ALL BINARY CYCLIC UEP CODES OF LENGTH LESS THAN OR EQUAL TO 39

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	21
	13	0,1,5	7,4,4,4,4,4,4,4,4,4,4,4	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	21
	17	3,1,5,7	4,4,4,2,2,2,2,2,2,2,2,2,2,2	21
25	21	0,1	5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	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	27
	20	9,1	6,6,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	27
	21	0,9,1	3,3,3,2,2,2,2,2,2,2,2,2,2,2,2,2,2	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	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	≥ 29
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	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	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	6,6,6,6,6,6,6,6,4,4,4,4,4,4,4,4,4,4,4,4,4,4	≥ 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	≥ 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	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	35

TABLE II
(*continued*)

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	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	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	35
39	13	0,1	15,12,12,12,12,12,12,12,12,12,12,12,12	≥ 37
	14	13,3	14,14,6,6,6,6,6,6,6,6,6,6,6,6	≥ 35
	15	0,1,13	13,10,10,10,10,10,10,10,10,10,10,10,10,10,10	≥ 36
	15	13,0,3	13,13,3,3,3,3,3,3,3,3,3,3,3,3,3	≥ 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	≥ 35
	25	0,1,7	13,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4	39
	27	1,13,0,3	6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,3,3,3,3,3,3,3,3,3,3,3,3,3,3	≥ 37
	27	0,1,7,13	12,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	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

If the weight of the generator polynomial of a cyclic code C equals the minimum distance d of the code, then all components of the separation vector are mutually equal, since $C = \langle C(d) \rangle$ (cf. Theorem 2). 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. A number of separation vectors in Table II were computed in this way.

Theorem 19: For $i = 1, 2$ let $\mathcal{M}_i$ be a minimal ideal in $\mathrm{GF}(q)[x]/(x^n - 1)$ with minimum distance d_i and weight distribution $(A_j^{(i)})_{j=0}^n$ such that $\mathcal{M}_1 \neq \mathcal{M}_2$ and $d_1 \geq d_2$; let $(A_j)_{j=0}^n$ be the weight distribution of their direct sum $\mathcal{M}_1 \oplus \mathcal{M}_2$. Then the components of the separation vector of $\mathcal{M}_1 \oplus \mathcal{M}_2$ are all equal to the minimum distance d of $\mathcal{M}_1 \oplus \mathcal{M}_2$ 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 $\min\{j | A_j^{(2)} < A_j\}$.

Proof: If $d < d_2$ or if $d = d_2$ and $A_d^{(2)} < A_d$ then a sum of an element in $\mathcal{M}_1 \setminus \{\mathbf{0}\}$ and one in $\mathcal{M}_2 \setminus \{\mathbf{0}\}$ 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 $\mathcal{M}_1 \setminus \{\mathbf{0}\}$ and one in $\mathcal{M}_2 \setminus \{\mathbf{0}\}$ exists such that its weight equals j; if $A_j^{(2)} = A_j$ it does not. Combining these observations with Theorem 16 and Corollary 18 proves the theorem.

B. A Majority Decoding Method for Certain Binary Cyclic UEP Code Classes

In this section we discuss certain classes of binary cyclic UEP codes which can be decoded by a majority decoding method. It is easy to implement this method and it is very useful whenever the number of independent votes on each message digit equals (or is not much less than) the separa-

tion component corresponding to that message position. For a cyclic $[n, k]$ code, we number the message positions from 0 to $k-1$, the code positions from 0 to $n-1$.

Unlike the usual definition [9, p. 6] we define an orthogonal check set of size δ on a code position j of a linear $[n, k]$ code C to be δ subsets $B_1^{(j)}, B_2^{(j)}, \cdots, B_\delta^{(j)}$ of $\{0,1,\cdots,n-1\}$ that satisfy the following three conditions.

$$B_r^{(j)} \cap B_s^{(j)} = \varnothing, \qquad \text{for } r \neq s, \tag{25}$$

$$\bigcup_{r=1}^{\delta} B_r^{(j)} \subset \{0,1,\cdots,n-1\} \setminus \{j\}, \tag{26}$$

$$c_j = \sum_{p \in B_r^{(j)}} c_p, \qquad \text{for all } \boldsymbol{c} \in C \text{ and all } r \in \{1,\cdots,\delta\}. \tag{27}$$

We define the weight of a check $B_r^{(j)}$ as the number of elements in $B_r^{(j)}$. We extend this definition of orthogonal check sets on code positions to orthogonal check sets on message positions. For a binary linear $[n, k, s]$ code C that uses an optimal generator matrix G for its encoding, we define an orthogonal check set of size δ on a message position j to be δ subsets $D_1^{(j)}, D_2^{(j)}, \cdots, D_\delta^{(j)}$ of $\{0,1,\cdots,n-1\}$ and δ linear functions $f_1, f_2, \cdots, f_\delta$ of $m_0, m_1, \cdots, m_{j-1}$ that satisfy the following three conditions, similar to (25)–(27),

$$D_r^{(j)} \cap D_s^{(j)} = \varnothing, \qquad \text{for } r \neq s, \tag{28}$$

$$\bigcup_{r=1}^{\delta} D_r^{(j)} \subset \{0,1,\cdots,n-1\}, \tag{29}$$

$$m_j = \sum_{p \in D_r^{(j)}} c_p + f_r(m_0, m_1, \cdots, m_{j-1}),$$

$$\text{for all } \boldsymbol{c} \in C \text{ and all } r \in \{1,\cdots,\delta\}. \tag{30}$$

If we have an orthogonal check set of size δ_j on message position j for $j = 0,1,\cdots,k-1$ such that $\delta_0 \geq \delta_1 \geq \cdots \geq \delta_{k-1}$, then we perform the following decoding rule.

Whenever we receive a vector $\boldsymbol{u}$, we estimate the message digit m_j for $j = 0,1,\cdots,k-1$, starting with $j = 0$ and increasing j by 1 until j equals $k-1$. We estimate m_j by $\hat{m}_j$ which is defined as the majority of the votes

$$\sum_{p \in D_r^{(j)}} u_p + f_r(\hat{m}_0, \hat{m}_1, \cdots, \hat{m}_{j-1}), \qquad r = 1,\cdots,\delta_j. \tag{31}$$

It is easily seen that this majority decoding method guarantees the correct estimation of the jth message digit (i.e., $\hat{m}_j = m_j$), whenever the Hamming distance between the transmitted codeword $\boldsymbol{c} = \boldsymbol{m}G$ and the received word $\boldsymbol{u}$ is less than or equal to $\lfloor(\delta_j - 1)/2\rfloor$. For δ_j being even, an error in the jth message position is detected whenever the error pattern has Hamming weight $\delta_j/2$. Hence, the jth component s_j of the separation vector of the code C satisfies $s_j \geq \delta_j$ $(j = 0,1,\cdots,k-1)$.

Now, we consider a special class of binary cyclic codes. Let $p(x)$ and $q(x)$ be two irreducible polynomials in GF(2)$[x]$ 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. These strong restrictions are sufficient to build codes that have orthogonal check sets on their message positions of "large" size. Define C to be the binary cyclic $[n = T_pT_q, k_p + k_q]$ 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 first k_p components of the separation vectors $\boldsymbol{s} = \boldsymbol{s}((M_p^T|M_q^T)^T$ and $\boldsymbol{s}^* = \boldsymbol{s}((M_p^T|M_q^T|M_0^T)^T)$ of resp. C and C^*, where M_p, M_q, and M_0 are generator matrices of the minimal ideals in GF(2)$[x]/(x^n+1)$ with check polynomials resp. $p(x)$, $q(x)$, and $(x+1)$.

Theorem 20: a) The separation vector $\boldsymbol{s}$ of the code C defined above satisfies

$$s_i \geq \begin{cases} T_q\delta_p + 1, & \text{if } C_q \text{ is an even-weight code,} \\ & \text{for } i = 0,1,\cdots,k_p-1; \\ T_q\delta_p, & \text{otherwise, for } i = 0,1,\cdots,k_p-1. \end{cases}$$

b) The separation vector $\boldsymbol{s}^*$ of the code C^* defined above satisfies

$$s_i^* \geq T_q\delta_p, \qquad \text{for } i = 0,1,\cdots,k_p-1. \tag{33}$$

Proof: a) Without loss of generality we consider the first message digit m_0. Let G_p and $G_q = [\,y_0|\,y_1|\cdots|\,y_{T_q-1}]$ be systematic generator matrices of C_p resp. C_q. Without loss of generality the first column of G_p equals $\boldsymbol{e}_0 = (1,0,0,\cdots,0)^T$.

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 $\boldsymbol{a}_i^{(1)}, \boldsymbol{a}_i^{(2)}, \cdots, \boldsymbol{a}_i^{(2w)}$, $i = 1,\cdots,\delta_p$ of G_p such that

$$\boldsymbol{a}_i^{(1)} + \boldsymbol{a}_i^{(2)} + \cdots + \boldsymbol{a}_i^{(2w)} = \boldsymbol{e}_0 \tag{34}$$

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

$$G = \left[\begin{array}{c|c|c|c|c} G_p & G_p & \cdots & G_p & \\ \hline G_q & G_q & G_q & \cdots & G_q \end{array}\right] \tag{35}$$

is an optimal generator matrix of C. Since the greatest common division (gcd) $(T_p, T_q) = 1$, any pair $(\boldsymbol{x}^T|\,\boldsymbol{y}^T)^T$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are columns of resp. G_p and G_q, occurs exactly once as a column of G. By combining this fact with formula (34), we get that for any $i \in \{1,\cdots,\delta_p\}$ and any $j \in \{0,\cdots,T_q-1\}$ the equality

$$\begin{bmatrix} \boldsymbol{a}_i^{(1)} \\ \boldsymbol{y}_j \end{bmatrix} + \begin{bmatrix} \boldsymbol{a}_i^{(2)} \\ \boldsymbol{y}_j \end{bmatrix} + \cdots + \begin{bmatrix} \boldsymbol{a}_i^{(2w)} \\ \boldsymbol{y}_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_0 \\ \boldsymbol{0} \end{bmatrix} \tag{36}$$

holds, where the $2w\delta_pT_q$ columns occuring in the left-hand side (LHS) in (36) are mutually different columns of G.

Eq. (36) implies $T_q\delta_p$ orthogonal checks on the message digit m_0. (In this case, the linear functions $f_r(r =$

$1,\cdots,T_q\delta_p$) in formula (30) are all equal to the zero mapping.) If C_q is an even-weight code, then $m_0 = c_0 + c_{T_p} + c_{2T_p} + \cdots + c_{(T_q-1)T_p}$ is an additional check for m_0, orthogonal to the $T_q\delta_p$ previous ones.

b) Follows as in the proof of a).

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 s.

Theorem 21: a) If $\mathrm{wt}((x^{T_q}+1)/(q(x))$ is even and $T_q\delta_p + 1 > T_p(\delta_q + 1)$, then the separation vector s of the $[T_pT_q, k_p + k_q]$ code with check polynomial $p(x)q(x)$ satisfies

$$s_i \geq \begin{cases} T_q\delta_p + 1, & \text{for } i = 0,\cdots,k_p - 1; \\ T_p(\delta_q + 1), & \text{for } i = k_p,\cdots,k_p + k_q - 1. \end{cases} \tag{37}$$

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

$$s_i \geq \begin{cases} T_q\delta_p, & \text{for } i = 0,\cdots,k_p - 1; \\ T_p(\delta_q + 1), & \text{for } i = k_p,\cdots,k_p + k_q - 1. \end{cases} \tag{38}$$

Proof: a) For $i = 0,\cdots,k_p - 1$, (37) was shown in Theorem 20. Without loss of generality we consider the message digit m_{k_p}. For $j = 0,1,\cdots,T_p - 1$, m_{k_p} equals

$$m_{k_p} = c_{jT_q} + \sum_{i=0}^{k_p-1} m_i G_{i,jT_q}, \tag{39}$$

where G is the matrix of (35). If an error pattern of weight less than or equal to $\lfloor T_p(\delta_q + 1)/2 \rfloor$ occurs, then the message digits $m_0,\cdots,m_{k_p-1}$ are correctly decodable, since $T_q\delta_p + 1 > T_p(\delta_q + 1)$. If we fill in these values of $m_0,\cdots,m_{k_p-1}$ in (39), then the $T_p(\delta_q + 1)$ checks on m_{k_p} obtained from (36) and (39) are mutually orthogonal (i.e., satisfy formula (28)–(30)). Hence $s_{k_p} \geq T_p(\delta_q + 1)$.

b) Follows as in a).

Example 1: 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 21 the [35,7] code C with check polynomial $p(x)q(x)$ has a separation vector s which satisfies $s \geq (16,16,16,14,14,14,14)$.

Fig. 1 shows an optimal generator matrix for C (points $(\cdot)$ should be read as zeros (0)), together with two rows of characters, corresponding with the orthogonal checks on m_0 and m_3 given below. Note that, we did not take G_p and G_q to be systematic as we did in the proof of Theorem 20,

```
pabcdefpghifjkpelbkmnpjchnaopmhlogd

111.1..111.1..111.1..111.1..111.1..
.111.1..111.1..111.1..111.1..111.1.
..111.1..111.1..111.1..111.1..111.1
11...11...11...11...11...11...11...
.11...11...11...11...11...11...11..
..11...11...11...11...11...11...11.
...11...11...11...11...11...11...11

ABCD EFAB DGEF BCDG FABC GEFA CDGE
```

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

because this is not necessary, but only convenient for the proof of Theorem 20. The generator matrix in Fig. 1 directly shows that C_p and C_q are cyclic codes. The message bit m_0 is equal to the following sixteen orthogonal checks:

$$\begin{aligned}
&a\colon c_1 + c_{26} \quad f\colon c_6 + c_{11} \quad k\colon c_{13} + c_{18} \\
&b\colon c_2 + c_{17} \quad g\colon c_8 + c_{33} \quad l\colon c_{16} + c_{31} \\
&c\colon c_3 + c_{23} \quad h\colon c_9 + c_{24} \quad m\colon c_{19} + c_{29} \\
&d\colon c_4 + c_{34} \quad i\colon c_{10} + c_{30} \quad n\colon c_{20} + c_{25} \\
&e\colon c_5 + c_{15} \quad j\colon c_{12} + c_{22} \quad o\colon c_{27} + c_{32} \\
&p\colon c_0 + c_7 + c_{14} + c_{21} + c_{28}.
\end{aligned}$$

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

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

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 $(15,15,15,7,7,7,7)$. For C^*, $a, b, c,\cdots,o$ are fifteen orthogonal checks on m_0; $A, B, C,\cdots,G$ are seven orthogonal checks on m_3. For the message bit m_7 we have the following seven checks:

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

We can extend Theorem 20 to codes with a check polynomial that is a product of more than two irreducible polynomials in GF(2)[x].

Theorem 22: For $i = 1,\cdots,v$ let $p_i(x)$ be an irreducible polynomial in GF(2)[x] of degree k_i and exponent T_i such that gcd $(T_i, T_j) = 1$ 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 s that satisfies

$$s_j \geq n\delta_i/T_i, \tag{40}$$

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

Proof: Analogous to Theorem 20.

In many cases we can do much better than formula (40) by adding other checks, e.g., checks like the ones in (39). This will be shown in the next example.

Example 2: Take $p(x) = x^3 + x + 1$, $q(x) = x^2 + x + 1$, and $r(x) = x^4 + x^3 + x^2 + x + 1$. $T_p = 7$, $T_q = 3$, $T_r = 5$. 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 respectively

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

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 $(x^T | y^T | z^T)^T$, where x, y, and z are columns of resp. P, Q, and R, occurs exactly once as a column of G. Now for any $i \in \{0,\cdots,T_q - 1\}$ and $j \in \{0,\cdots,T_r - 1\}$ we have

$$\begin{bmatrix} P_{*1} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*5} \\ Q_{*i} \\ R_{*j} \end{bmatrix} = \begin{bmatrix} P_{*2} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*3} \\ Q_{*i} \\ R_{*j} \end{bmatrix}$$

$$= \begin{bmatrix} P_{*4} \\ Q_{*i} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*6} \\ Q_{*i} \\ R_{*j} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{41}$$

This implies $T_qT_r\delta_p = 45$ orthogonal checks on m_0. These checks do not contain the fifteen code digits $\{c_j | j \equiv 0 \bmod 7\}$, which correspond to the columns of G given in Fig. 2.

From Fig. 2 it is easy to see that

$$a\colon\ c_0 + c_{21} + c_{42} + c_{49} + c_{98}$$

$$b\colon\ c_7 + c_{14} + c_{63} + c_{70} + c_{91}$$

$$c\colon\ c_{28} + c_{35} + c_{56} + c_{77} + c_{84}$$

are three additional orthogonal checks on m_0, which are orthogonal to the 45 checks implied by formula (41). Hence we have 48 orthogonal checks on m_0. Analogously we can find 48 orthogonal checks on m_1 and m_2.

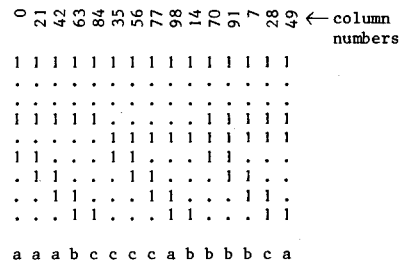

Fig. 2. Columns G_{*j} of G where $j \equiv 0 \bmod 7$.

For any $i \in \{0,\cdots,T_p - 1\}$ and any $j \in \{0,\cdots,T_r - 1\}$ we have

$$\begin{bmatrix} P_{*i} \\ Q_{*1} \\ R_{*j} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*2} \\ R_{*j} \end{bmatrix} = e_3 = (0,0,0,1,0,0,0,0,0)^T. \tag{42}$$

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

From Fig. 3 it is easy to see that

$$a\colon\ c_0 + c_9 + c_{12} + c_{36} + c_{93}$$

$$b\colon\ c_{18} + c_{21} + c_{39} + c_{45} + c_{72}$$

$$c\colon\ c_{30} + c_{42} + c_{48} + c_{66} + c_{69}$$

$$d\colon\ c_{51} + c_{63} + c_{75} + c_{84} + c_{102}$$

$$e\colon\ c_{15} + c_{33} + c_{54} + c_{81} + c_{87}$$

$$f\colon\ c_3 + c_{57} + c_{90} + c_{96} + c_{99}$$

$$g\colon\ c_6 + c_{24} + c_{27} + c_{60} + c_{78}$$

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

For any $i \in \{0,\cdots,T_p - 1\}$ and any $j \in \{0,\cdots,T_q - 1\}$ we have

$$\begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*1} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*2} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*3} \end{bmatrix} + \begin{bmatrix} P_{*i} \\ Q_{*j} \\ R_{*4} \end{bmatrix}$$

$$= e_5 = (0,0,0,0,0,1,0,0,0)^T. \tag{43}$$

This implies $T_pT_q\delta_r = 21$ orthogonal checks on m_5. These checks do not contain the 21 code digits $\{c_j | j \equiv 0 \bmod 5\}$. Since $s_0, s_1,\cdots,s_4 \geq 42$,

$$c_j + \sum_{i=0}^{4} m_i G_{ij},$$

for $j \in \{0, 5, 10,\cdots,100\}$ build 21 additional checks on m_5 (cf. the proof of Theorem 21). Hence we have 42 orthogonal checks on m_5. Analogously we can find 42 checks on m_6, m_7, and m_8.

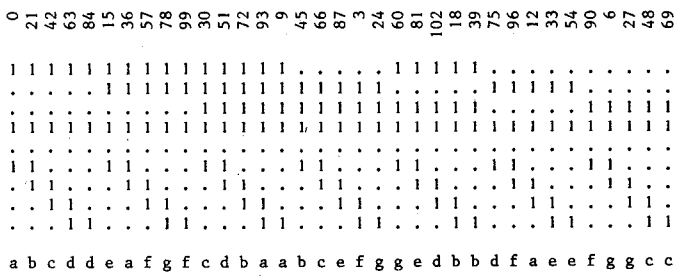

Fig. 3. Columns G_{*j} of G where $j \equiv 0 \bmod 3$.

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 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.

Theorem 23: For the primitive polynomials $p(x), q(x) \in \mathrm{GF}(2)[x]$ of degrees resp. k_p and k_q such that $k_p > k_q$ and gcd $(k_p, k_q) = 1$, 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 $\boldsymbol{s}$ that satisfies

$$s_i = \begin{cases} (2^{k_q} - 1)(2^{k_p - 1} - 1), & \text{for } i = 0, \cdots, k_p - 1 \\ (2^{k_p} - 1)(2^{k_q - 1} - 1), & \text{for } i = k_p, \cdots, k_p + k_q. \end{cases} \tag{44}$$

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 (also called a maximal-length feedback shift register code (cf. [7, p. 31]), i.e., all elements of $\mathrm{GF}(2)^{k_p} \setminus \{0\}$ 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 gcd $(k_p, k_q) = 1$, we may apply Theorem 20 b) to the first k_p message bits as well as to the message bits $m_{k_p}, \cdots, m_{k_p + k_q - 1}$.

Furthermore

$$\mathrm{wt}\left((x^n + 1)/p(x) + \sum_{i=0}^{n-1} x^i\right) = (2^{k_q} - 1)(2^{k_p - 1} - 1)$$

and

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

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

These observations imply (44) for $i = 0, 1, \cdots, k_p + k_q - 1$. From the observations above it also follows that

$$\begin{aligned} s_{k_p + k_q} &= \min\{(2^{k_p} - 1)(2^{k_q} - 1), (2^{k_p} - 1) \\ &\quad \cdot (2^{k_q - 1} - 1), (2^{k_q} - 1)(2^{k_p - 1} - 1)\} \\ &= (2^{k_p} - 1)(2^{k_q - 1} - 1), \end{aligned}$$

since $k_p > k_q$.

Dyn'kin and Togonidze [3] mentioned Theorem 23 without a proof.

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

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

is an optimal generator matrix for the [21, 6] cyclic code C with check polynomial $(x + 1)p(x)q(x)$. $\boldsymbol{s}(G) = (9, 9, 9, 7, 7, 7)$. m_0, m_3, and m_5 have the following checks:

$$\begin{aligned} m_0 &= c_1 + c_{19} = c_2 + c_{17} = c_3 + c_9 \\ &= c_4 + c_{13} = c_5 + c_8 = c_6 + c_{18} \\ &= c_{10} + c_{16} = c_{11} + c_{20} = c_{12} + c_{15}, \\ m_3 &= c_1 + c_8 = c_2 + c_{16} = c_4 + c_{11} \\ &= c_5 + c_{19} = c_7 + c_{14} \\ &= c_{10} + c_{17} = c_{13} + c_{20}, \\ m_5 &= c_0 + c_1 + c_2 + m_0 + m_2 \\ &= c_3 + c_4 + c_5 + m_0 \\ &= c_6 + c_7 + c_8 + m_1 + m_2 \\ &= c_9 + c_{10} + c_{11} + m_2 \\ &= c_{12} + c_{13} + c_{14} + m_0 + m_1 + m_2 \\ &= c_{15} + c_{16} + c_{17} + m_1 \\ &= c_{18} + c_{19} + c_{20} + m_0 + m_1. \end{aligned}$$

The generator matrix G' for this code, whose rows are the cyclic shifts of the generator polynomial, has separation vector $\boldsymbol{s}(G') = (7, 7, 7, 7, 7, 7)$. A binary [21, 6] code has a minimum distance of at most 8 (cf. [5]).

We can also extend Theorem 23 to the following theorem. The proof is analogous to that of Theorem 23.

Theorem 24: For the primitive polynomials $p_i(x) \in \mathrm{GF}(2)[x]$, $i = 1, \cdots, v$ of degrees resp. k_i, $i = 1, \cdots, v$ such that $k_1 > k_2 > \cdots > k_v$ and gcd $(k_i, k_j) = 1$ for all i, j, $i \neq j$, the binary cyclic

$$\left[\prod_{i=1}^{v} (2^{k_i} - 1), 1 + \sum_{i=1}^{v} k_i\right]$$

code with check polynomial

$$(x + 1)\prod_{i=1}^{v} p_i(x)$$

has a separation vector $\boldsymbol{s}$ which satisfies

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

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

$$s_i = (2^{k_v - 1} - 1)\prod_{u=1}^{v-1} (2^{k_u} - 1)$$

for $i = \sum_{u=1}^{v} k_u$.

Table II contains the parameters of all binary cyclic UEP codes of length less than or equal to 39. In this table the exponents $i, j, k, \cdots$ of a primitive nth root of unity α are given for each code of length n such that $\alpha^i, \alpha^j, \alpha^k, \cdots$ 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 $\{\alpha^i | i \in C_5 \cup C_0 \cup C_3\}$, 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 $i, j, k, \cdots$, then the separation vector equals $s((M_i^T | M_j^T | M_k^T | \cdots)^T)$, where M_t $(t = i, j, k, \cdots)$ denotes a generator matrix of the minimal ideal in $GF(2)[x]/(x^n + 1)$ with nonzeros $\{\alpha^y | y \in C_t\}$. In the above example $s((M_5^T | M_0^T | M_3^T)^T) = (5, 5, 3, 3, 3, 3, 3)$.

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.

Acknowledgment

This paper is based on the author's Master's thesis [4], written under supervision of Professor J. H. van Lint at the Eindhoven University of Technology. The subject "unequal error protection" was brought to the author's attention by L. M. H. E. Driessen of Philips Research Laboratories, Eindhoven, The Netherlands.

The author wishes to thank Professor J. H. van Lint, Dr. H. C. A. van Tilborg, Professor J. M. Goethals, and L. M. H. E. Driessen for their stimulating comments during the preparation of this paper. Thanks are also due to the anonymous referees for their prompt and excellent reviews.

References

[1] I. M. Boyarinov and G. L. Katsman, "Linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 168–175, Mar. 1981.

[2] L. A. Dunning and W. E. Robbins, "Optimal encodings of linear block codes for unequal error protection," *Inform. Contr.*, vol. 37, pp. 150–177, 1978.

[3] V. N. Dyn'kin and V. A. Togonidze, "Cyclic codes with unequal symbol protection," *Problemy Peredach. Informatsii*, vol. 12, no. 1, pp. 24–28, 1976.

[4] W. J. van Gils, "On linear unequal error protection codes," Eindhoven Univ. Tech. EUT-Rep. 82-WSK-02, June 1982. (This report is available on request from the Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.)

[5] H. J. Helgert and R. D. Stinaff, "Minimum-distance bounds for binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 344–356, May 1973.

[6] G. L. Katsman, "Bounds on volume of linear codes with unequal information-symbol protection," *Problemy Peredach. Informatsii*, vol. 16, no. 2, pp. 25–32, 1980.

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, Vol. 16. Amsterdam: North Holland 1978.

[8] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. 13, pp. 600–607, Oct. 1967.

[9] J. L. Massey, *Threshold Decoding*. Cambridge, MA: MIT Press, 1963.

EXHIBIT B

Modified generalized concatenated codes and their application to the construction and decoding of LUEP codes

<< Results

3 Author(s) U. Dettmar Yan Gao U.K. Sorger U. Dettmar ; Yan Gao ; U.K. Sorger View All Authors

Abstract | Authors | Figures | References | Citations | Keywords

Abstract: Proposes a modification of generalized concatenated codes, which allows to construct some of the best known binary codes in a simple way. Furthermore, a large class of optimal linear unequal error protection codes (LUEP codes) can easily be generated. All constructed codes can be efficiently decoded by the Blokh-Zyablov-Zinov'ev algorithm if an appropriate metric is used.<>

Metadata

Abstract:
Proposes a modification of generalized concatenated codes, which allows to construct some of the best known binary codes in a simple way. Furthermore, a large class of optimal linear unequal error protection codes (LUEP codes) can easily be generated. All constructed codes can be efficiently decoded by the Blokh-Zyablov-Zinov'ev algorithm if an appropriate metric is used.<>

Published in: IEEE Transactions on Information Theory (Volume: 41, Issue: 5, Sep 1995)

Page(s): 1499 - 1503

Date of Publication: Sep 1995

ISSN Information:

INSPEC Accession Number: 5053389

DOI: 10.1109/18.412696

Publisher: IEEE

Sponsored by: IEEE Information Theory Society

Authors

References

Citations

Keywords

Related Articles

Back to Top

Download PDF

Download Citation

View References

Email

Print

Request Permissions

Export to Collabratec

Alerts

Download PDF

Download Citation

View References

Email

Print

Request Permissions

Export to Collabratec

Alerts

Authors

References

Citations

Keywords

Related Articles

Authors

References

Citations

Keywords

Related Articles

Back to Top

IEEE Account

Profile Information

Purchase Details

The author would also like to acknowledge stimulating discussions with O. Amrani and F.-W. Sun. Finally, the author wishes to thank Hagit Itzkowitz for her invaluable help.

REFERENCES

[1] A. D. Abbaszadeh and C. K. Rushforth, "VLSI implementation of a maximum likelihood decoder for the Golay (24, 12) code," *IEEE J. Selected Areas Commun.*, vol. 6, pp. 558–565, 1988.

[2] O. Amrani and Y. Be'ery, "Efficient bounded-distance decoding of the hexacode and associated decoders for the Leech lattice and the Golay code," in *Proc. IEEE Int. Symp. on Information Theory* (Trondheim, Norway, 1994), p. 400.

[3] O. Amrani, Y. Be'ery, and A. Vardy, "Bounded-distance decoding of the Leech lattice and the Golay code," *Lecture Notes Comput. Sci.*, vol. 781, pp. 236–247, 1993.

[4] O. Amrani, Y. Be'ery, A. Vardy, F.-W. Sun, and H. C. A. van Tilborg, "The Leech lattice and the Golay code: bounded-distance decoding and multilevel constructions" *IEEE Trans. Inform. Theory*, vol. 40, pp. 1030-1043, 1994.

[5] E. F. Assmus, Jr. and H. F. Mattson, Jr., "Algebraic theory of codes," Rep. 1, Contract F19628–69C0068, Air Force Cambridge Res. Labs., Bedford, MA, 1969.

[6] Y. Be'ery and B. Shahar, "VLSI architectures for soft decoding of the Leech lattice and the Golay codes," in *Proc. IEEE Int. Workshop on Microelectronics in Communications* (Interlaken, Switzerland, Mar. 1991).

[7] Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the Leech lattice," *IEEE J. Selected Areas Commun.*, vol. 7, pp. 959–967, 1989.

[8] Y. Be'ery and J. Snyders, "Optimal soft decision block decoders based on Fast Hadamard Transform," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 355–364, 1986.

[9] A.R. Calderbank, *Bandwidth Efficient Communication*, in preparation.

[10] J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 41–50, 1986.

[11] ——, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1988.

[12] M. V. Eyuboğlu and G. D. Forney, Jr., "Lattice and trellis quantization with lattice- and trellis-bounded codebooks—high-rate theory for memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 46–59, 1993.

[13] G. D. Forney, Jr., "Generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 125–131, 1966.

[14] ——, "Coset codes I: Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123-1151, 1988.

[15] ——, "Coset codes II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.

[16] ——, "A bounded distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Trans. Inform. Theory*, vol. 35, pp. 906–909, 1989.

[17] ——, "Density/length profiles and trellis complexity of lattices," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1753–1772, 1994.

[18] G. R. Lang and F. M. Longstaff, "A Leech lattice modem," *IEEE J. Selected Areas Commun.*, vol. 7, pp. 968–973, 1989.

[19] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 963–975, 1989.

[20] F.-W. Sun and H. C. A. van Tilborg, "More efficient bounded-distance decoding of the Golay code and the Leech lattice," in *Proc. IEEE Int. Symp. on Information Theory* (Trondheim, Norway, 1994), p. 399.

[21] ——, "The Leech lattice, the octacode, and decoding algorithms," *IEEE Trans. Inform. Theory*, vol. 41, no. 4, pp. 1097–1106, July 1995.

[22] A. Vardy, "A new sphere packing in 20 dimensions," *Inventiones Mathematicae*, vol. 121, no. 1, July 1995.

[23] A. Vardy and Y. Be'ery, "More efficient soft-decision decoding of the Golay codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 667–672, 1991.

[24] ——, "Maximum-likelihood decoding of the Leech lattice," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1435–1444, 1993.

Modified Generalized Concatenated Codes and their Application to the Construction and Decoding of LUEP Codes

Uwe Dettmar, Yan Gao, and Ulrich K. Sorger

***Abstract*—We propose a modification of generalized concatenated codes, which allows us to construct some of the best known binary codes in a simple way. Furthermore, a large class of optimal linear unequal error protection codes (LUEP codes) can easily be generated. All constructed codes can be efficiently decoded by the Blokh–Zyablov–Zinov'ev algorithm if an appropriate metric is used.**

***Index Terms*—Linear unequal error protection codes, generalized concatenated codes, multistage decoding.**

I. INTRODUCTION

Many of the best known codes can be constructed as Generalized Concatenated (GC) codes [2], [3]. Generally, the constructions of GC codes use different outer codes A_i of constant length n_a but only one inner code $B^{(1)}$ together with its partition. However, this restriction is not necessary. In this correspondence we construct GC codes consisting of outer codes A_i with different lengths $n_{a,i}$, and inner codes $B_j^{(1)}$ in the columns of the code matrix with different lengths $n_{b,j}$ and distances $d_{b,j}^{(i)}$ together with their partitions.

In Section II, modified GC codes are defined and a lower bound on their minimum distance, a designed minimum distance, is derived. Two examples of the construction of good binary codes as modified GC codes are given. In Section III, the modification is used to construct binary optimal linear unequal error protection (LUEP) codes. In Section IV, a decoding algorithm for the modified GC codes is presented, which allows decoding up to half their designed minimum distance. Moreover, decoding of the constructed LUEP codes up to half the separation vector is discussed.

For the sake of simplicity only binary codes are considered in this correspondence. An extension to the nonbinary case is straightforward.

II. DEFINITION OF MODIFIED GC CODES

GC codes are a generalization of concatenated codes defined by Forney [4]. The inner code is multiply partitioned, and this partitioning into subcodes is protected by different outer codes. Denote the m outer codes by $A_i = A_i(q_i; n_a, k_{a,i}, d_{a,i})$, where $q_i = 2^{r_i}$ is the alphabet size, n_a is the code length, $k_{a,i}$ is the number of information symbols, and $d_{a,i}$ is the Hamming distance of the code. Further, denote by $B^{(i)} = B^{(i)}(2; n_b, k_{b,i}, d_{b,i})$, for $i = 1, 2, \cdots, m$, the binary inner codes, where

$$B^{(i+1)} \subset B^{(i)} \quad \text{and} \quad k_b^{(i)} - k_b^{(i+1)} = \log_2(q_i).$$

By concatenating inner and outer codes we get a GC code with the

Manuscript received February 11, 1994; revised March 5, 1995. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, San Antonio, TX, 1993..

The authors are with the Institut für Netzwerk- und Signaltheorie, Technische Hochschule Darmstadt, D-64283 Darmstadt, Germany.

IEEE Log Number 9413060.

TABLE I
EXAMPLE 1

Inner codes				Outer codes			GC code
	$j=1,2,\ldots,8$	$j=9$	$j=10$				
$B_j^{(1)}$	$(8,4,4)$	$(4,1,4)$	$(7,3,4)$	A_1	$(2^3;10,7bit,8)$	$\mathcal{J}_1=\{1,\ldots,10\}$	$(75,11,32)$
$B_j^{(2)}$	$(8,1,8)$	$(4,0,\infty)$	$(7,0,\infty)$	A_2	$(2;8,4,4)$	$\mathcal{J}_2=\{1,\ldots,8\}$	
$B_j^{(1)}$	$(8,4,4)$	$(4,3,2)$	$(7,3,4)$	A_1	$(2^3;10,3,8)$	$\mathcal{J}_1=\{1,\ldots,10\}$	$(75,13,30)$
$B_j^{(2)}$	$(8,1,8)$	$(4,0,\infty)$	$(7,0,\infty)$	A_2	$(2;8,4,4)$	$\mathcal{J}_2=\{1,\ldots,8\}$	

parameters [3]

$$n = n_a n_b,$$
$$k = \sum_{i=1}^{m} k_{a,i} \log_2 (q_i)$$
$$d \geq \min \{d_{a,1} d_{b,1}, d_{a,2} d_{b,2}, \cdots, d_{a,m} d_{b,m}\}.$$

The lower bound on the minimum distance is derived as follows [3]: since the codes are linear, the minimum weight and distance are equal. If $a_1 \neq 0$, with $a_1 \in A_1$, then not less than $d_{a,1}$ codewords of the inner code $B^{(1)}$ are different from the zero word. However, this inner code has a minimum weight $d_{b,1}$ which leads to a minimum weight of $d_{a,1} d_{b,1}$ for the appropriate codeword of the GC code. Similar considerations hold for the other stages.

In the above definition the outer codes all have equal length n_a. However, this restriction is not necessary: denote the outer codes by

$$A_i = A_i(q_i;\, n_{a,i},\, k_{a,i},\, d_{a,i})$$

where the $n_{a,i}$ now can be different. Define a set of indices $\mathcal{J}_i$ with $|\mathcal{J}_i| = n_{a,i}$ and $j_{\max}$ so that

$$1 \leq j \leq j_{\max}, \quad \forall j \in \mathcal{J}_i$$

holds. The inner codes are given by

$$B_j^{(i)} = B_j^{(i)}(2;\, n_{b,j},\, k_{b,j}^{(i)},\, d_{b,j}^{(i)})$$

for $i = 1, 2, \cdots, m$ and $j = 1, \cdots, j_{\max}$, with

$$B_j^{(i+1)} \subseteq B_j^{(i)} \text{ and } k_{b,j}^{(i)} - k_{b,j}^{(i+1)} = \begin{cases} \log_2 (q_i), & \text{if } j \in \mathcal{J}_i \\ 0, & \text{if } j \notin \mathcal{J}_i. \end{cases}$$

We assume that $\cup_i \mathcal{J}_i = \mathcal{J}$, with $j \in \mathcal{J}$ for $j = 1, 2, \cdots, j_{\max}$, because in this case all inner codes are concatenated with some outer codes.

The GC code has the parameters

$$n = \sum_{j=1}^{j_{\max}} n_{b,j} \qquad k = \sum_{i=1}^{m} k_{a,i} \log_2 (q_i).$$

The minimum distance of the GC code is given by

$$d \geq \min_i \min_{\mathcal{S}_i} \sum_{j \in \mathcal{S}_i} d_{b,j}^{(i)} \tag{1}$$

where $\mathcal{S}_i \subseteq \mathcal{J}_i$ with $|\mathcal{S}_i| = d_{a,i}$.

The lower bound for the minimum distance is derived similarly to that for conventional GC codes: the minimum weight in stage i is attained if the positions where $a_i \neq 0$ coincide with codewords of the inner codes $B_j^{(i)}$ with the smallest minimum distances. This leads to (1).

To simplify this expression, we define by $\Pi(j)$ the permutation of the indices $j = 1, \cdots, n_{a,i}$ so that

$$d_{b,\Pi(1)} \leq d_{b,\Pi(2)} \leq \cdots \leq d_{b,\Pi(n_{a,i})}.$$

This results in

$$d \geq \min_{\forall i} \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)}^{(i)}.$$

TABLE II
CONSTRUCTION 1

Inner codes		Outer codes		
$j=1,\ldots,n_1$				
$B_j^{(1)}$	$(n_2, k_{21}+k_{22}, d_{21})$	A_1	$(2^{k_{21}}; n_1, k_{11}, s_1)$	$\mathcal{J}_1=\{1,\ldots,n_1\}$
$B_j^{(2)}$	(n_2, k_{22}, d_{22})	A_2	$(2^{k_{22}}; n_1, k_{12}, s_2)$	$\mathcal{J}_2=\{1,\ldots,n_1\}$

Example 1: The $(75, 11, 32)$ and the $(75, 13, 30)$ code from [5] can be constructed as given in Table I.

III. OPTIMAL LUEP CODES CONSTRUCTED AS GC CODES

Linear unequal error protection (LUEP) codes can be useful if different information symbols have different importance. In [1] and [6], van Gils proposed constructions for some special classes of LUEP codes (some of them based on product or concatenated codes). In [7], Zinov'ev investigated the application of GC codes on the construction of LUEP codes, but these constructions only work for composite code length, i.e., $n = n_a n_b$. The modified construction yields a large class of binary LUEP codes which contains most of van Gils constructions and which can be easily decoded.

The LUEP code is characterized by its separation vector [6].

Definition 1: For a linear (n, k) code C over the alphabet GF(q), the separation vector $\boldsymbol{s} = (s_1, s_2, \cdots, s_k)$ with respect to a generator matrix G of C, is defined as

$$s_i := \min\{\text{wt}\{\boldsymbol{m}G | \boldsymbol{m} \in \text{GF}(q)^k,\, m_i \neq 0\},\, i = 1, \cdots, k\}$$

where $\boldsymbol{m}$ is an information vector and wt$\{\cdot\}$ denotes the Hamming weight function.

We assume, without loss of generality, that $\boldsymbol{s}$ is nonincreasing, i.e., $s_i \geq s_j$ if $i < j$ $\forall i, j \in \{1, \cdots, k\}$. Note that this definition is different from that in [8] as it deals with information symbols instead of code symbols. The separation vector guarantees the correct interpretation of the ith information symbol whenever nearest neighbor decoding [9] is applied and not more than $\lfloor (s(G)_i - 1)/2 \rfloor$ errors have occurred in the transmitted codeword [10].

An $(n, k, \boldsymbol{s})$ code is called optimal if an $(n, k, \boldsymbol{t})$ code with $\boldsymbol{t} > \boldsymbol{s}$, i.e., $t_i \geq s_i$ $\forall i \in \{1, \cdots, k\}$ and $\exists j \in \{1, \cdots, k\}: t_j > s_j$, does not exist. Denote by $n(\boldsymbol{s})$ the length of the shortest linear binary code of dimension k with separation vector at least $\boldsymbol{s}$ and denote $n^{\text{ex}}(\boldsymbol{s})$ the length of the shortest linear binary code of dimension k with separation vector (exactly) $\boldsymbol{s}$. Van Gils [1], [11], has derived the following lower bounds on $n(\boldsymbol{s})$:

Theorem 1: For any $k \in \mathcal{N}$, and nonincreasing $\boldsymbol{s} \in \mathcal{N}^k$

$$n^{ex}(s_1, s_2, \cdots, s_k) \geq s_i + n(\hat{s}_1, \cdots, \hat{s}_{i-1}, \hat{s}_{i+1}, \cdots, \hat{s}_k) \tag{2}$$

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

$$\hat{s}_j := \begin{cases} s_j - \left\lfloor \frac{s_i}{2} \right\rfloor, & \text{for } j < i \\ \left\lceil \frac{s_j}{2} \right\rceil, & \text{for } j > i. \end{cases} \tag{3}$$

$\lceil x \rceil$ denotes the smallest integer larger than or equal to x.

TABLE III
CONSTRUCTION 2

Inner codes			Outer codes		
	$j=1,\ldots,n_1$	$j=n_1+1,\ldots,n_1+n'$			
$B_j^{(0)}$	$(n_2,n_2,1)$	$(n_2-k_2,n_2-k_2,1)$	A_1	$(2^{n_2-k_2};n_1+n',k_{11},\mathbf{s}_1)$	$\mathcal{J}_1=\{1,\ldots,n_1+n'\}$
$B_j^{(1)}$	(n_2,k_2,d_{22})	$(n_2-k_2,0,\infty)$	A_2	$(2^{k_2};n_1,k_{12},\mathbf{s}_2)$	$\mathcal{J}_2=\{1,\ldots,n_1\}$

TABLE IV
CONSTRUCTION 3

Inner codes			Outer codes		
	$j=1,\ldots,n_1$	$j=n_1+1$			
$B_j^{(1)}$	(n_2,k_2+1,d_2)	$(1,1,1)$	A_1	$(2^{k_2};n_1,k_{11},\mathbf{s}_1)$	$\mathcal{J}_1=\{1,\ldots,n_1\}$
$B_j^{(2)}$	$(n_2,1,n_2)$	$(1,1,1)$	A_2	$(2;n_1+1,k_{12},2\lceil s_2'/2\rceil)$	$\mathcal{J}_2=\{1,\ldots,n_1+1\}$

Theorem 2: For any $k\in\mathcal{N}$, and nonincreasing $\boldsymbol{s}\in\mathcal{N}^k$, $n(\boldsymbol{s})$ satisfies the inequalities

$$n(s_1,\cdots,s_k)\geq s_1+n\left(\left\lceil\frac{s_2}{2}\right\rceil,\cdots,\left\lceil\frac{s_k}{2}\right\rceil\right) \tag{4}$$

$$n(s_1,\cdots,s_k)\geq\sum_{i=1}^{k}\left\lceil\frac{s_i}{2^{i-1}}\right\rceil. \tag{5}$$

Construction 1: First we construct a two-level GC code as shown in Table II. A_1 and A_2 are LUEP codes with nonincreasing separation vectors

$$\boldsymbol{s}_1=(s_{11},s_{12},\cdots,s_{1k_{11}})\quad \boldsymbol{s}_2=(s_{21},s_{22},\cdots,s_{2k_{12}}).$$

As a special case, both A_1 and A_2, or one of them, may be chosen as equal error protection codes. If $d_{21}s_{1k_{11}}\geq d_{22}s_{21}$, then the GC code is a binary $(n_1n_2,\ k_{11}k_{21}+k_{12}k_{22},\ \boldsymbol{s})$ LUEP code, where

$$\boldsymbol{s}=(d_{21}s_{11}\mathbf{1}_{k_{21}},\cdots,d_{21}s_{1k_{11}}\mathbf{1}_{k_{21}},d_{22}s_{21}\mathbf{1}_{k_{22}},\cdots,d_{22}s_{2k_{12}}\mathbf{1}_{k_{22}})$$

where $s\mathbf{1}_{k_{2i}}$ denotes the k_{2i} vector with all components equal to s.

Obviously, [6, Construction 5] is a special case of the above construction.

If A_1 is an $(n,\ 1,\ n)$ repetition code, A_2 is an optimal $(n,\ k,\ \boldsymbol{s})$ LUEP code, $B_j^{(1)}$ and $B_j^{(2)}$ are Reed–Muller codes with parameters $(2^m,\ m+1,\ 2^{m-1})$ and $(2^m,\ 1,\ 2^m)$ an optimal LUEP code equivalent to the code of [6, Construction 1] is obtained. Choosing $B_j^{(1)}$ as the $(2,\ 2,\ 1)$ code and $B_j^{(2)}$ as the $(2,\ 1,\ 2)$ code, the above construction is equivalent to [6, Construction 3A].

Construction 2: The GC Code given in Table III has the parameters

$$n=n_1n_2+(n_2-k_2)n'$$
$$k=(n_2-k_2)k_{11}+k_2k_{12}$$

and

$$\boldsymbol{s}=(s_{11}\mathbf{1}_{(n_2-k_2)},\cdots s_{1k_{11}}\mathbf{1}_{(n_2-k_2)},\ d_2s_{21}\mathbf{1}_{k_2},\cdots,d_2s_{2k_{21}}\mathbf{1}_{k_2})$$

where $s_{1k_{11}}\geq d_2s_{21}$. If $B_j^{(1)}$ is a $(2,\ 2,\ 1)$ code and $B_j^{(2)}$ is a $(2,\ 1,\ 2)$ code for $j=1,\cdots,n_1$, we obtain all the LUEP codes of *Constructions A,C,E,F,I,J* and *K* from van Gils in [1] and a class of LUEP code which is better than the codes of [6, Construction 2] with the same code rate.

In fact, choosing for A_1 the $(n_1+n',\ 1,\ n_1+n')$ repetition code and for A_2 an optimal $(n_1,\ k,\ \boldsymbol{s})$ LUEP code, we get with the inner codes of length 2, as above, an optimal $(2n_1+n',\ 1+k,\ (n_1+n',\ 2\boldsymbol{s}))$ GC code. The optimality can be shown as follows:

TABLE V
EXAMPLE 2

Inner codes				Outer codes		
	$j=1$	$j=2,3$	$j=4$			
$B_j^{(1)}$	$(4,3,2)$	$(3,2,2)$	$(2,2,1)$	A_1	$(2^2;4,2,3)$	$\mathcal{J}_1=\{1,\ldots,4\}$
$B_j^{(2)}$	$(4,1,4)$	$(3,0,\infty)$	$(1,0,\infty)$	A_2	$(2;1,1,1)$	$\mathcal{J}_2=\{1\}$

TABLE VI
EXAMPLE 3

Inner codes				Outer codes		
	$j=1,2$	$j=3$	$j=4$			
$B_j^{(1)}$	$(4,3,2)$	$(3,2,2)$	$(2,2,1)$	A_1	$(2^2;4,2,3)$	$\mathcal{J}_1=\{1,\ldots,4\}$
$B_j^{(2)}$	$(4,1,4)$	$(3,0,\infty)$	$(1,0,\infty)$	A_2	$(2;2,2,1)$	$\mathcal{J}_2=\{1,2\}$

Proof: For the proof we use Theorem 1. First we show that the GC code is optimal in length

$$n^{ex}(n_1+n',\ 2\boldsymbol{s})\geq n_1+n'+n(\boldsymbol{s})\geq 2n_1+n'.$$

We now have to show that all codes with a greater separation vector also have greater length

$$\begin{aligned}n^{ex}(n_1+n'+1,\ 2\boldsymbol{s})&\geq n_1+n'+1+n(\boldsymbol{s})\\&\geq 2n_1+n'+1\\&>2n_1+n'\end{aligned}$$

and

$$\begin{aligned}n^{ex}(n_1+n',\ 2\boldsymbol{s}+\boldsymbol{u})&\geq n_1+n'+n\\&\quad\cdot\left(\left\lceil\frac{s_1+u_1}{2}\right\rceil,\cdots,\lceil s_k+u_k\rceil\right)\\&\geq 2n_1+n'+1.\end{aligned}$$

Construction 3: The construction is given in Table IV, where $\lceil\boldsymbol{s}_2/2\rceil$ denotes $\lceil s_{2i}/2\rceil$ for all $i=1,\ 2,\cdots,k_{12}$.

We suppose here that the outer code A_2 is the code A_2' : $(2;\ n_1,\ k_{12},\ \boldsymbol{s}_2')$ with an added overall parity bit. With $s_{1k_1}d_2'>n_2$ we obtain a new

$$\{n_1n_2+1,\ k_{11}k_2+k_{12},\ [s_1d_2\mathbf{1}_{k_2},\cdots,s_{k_1}d_2\mathbf{1}_{k_2}\ (n_2-1)\boldsymbol{s}_2'+2\lceil\boldsymbol{s}_2'/2\rceil]\}$$

LUEP code. Using the $(2,\ 2,\ 1)$ and the $(2,\ 1,\ 2)$ code as inner codes for $j=1,\cdots,n_1$, we obtain the LUEP codes [6, Construction 3B] of van Gils. If at the same time A_1 is a repetition code and A_2' is uncoded, the LUEP codes are the same as in [1, Construction B], i.e., they are optimal.

Choosing A_1 as a repetition code and A_2 as uncoded and the $(4,\ 3,\ 2)$ and $(4,\ 1,\ 4)$ codes as inner codes for $j=1,\cdots,n_1$, results in [1, Construction H] of van Gils.

In the following examples we show the construction of some codes from [1] as GC codes.

Example 2: The GC code given by Table V has the parameters $n=12,\quad k=6\quad \boldsymbol{s}=(555544)$.

Example 3: The GC code given by Table VI has the parameters $n=13,\quad k=6\quad \boldsymbol{s}=(555544)$.

Example 4: The GC code given by Table VII has the parameters $n=14,\quad k=7\quad \boldsymbol{s}=(5555444)$.

For these three codes, see [1, Construction M].

TABLE VII
EXAMPLE 4

Inner codes			Outer codes		
	$j=1,2,3$	$j=4$			
$B_j^{(1)}$	$(4,3,2)$	$(2,2,1)$	A_1	$(2^2;4,2,3)$	$\mathcal{J}_1=\{1,\ldots,4\}$
$B_j^{(2)}$	$(4,1,4)$	$(1,0,\infty)$	A_2	$(2;3,3,1)$	$\mathcal{J}_2=\{1,2,3\}$

TABLE VIII
EXAMPLE 5

Inner codes			Outer codes		
	$j=1,2,3$	$j=4$			
$B_j^{(1)}$	$(4,3,2)$	$(3,3,1)$	A_1	$(2^2;4,2,3)$	$\mathcal{J}_1=\{1,\ldots,4\}$
$B_j^{(2)}$	$(4,1,4)$	$(3,1,3)$	A_2	$(2;4,4,1)$	$\mathcal{J}_2=\{1,\ldots,4\}$

Example 5: The GC code given by Table VIII has the parameters $n=15,\quad k=8\quad \boldsymbol{s}=(55554443)$ (See [1, Construction R]).

IV. A Decoding Algorithm

GC codes can be decoded by the well-known Blokh–Zyablov–Zinov'ev (BZZ) algorithm up to half their designed minimum distance.

For decoding GC codes with different inner codes we use the BZZ algorithm together with an appropriate metric.

As shown in Section II, the designed minimum distance of the constructed codes is given by

$$d \geq \min_{\forall i} \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)}^{(i)}.$$

Define $\boldsymbol{c}$ to be the transmitted codeword, with $c_j \in B_j^{(1)}$ for $j=1,\cdots,j_{\max}$. Denote by $\check{a}_i$ the transmitted codeword of the outer code A_i with respect to $\boldsymbol{c}$, and by $\hat{a}_i$ the estimate for $\check{a}_i$ calculated by decoding the inner codes of stage i and mapping the result to symbols of the outer code alphabet. Let a_i be a codeword of the outer code A_i and $\mathcal{E}(a_i,\hat{a}_i)$ be the set of indices j such that $a_i \neq \hat{a}_i$. Define w to be the cardinality of $\mathcal{E}(a_i,\hat{a}_i)$. Denote by $\bar{\mathcal{E}}^*(a_i,\hat{a}_i)$ the $d_{a,i}-w$ components in

$$\bar{\mathcal{E}} = \{1,\cdots,n_{a,i}\}/\mathcal{E}(a_i,\hat{a}_i)$$

with the smallest $\alpha_j^{(i)}$.

An appropriate reliability function is given by the following definition:

Definition 2: Let $d_{b,j}^{(i)}$ be the minimum distance of the inner code $B_j^{(i)}$, and $d_H(r_j,b_j^{(i)})$ be the distance between the received word r_j and the estimated codeword $b_j^{(i)} \in B_j^{(i)}$. Then the reliability $\alpha_j^{(i)}$ for the jth position in $\hat{a}_i$ is given by

$$\alpha_j^{(i)} = \max\left[0,\ d_{b,j}^{(i)} - 2d_H(r_j,b_j^{(i)})\right].$$

This definition takes into account the different distances of the inner codes. It may be interpreted as the minimum number of additional errors that occurred in case of a wrong decision in the inner code. During the decoding process the following condition has to be checked:

$$\text{(Condition 1)} \qquad \sum_{j\in\mathcal{E}^*(a_i,\hat{a}_i)} \alpha_j^{(i)} > \sum_{j\in\mathcal{E}(a_i,\hat{a}_i)} \alpha_j^{(i)}. \qquad (6)$$

Decoding Algorithm: For each stage i, with $i=1,2,\cdots,m$, do:

1) Decode the received word $\boldsymbol{r}$ in the inner codes to get an estimated codeword $b_j^{(i)}$ of code $B_j^{(i)}$ for $j\in\mathcal{J}_i$.
2) Map these estimates $b_j^{(i)}$, $j=1,\cdots,n_{a,i}$ to the outer code symbols to get $\hat{a}_i$ and use $\alpha_j^{(i)}$ as reliability for position j of $\hat{a}_i$.
3) For $l=1,\cdots,d_{a,i}$ set the l positions with smallest reliabilities $\alpha_j^{(i)}$ to erasures and use an Error-and-Erasure Decoder (EED) to find $a_i\{l\}$.
4) Check if any $a_i\{l\}$ satisfies *Condition 1*. If yes, $a_i=a_i\{l\}$ is the final decision. If no, signal decoding failure.
5) Continue with the next stage.

In the Appendix, we prove the following theorem:

Theorem 3: The algorithm will find the transmitted codeword as long as less than $d/2$ errors have occurred during transmission.

For an LUEP code constructed as a (modified) GC code, it is desirable to be able to guarantee a correct decoding as long as at most $\lfloor (s(G)_i-1)/2\rfloor$ errors have occurred in the transmitted codeword. If the outer codes in the above constructions have an equal protection of their information symbols, the described decoding algorithm can be applied directly and guarantees a decoding up to $\lfloor (s(G)_i-1)/2\rfloor$ errors. This is due to the properties of the multistage decoding. Similar to the BZZ algorithm, many error patterns of higher weight are decoded too.

The authors are not aware of a general BMD decoder for LUEP codes that corrects errors and erasures, and issues complete codewords. Such a decoder would be necessary for the general case. However, if we again assume that the outer codes A_i in the constructions are also GC LUEP codes with inner codes that are not UEP codes, the decoding up to $\lfloor (s(G)_i-1)/2\rfloor$ errors can be achieved by the algorithm for modified GC codes as described above. This is demonstrated for the first information symbol m_1 with separation $s(G)_1$: first the inner codes $B_j^{(0)'}$, for $j=1,\cdots,n_{a,1}$, are BMD decoded and an estimation for the outer code symbols $a_{1,j}$, together with the estimation of the reliability for these symbols $\alpha_j^{(0)}$, are transmitted to the outer code A_1. Since this code is again a GC LUEP code with inner codes that are not UEP codes, the $\alpha_j^{(0)}$ represent the reliabilities for the code symbols of the inner codes $B_j^{(0)'}$ of the GC code A_1. Proceeding in this way until the outer code is no longer an UEP code (this occurs in the worst case after $k_{a,1}$ steps) finally allows the application of the above algorithm and a decoding of m_1.

V. Summary

In this correspondence we use modified GC codes for the construction of binary codes of noncomposite length and of LUEP codes. Using this construction, it is not only possible to construct good codes but also to decode them efficiently. Especially, this holds true for the constructed LUEP codes. A decoding algorithm which guarantees decoding up to half the designed minimum distance, similar to the BZZ algorithm, is derived.

Appendix
Proof of Theorem 3

We will prove Theorem 3 by using the following two theorems from [12]:

Theorem 4: There is only one codeword a_i in a code with minimum distance $d_{a,i}$ that satisfies *Condition 1*.

Theorem 5: If any codeword a_i satisfies *Condition 1*, it will be found by an Error-and-Erasure Decoder.

Notice that for the proof of these two theorems, the reliability $\alpha_j^{(i)}$ does not have to be specified. It only has to be greater or equal to zero. Using *Condition 1* we prove Theorem 3:

Proof: Denote by T the total number of errors and consider the following inequalities:

$$2T < \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)}^{(i)}$$

$$\Leftrightarrow \quad 2\sum_{j=1}^{n_{a,i}} d_H[r_j, c_j] < \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)}^{(i)}$$

$$\Rightarrow \quad 2\sum_{j=1}^{d_{a,i}} d_H[r_{\Pi(j)}, c_{\Pi(j)}] < \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)}^{(i)}$$

$$\Rightarrow \quad \sum_{j\in\mathcal{E}(\hat{a}_i,\hat{a}_i)} \alpha_j^{(i)} < \sum_{j\in\bar{\mathcal{E}}^*(\hat{a}_i,\hat{a}_i)} \alpha_j^{(i)}.$$

The last step follows from the following considerations:

1) If $b_j^{(i)} = c_j$ then $\alpha_j^{(i)} \geq d_{b,j} - 2d_H(c_j, r_j)$.
2) If $b_j^{(i)} \neq c_j$ then $\alpha_j^{(i)} \leq 2d_H(c_j, r_j) - d_{b,j}$, because

$$d_{b,j} \leq d_H(c_j, r_j) + d_H[b_j^{(i)}, r_j]$$

$$\Leftrightarrow \quad d_{b,j} - 2d_H[b_j^{(i)}, r_j] \leq 2d_H(c_j, r_j) - d_{b,j}.$$

So

$$\begin{aligned}
&\sum_{j\in\bar{\mathcal{E}}^*(\hat{a}_i,\hat{a}_i)} \alpha_j^{(i)} - \sum_{j\in\mathcal{E}(\hat{a}_i,\hat{a}_i)} \alpha_j^{(i)} \\
&\quad \geq \sum_{j\in\bar{\mathcal{E}}^*(\hat{a}_i,\hat{a}_i)} [d_{b,j} - 2d_H(b_j^{(i)}, r_j)] \\
&\qquad - \sum_{j\in\mathcal{E}(\hat{a}_i,\hat{a}_i)} [2d_H(c_j, r_j) - d_{b,j}] \\
&\quad = \sum_{j\in\bar{\mathcal{E}}^*(\hat{a}_i,\hat{a}_i)\cup\mathcal{E}(\hat{a}_i,\hat{a}_i)} d_{b,j} - 2d_H(c_j, r_j) \\
&\quad = \sum_{j=1}^{d_{a,i}} d_{b,\Pi(j)} - 2d_H[r_{\Pi(j)}, c_{\Pi(j)}]
\end{aligned}$$

i.e., as long as less than $d/2$ errors have occurred, only the transmitted codeword satisfies *Condition 1* and the EED will find it. □

For LUEP codes the following theorem holds:

Corollary 1: All codewords a_i in an LUEP code with separation vector $\boldsymbol{s} = (s_1, \cdots, s_k)$ that satisfy *Condition 1* with $d = s_1$, have the same information symbol m_1.

Proof: Codewords generated from information vectors which differ in component m_1 have minimum distance s_1 by definition of $\boldsymbol{s}$. The theorem follows from the same arguments as in [12, proof of Theorem 1]. □

References

[1] W. van Gils, "Design of error-control coding schemes for three problems of noisy information transmission, storage and processing," Ph.D. dissertation, Eindhoven Univ. of Technology, Eindhoven, The Netherlands, 1988.

[2] E. L. Blokh and V. Zyablov, "Coding of generalized concatenated codes," *Probl. Inform. Transm.*, vol. 10, no. 3, pp. 218–222, 1974.

[3] V. A. Zinov'ev, "Generalized concatenated codes for channels with error bursts and independent errors," *Probl. Inform. Transm.*, vol. 17, no. 4, pp. 53–56, 1981.

[4] G. D. Forney Jr., *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.

[5] L. Tolhuizen, "Two new binary codes obtained by shortening a generalized concatenated code," *IEEE Trans. Inform. Theory*, vol. 37, p. 1705, 1991.

[6] W. van Gils, "Linear unequal error protection codes from shorter codes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 544–546, May 1984.

[7] V. A. Zinov'ev and V. Zyablov, "Codes with unequal protection of information symbols," *Probl. Inform. Transm.*, vol. 15, no. 3, pp. 197–205, 1980.

[8] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-3, pp. 600–607, Oct. 1967.

[9] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. New York: North-Holland, 1983.

[10] L. A. Dunning and W. E. Robbins, "Optimum encoding of linear block codes for unequal error protection," *Inform. Contr.*, vol. 37, pp. 150–177, 1978.

[11] W. van Gils, "Two topics on linear unequal error protection codes: Bounds on their length and cyclic code classes," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 866–876, Nov. 1983.

[12] D. Taipale and M. Pursley, "An improvement to generalized minimum distance decoding," *IEEE Trans. Inform. Theory*, vol. 37, pp. 167–172, Jan. 1991.

A New Approach to the Design of Codes for Synchronous-CDMA Systems

Gurgen H. Khachatrian and Samvel S. Martirossian

***Abstract*— In this correspondence a new approach to increase the sum rate for conventional synchronous code-division multiple-access (S-CDMA) systems is presented. It is shown that it can be done by joint processing of the outputs of matched filters, when one considers the system of codes for S-CDMA to be the codes for the usual adder channel. An example of construction and decoding of such a system is also given.**

***Index Terms*—Multiuser spread-spectrum system, code-division multiple access, adder channel, matched filter.**

I. Introduction

Recent developments in multiuser spread-spectrum communication systems show the need to increase their sum rate. In cellular systems this means increasing the number of users, that can be simultaneously active inside each cell. In code-division multiple-access (CDMA) systems each of the users is assigned a binary ± 1-valued spreading sequence of the same length.

In a synchronous CDMA (S-CDMA) system, all users are in exact synchronism in the sense that not only are their carrier frequencies and phases the same, but also their expanded data symbols are aligned in time. It is also assumed that all the sequences have equal energy.

In a conventional CDMA receiver, the demodulator output for each symbol interval is further processed separately by each user. This procedure is called matched filtering. Mathematically, this corresponds to computing the scalar product between the spreading sequence of the ith user and the vector which represents the demodulator output of the CDMA system. Interuser interference then is defined by the crosscorrelation function between the spreading sequences of

Manuscript received March 1, 1993; revised February 10, 1995. The research reported has been performed during a visit at the Institut für Netzwerk- und Signaltheorie, Technische Hochschule Darmstadt, Darmstadt, Germany, and supported by DFG in cooperation with DFG and the Academy of Sciences of the Republic of Armenia.

The authors are with the Institute for Problems of Informatics and Automation, the Academy of Sciences of Armenia, 375044 Yerevan, Armenia.

IEEE Log Number 9413057.