ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2008 (665-669)
VOLUME : 8
NUMBER :2

GEOMETRIC CONSTRUCTION CODES OF HAMMING DISTANCE-8

Gökmen Altay

Bahcesehir University, Department of Electrical & Electronics Engineering,

Besiktas, Istanbul, 34349, Turkey

galtay@bahcesehir.edu.tr

ABSTRACT

A new high rate binary linear block code construction technique, named as Geometric Construction (GC) codes, was proposed recently [1]. It generates all the even full information rate (optimal) Hamming distance – 4 codes. In this paper, we have enhanced the construction of GC codes with respect to code rate and derived a code family of hamming distance-8 GC codes.

Keywords: *Linear block codes, code construction*

1. Introduction

The goal of channel coding is to find a code that is easy to encode and decode, and at the same time gives a high code rate for the largest minimum distance [2]. Binary codes of optimal or near optimal sizes can be used for power limited or bandwidth limited applications. Constructing long and powerful codes from small and simple codes is an old and popular technique. The |u|u+v| construction [3], squaring construction [4], block turbo codes [5], augmented product codes [6, 7] are some of these sorts of codes. They are advantages from encoding and decoding point of view as they have decomposable structure.

A similar type of binary linear block code construction technique, Geometric Construction (GC) codes were proposed recently [1], which are capable of generating all the even length and optimal binary linear block codes of Hamming distance-4, and that can also generate some higher distance good codes. GC technique basically uses simple component generator matrices to form (in a specified fashion) a larger generator matrix. In [1], the component generator matrices were specified for Hamming distance-4 codes. Additionally, some component generator matrices were specified for Hamming distance-8 and Hamming distance-16 codes but they are not able to generate neither mostly good codes nor a code family. In other words, the GC technique did not define a general way to find the necessary component generator matrices for higher Hamming distances. This was the week point of the construction in [1].

In this paper, we enhance the GC method of [1] for Hamming distance-8 code construction and derive an optimal or near optimal Hamming distance-8 linear block code family. The optimal sizes of block codes can be looked up from the table of best known codes [8]. The codes constructed by the proposed GC technique have great flexibility with respect to adjusting the length of a code. Additionally, the constructed generator matrices of GC codes contain the lowest density of ones (that is $k*d$ where k and d are the code dimension and length, respectively) since each row of a GC generator matrix contains binary 1's of size equal to the desired Hamming distance of the code. The rows of a GC generator matrix have also quasi-cyclic property and it is known that a code with cyclic or quasi-cyclic property can be encoded with less complexity [9, 10] in a similar way of cyclic codes. The GC codes also incorporates the advantages of low-density,

Received Date: 01.02.2007
Accepted Date: 01.07.2008

quasi-cyclic, regular structure generator matrices and therefore practical for encoding and decoding.

2. Hamming distance-8 GC code construction

The generic binary generator matrix G of the Hamming distance-8 GC codes C is proposed as in (1).

$$G = \left(\begin{array}{l}
G_1G_1 \\
\quad G_1G_1 \\
\qquad\qquad\qquad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad G_1G_1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad G_1G_1 \\
G_2G_2G_2G_2 \\
\qquad G_2G_2G_2G_2 \\
\qquad\qquad\qquad\quad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad G_2G_2G_2G_2 \\
G_2 \quad G_2 \quad G_2 \quad G_2 \\
\qquad\quad G_2 \quad G_2 \quad G_2 \quad G_2 \\
\qquad\qquad\qquad\qquad\qquad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\qquad\qquad\qquad G_2 \quad G_2 \quad G_2 \quad G_2 \\
G_2 \qquad G_2 \qquad G_2 \qquad G_2 \\
\qquad\qquad\quad\;\; G_2 \qquad G_2 \quad\;\; G_2 \qquad G_2 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\quad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad G_3 \equiv G_2 \;\; (G_3 \text{ is placed just as } G_2\,) \\
G_4G_4G_4G_4G_4G_4G_4G_4 \\
\qquad\quad G_4G_4G_4G_4G_4G_4G_4G_4 \\
\qquad\qquad\qquad\qquad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\qquad\qquad\qquad G_4G_4G_4G_4G_4G_4G_4G_4 \\
G_4G_4 \qquad G_4G_4 \qquad G_4G_4 \qquad G_4G_4 \\
\qquad\qquad\qquad\qquad\qquad\qquad \cdot \quad \cdot \quad \cdot \\
G_4 \quad G_4 \quad G_4 \quad G_4 \quad G_4 \quad G_4 \quad G_4 \quad G_4 \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \cdot \quad \cdot \quad \cdot \\
\qquad\qquad\qquad\quad \cdot \quad \cdot \quad \cdot
\end{array} \right) \tag{1}$$

Where we specify the component matrices as G_1=[1 1 1 1], G_2=[1 1 0 0], G_3=[1 0 1 0] and G_4=[1 0 0 0]. The placement of these component generator matrices in GC generator matrix is similar to GC construction of [1]; but here, we need to set some additional placement rules. We modify the construction in [1] to obtain higher code rate GC codes of Hamming distance-8. The difference between the construction of existing GC codes of [1] and the proposed construction of (1) of this paper is pointed out by writing the enhanced part in italic character in (1). After illustrating the construction we proceed by describing it in detail.

For all the GC codes the ultimate generator matrix structure can only be formed basically as in (2).

$$G = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_t \end{bmatrix} \tag{2}$$

Here, $D_1, D_2, ..., D_t$ are the *group generator matrices* of the component generator matrices (G_1, G_2, G_3, G_4). Each group generator matrix, D_i (i=1,2,…,t), is separated with respect to the placements of the component generator matrices G_j (j=1,…,4). As seen in (2), the ultimate generator matrix of Hamming distance-8 GC codes is obtained by augmenting group generator matrices, D_is. The reason for grouping the component matrices is because, otherwise, the whole matrix G looks quite complex and so confusing to realise. The number of group generator matrices (t), depends on the size of code (*n, k, 8*), where n, k and 8 is the code length, dimension and distance, respectively. The length of the GC code can be chosen as multiple of 4 and greater than or equal to 16.

Construction continues as follows: Each group generator matrix D_i is obtained by placing a number of an identical component matrices (G_j) , which means each group contains only one of G_j*s*. **An important condition that needs to be satisfied in all the group generator matrices is that the number of component matrices, which are placed in a row of D_i, is adjusted so that the Hamming weight of the row becomes 8**. As an example, if a D_i is formed using the G_4, then we must place 8 of G_4s in the rows of D_i. By doing so, the ultimate generator matrix of the GC code contains the least possible binary 1's in it, which is k×8, and becomes the lowest density generator matrix for a binary linear block code of Hamming distance 8. For the ease of descriptions, we denote the number of component generator matrices, in a row of a group matrix, as '*m*', where *m* varies for different groups.

Keeping the described constraints in mind, placing and shifting of G_j*s* in D_i*s* can be expressed more easily. Firstly, 2 of G_1*s* are placed consecutively in (1), without interval, into the first row of D_1. From above expressions, it is easy to know that 2 of G_1*s* should be placed so that the Hamming distance of the row becomes *8*. Note that the length of columns is *n* and 8 is less than *n*, so the rest of columns are filled by binary zeros. The columns of the first row is shifted to the right by a scale of one G_1 and placed to the second row. This cyclic process is repeated for the following rows of the group matrix D_1 until the G_1 matrices arrive to the last column of a row. When D_1 is constructed, it is placed in G as shown in (2). The rest of the process is slightly different than of D_1 but all of them have the similar rule as follows.

We start from G_2 and continue until G_4. G_2 and G_3 are placed in exactly the same way since they contain the same number of ones in them, whereas G_4 is placed in a slightly different way as will be explained later. Let's start constructing from D_2; we place m (obviously $m = 4$) of G_2s consecutively without interval in the first row of the group D_2. The columns of the first row is shifted to the right by a scale of $m/2$ G_2s and placed to the second row of D_2. For the first group of G_2s, the number of shifting is calculated as 2^1. **It is important to emphasize the rule that the scale of shifting must be arranged to satisfy that the number of overlapping G_2s between the consecutive rows is $m/2$ and also component matrices must not overlap between non-consecutive rows of the same group.** The process of shifting to the right and placing the G_2 matrices, which is also a cyclic process, is performed as long as there is room to place m of G_2 matrices in the row of G. If there are not enough columns to place m of G_2s, then we stop placing G_2s. When the placement of the group D_2 is complete, then we start placing the same G_2s of group D_3, if there is room to place them. In this case, there will be 1 interval among m of G_2s in a row. At the second row of D_3, the shifting to the right will be by 4 times. For this second group of G_2s, the number of shifting is calculated as 2^2 and for the following groups of G_2s the number of shifting is calculated as 2^i (i=1,2,…). The number of intervals can be calculated as the power of 2 minus 1 in order. Similarly, this process is performed until there will be no room left for placing G_2s in a row. After the placement for D_3 is ended, if there is still room for placing G_2s of the group D_4, then, m of G_2s are placed with 3 intervals among them. In this group, shifting to right interval will be by 8 times (2^3) the scale of G_2. This process continues for other groups $\{D_4, D_5, \ldots\}$ and the whole process is ended when there is no room left to place m of the same G_2s in a row. Since G_3 is placed in exactly the same way as G_2 and augmented under them as in (2), we continue by describing the placement of G_4 that is one of the new enhancement parts of GC construction introduced by this paper.

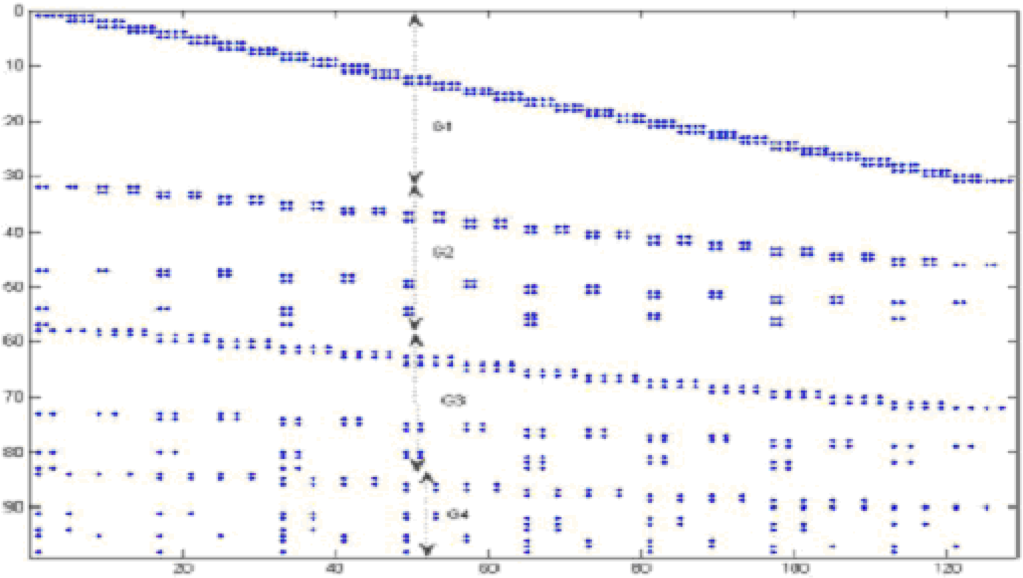
When placing m of (In this case clearly m = 8) G_4 matrices, they are placed consecutively without interval among them and shifted to right by $m/2$ of G_4 matrices for a group D_i. When this group is complete similar to the previous ones, we proceed by placing double G_4s with initially 2 intervals and cyclically shift them such that the number of overlapping G_4s between consecutive rows becomes $m/2$. Then for the following groups of double G_4s, the interval becomes 6 and so on as the number of shifting of previous group minus 2. When the groups of double G_4s are complete, we proceed by placing single G_4s with one interval among them and by shifting to right for the following rows, regarding the general rule, such that the number of overlapping G_js between the consecutive rows is $m/2$ and also component matrices must not overlap between non-consecutive rows of the same group. For other groups the interval becomes 3, 7,… as the power of 2 minus 1. At the end, all the constructed group matrices are placed in the ultimate GC code generator matrix G as shown in (2). In order to give a utility to the reader for verifying the above explanations of GC codes, we demonstrate some examples below.

Example 1. We construct (64, 42, 8) GC code as shown in (3). This GC code is in the table of best-known codes [8] and is considered as optimal. For making the illustration simpler, we denote G_1 = a, G_2 = b, G_3 = c and G_4 = d in the GC generator matrix.

$$G=\left[\begin{array}{l}
\text{aa} \\
\quad\text{aa} \\
\qquad\qquad\cdots \\
\qquad\qquad\qquad\quad\text{aa} \\
\qquad\qquad\qquad\qquad\text{aa} \\
\text{bbbb} \\
\quad\text{bbbb} \\
\qquad\qquad . \; . \; . \\
\qquad\qquad\qquad\quad\text{bbbb} \\
\text{b b b b} \\
\qquad\text{b b b b} \\
\qquad\qquad\qquad\text{b b b b} \\
\text{b}\quad\;\text{b}\quad\;\text{b}\quad\;\text{b} \\
\text{c} \equiv \text{b (same type)} \\
\text{dddddddd} \\
\qquad\text{dddddddd} \\
\qquad\qquad\quad\text{dddddddd} \\
\text{dd}\quad\text{dd}\quad\text{dd}\quad\text{dd} \\
\text{d d d d d d d d}
\end{array}\right]
\begin{array}{l}
\\ \\ \\ \\ \text{(}D_1\text{ has 15 rows)} \\ \\ \\ \\ \text{(}D_2\text{ has 7 rows)} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \{42\times 64\}
\end{array}
\qquad (3)$$

In Example 2 we illustrate the construction for a bigger GC code in order to support the description of the construction.

Example 2. We plot the constructed generator matrix of (128, 98, 8) GC code regarding their 1's in the matrix as below. In the figure of the matrix, the nonzero elements in the matrix are denoted by a square and the rest is left empty. The ranges of component matrices, G_js, are pointed out by arrows. Here the number of ones in the matrix is 128×8=784, which is the least possible number for the generator matrix of a (128, 98, 8) block code.



We continue by proving Theorem 1, which will be useful in finding the minimum Hamming distances of a GC code family.

Theorem 1: Let *G* be the binary generator matrix of the code *C* that is constructed using (1). Then the minimum Hamming distance of the code *C* is 8.

Proof: Recall firstly, the row vectors of the generator matrix of (1) have even weight 8 and the number of ones in common is at most 4. The codewords of *C* are obtained by linear combination of row vectors a_i of *G*, *(i=1, 2 ... s, where s ≤ k).* Then the minimum (Hamming) distance of *C* can be written as,

$$\mathrm{w}_{\min}(\mathrm{a}_1 + \mathrm{a}_2 + \ldots + \mathrm{a}_s) = \mathrm{w} = d \qquad (4)$$

where at least one of a_i's is not a zero vector and d denotesthe Hamming distance of the code and also w denotes the Hamming weight of a vector.

Case 1: If any two rows do not have 1 in common, then clearly $w \geq 8$ since each vector has weight 8.

Case 2: Since we look for the minimum value for *w*, then there must be common 1's among the rows.

Now consider the case (where the maximum overlapping occurs) there are 4 locations that have 1's in common. In this case if there are even number of rows then *w* becomes $w = j \times 4$ $(j \geq 2)$ and if there are odd number of rows then $w = (j+1) \times 4$ where *j* is an integer greater than 2.

In order to obtain the minimum value for *w* the other 4 locations should be in different places. The best situation occurs between the rows following one another, i.e. a_i and a_{i+1} for $i = 1, 2, \ldots s-1$.

$$\mathrm{w}_{\min}(\underbrace{a_1 + a_2}_{4} \underbrace{+ \ldots}_{4} \ldots \underbrace{+ a_{s-1} + a_s}_{4}) = w \qquad \underbrace{}_{4} \quad (5)$$

In this case, each common 1's are cancelled and so the summation in (4) is minimized. Therefore *w* is calculated as $w = s \times 8 - (s-1) \times 4 = (s+1) \times 4$ where $s \geq 1$. The minimum value is obtained when $s = 1$. Therefore $w = \mathrm{d} = 8$.

We constructed some of distance-8 GC codes utilizing a computer program and demonstrated them in Table 1.

Table 1. Some of the codes of the Hamming distance-8 GC code family. Codes are denoted as C = (n, k ,8) where n, k, 8 are the code length, dimension and distance, respectively.

(16,5,8)	(40,20,8)	(68,43,8)	. . .	. . .	. . .
(20,6,8)	(44,23,8)	(72,46,8)	(256,214,8)	(2048,1882,8)	(16384,15310,8)
(24,9,8)	(48,27,8)	(76,49,8)	. . .	. . .	. . .
(28,12,8)	(56,35,8)	(80,53,8)	(512,450,8)	(4096,3798,8)	(32768,30666,8)
(32,16,8)	(60,38,8)	. . .	. . .	. . .	. . .
(36,17,8)	(64,42,8)	(128,98,8)	(1024,926,8)	(8192,7634,8)	. . .

2. CONCLUSION

In this paper, we have enhanced and generalized the GC codes of [1] for Hamming distance-8 binary linear block codes. Some of the newly constructed codes have been demonstrated in Table 1 as a code family. As GC codes generator matrices have some useful properties like low density, regular and quasi-cyclic structure and also grouping utility. They are also practical for encoder and decoder design of very long length block codes. We currently work on finding efficient decoding algorithm for the proposed codes. A future work includes generalizing the method for higher distance GC codes.

References

[1] Altay, G.; Ucan, O.N.: Heuristic construction of high-rate linear block codes. Int. J. Electron. Commun. (AEÜ), vol.60, pp.663-666, 2006.

[2] Bossert, M.: Channel Coding, Wiley, 1999.

[3] Williams, M.F.J.; Sloane, N.J.A.: The Theory of Error-Correcting Codes. (North-Holland, Amsterdam, 1998.

[4] 4. Forney, Jr., G.D.: Coset codes II: Binary lattices and related codes. IEEE Trans. Inform. Theory, 1988, 34, pp. 1152-1187.

[5] R. M. Pyndiah.: Near-optimum decoding of product codes: Block turbo codes. IEEE Trans. Commun., vol.46, No. 8, pp.1003-1010, Aug. 1998.

[6] Peng, X.-H.; Farrell, P.G.: Optimal augmentation of product codes. Electronics Letters, 2004, 40, pp. 750-752.

[7] Salomon, A.J.; Amrani, O.: Augmented product codes and lattices: Reed-Muller codes and Barnes-Wall lattices. Trans. Inform. Theory, November 2005, vol.51, No. 11, pp. 1152-1187.

[8] Brouwer, A. E.; Verhoeff, T.: An updated table of minimum-distance bounds for binary linear codes. IEEE Trans. Inform. Theory, vol. 39, No. 2, pp. 664-677, March 1993.

[9] Lucas, R.; Fossorier, M.P.C.; Kou, Y.; Lin, S.: Iterative decoding of one-step majority logic decodable codes based on belief propagation. IEEE Trans. Commun., vol.48, pp.931-937, June 2000.

[10] Jhonson, S. J.; Weller, S.R.: A family of irregular LDPC codes with low encoding complexity. IEEE Comm. Letter, vol. 7, pp.79-81, No. 2, Feb. 2003.