A CLASS OF SOLID BURST ERROR CORRECTING CODES DERIVED FROM A REVERSIBLE CODE

Pankaj Kumar Das[1]

In this paper, we present a class of linear codes that are capable of correcting solid burst errors of certain length or less. We obtain the codes by modifying the parity check matrix of the reversible code given by Muttoo and Lal ("A reversible code over GF(q)", Kybernetika, *22(1), 85–91, 1986). Further, we present solid burst error uncorrected probability and weight distribution of the reversible code and the obtained codes for particular cases.*

Keywords: parity check matrix, syndromes, error correction, solid burst, weight distribution.

MSC2010: 94B 05, 94B 20.

1. Introduction

In some communication channels like semiconductor memory data, supercomputer storage systems [1, 2, 3, 6], it was found that errors do not occur independently, they occur consecutively. As a result in the received codeword (codevector), some consecutive (adjacent) components get disturbed and so there is a need to rectify them. This type of disturbances is known as solid burst error/adjacent error.

Definition 1.1. *A solid burst of length b is a vector whose all the b-consecutive components are non-zero and rest are zero.*

Muttoo and Lal [8] have presented a linear code over finite field $GF(q)$ (which is also reversible code), capable of correcting solid burst of odd length only. This paper modifies the parity check matrix of the code and obtains a class of linear codes that correct all solid burst errors of certain length or less, irrespective of even or odd length. The work of this paper is motivated from the paper [7] where the author obtains two codes by rearranging the columns of the parity check matrix of a systematic code given by [5]. One code can correct double errors as well as detect all triple-adjacent errors removing the restriction of detecting triple-adjacent errors within 8-bit bytes and another code is capable of correcting all 16 single errors, correct 113 of the 120 double

[1]Department of Mathematical Sciences, Tezpur University, Napaam, Sonitpur, Assam-784028, India, e-mail: pankaj4@tezu.ernet.in, pankaj4thapril@yahoo.co.in

errors, detect seven double errors, correct seven of 14 triple adjacent errors, detect seven triple adjacent errors, and correct all quadruple adjacent errors. In a very recent paper [4], it also obtains a class of solid burst error detecting and locating codes (with better information rate) derived from the famous Extended (8, 4) Hamming code.

As it is not always possible to detect/correct all errors, so it becomes important to know the probability of errors that goes undetected/uncorrected even after applying error detection/correction mechanism. In this direction, the probability of solid burst errors going undetected/uncorrected by the original code and by new derived codes is studied here. Further, for carefully characterizing the performance of codes, the weight distribution of a code is important. This paper also studies weight distribution of the codes for binary case.

The paper has been organized as follows. Section 1 is the introductory part and motivation of the paper. In Section 2, we derive a class of linear codes that are capable of correcting solid burst errors of certain length or less from the reversible code studied by Muttoo and Lal [8]. In Section 3, we obtain solid burst error probability that goes uncorrected by the original and new derived codes. It is followed by weight distribution of the codes for binary case.

2. **Code Construction**

The main result of the paper [8] is the following theorem.

Theorem 2.1. *An (n, k) linear code C_k over $GF(q)$, $n = 2k+1$, whose parity check matrix is H_k,*

$$H_k = \left[\begin{array}{ccccc c ccccc} \multicolumn{5}{c}{\overbrace{\hphantom{x_1 \quad x_2 \quad \dots \quad x_{k-1} \quad x_k}}^{k}} & & \multicolumn{5}{c}{\overbrace{\hphantom{0 \quad 0 \quad \dots \quad 0 \quad 0}}^{k}} \\ x_1 & x_2 & \dots & x_{k-1} & x_k & y & 0 & 0 & \dots & 0 & 0 \\ 0 & x_2 & \dots & x_{k-1} & x_k & y & x_k & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & x_{k-1} & x_k & y & x_k & x_{k-1} & \dots & 0 & 0 \\ . & . & \dots & . & . & . & . & . & \dots & . & . \\ . & . & \dots & . & . & . & . & . & \dots & . & . \\ 0 & 0 & \dots & 0 & x_k & y & x_k & x_{k-1} & \dots & x_2 & 0 \\ 0 & 0 & \dots & 0 & 0 & y & x_k & x_{k-1} & \dots & x_2 & x_1 \end{array}\right],$$

where $y, x_j \in \{1, 2, \dots q-1\}$, $(x_i, y) = 1$ and $(x_i, x_j) = 1$ for $i \neq j$, $i, j = 1, 2, \dots k$.

is capable to correct,
(i) all solid bursts of odd lengths upto $2k-1$, if $n-k$ is even, i.e. if k is odd,
(ii) all solid bursts of odd lengths upto $k-1$, if $n-k$ is odd, i.e. if k is even.

The matrix H_k in the theorem can be written as

$$H_k = \begin{bmatrix} A_k & Y & B_k \end{bmatrix},$$

where

$$A_k = \begin{bmatrix} x_1 & x_2 & \dots & x_{k-1} & x_k \\ 0 & x_2 & \dots & x_{k-1} & x_k \\ 0 & 0 & \dots & x_{k-1} & x_k \\ . & . & \dots & . & . \\ . & . & \dots & . & . \\ 0 & 0 & \dots & 0 & x_k \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} y \\ y \\ y \\ . \\ . \\ y \\ y \end{bmatrix} \text{ and}$$

$$B_k = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ x_k & 0 & \dots & 0 & 0 \\ x_k & x_{k-1} & \dots & 0 & 0 \\ . & . & \dots & . & . \\ . & . & \dots & . & . \\ x_k & x_{k-1} & \dots & x_2 & 0 \\ x_k & x_{k-1} & \dots & x_2 & x_1 \end{bmatrix}.$$

The main drawback of the code is that it can not correct solid burst of even length (less than certain value given in Theorem 2.1). This is due to the fact that syndromes of solid burst of even length are not distinct with syndromes of other solid bursts. For example, the syndrome of solid bursts confined to the k^{th} and $(k+1)^{th}$ components may coincide with the syndrome resulting from the error at last position.

In this correspondence, we modify the matrix H_k such that the syndromes of all solid bursts of length $2k-1$ (or $k-1$) or less, irrespective of even or odd length, become all distinct and the resulting code can correct all such errors. We modify the matrix H_k by keep on adding rows to H_k and finally obtaining the general matrix, denoted by H_k^t which gives rise to a linear code C_k^t that can correct all solid bursts of certain length or less, irrespective of even or odd length.

First, we add a single row to H_k after the $(k+1)^{th}$ row as follows.

$$H_k^1 = \begin{bmatrix} x_1 & x_2 & \dots & x_{k-1} & x_k & y & 0 & 0 & \dots & 0 & 0 \\ 0 & x_2 & \dots & x_{k-1} & x_k & y & x_k & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & x_{k-1} & x_k & y & x_k & x_{k-1} & \dots & 0 & 0 \\ . & . & \dots & . & . & . & . & . & \dots & . & . \\ . & . & \dots & . & . & . & . & . & \dots & . & . \\ 0 & 0 & \dots & 0 & x_k & y & x_k & x_{k-1} & \dots & x_2 & 0 \\ 0 & 0 & \dots & 0 & 0 & y & x_k & x_{k-1} & \dots & x_2 & x_1 \\ \\ 0 & 0 & \dots & 0 & 0 & y & 0 & x_{k-1} & \dots & 0 & 0 \end{bmatrix},$$

or

$$H_k^1 = \begin{bmatrix} A_k & Y & B_k \\ 0 & y & a_1 \end{bmatrix},$$

where a_1 represents the k-tuple $(0, x_{k-1}, 0, x_{k-3}, 0, \ldots, c)$ and the value of c is given as follows.

$$c = \begin{cases} x_1 & \text{if } k-2 \text{ is a multiple of } 2 \\ 0 & \text{if } k-2 \text{ is not a multiple of } 2. \end{cases}$$

This matrix gives rise to a $(2k+1, k-1)$ linear code C_k^1 which will correct all solid burst error of length upto 3. This can be confirmed by showing that all the syndromes produced by solid bursts of length 3 or less are distinct and non-zero. Since the code obtained from H_k correct all solid bursts of length 1 and 3, so the syndromes resulting from such errors are distinct and non-zero. The situation of syndromes of solid burst of length 2 coinciding with other syndromes of solid bursts of length upto 3 is only possible when one burst comes from the first $k+1$ components and the second comes from the last $k+1$ components. We have added the $(k+2)^{th}$ row to H_k in such a way even if the first $k+1$ components of the syndromes of solid burst errors of length 3 or less coincide, the last component i.e. $(k+2)^{th}$ component gets different.

We now add another row to H_k^1 as follows.

$$H_k^3 = \begin{bmatrix} A_k & Y & B_k \\ 0 & y & a_1 \\ 0 & y & a_3 \end{bmatrix},$$

where a_3 represents the k-tuple $(0, 0, 0, x_{k-3}, 0, 0, 0, x_{k-7} \ldots, c)$ and the value of c is given as follows.

$$c = \begin{cases} x_1 & \text{if } k-4 \text{ is a multiple of } 4 \\ 0 & \text{if } k-4 \text{ is not a multiple of } 4. \end{cases}$$

This matrix gives rise to a $(2k+1, k-2)$ linear code C_k^3 that will correct all solid bursts of length 5 or less. This is again justified by showing that syndromes of such solid burst errors are all distinct and non-zero. The pattern of the tuple a_3 is such that the syndromes of all such solid bursts gets distinct from all syndromes of solid burst of length 5 or less.

If we continue the process of adding row in the above manner, we obtain the following matrix H_k^t (t is odd) that gives rise to a $(2k+1, k - \frac{t+1}{2})$ linear code C_k^t which corrects all solid bursts of length $t+2$ or less.

$$H_k^t = \begin{bmatrix} A_k & Y & B_k \\ 0 & y & a_1 \\ 0 & y & a_3 \\ 0 & y & a_5 \\ . & . & . \\ . & . & . \\ 0 & y & a_t \end{bmatrix},$$

where a_t represents the k-tuple $(\underbrace{0, 0, ..., 0}_{t},\ x_{k-t}, \underbrace{0, 0, ..., 0}_{t},\ x_{k-2t-1}, \ldots, c)$ and the value of c is given as follows:

$$c = \begin{cases} x_1 & \text{if } \ k - t - 1 \text{ is a multiple of } t + 1 \\ 0 & \text{if } \ k - t - 1 \text{ is not a multiple of } t + 1. \end{cases}$$

Observation: The number of non-zero components in a_t is given by $\lceil \frac{k-t}{t+1} \rceil$, where $\lceil x \rceil$ means the smallest integer greater than or equal to x.

By Theorem 2.1, the code C_k corrects (i) all solid bursts of odd lengths upto $2k-1$, if k is odd, (ii) all solid bursts of odd lengths upto $k-1$, if k is even. So, the value of t in the length of solid burst correction is restricted to (i) $t \leq 2k-3$ if k is odd, and (ii) $t \leq k-3$ if k is even.

Note that if $t = 2k-1$, then $n - k = k + 1 + \frac{2k-3+1}{2} = 2k + 1 = n$.

We can summarise the above discussion as follows.

Theorem 2.2. *An* $(2k+1, k - \frac{t+1}{2})$ *linear code* C_k^t *over* $GF(q)$, t *is an odd number, whose parity check matrix is* H_k^t, *is capable to correct*
(i) all solid bursts of length $t+2$ *or less* ($t \leq 2k-3$) *and all solid bursts of odd lengths upto* $2k-1$, *if* k *is odd,*
(ii) all solid bursts of length $t+2$ *or less* ($t \leq k-3$) *and all solid bursts of odd lengths upto* $k-1$, *if* k *is even.*

Example 2.1. *Consider the following* (5×7) *matrix* H_3^1 *over GF(2):*

$$H_3^1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

We can easily check that the syndromes produced by all solid bursts of length 3 or less are non-zero and distinct. Therefore, the null space of H_3^1 *over GF(2) corrects all solid bursts of length 1, 2, 3.*

3. **Error Probability and Weight Distribution**

In this section, we study solid burst error probability and weight distribution of the codes C_k and C_k^t discussed in Section 2. The generator matrix of the $(2k+1, k)$ linear code C_k over $GF(q)$ can be obtained as follows.

$$G_k = \begin{bmatrix} \overbrace{x_k \quad 0 \ \ldots \quad 0 \quad 0}^{k} & -y^{-1}x_1x_k & \overbrace{x_1 \quad 0 \ \ldots \quad 0 \quad 0}^{k} \\ 0 \quad x_{k-1} \ \ldots \quad 0 \quad 0 & -y^{-1}x_2x_{k-1} & 0 \quad x_2 \ \ldots \quad 0 \quad 0 \\ . \quad . \ \ldots \quad . \quad . & . & . \quad . \ \ldots \quad . \quad . \\ . \quad . \ \ldots \quad . \quad . & . & . \quad . \ \ldots \quad . \quad . \\ 0 \quad 0 \ \ldots \quad x_2 \quad 0 & -y^{-1}x_{k-1}x_2 & 0 \quad 0 \ \ldots \quad x_{k-1} \quad 0 \\ 0 \quad 0 \ \ldots \quad 0 \quad x_1 & -y^{-1}x_kx_1 & 0 \quad 0 \ \ldots \quad 0 \quad x_k \end{bmatrix},$$

where $y, x_j \in \{1, 2, \ldots q-1\}$, $(x_i, y) = 1$ and $(x_i, x_j) = 1$ for $i \neq j$, $i, j = 1, 2, \ldots k$.

Remark 3.1. From the generator matrix G_k, we see that no codeword is of the form-solid burst of length $2k$ or less. Therefore, the code C_k can detect any solid burst of length $2k$ or less.

Solid Burst Error Probability: Now we find the probability of solid burst error that goes undetected/ uncorrected by the codes C_k and C_k^t. Let us consider a binary symmetrical channel (BSC) which has error probability p.

For the C_k code, since the code can correct (i) all solid burst of odd length upto $2k-1$, if k is odd, and (ii) all solid bursts of odd lengths upto $k-1$, if k is even. So, the probability that solid burst goes uncorrected for k being odd is $\sum_{i=1}^{k}(2k+2-2i)p^{2i}(1-p)^{2k+1-2i} + p^{2k+1}$ and for k being even is $\sum_{i=1}^{\frac{k}{2}}(2k+2-2i)p^{2i}(1-p)^{2k+1-2i} + \sum_{i=k+1}^{2k+1}(2k+2-i)p^i(1-p)^{2k+1-i}$.

For the C_k^t code (t being odd), the code can correct (i) all solid bursts of length $t+2$ or less ($t \leq 2k-3$) and all solid bursts of odd lengths upto $2k-1$, if k is odd, (ii) all solid bursts of length $t+2$ or less ($t \leq k-3$) and all solid bursts of odd lengths upto $k-1$, if k is even. So, the probability that solid burst goes uncorrected for k being odd is $\sum_{i=\frac{t+3}{2}}^{k}(2k+2-2i)p^{2i}(1-p)^{2k+1-2i} + p^{2k+1}$ and for k being even is $\sum_{i=\frac{t+3}{2}}^{\frac{k}{2}}(2k+2-2i)p^{2i}(1-p)^{2k+1-2i} + \sum_{i=k+1}^{2k+1}(2k+2-i)p^i(1-p)^{2k+1-i}$.

Remark 3.2. As the code C_k can detect any solid burst of length $2k$ or less and probability that solid burst goes undetected is p^{2k+1}.

Now consider the value of p is 0.1. In Table 3.1, we list the probabilities of solid burst error going uncorrected by the codes C_k and C_k^t for different values of k and t.

Table 3.1: Error Probability

k	t	Probability that solid burst goes uncorrected by C_k for $p = 0.1$ (appr. value)	Probability that solid burst goes uncorrected by C_k^t for $p = 0.1$ (appr. value)
(Odd value)			
5	1	0.0391282587	0.00038620980
5	3	0.0391282587	0.00000357229
5	5	0.0391282587	0.00000002935
5	7	0.0391282587	0.00000000019
(Even value)			
6	1	0.0469729930	0.00039165790
6	3	0.0469729930	0.00000423741

Weight Distribution: Here we give the weight distribution of the codes C_k and C_k^1. Clearly the distance of the codes C_k and C_k^1 is 3. If $wt(j)$ represents the number of codevector of the code C_k of hamming weight j, then weight distribution of the code C_k for the binary case is given by

$$\begin{gathered} wt(3) = \binom{k}{1}, \\ wt(4) = \binom{k}{2}, \\ wt(5) = 0, \\ wt(6) = 0, \\ wt(7) = \binom{k}{3}, \\ wt(8) = \binom{k}{4}, \\ wt(9) = 0, \\ wt(10) = 0, \\ wt(11) = \binom{k}{5}, \\ wt(12) = \binom{k}{6}, \\ wt(13) = 0, \\ wt(14) = 0, \\ - - - - - - - \\ - - - - - - - \end{gathered}$$

This can be summarised as follows.

Theorem 3.1. *If $wt(j)$ represents the number of codevector of the binary code $C_k(2k+1, k)$ having hamming weight j where $j \leq 2k+1$, then for $i \in \mathbb{N}$ and assuming $\binom{n}{r} = 0$ if $r > n$, we have*

$$wt(j) = \begin{cases} \binom{k}{i} & \text{if } j = 4i-1 \\ \binom{k}{2i} & \text{if } j = 4i \\ 0 & \text{if } j = 4i-2 \text{ or } 4i-3. \end{cases}$$

Using the MacWilliams Identity (Theorem 3.14, [9]), we get the following theorem.

Theorem 3.2. *If $wt(i)$ and $wt'(i)$ denote the number of codevectors of weight i in the binary code $C_k(n, k)$, where $n = 2k+1$ and its dual code, then*

$$wt'(i) = q^{-k} \sum_{j=0}^{n} wt(j) \sum_{s=0}^{n} \binom{j}{s} \binom{n-j}{i-s} (-1)^s (q-1)^{i-s},$$

where $wt(j)$ is given by Theorem 3.1.

Next, we give the weight distribution of C_k^t for $t = 1$. The code C_k^1 is more important than other codes C_k^t for $t \neq 1$ due to high probability of occurring of solid burst error of length upto 3. The generator matrix of the code C_k^1 for binary case is given by

$$G_k^1 = \begin{bmatrix} J & I_{k-1} & L & I_{k-1} \end{bmatrix},$$

where I_{k-1} represents the identity matrix of order $(k-1)$, J is a matrix of order $(k-1) \times 1$, the single column consisting of elements 0 and 1 alternatively with 0 first; and L is a matrix of order $(k-1) \times 2$ with the rows 10 and 01 alternatively with 10 as first row. For example the generator matrix of binary C_6^1 is given by

$$G_6^1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The generator matrix of the binary code C_k^1 in the above form helps us to obtain the weight distribution of the code. If A_j represents the number of codevectors of hamming weight j in the code $C_k^1(n, k-1)$, then the weight distribution of the C_k^1 is given by

$$\begin{aligned}
A_3 &= \binom{\lceil \frac{k-1}{2} \rceil}{1} . \binom{\lfloor \frac{k-1}{2} \rfloor}{0}, \\
A_4 &= \binom{\lceil \frac{k-1}{2} \rceil}{0} . \binom{\lfloor \frac{k-1}{2} \rfloor}{2} + \binom{\lceil \frac{k-1}{2} \rceil}{2} . \binom{\lfloor \frac{k-1}{2} \rfloor}{0} + \binom{\lceil \frac{k-1}{2} \rceil}{0} . \binom{\lfloor \frac{k-1}{2} \rfloor}{1}, \\
A_5 &= 0, \\
A_6 &= 0, \\
A_7 &= \binom{\lceil \frac{k-1}{2} \rceil}{1} . \binom{\lfloor \frac{k-1}{2} \rfloor}{2} + \binom{\lceil \frac{k-1}{2} \rceil}{3} . \binom{\lfloor \frac{k-1}{2} \rfloor}{0} + \binom{\lceil \frac{k-1}{2} \rceil}{1} . \binom{\lfloor \frac{k-1}{2} \rfloor}{1}, \\
A_8 &= \binom{\lceil \frac{k-1}{2} \rceil}{0} . \binom{\lfloor \frac{k-1}{2} \rfloor}{4} + \binom{\lceil \frac{k-1}{2} \rceil}{2} . \binom{\lfloor \frac{k-1}{2} \rfloor}{2} + \binom{\lceil \frac{k-1}{2} \rceil}{4} . \binom{\lfloor \frac{k-1}{2} \rfloor}{0} \\
&\quad + \binom{\lceil \frac{k-1}{2} \rceil}{0} . \binom{\lfloor \frac{k-1}{2} \rfloor}{3} + \binom{\lceil \frac{k-1}{2} \rceil}{2} . \binom{\lfloor \frac{k-1}{2} \rfloor}{1}, \\
A_9 &= 0, \\
A_{10} &= 0, \\
A_{11} &= \binom{\lceil \frac{k-1}{2} \rceil}{1} . \binom{\lfloor \frac{k-1}{2} \rfloor}{4} + \binom{\lceil \frac{k-1}{2} \rceil}{3} . \binom{\lfloor \frac{k-1}{2} \rfloor}{2} + \binom{\lceil \frac{k-1}{2} \rceil}{5} . \binom{\lfloor \frac{k-1}{2} \rfloor}{0} \\
&\quad + \binom{\lceil \frac{k-1}{2} \rceil}{1} . \binom{\lfloor \frac{k-1}{2} \rfloor}{3} + \binom{\lceil \frac{k-1}{2} \rceil}{3} . \binom{\lfloor \frac{k-1}{2} \rfloor}{1},
\end{aligned}$$

$$A_{12} = \binom{\lceil\frac{k-1}{2}\rceil}{0}.\binom{\lfloor\frac{k-1}{2}\rfloor}{6} + \binom{\lceil\frac{k-1}{2}\rceil}{2}.\binom{\lfloor\frac{k-1}{2}\rfloor}{4} + \binom{\lceil\frac{k-1}{2}\rceil}{4}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2} + \binom{\lceil\frac{k-1}{2}\rceil}{6}.\binom{\lfloor\frac{k-1}{2}\rfloor}{0}$$
$$+ \binom{\lceil\frac{k-1}{2}\rceil}{0}.\binom{\lfloor\frac{k-1}{2}\rfloor}{5} + \binom{\lceil\frac{k-1}{2}\rceil}{2}.\binom{\lfloor\frac{k-1}{2}\rfloor}{3} + \binom{\lceil\frac{k-1}{2}\rceil}{4}.\binom{\lfloor\frac{k-1}{2}\rfloor}{1},$$

$A_{13} = 0,$

$A_{14} = 0,$

$- - - - - - - - - - -$

$- - - - - - - - - - -$

$- - - - - - - - - - -.$

This leads to the following result.

Theorem 3.3. *If A_j represents the number of codevectors of hamming weight j in the binary code $C_k^1(n, k-1)$, where $n = 2k+1$ and $j \leq 2k+1$, then for $i \in \mathbb{N}$ and assuming $\binom{n}{r} = 0$ if $n < r$, we have*

$$A_{4i-1} = \left[\binom{\lceil\frac{k-1}{2}\rceil}{1}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-2} + \binom{\lceil\frac{k-1}{2}\rceil}{3}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-4} + \ldots + \binom{\lceil\frac{k-1}{2}\rceil}{2i-1}.\binom{\lfloor\frac{k-1}{2}\rfloor}{0}\right]$$
$$+\left[\binom{\lceil\frac{k-1}{2}\rceil}{1}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-3} + \binom{\lceil\frac{k-1}{2}\rceil}{3}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-5} + \cdots + \binom{\lceil\frac{k-1}{2}\rceil}{2i-3}.\binom{\lfloor\frac{k-1}{2}\rfloor}{1}\right],$$

$$A_{4i} = \left[\binom{\lceil\frac{k-1}{2}\rceil}{0}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i} + \binom{\lceil\frac{k-1}{2}\rceil}{2}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-2} + \ldots + \binom{\lceil\frac{k-1}{2}\rceil}{2i}.\binom{\lfloor\frac{k-1}{2}\rfloor}{0}\right]$$
$$+\left[\binom{\lceil\frac{k-1}{2}\rceil}{0}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-1} + \binom{\lceil\frac{k-1}{2}\rceil}{2}.\binom{\lfloor\frac{k-1}{2}\rfloor}{2i-3} + \cdots + \binom{\lceil\frac{k-1}{2}\rceil}{2i-2}.\binom{\lfloor\frac{k-1}{2}\rfloor}{1}\right],$$

$A_{4i-2} = 0,$

$A_{4i-3} = 0.$

By the MacWilliams Identity (Theorem 3.14, [9]), the following theorem follows.

Theorem 3.4. *If A_i and B_i denote the number of codevectors of weight i in the binary code $C_k^1(n, k-1)$, where $n = 2k+1$ and its dual code, then*

$$B_i = q^{-(k-1)} \sum_{j=0}^{n} A_j \sum_{s=0}^{n} \binom{j}{s}\binom{n-j}{i-s}(-1)^s(q-1)^{i-s}.$$

where A_j is given by Theorem 3.3.

4. Conclusion and scope for further study

The codes C_k^t are obtained from C_k by adding rows to the parity check matrix H_k. The new codes C_k^t are capable of correcting all solid bursts of certain length or less. The new codes are better codes from the point of view of correction of solid burst error. But the new codes are no longer reversible code and have less information rate. Further, we have studied solid burst error uncorrected probability of the codes C_k and C_k^t and weight distribution of the codes C_k and C_k^1 in binary case. The weight distribution for the codes for other cases remains a further study.

REFERENCES

[1] *C.A. Argyrides, P. Reviriego, D.K. Pradhan and J.A. Maestro*, Matrix-Based Codes for Adjacent Error Correction, *IEEE Transactions on Nuclear Science*, **57**(2010), No. 4, 2106–2111.

[2] *J. Arlat and W.C. Carter*, Implementation and Evaluation of a (b,k)-Adjacent Error-Correcting/Detecting Scheme for Supercomputer Systems, IBM J. Res. Develop., **28**(1984), No. 2, 159–169.

[3] *D.C. Bossen*, *b*-Adjacent Error Correction, IBM Journal of Research and Development, **14**(1970), No. 4, 402–408.

[4] *P. K. Das*, Error locating codes and extended Hamming code, Mathematicki Vesnik, **70**(2018), No. 1, 89–94.

[5] *T.A. Gulliver and V.K. Bhargava*, A systematic (16, 8) code for cor- recting double errors and detecting triple-adjacent errors, IEEE Trans. Computers, **42**(1993), No. 1, 109–112.

[6] *D.W. Jensen*, Block code to efficiently correct adjacent data and/or check bit errors, Patent number: US 6604222 B1, Date of Patent Aug 5, 2003, (www.google.com/patents/US6604222).

[7] *R.S. Katti*, Comments on 'A systematic (16, 8) code for correcting double errors and detecting triple-adjacent errors', IEEE Trans. Computers, **44**(1995), No. 12, 1472–1473.

[8] *S.K. Muttoo and S. Lal*, A reversible Code over GF(q), Kybernetika, **22**(1986), No. 1, 85–91.

[9] *W. W. Peterson, E. J. Weldon(Jr.)*, Error-Correcting Codes, 2nd edition, The MIT Press, Mass, 1972.