Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 01 (2024)

Solid Restricted Burst Error Locating and Correcting Linear Optimal Codes

Subodh Kumar[1], Tarun Lata[2], Hari Pratap[3], Gajraj Singh[4], Anshul Srivastava[5]
[1]Shyam Lal College, University of Delhi, Delhi, India
skumarmath@shyamlal.du.ac.in
[2]Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi, India
tarunlata1983@gmail.com
[3]P.G.D.A.V. College (Evening), University of Delhi, Delhi, India
haripratap@pgdave.du.ac.in
[4]Indira Gandhi National Open University, Delhi, India
gajrajsingh@ignou.ac.in
[5]Amity International Business School, Amity University, Noida, Uttar Pradesh, India
anshul_sriv@rediffmail.com
Corresponding author: - Hari Pratap

Abstract:- In this communication we are presenting the codes that deal with location of the solid restricted bursts errors occurring in byte oriented communication channels/ memory systems during the transmission of information. The code-length for such codes is assumed to be subdivided into bytes of unequal lengths. The necessary and sufficient conditions for the codes those are capable to locate the solid burst errors lying in a single byte. We also provide a construction technique of the solid restricted b_i-byte correcting non-binary linear optimal codes.
Keywords: *b_i-byte correcting codes, optimal codes, Syndromes, Byte oriented channels, Parity check matrix, Solid burst errors, restricted burst errors.*

1. Introduction

The main purpose of coding theory is construction of codes for error-free transmission of information by detecting and correcting errors that occur during data transmission. The nature of error and the communication channel are known to be related. Study of solid burst errors is an important area in coding theory. Schillinger [3] developed codes that correct solid burst errors in binary system. Solid burst errors can be found in various storage systems (viz. super computer storage system, semiconductor memory data). The primary goal of the early development of solid burst error correcting codes was to detect and correct solid bursts in binary memory systems. We present in this paper the concept of solid restricted burst (SRB Error), which is a modification of Schillinger's [3] definition of a solid burst. The coding theorists are supposed to provide the codes that are helpful to perform the error free transmission of the information through a communication channel. Initially, communication channels used the codes having the capability to detect and correct the errors occurring when the information is transmitted to one place to another place. The nature of occurrence of the errors may vary from a channel to another channel. Errors can be classified into a number of different categories, including random error, burst error, solid burst error, repeated burst error and key error. Abramson [1] obtained the codes that deal with the errors that are not in a random occurrence in other words these errors occur in clusters. At first, Fire [2] gave the concept of a particular type of clustered errors known as "burst errors". According to him "A burst of length *b* is considered as an n-tuple whose only non-zero components are confined to some *b* consecutive positions, the first and the last of which is non-zero". The vectors 0012100, 1010000, 0111000, 000201are examples of bursts of length 3 over GF(3). If all the components of a burst of length *b* are non- zero then this burst called a "solid burst". Initially, only those codes were developed that were capable to detect and correct the different types of errors. The error detecting codes use short number of parity check digits, but provide the information about the presence of an error and no other information. The error correcting codes help to remove the errors, but these codes have large number of parity check digits which effects the transmission rate of a channel where these are being used. Wolf and Elspas presented a novel idea called "location of errors" in order to prevent these issues. They were able to locate the error-causing portion of the code length by using the codes they had obtained. These codes have a parity check digit count that falls between the number of parity check digits with codes that can detect and correct these errors, respectively. In other words location is a midway of detection and correction. The study of the codes dealing with the solid bust errors was started by Schillinger [3]. Shiva and Sheng worked on multiple solid bursts. Subsequently, a class of solid burst error-correcting codes derived from a reversible code was obtained by Das. He gave the codes capable of correcting the solid bursts lying in sub-blocks. Etzion initiated to work on the byte oriented channels and obtained the codes that were capable of correct the solid bursts occurring in a byte. Chen [6] developed byte oriented error correcting codes. Recently, Tyagi and Lata [10] gave the perfect binary codes based bytes. Tyagi and Lata [11] introduced the restricted burst errors and in [11] they obtained the byte oriented restricted burst error correcting non-binary optimal codes. According to them a restricted burst is defined as:

Definition-1 A restricted burst of length *b* is a vector of length *n* whose all non-zero components are confined to some b consecutive positions, the first and last of which is nonzero with the restriction that all the non-zero consecutive positions contain same field element.

As SRB errors correcting non-binary linear codes for byte-oriented communication channels are of interest to us. The following defines a SRB error of length *b*:

Definition-2 A solid restricted burst of length *b* is defined as a *n*-tuple vector with non-zero entries in some b consecutive positions and zero elsewhere with the restriction that all the non-zero consecutive positions contain same field element.

For example: For q=3, n=4 and b=3, solid restricted burst of length 3 or less are 1110, 2220, 0111, 0222, 1100, 2200, 0110, 0220, 0011, 0022, 1000, 2000,0100, 0200, 0010, 0020, 0001, 0002.

Reciting that communication channel behaves differently in the presence of different types of errors. Due to having comparatively less parity check digits or in other words have good code rate, the codes capable of detect, correct or locate the solid restricted burst errors will help to enhance the transmission speed of a channel. In this paper, we assume that the whole code length n is subdivided into f number of bytes of different size. Here we obtain the codes that have capability to locate the solid restricted burst errors which are occurring in a single byte. We also provide a construction technique of the solid restricted b_i-byte correcting non-binary linear optimal codes. This paper is divided as follows: Present paper, introduction is given in Section 1. The necessary and sufficient bounds for the existence of SRB error locating codes are provided in Section 2. The solid restricted b_i-byte correcting non-binary linear optimal codes construction method is presented in Section 3.

2. Location of solid restricted burst errors in the bytes

In this part of the paper, we derive two conditions in the form of inequalities for the codes that have capability to locate the solid restricted burst (SRB) errors occurred in one the f bytes of different lengths. We will represent the SRB error locating codes by SRBEL-codes satisfy the following conditions:

(i) The syndromes due to the SRB errors in a single byte must be different from zero.

(ii) The syndromes due to the SRB errors in a single byte must be different from the syndromes due to the SRB errors in any other bytes.

To verify these codes, we have also provided an example.

Theorem 1: *An* $\left(n = \sum_{s=0}^{f} \beta_s, k\right)$ *SRBEL-code that locates the SRB errors of length b or less occurring in a single byte satisfies the following condition*

$$n - k \geq \log_q \{1 + f\, b(q-1)\}$$

Proof: The enumeration of all SRB errors that have to be located in a single byte will give the right hand side of the required result.

Let L be a set that contains all of the vectors that contain all of the restricted non-zero entries in at least b consecutive positions. We assert that no more than two SRB can be found in the same coset. We will use the fact that a code vector is the sum or difference of two bursts in a coset to support this claim. Let x_1 and x_2 be two elements of L lying in the same coset. The sum of x_1 and x_2 i.e. $x_1 + x_2$ or their difference

i.e. $x_1 - x_2$ is a SRB errors. Since x_1 and x_2 are the same coset therefore $x_1 + x_2$ or $x_1 - x_2$ is a code vector. This is a contradiction. Hence our assumption that x_1 and x_2 are in the same coset, was wrong. This proves our claim.

The number of elements in the set L is given by $b(q-1)$.

Given that the entire code length is divided into f different-length bytes. Therefore the total count of the locatable SRB errors is given as:

$$fb(q-1) \quad ...(1)$$

We will get the required result by putting the expression (1) less than q^{n-k}.

$$q^{n-k} \geq f\, b(q-1).$$

The necessary condition for the existence of SRBEL codes is provided by the following theorem.

Theorem 2: *The existence of an* $\left(n = \sum_{s=0}^{f} \beta_s\, , k\right)$ *SRBEL-code is ensured if the following inequality holds*

$$q^{n-k} > (b-1)(q-1)\left\{1 + \sum_{s=1}^{f-1}\sum_{i=1}^{b}\{(\beta_s - i + 1)(q-1)\}\right\}$$

Proof: Constructing a parity check matrix H with dimension $(n-k)\times n$ is sufficient to derive this theorem. We employ the same methodology as [12] in order to prove the Varshamov Gilbert-Sacks bound. Let us begin by creating the matrix using *n-k* tuples. Assume that we have selected all of the columns in the first *f-1* bytes as well as the first *j-1* columns in the f^{th}. If the conditions *(i)* - *(ii)* are met, the last byte β_s's j^{th} column h_j cannot be expressed as a linear combination of the columns $h_{j-1}, h_{j-2}, \ldots, h_{j-(b-1)}$, in accordance with the condition *(i)*. Stated differently, h_j should not equal the linear combination of *b-1* or fewer columns immediately before it i.e

$$h_j \neq \sigma_1 h_{j-1} + \sigma_2 h_{j-2} + \ldots + \sigma_{b-1} h_{j-(b-1)} \quad \ldots (2)$$

In the expression (2), the coefficients σ_i are similar as the non-zero entry in a solid restricted burst of length $b-1$ or less. Therefore the total number of linear combinations is given by

$$(b-1)(q-1) \quad \ldots (3)$$

Now according to the condition (*ii*), the column h_j should not be written in the form of linear sum of immediately preceding $b-1$ columns together with the linear sum of any b columns of any other of $j-1$ bytes i.e.

$$h_j \neq \sigma_1 h_{j-1} + \sigma_2 h_{j-2} + \ldots + \sigma_{b-1} h_{j-(b-1)} + \delta_1 h_1 + \delta_2 h_2 + \cdots + \delta_b h_b. \quad \ldots (4)$$

The number of σ_i is same as in the expression (2). That is $(b-1)(q-1)$. The enumeration of the δ coefficients is equivalent to find the solid restricted burst error in a vector of length β_s. Therefore the total number of linear sums due to the expression (4) is given by

$$(b-1)(q-1) \times \sum_{i=1}^{b}\{(\beta_s - i + 1)(q-1)\} \quad \ldots (5)$$

Since there are $j-1$ such bytes, therefore due to the condition (*ii*) the total number of linear sums is given by

$$(b-1)(q-1) \times \sum_{s=1}^{j-1}\sum_{i=1}^{b}\{(\beta_s - i + 1)(q-1)\} \quad \ldots (6)$$

Therefore h_j should not be equal to the linear sums given by expression (3) plus linear sums given by expression (6) i.e.

$$(b-1)(q-1)\left\{1+\sum_{s=1}^{j-1}\sum_{i=1}^{b}\{(\beta_s-i+1)(q-1)\}\right\} \quad \ldots (7)$$

Since there are atmost q^{n-k} cosets, there we have

$$q^{n-k} > (b-1)(q-1)\left\{1+\sum_{s=1}^{j-1}\sum_{i=1}^{b}\{(\beta_s-i+1)(q-1)\}\right\}$$

The required result can be obtained by replacing j by *f*. We conclude this section by giving an example of a code that is capable of locating SRB errors occurring in a single burst.

Example 2.1: *For $\beta_1 = 4$, $\beta_2 = 5$, $\beta_3 = 6$ and $\beta_4 = 5$, $b = 2$, consider a (20, 15) linear code over GF(3) with parity check matrix*

$$H = \begin{bmatrix} 1\,0\,0\,0 & 0\,0\,2\,0\,1 & 0\,1\,1\,1\,1\,1 & 1\,1\,1\,1\,0 \\ 0\,2\,0\,0 & 0\,1\,0\,1\,1 & 2\,0\,2\,2\,2\,2 & 0\,2\,1\,1\,0 \\ 0\,0\,1\,0 & 0\,0\,1\,1\,0 & 2\,0\,0\,1\,1\,1 & 1\,1\,1\,0\,1 \\ 0\,0\,0\,2 & 0\,2\,0\,2\,2 & 0\,1\,1\,0\,2\,2 & 2\,0\,1\,1\,1 \\ 0\,0\,0\,0 & 1\,2\,1\,0\,1 & 1\,2\,2\,0\,0\,1 & 0\,0\,0\,0\,0 \end{bmatrix}$$

The error patterns and syndromes for the parity check matrix given above can be obtained with the help of MS-EXCEL. It can be seen that all the syndrome corresponding to all solid restricted burst errors are non-zero and distinct in different bytes. This assures that this code locates all the solid restricted burst errors occurring in a single byte.

1. Construction of Solid Restricted b_i-Byte Correcting Non-Binary Linear Optimal Codes

In this section, firstly we give a construction method of an $\left(n=\sum_{s=0}^{f}\beta_s\,,k\right)$ byte oriented linear optimal code that can correct solid restricted bursts of length b_i or less within the bytes of size β_i, i = 1,2,3,...,m, over GF(q), q is prime and $q \geq 3$. The parity check matrix for the requisite code has been constructed by using the construction method given by Etzion [7] (p. 2554).

Let H = [H_1 H_2 H_3 ... H_m] be the parity check matrix for an $\left(n=\sum_{s=0}^{f}\beta_s\,,k\right)$ code C that corrects all solid restricted bursts of length b_i or less within the bytes of size β_i, i = 1,2,3,...,m, over GF(q), q is prime and $q \geq 3$, where $H_i = [h_{i1}, h_{i2}, h_{i3}, \ldots, h_{i\beta_i}]$ denotes the sub matrix of parity check matrix H corresponding to i-th byte of size β_i, $1 \leq i \leq m$.

Let $S(H_i)$ denote the set of distinct non-zero syndromes corresponding to all solid restricted bursts of length b_i or less within the bytes of size β_i for i = 1,2,3,...,m and S(H) denote the union of sets of distinct non-zero syndromes corresponding to all solid restricted bursts of length b_i or less within the bytes of size β_i for i = 1,2,3,...,m. In other words we can say that, $S(H) = \cup_{i=1}^{m} S(H_i)$

Clearly, each $S(H_i)$ is the union of the sets of the sum of b_i or less adjacent columns of H_i by multiplying the columns of H_i, i = 1,2,3,...,m with each non-zero field element viz. 1,2,3,...,(q − 1) ∈ GF(q). In other words,

$$S(H_i) = \{h_j + h_{j+1} + h_{j+2} + \cdots + h_{j+bi-1} : 1 \le j \le \beta_i - (b_i - 1)\} \cup \{h_j + h_{j+1} + h_{j+2} + \cdots + h_{j+bi-2} : 1 \le j \le \beta_i - (b_i - 2)\} \cup \ldots \cup \{h_j + h_{j+1} : 1 \le j \le \beta_i - 1\} \cup \{h_j : 1 \le j \le \beta_i\} \cup \{2h_j + 2h_{j+1} + 2h_{j+2} + \cdots +$$

size β_i, therefore each subset of $S(H_i)$ must be distinct for i = 1,2,3,...,m i.e

$$\{hj + hj{+}1 + hj{+}2 + \cdots + hj{+}bi{-}1 : 1 \le j \le \beta i - (bi - 1)\} \cap \{h_j + h_{j+1} + h_{j+2} + \cdots + h_{j+bi-2} : 1 \le j \le \beta_i - (b_i - 2)\} \cap \ldots. \cap \{h_j + h_{j+1} : 1 \le j \le \beta_i - 1\} \cap \{h_j : 1 \le j \le \beta_i\} \cap \{2h_j + 2h_{j+1} + 2h_{j+2} + \cdots + 2h_{j+bi-1} : 1 \le j \le \beta_i - (b_i - 1)\} \cap \{2h_j + 2h_{j+1} + 2h_{j+2} + \cdots + 2h_{j+bi-2} : 1 \le j \le \beta_i - (b_i - 2)\} \cap \ldots \cap \{2h_j + 2h_{j+1} : 1 \le j \le \beta_i - 1\} \cap \{2h_j : 1 \le j \le \beta_i\} \cap \ldots \cap \{(q-1)h_j + (q-1)h_{j+1} + \cdots + (q-1)h_{j+b}i_{-1} : 1 \le j \le \beta_i - (b_i - 1)\} \cap \{(q-1)h_j + (q-1)h_{j+1} + \cdots + (q-1)h_{j+b}i_{-2} : 1 \le j \le \beta_i - (b_i - 2)\} \cap \ldots \cap \{(q-1)h_j + (q-1)h_{j+1} : 1 \le j \le \beta_i - 1\} \cap \{(q-1)h_j : 1 \le j \le \beta_i\} = \Phi \quad \ldots (3.2)$$

The non-binary code C is a solid restricted b_i-byte correcting linear optimal code if every column vector of H of length r belongs to $S(H) = \cup_{i=1}^{m} S(H_i)$ and all the column vectors of H are distinct.

We now illustrate the construction of codes using the technique discussed above. In the following, several examples of solid restricted b_i-byte correcting non-binary linear optimal codes have been provided.

Example 3.1: For $\beta_1 = 4$, $\beta_2 = 9$, $\beta_3 = 5$ and $b_1 = 3$, $b_2 = 2$, $b_3 = 1$, consider a (18, 15) linear code over *GF* (5) *with parity check matrix*

$$H = \begin{bmatrix} 1\,1\,0\,0 & 1\,0\,1\,1\,1\,1\,1\,1\,1 & 1\,1\,1\,0\,0 \\ 2\,0\,1\,0 & 0\,1\,2\,0\,3\,0\,4\,0\,3 & 2\,3\,3\,1\,1 \\ 1\,0\,0\,1 & 2\,2\,0\,4\,0\,3\,0\,1\,1 & 2\,3\,4\,3\,4 \end{bmatrix} \quad \ldots (3.3)$$

This matrix has been obtained by the construction method given in the equations (3.1, 3.2) by taking $\beta_1 = 4$, $\beta_2 = 9$, $\beta_3 = 5$ and $b_1 = 3$, $b_2 = 2$, $b_3 = 1$ over *GF* (5). It can be easily verified from error-vector and syndrome table (Table 1 and 2) that the syndromes of different solid restricted bursts of length b_i or less with in the bytes of size β_i for i = 1, 2, 3 are distinct, showing thereby that the linear code that is the null space of the matrix (3.3), corrects all solid restricted bursts of length b_i or less within the bytes of size β_i, for i = 1, 2, 3 over *GF* (5).

Table-1: Error-Vector and Syndrome

Error-Vectors	Syndromes	Error-Vectors	Syndromes
1110 000000000 00000	231	0000 000011000 00000	233
0111 000000000 00000	111	0000 000001100 00000	243
1100 000000000 00000	221	0000 000000110 00000	241
0110 000000000 00000	110	0000 000000011 00000	232
0011 000000000 00000	011	0000 100000000 00000	102
1000 000000000 00000	121	0000 010000000 00000	012
0100 000000000 00000	100	0000 001000000 00000	120
0010 000000000 00000	010	0000 000100000 00000	104
0001 000000000 00000	002	0000 000010000 00000	130
2220 000000000 00000	412	0000 000001000 00000	103
0222 000000000 00000	222	0000 000000100 00000	140
2200 000000000 00000	442	0000 000000010 00000	101
0220 000000000 00000	440	0000 000000001 00000	131
0022 000000000 00000	044	0000 220000000 00000	223
2000 000000000 00000	242	0000 022000000 00000	214
0200 000000000 00000	200	0000 002200000 00000	443
0020 000000000 00000	020	0000 000220000 00000	413
0002 000000000 00000	002	0000 000022000 00000	411
3330 000000000 00000	143	0000 000002200 00000	431
0333 000000000 00000	333	0000 000000220 00000	432
3300 000000000 00000	113	0000 000000022 00000	414
0330 000000000 00000	330	0000 200000000 00000	204
0033 000000000 00000	033	0000 020000000 00000	024
3000 000000000 00000	313	0000 002000000 00000	240
0300 000000000 00000	300	0000 000200000 00000	203
0030 000000000 00000	030	0000 000020000 00000	210
0003 000000000 00000	003	0000 000002000 00000	201
4440 000000000 00000	324	0000 000000200 00000	230
0444 000000000 00000	444	0000 000000020 00000	202
4400 000000000 00000	334	0000 000000002 00000	212
0440 000000000 00000	440	0000 330000000 00000	332
0044 000000000 00000	044	0000 033000000 00000	341
4000 000000000 00000	434	0000 003300000 00000	112
0400 000000000 00000	400	0000 000330000 00000	142
0040 000000000 00000	040	0000 000033000 00000	144
0004 000000000 00000	004	0000 000003300 00000	124
0000 110000000 00000	114	0000 000000330 00000	123
0000 011000000 00000	132	0000 000000033 00000	141
0000 001100000 00000	224	0000 300000000 00000	301

Table-2

Error-Vectors	Syndromes	Error-Vectors	Syndromes
0000 000110000 00000	234	0000 030000000 00000	031
0000 003000000 00000	310	0000 000000040 00000	404
0000 000300000 00000	302	0000 000000004 00000	424
0000 000030000 00000	340	0000 000000000 10000	122
0000 000003000 00000	304	0000 000000000 01000	133
0000 000000300 00000	320	0000 000000000 00100	134
0000 000000030 00000	303	0000 000000000 00010	013
0000 000000003 00000	343	0000 000000000 00001	014
0000 440000000 00000	441	0000 000000000 20000	244
0000 044000000 00000	423	0000 000000000 02000	211
0000 004400000 00000	331	0000 000000000 00200	213
0000 000440000 00000	321	0000 000000000 00020	021
0000 000044000 00000	322	0000 000000000 00002	023
0000 000004400 00000	312	0000 000000000 30000	311
0000 000000440 00000	314	0000 000000000 03000	344
0000 000000044 00000	323	0000 000000000 00300	342
0000 400000000 00000	403	0000 000000000 00030	034
0000 040000000 00000	043	0000 000000000 00003	032
0000 004000000 00000	430	0000 000000000 40000	433
0000 000400000 00000	401	0000 000000000 04000	422
0000 000040000 00000	420	0000 000000000 00400	421
0000 000004000 00000	402	0000 000000000 00040	042
0000 000000400 00000	410	0000 000000000 00004	041

Example 3.2: For $\beta_1 = \beta_2 = 3$, $\beta_3 = 2$ and $b_1 = 3$, $b_2 = 2$, $b_3 = 1$, consider a (8, 5) linear code over *GF* (3) with parity check matrix

$$H = \begin{bmatrix} 1\,1\,1 & 1\,0\,1 & 0\,0 \\ 1\,1\,2 & 2\,1\,2 & 1\,0 \\ 1\,2\,1 & 2\,2\,0 & 0\,1 \end{bmatrix} \qquad \ldots (3.4)$$

This matrix has been obtained by the construction method given in the equations (3.1, 3.2) by taking $\beta_1 = \beta_2 = 3$, $\beta_3 = 2$ and $b_1 = 3$, $b_2 = 2$, $b_3 = 1$ over *GF*(3). It can be verified from error-vector and syndrome table (Table 3) that the syndromes of different solid restricted bursts of length b_i or less with in the bytes of size β_i for $i = 1, 2, 3$ are distinct, showing thereby that the linear code that is the null space of the matrix (3.4), corrects all solid restricted bursts of length b_i or less within the bytes of size β_i for $i = 1, 2, 3$ over *GF* (3).

Table-3 Error-Vector and Syndrome

Error-Vectors	Syndromes	Error-Vectors	Syndromes
111 000 00	011	000 011 00	102
110 000 00	220	000 100 00	122
011 000 00	200	000 010 00	012
100 000 00	111	000 001 00	120
010 000 00	112	000 220 00	202
001 000 00	121	000 022 00	201
222 000 00	022	000 200 00	211
220 000 00	110	000 020 00	021
022 000 00	100	000 002 00	210
200 000 00	222	000 000 10	010
020 000 00	221	000 000 01	001
002 000 00	212	000 000 20	020
000 110 00	101	000 000 02	002

Example 3.3: For $\beta_1 = \beta_2 = \beta_3 = 3$ and $b_1 = 1$, $b_2 = b_3 = 2$, consider a (9, 6) linear code over GF (3) with parity check matrix

$$H = \begin{bmatrix} 2\,1\,2 & 2\,0\,1 & 1\,0\,0 \\ 1\,0\,2 & 2\,1\,2 & 1\,1\,0 \\ 0\,0\,1 & 2\,0\,2 & 0\,1\,1 \end{bmatrix} \quad \dots (3.5)$$

This matrix has been obtained by the construction method given in the equations (3.1, 3.2) by taking $\beta_1 = \beta_2 = \beta_3 = 3$ and $b_1 = 1$, $b_2 = b_3 = 2$ over GF (3). It can be easily verified from error-vector and syndrome table that the syndromes of different solid restricted bursts of length b_i or less with in the bytes of size β_i for $i = 1, 2, 3$ are distinct, showing thereby that the linear code that is the null space of the matrix (3.5), corrects all solid restricted bursts of length b_i or less within the bytes of size β_i for $i = 1, 2, 3$ over GF (3).

2. Construction

As we know that the byte correcting optimal codes can enhance the rate of transmission of data and also improve the efficiency of the byte-oriented communication systems. The contribution of this paper is to find out the possibility of existence of the non-binary linear codes that can correct solid restricted burst of length b_1 or less in the first byte of size β_1, solid restricted burst of length b_2 or less in the second byte of size β_2 and so on and solid restricted burst of length b_i or less in the i-th byte of size β_i. We have been able to obtain some codes. This justifies the existence of such solid restricted b_i-byte correcting non-binary linear optimal code.

References

1. N.M. Abramson, A class of systematic codes for non-independent errors, IRE Trans. Information Theory, Vol.5 (1959), 150-157.
2. P. Fire, A class of multiple error correcting binary codes for non-independent errors, Sylvania Reports R SL-E-2, (1959).
3. A. G. Schillinger, A class of solid burst error correcting codes, Polytechnic Institute of Brooklyn, N.Y., Research Rept. PIBMRI, April (1964), 1223-1264.
4. S. G. S. Shiva, C. L. Sheng, Multiple solid burst-error-correcting binary codes, IEEE Transactions on Information Theory, Vol.15, (1969), 188-189.
5. W.W. Peterson, E.J. Weldon, (Jr.), Error Correcting Codes, 2nd Edition, M.I.T. Press, Cambridge, 1972.
6. C.L. Chen, Byte oriented error correcting codes with semiconductor memory systems, IEEE Trans.Comput., Vol.35 (1986), 646-648.
7. Tuvi Etzion, Perfect byte correcting codes, IEEE Trans. Information Theory, Vol.44 (1998), 3140-3146.
8. Tuvi Etzion, Construction for perfect 2-burst correcting codes, IEEE Trans. Information Theory Vol.47, Number 6 (2001), 2553 - 2555.
9. P. K. Das, Blockwise Solid Burst Error Correcting Codes, International Journal on Information Theory (IJIT), Vol. 1(1), 2012.
10. Vinod Tyagi, Tarun Lata, bi- Type perfect byte correcting binary codes, Global Journal of Pure and Applied Mathematics, Vol.11, Number 5 (2015), 2419-2427.
11. Vinod Tyagi, Tarun Lata, Restricted 2- Burst Correcting Non-Binary Optimal Codes, Journal of Combinatorics, Information and System Sciences, Vol.42, (2017), 143-152.
12. P. K. Das, A Class of Solid Burst Error Correcting Codes Derived from a Reversible Code, U.P.B. Sci. Bull., Series A, Vol. 80(4), 2018.
13. Vinod Tyagi, Tarun Lata, Restricted bi-Byte Correcting Non-Binary Optimal Codes, Discrete Mathematics, Algorithms and Applications, Vol.11, Number 2 (2019), 1950019:1-1950019:15