Mathematical Problems of Computer Science 48, 105--111, 2017.

Further Results for Encoding and Decoding Procedures of Asymmetric Low Magnitude Error Correcting Codes

Hamlet K. Khachatrian

Institute for Informatics and Automation Problems of NAS RA
e-mail: hamletkh@ipia.sci.am

Abstract

In this paper an implementation of encoding and decoding procedures for double ±1 error correcting optimal linear codes over rings Z_7 and Z_9 is presented.
Keywords: Error correcting codes, Asymmetric low magnitude error correcting codes, Encoding and Decoding procedures.

1. Introduction

Codes over finite rings, particularly over integer residue rings and their applications in coding theory, have been studied for a long time. Errors happening in the channel are basically asymmetrical; they also have a limited magnitude, and this effect is particularly applicable to flash memories. There have been a couple of papers regarding the optimal ±1 single error correcting codes over the alphabet Z_m [1, 2]. Also there are some papers regarding the construction of optimal double ±1 error correcting codes [3, 4]. Here, we propose to construct encoding and decoding algorithms for the optimal codes correcting double **±1** errors. In [5] you can see the construction of encoding and decoding procedures for the optimal linear code (12, 8) over ring Z_5, which was given by parity check matrix H_5:

$$H_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 2 & 3 & 4 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 3 & 2 & 4 & 4 & 2 & 3 & 2 & 4 & 4 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 3 & 2 & 4 & 4 & 2 & 0 & 4 \end{bmatrix}.$$

In this case the number of combinations for each code word that can be corrected is:

$$(1 + 12 * 2 + (12\ choose\ 2) * 4) = 289.$$

Implementation of codes over large alphabets is much more difficult compared with small alphabets. In this paper we construct encoding and decoding procedures for the codes (16, 12) and (20, 16) over rings Z_7 and Z_9, which are developed in [4]. Using this codes we can correct consequently 512 and 800 errors of type ±1 in any vectors from Z_7 and Z_9 with lengths 12 and 16 by adding only 4 parity check symbols.

2. Presentation of the Codes (16, 12) and (20, 16) over Rings Z_7 and Z_9

In [4] you can see the construction of optimal linear codes over Rings Z_7 and Z_9 correcting double ± 1 errors.

2.1 Code (16, 12) over Ring Z_7

Let a linear (16, 12) code over ring Z_7 be given by the following parity check matrix H_7:

$$H_7 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 1 & 1 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 4 & 3 & 6 & 6 & 3 & 4 & 2 & 4 & 3 & 6 & 6 & 3 & 4 & 2 & 1 & 6 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 3 & 6 & 6 & 3 & 4 & 2 & 0 & 0 \end{bmatrix}.$$

A linear code over ring Z_7 , with 12 information and 4 parity check symbols, given by the parity check matrix H_7 can correct up to two errors of the type ± 1, because H_7 has a property according to which all the syndromes resulting from adding and subtracting operations between any two columns of the matrix H_7 are different ($\pm h_i \pm h_j$ and $h_i \neq h_j$).
This code is optimal in the sense that it has a minimal possible number of parity check symbols. In this case the number of combinations for each code word that can be corrected is:

$$16 * 2 + (16 \, choose \, 2) * 4 = 512.$$

2.2 Code (20, 16) over Ring Z_9

The parity check matrix H_9 for an optimal linear code (20, 16) correcting double errors of the type ± 1 over ring Z_9 has the following form:

$$H_9 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 1 & 1 & 2 & 4 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 2 & 4 \\ 7 & 3 & 2 & 4 & 4 & 2 & 3 & 7 & 7 & 3 & 2 & 4 & 4 & 2 & 3 & 7 & 1 & 1 & 2 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 7 & 3 & 2 & 4 & 4 & 2 & 3 & 7 & 6 & 3 & 7 & 2 \end{bmatrix}.$$

A linear code over ring Z_9 , with 16 information and 4 parity check symbols, given by the parity check matrix H_9 can correct up to two errors of the type ± 1.

This code is optimal too in the sense that it has a minimal possible number of parity check symbols. In this case the number of combinations for each code word that can be corrected is:

$$(20 * 2 + (20 \, choose \, 2) * 4) = 800 .$$

In the next chapter we will construct encoding and decoding procedures for these two optimal linear codes.

3. Encoding and Decoding Procedures

3.1 Code (16, 12)

For encoding every message in Z_7 we must have the generator matrix G_7. For this we should construct a combinatorial equivalent matrix H'_7 from parity check matrix H_7 of the code (16, 12):

$$H'_7 = \begin{bmatrix} 1 & 0 & 0 & 0 & 5 & 2 & 5 & 1 & 5 & 2 & 5 & 0 & 1 & 1 & 6 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 & 5 & 5 & 0 & 6 & 4 & 1 & 4 & 6 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 4 & 1 & 6 & 5 & 5 & 6 & 5 & 5 \\ 0 & 0 & 0 & 1 & 1 & 5 & 5 & 2 & 3 & 1 & 1 & 3 & 0 & 6 & 2 & 3 \end{bmatrix}.$$

Here all syndromes will be different, too. We know the theorem, which says, that if $H' = [-P^T|I_{n-k}]$, then $G = [I_k|P]$ (the reverse statement is also true), where I_k is a $k * k$ identity matrix and P is a $k * (n-k)$ matrix,

$$GH'^T = 0. \tag{1}$$

Thus, we can construct the generator matrix G_7:

$$G_7 = \begin{bmatrix} 2 & 5 & 0 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 6 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 2 & 0 & 5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 6 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 1 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 6 & 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 6 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 6 & 3 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Encoding procedure:

In our scheme the message was presented by 12-tuples in Z_7. $v = (a_1, a_2, a_3, \dots, a_{12})$ is an arbitrary 12-tuple, and consider the vector u that is the linear combination of columns G_7 with a_i is the i^{th} coefficient.

$$u = vG = (c_1, c_2, c_3, c_4, a_1, a_2, a_3, \dots, a_{12}),$$

where the first 4 components of the code vector are the parity check symbols and the next 12 components are information symbols, where

$$c_j = \left(\sum_{i=1}^{k} a_i p_{ij}\right) mod 7. \tag{2}$$

Let us show the example to describe how we do these procedures.

Example.

Let (0 1 2 6 4 0 6 5 4 1 2 2) be the message vector in Z_7. From (2) we can obtain parity check symbols by multiplying this message vector with the columns of the matrix G_7.
For example, the first parity check symbol is c_1:
$c_1 = (0*2) + (1*5) + (2*2) + (6*6) + (4*2) + (0*5) + (6*2) + (5*0) + (4*6) + (1*6) + (2*1) + (2*6) = 0 + 5 + 4 + 1 + 1 + 0 + 5 + 0 + 3 + 6 + 2 + 5 = 4(mod 7)$.
(All operations are in Z_7.)
Similarly, we can find other 3 parity check symbols:

$$c_2 = 5, \; c_3 = 3, \; c_4 = 1.$$

After performing other multiple operations with matrix G_7 we obtain this encoded vector:
(4 5 3 1 0 1 2 6 4 0 6 5 4 1 2 2). As we can see in this code, the encoded message (codeword) has the length 16, from which the first 4 are parity check symbols, and the last 12 are information symbols.

Decoding procedure:

Now we will show how a decoding procedure will be implemented using the parity check matrix H'_7, if during the message sending process the errors occured in the codewords.
We will describe the decoding procedure by 3 steps:

1. Receiver multiplies the vector with every row of matrix H'_7 and gets the syndrome $\mathrm{S} = \mathbf{v}\mathrm{H}'$. If $\mathrm{S} = (0,0,0,0)$ then there were not any errors in the received vector.
2. If the calculated syndrome S is a nonzero vector, then there are some occurred errors. These codes can correct only up to two errors with magnitude ±1. We know that all possible syndromes of matrix H'_7 are different **(**$\pm h_i \pm h_j$ and $h_i \neq h_j$). After calculating the syndrome the receiver knows from which two columns of the matrix H'_7 the syndrome was resulted, consequently, it can find the two corresponding components of the vector, where the error was occurred and the direction of the error (if $+h_i$, then upward direction or if $-h_i$ downward direction). On the other hand, if in the table of syndromes we do not have the resulted syndrome, then we cannot correct this kind of errors.
3. After finding the error components the receiver adds or subtracts 1 from them and obtains the sent code vector $(c_1, c_2, c_3, c_4, a_1, a_2, a_3, \ldots, a_{12})$. So $(a_1, a_2, a_3, \ldots, a_{12})$ is our message vector.

An example.
(4 5 3 1 0 1 2 6 4 0 6 5 4 1 2 2) is an encoded vector from the previous example. Let 2 errors occur in the channel, and the receiver gets the vector (4 5 **2** 1 0 1 2 6 4 0 6 5 4 1 2 **1**). After performing multiple operations with rows of matrix H'_7 the receiver obtains the syndrome(6 3 1 4). Next from the table of syndromes it finds the corresponding columns, now they are 3 and 16. Hence, the syndrome (6 3 1 4) was resulted from adding a negated column 3 of matrix H'_7 to the negated column 16:

$$\begin{matrix} 0 & -1 & -1 \\ 0 & -4 & -4 \\ -1 & -5 & -6 \\ 0 & -3 & -3 \end{matrix} = \begin{matrix} \\ \\ \end{matrix} (mod 7) = (6\ 3\ 1\ 4)$$

(Because in Z_7 $0 = 7, -1 = 6, -2 = 5, -3 = 4, -5 = 2, -6 = 1$).
Hence, the error positions of encoded vector are 3 and 16 (both have a downward direction).

So, it adds 1 to *the* 3rd component and 1 to 16^{th} of vector (4 5 **2** 1 0 1 2 6 4 0 6 5 4 1 2 **1**) and obtains the sent encoded vector (4 5 3 1 0 1 2 6 4 0 6 5 4 1 2 2).
Consequently, the message vector (code word) is (0 1 2 6 4 0 6 5 4 1 2 2) as we have in the example of the encoding procedure.
Using this code we can find and correct all possible 512 errors of the type ± 1 in every vector over ring Z_7.

3.2 Encoding and Decoding for the Code (20, 16)

For the code (20, 16) over the ring Z_9 correcting double errors of the type ± 1 we can do the same encoding and decoding processes as we did for the code (16, 12). In this case, the parity check matrix H'_9 and the generator matrix G_9 will have the following form:

$$H'_9 = \begin{bmatrix} 1&0&0&0&6&7&8&5&0&6&6&0&7&3&5&4&7&4&7&4 \\ 0&1&0&0&0&6&0&2&4&7&1&4&1&1&1&1&7&4&3&3 \\ 0&0&1&0&6&4&5&6&6&3&0&6&3&5&6&3&7&1&4&5 \\ 0&0&0&1&1&4&1&5&2&1&6&8&6&3&0&6&4&1&7&0 \end{bmatrix},$$

$$G_9 = \begin{bmatrix} 3&0&3&8&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 2&3&5&5&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&4&8&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 4&7&3&4&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&5&3&7&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 3&2&6&8&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 3&8&0&3&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&5&3&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 2&8&6&3&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 6&8&4&6&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 4&8&3&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 5&8&6&3&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 2&2&2&5&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 5&5&8&8&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 2&6&5&2&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 5&6&4&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

Unlike the previous case for the code (16, 12) over ring Z_7, in this case the message was presented by 16-tuples in Z_9. The encoded vector u (codeword) has a length 20:

$$u = vG = (c_1, c_2, c_3, c_4, a_1, a_2, a_3, \ldots, a_{16}),$$

where the first four are the parity check symbols: $c_j = \left(\sum_{i=1}^{k} a_i p_{ij}\right) mod 9$ and the next 16 are information symbols.

Using this code we can find and correct all possible 800 errors of the type ± 1 in every vector over ring Z_9.

4. Conclusion

In this paper an implementation of encoding and decoding procedures of optimal (16, 12) and (20, 16) linear codes over ring Z_7 and Z_9 correcting double ± 1 errors is presented. We propose that this approach can be extended for implementation of similar procedures for the optimal codes over other rings Z_n and the research in this direction will follow.

References

[1] S. Martirossian, "Single error correcting close packed and perfect codes", *Proc.1st INTAS Int. Seminar Coding Theory and Combinatorics*, Armenia, pp. 90-115, 1996.

[2] H. Kostadinov, N.Manev and H.Morita, "On ± 1 error correctable codes", *IEICE Trans.Fundamentals*, vol. E93-A, pp. 2578-2761, 2010.

[3] G. Khachatrian and H. Morita, "Construction of optimal 1 double error correcting linear codes over ring Z5 ", *3rd International Workshop on Advances in Communications, Boppard*, Germany, pp. 10-12, May 2014.

[4] G. Khachatryan and H. Khachatryan, "Construction of double ± 1 error correcting linear optimal codes over rings Z_7 and Z_9" , *Mathematical Problems of Computer Science*, vol. 45, pp. 106—110, 2016.

[5] H. Khachatryan, "Encoding and decoding procedures for double ±1 error correcting linear code over ring *Z*5", *Mathematical Problems of Computer Science*, vol. 43, pp. 57—61, 2015.

Submitted 17.09.2017, accepted 04.12.2017.

Այլ արդյունքներ ասիմետրիկ փոքր մեծությամբ սխալներ ուղղող կոդերով կոդավորման և ապակոդավորման ալգորիթմների համար

Հ. Խաչատրյան

Ամփոփում

Այս հոդվածի շրջանակներում ներկայացված են կոդավորման և ապակոդավորման ալգորիթմները Z_7 և Z_9 օղակներում կառուցված ասիմետրիկ փոքր ամպլիտուդայով սխալներ ուղղող կոդերի համար:

Алгоритмы кодирования и декодирования для кодов исправляющих асимметричные двойные ошибки

Г. Хачатрян

Аннотация

В данной статье представлены алгоритмы кодирования и декодирования **для кодов** в кольцах Z_7 и Z_9 исправляющих двойные асимметричные ошибки.