Shortened Weight Spectrum of [45,42,3] Modified Generalized Weighted Sum Code for Error Control Over GF(8)

Peter Farkaš, *Member IEEE* and M. Patarák and Martin Rakús, *Student member IEEE*

Abstract— **In [1] a new family of error detection codes called Weighted Sum Codes was proposed. In [2] it was noted, that these codes are equivalent to lengthened Reed Solomon Codes, resp. shortened version of lengthened Reed Solomon Codes over $GF(2^{(h/2)})$. It was also shown, that it is possible to use these codes for an error correction of one error in the codeword over $GF(2^{(h/2)})$. In [3] and [4] a generalization of Weighted Sum Codes was done which allows the correction of more than two erasures in each codeword. In [5] a new family of error – correcting modified Generalized Weighted Sum Codes was presented.**

I. Introduction

One form of the Generalized Weighted Sum Codes constructed over finite fields in [3] possess the **H** matrix (1) where **I** is an identity submatrix and α is a primitive element of the finite field $GF(q)$.

In general the matrix defined above will have sets of $2t$ dependent columns so the minimum distance of the linear code it defines is at most $2t$, and the code is therefore not $2t$ - error - correcting. However, it does have the slightly weaker property defined below:
Definition:

A code is called t - *information - error - correcting* if it can correct all instances of up to t errors per received word, provided that no errors occur in the check symbols.

The code defined by (1) has this property. To see this, note that any $2t$ columns of **A** form a Vandermonde matrix, so these columns are independent. Now consider the set of all error patterns of weight at most t in which all the non-zero components are in the first $q-1$ symbols (the information symbols). If any two of these error patterns have the same syndrome there are two distinct linear combinations of t columns of **A** which are the same, and hence a linear combination of $2t$ columns is equal to zero. This contradicts the Vandermonde property of **A**.

Hence the error patterns described above may all be chosen as coset leaders, so that all received words with these errors are correctly decoded. The error correction may be done via a syndrome look-up table or by solving the set of syndrome equations. If Y_i denotes the magnitude of the ith error and $X_{i'}$ is its locator, these equations are:

$$\begin{aligned} S_0 &= Y_1 + Y_2 + \cdots + Y_t \\ S_1 &= \sum_{i=1}^{t} Y_i X_{i'} \\ S_{2t-1} &= \sum_{i=1}^{t} Y_i X_{i'}^{2t-1} \end{aligned} \quad (2)$$

Here, error Y_i occurs in the position i' corresponding to the entry $\alpha^{i'}$ in row 2 of **H**, so X_i is just the field element $\alpha^{i'}$.

The codes introduced in [3] for erasure correction can have **H** matrices of the following form:

$$\mathbf{H} = [\mathbf{A} \cdots \mathbf{AI}] \quad (3)$$

where the **A** submatrix can be repeated more than once. These codes cannot be used for error correction because the position of the errors cannot be evaluated exactly from similar control equations as (2). In Section II a modification is introduced which overcomes this disadvantage and a class of single error correcting codes is introduced. In Section III a similar class of conditionally double error correcting codes is introduced.

II. Single Error Correcting Codes Based on Generalized Weighted Sum Codes

A new family of codes which allow the correction of one error in each codeword was described in [5] using the following **H** matrix:

$$\mathbf{H} = \left[\begin{array}{cccccccc} \mathbf{A} & \mathbf{A} & \cdots & \mathbf{A} & \cdots & \mathbf{A} & & \\ & & & & & & \mathbf{I} & \\ \bar{\mathbf{a}}_0 & \bar{\mathbf{a}}_1 & \cdots & \bar{\mathbf{a}}_j & \cdots & \bar{\mathbf{a}}_{q-2} & & \end{array} \right] = [\mathbf{BI}] \quad (4)$$

where the vectors: $\bar{\mathbf{a}}_0 = (1,1,\ldots,1)$, $\bar{\mathbf{a}}_1 = (\alpha,\alpha,\ldots,\alpha)$, $\bar{\mathbf{a}}_2 = (\alpha^2,\alpha^2,\ldots,\alpha^2),\ldots,$ $\bar{\mathbf{a}}_{q-2} = (\alpha^{q-2},\alpha^{q-2},\ldots,\alpha^{q-2})$ all have $q-1$ coordinates and **A** is:

$$\mathbf{A} = \left[\begin{array}{cccccc} 1 & \cdots & 1 & 1 & 1 & 1 \\ \alpha^{q-2} & \cdots & \alpha^3 & \alpha^2 & \alpha^1 & \alpha^0 \end{array} \right] \quad (5)$$

or the positions of the vectors $\bar{\mathbf{a}}_i$ may be permuted freely, so that, for example:

P. Farkaš is with the Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak Technical University in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia, IEEE Log Number 3495181,e-mail:farkas@ktl.elf.stuba.sk

Martin Patarák is a graduate student of Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak Technical University in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia

M. Rakús is with the Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak Technical University in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia, IEEE Log Number 40254330,e-mail:rakus@ktl.elf.stuba.sk

$$\mathbf{H} = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \cdots & 0 \\ \alpha^{(q-2)} & \cdots & \alpha^3 & \alpha^2 & \alpha^1 & \alpha^0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha^{(q-2)(2t-1)} & \cdots & \alpha^{3(2t-1)} & \alpha^{2(2t-1)} & \alpha^{2t-1} & \alpha^0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = [\mathbf{AI}] \quad (1)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{A} & \cdots & \mathbf{A} & \cdots & \mathbf{A} & \\ & & & & & & \mathbf{I} \\ \bar{\mathbf{a}}_{q-2} & \bar{\mathbf{a}}_{q-3} & \cdots & \bar{\mathbf{a}}_j & \cdots & \bar{\mathbf{a}}_0 & \end{bmatrix} \quad (6)$$

or:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{A} & \cdots & \mathbf{A} & \cdots & \mathbf{A} & \\ & & & & & & \mathbf{I} \\ \bar{\mathbf{a}}_1 & \bar{\mathbf{a}}_0 & \cdots & \bar{\mathbf{a}}_j & \cdots & \bar{\mathbf{a}}_{q-2} & \end{bmatrix} \cdots \quad (7)$$

Theorem 1: The codes given by (4) are one error correcting.

Proof: If the error has occurred in an information position then the syndromes for the code defined by (4) are:

$$S_0 = Y_1 \quad (8)$$
$$S_1 = Y_1\alpha^i \quad (9)$$
$$S_2 = Y_1\alpha^j \quad (10)$$

It is obvious, that (8) gives the value of the error, (9) determines that the error has occurred in the position corresponding to the locator α^i in one of the submatrices $\mathbf{A}$ and j in (10) determines that the error position is one covered by the vector $\bar{\mathbf{a}}_j$. Solving these equations we get the needed information about the position and the value of the error:

$$Y_1 = S_0 \quad (11)$$
$$\alpha^i = S_1/S_0 \quad (12)$$
$$\alpha^j = S_2/S_0 \quad (13)$$

If the error has occurred in a control position, then only one syndrome is different from zero and that syndrome determines the position of the error and also the value of the error.

Corollary: Any code matrix having an $\mathbf{H}$ matrix with "permuted" vectors $\bar{\mathbf{a}}_0, \bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2$ e. g. (6) or (7) etc. is one error correcting.

Proof: is straightforward.

The "expansion" made in (4) can be continued further:

$$\mathbf{H} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{B}_j & \cdots & \mathbf{B}_{q-2} & \\ & & & & & & \mathbf{I} \\ \bar{\mathbf{b}}_0 & \bar{\mathbf{b}}_1 & \cdots & \bar{\mathbf{b}}_j & \cdots & \bar{\mathbf{b}}_{q-2} & \end{bmatrix} = [\mathbf{CI}] \quad (14)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_j & \cdots & \mathbf{C}_{q-2} & \\ & & & & & & \mathbf{I} \\ \bar{\mathbf{c}}_0 & \bar{\mathbf{c}}_1 & \cdots & \bar{\mathbf{c}}_j & \cdots & \bar{\mathbf{c}}_{q-2} & \end{bmatrix} = [\mathbf{DI}] \quad (15)$$

and where the vectors $\bar{\mathbf{b}}_i, \bar{\mathbf{c}}_i, \ldots$ etc. have all their components equal to α^i and the length of $\bar{\mathbf{b}}_i$ is $(q-1)$, the length of the vectors $\bar{\mathbf{c}}_i$ is $(q-1)$ etc.. We will call the codes define by (4) *squared* or *of exponent 2*, the codes given by (14) *cubed* or *of exponent 3* and codes given by (15) *quadrupled* or *of exponent 4*, and of exponent e for the eth expansion. It is obvious, that such codes exist for any positive integer i, and all such codes can correct one symbol error in a codeword.

The code rate of the new single error correcting codes of exponent e is:

$$R = \frac{(q-1)^e}{(q-1)^e + e + 1} \quad (16)$$

The code rate is higher than the code rate of ordinary Reed Solomon codes correcting one error over the same field for any $e > 0$. For example in GF(8) the code rate of ordinary RS codes correcting one error is $6/7 \simeq 0.86$ and if $e = 5$, then from (16) $R \simeq 0.9996$. This improvement of code rate has a very small cost of slightly increased decoder complexity. Specifically, in the chosen example it means that the decoder, in order to get the location of the error, must evaluate 4 syndromes more plus 4 divisions in GF(8).

IV. Shortened weight spectra

The weight spectrum known also as the weight distribution is an important characteristics of the code. It could be used for error performance evaluation and other purposes [6],[7]. The weight spectrum could be presented as a set of constants a_i. Each element of the set a_i represents the number of codewords with Hamming weight of i. The code distance is identical to the nonzero a_i with minimum value of i. It results from the linearity of the analyzed codes. In some cases the complete weight spectrum of an analyzed code can not be calculated due to the insufficient available computing power. For instance in order to calculate the complete weight spectrum of the GWSC code as was described in [3] it is necessary to search $8^6 = 262144$ codewords. This code was constructed over $GF(8)$. To do the same analysis in larger field e.g. $GF(16)$ it would mean to search $16^{14} \doteq 7.2.10^{16}$ codewords. Even if we dispose with the computing power enabling us to search 10^9 codewords per second, it will take approx. 2 years to do a complete search.

$$
\mathbf{G} = \begin{bmatrix}
1000000000000000000000000000000000000000000000000 & 1 & 1 & 1 \\
0100000000000000000000000000000000000000000000000 & 1 & \alpha & 1 \\
0010000000000000000000000000000000000000000000000 & 1 & \alpha^2 & 1 \\
0001000000000000000000000000000000000000000000000 & 1 & \alpha^3 & 1 \\
0000100000000000000000000000000000000000000000000 & 1 & \alpha^4 & 1 \\
0000010000000000000000000000000000000000000000000 & 1 & \alpha^5 & 1 \\
0000001000000000000000000000000000000000000000000 & 1 & \alpha^6 & 1 \\
0000000100000000000000000000000000000000000000000 & 1 & 1 & \alpha \\
0000000010000000000000000000000000000000000000000 & 1 & \alpha & \alpha \\
0000000001000000000000000000000000000000000000000 & 1 & \alpha^2 & \alpha \\
0000000000100000000000000000000000000000000000000 & 1 & \alpha^3 & \alpha \\
0000000000010000000000000000000000000000000000000 & 1 & \alpha^4 & \alpha \\
0000000000001000000000000000000000000000000000000 & 1 & \alpha^5 & \alpha \\
0000000000000100000000000000000000000000000000000 & 1 & \alpha^6 & \alpha \\
0000000000000010000000000000000000000000000000000 & 1 & 1 & \alpha^2 \\
0000000000000001000000000000000000000000000000000 & 1 & \alpha & \alpha^2 \\
0000000000000000100000000000000000000000000000000 & 1 & \alpha^2 & \alpha^2 \\
0000000000000000010000000000000000000000000000000 & 1 & \alpha^3 & \alpha^2 \\
0000000000000000001000000000000000000000000000000 & 1 & \alpha^4 & \alpha^2 \\
0000000000000000000100000000000000000000000000000 & 1 & \alpha^5 & \alpha^2 \\
0000000000000000000010000000000000000000000000000 & 1 & \alpha^6 & \alpha^2 \\
0000000000000000000001000000000000000000000000000 & 1 & 1 & \alpha^3 \\
0000000000000000000000100000000000000000000000000 & 1 & \alpha & \alpha^3 \\
0000000000000000000000010000000000000000000000000 & 1 & \alpha^2 & \alpha^3 \\
0000000000000000000000001000000000000000000000000 & 1 & \alpha^3 & \alpha^3 \\
0000000000000000000000000100000000000000000000000 & 1 & \alpha^4 & \alpha^3 \\
0000000000000000000000000010000000000000000000000 & 1 & \alpha^5 & \alpha^3 \\
0000000000000000000000000001000000000000000000000 & 1 & \alpha^6 & \alpha^3 \\
0000000000000000000000000000100000000000000000000 & 1 & 1 & \alpha^4 \\
0000000000000000000000000000010000000000000000000 & 1 & \alpha & \alpha^4 \\
0000000000000000000000000000001000000000000000000 & 1 & \alpha^2 & \alpha^4 \\
0000000000000000000000000000000100000000000000000 & 1 & \alpha^3 & \alpha^4 \\
0000000000000000000000000000000010000000000000000 & 1 & \alpha^4 & \alpha^4 \\
0000000000000000000000000000000001000000000000000 & 1 & \alpha^5 & \alpha^4 \\
0000000000000000000000000000000000100000000000000 & 1 & \alpha^6 & \alpha^4 \\
0000000000000000000000000000000000010000000000000 & 1 & 1 & \alpha^5 \\
0000000000000000000000000000000000001000000000000 & 1 & \alpha & \alpha^5 \\
0000000000000000000000000000000000000100000000000 & 1 & \alpha^2 & \alpha^5 \\
0000000000000000000000000000000000000010000000000 & 1 & \alpha^3 & \alpha^5 \\
0000000000000000000000000000000000000001000000000 & 1 & \alpha^4 & \alpha^5 \\
0000000000000000000000000000000000000000100000000 & 1 & \alpha^5 & \alpha^5 \\
0000000000000000000000000000000000000000010000000 & 1 & \alpha^6 & \alpha^5 \\
0000000000000000000000000000000000000000001000000 & 1 & 1 & \alpha^6 \\
0000000000000000000000000000000000000000000100000 & 1 & \alpha & \alpha^6 \\
0000000000000000000000000000000000000000000010000 & 1 & \alpha^2 & \alpha^6 \\
0000000000000000000000000000000000000000000001000 & 1 & \alpha^3 & \alpha^6 \\
0000000000000000000000000000000000000000000000100 & 1 & \alpha^4 & \alpha^6 \\
0000000000000000000000000000000000000000000000010 & 1 & \alpha^5 & \alpha^6 \\
0000000000000000000000000000000000000000000000001 & 1 & \alpha^6 & \alpha^6
\end{bmatrix} \tag{17}
$$

One possibility how to overcome this problem is to calculate *shortened weight spectrum*. It is based on the property of a weight spectrum described above. If the code is constructed to correct e.g 4 errors, the most important part of the weight spectra include weights: $a_0, a_1, \ldots, a_5$. Therefore it is necessary to generate all information words with weight up to 5. The rest of the spectra is not calculated since it doesn‘t bear the most significant information regarding correction capabilities of the analyzed code which is already contained in the weight spectrum entries with lower indexes. The correction capabilities of large codes could be analyzed this way. For instance modified GWSC code with $n = 45, k = 42$ for one error correction described by its $\mathbf{G}$ matrix (17) has the shortened weight spectrum (18). Computing time for calculation of shortened weight spectrum of a code (17) was several minutes. To calculate a complete weight spectrum of such code is today practically impossible.

$$
\begin{aligned}
a_0 &= 1 \\
a_3 &= 15092 \\
a_4 &= 1279047 \\
a_5 &= 85250592 \\
\textstyle\sum_{i=6}^{45} a_i &= 2^{126} - 86544732
\end{aligned} \tag{18}
$$

IV. Conclusion

The shortened weight spectrum of the analyzed code can be an useful information which gives us the code distance. The main advantage of the mentioned method of the calculation of the shortened weight spectrum is that this way it is possible to obtain the minimum code distance of a large codes in reasonable time interval, for which to calculate their complete weight spectra is practically impossible. For our [45,42,3] code constructed over *GF(8)* we can see that the code distance is 3.

References

[1] A. J. McAuley, "Weighted sum codes for error detection and their comparison with existing codes," *IEEE/ACM Transaction on Networking,* vol. 2, no. 1, pp. 16-22, 1994.

[2] P. Farkaš, "Comments on "Weighted sum codes for error detection and their comparison with existing codes," *IEEE/ACM Transaction on Networking,* vol. 3, no. 2, pp. 222-223, 1995.

[3] P. Farkaš, M. Rakús. "Weight spectra of two and three erasure correcting weighted sum codes over GF(8)," *Proc. DSP'99*, Herľany, pp. 1-3, September 1999.

[4] P. Farkaš, M. Rakús. "Weight Distribution of Some Weighted Sum Codes for Erasure Correction," *Proc. TSP'99*, Brno, pp. 106-190, September 1999.

[5] P. Farkaš, J. Baylis, "Modified generalized weighted sum codes for error control", *Proc. Communication Theory and applications*, pp. 35-36, July 1999.

[6] A. M. Michelson, A.H. Levesque, "Error–control techniques for digital communication", *J. Wiley & Sons*, New York, 1986.

[7] J. Bland, J. Baylis, "Minimum distance as a combinatorial search problem", *Proc. Communication Theory and applications*, pp. 336-337, July 1995.