VOL. 14, NO. 19, OCTOBER 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

PERFORMANCE OF BASIC GEOMETRIC CODES OVER FADING CHANNELS

Osama Nashwan, Samer Issa and Khalil Saleh
Electrical Engineering Department, Faculty of Engineering, Applied Science Private University, Amman, Jordan
E-Mail: os_nashwan@asu.edu.jo

ABSTRACT

Forward error correcting codes while having good efficiency on normal channels, their performance is very poor over noisy (fading) channels. A new group of codes, called Geometric Codes that has a greater efficiency over fading channels has been designed. This paper presents the simulated results for Basic Geometric Code and studies the effects of its various parameters such as the number of bits per symbol, the number of parity lines and the code rate.

Keywords: geometric codes, fading channels, code rate, parity symbol.

1. INTRODUCTION

Most error correcting codes perform well on channels with low error probability, but their efficiency drops as the channel becomes noisy (fades). Many techniques have been presented to fix this problem, including using adjustable rate codes. In this paper, we present a group of codes called the Geometric Codes, and show that though geometric codes are not optimal; they outperform many codes, particularly over noisy or fading channels (input error rates of 10^{-2} to 10^{-3}). We have tested our codes on the systems presented in [2] [3] [4] [5] [6] [7] and [8] and have gotten significant improvement.

We presented the encoding and decoding algorithms as well as possible hardware implementations. We studied the performance of the Basic Geometric Code over a noisy channel. We then presented the simulation results for this code and have shown that the code performance does improve with the increase of the number of bits per symbol and with the coding rate.

2. BASIC GEOMETRIC CODES

Figure-1 shows a typical code block. The data bits are arranged in k rows of w bits. The r rows of parity bits are placed below the data block. The coding efficiency is k/k+r. We will describe the binary basic geometric code and later extend it for the nonbinary case.

In the Binary Basic Geometric Code, the location of each code bit is identified with a vertex of a uniformed lattice in the plane. Slopes of the form $1/m_1$, $1/m_2$, $1/m_3$, …, $1/m_r$ are chosen. To determine the jth parity row, a line having the slope $1/m_j$ is drawn through each data bit. The lines wrap cyclically upon reaching the left end of the row as though the block formed a cylinder. These lines are extended into the parity block, and each terminates at a particular parity bit in the jth parity row. This bit is determined so that the sum (mod 2) of the data bits on the line plus the single parity bit is zero. Thus the jth parity row is not affected by the other parity rows.

It follows from the construction that each data bit has r orthogonal estimators, corresponding to the r lines passing through it. Thus, if r is even, the code can correct r/2 errors by majority logic, while if r is odd (r-1)/2 errors may be corrected [1]

We impose a constraint on the slopes: no triangle with vertices at the bit positions and sides with slopes equal to $1/m_1$, $1/m_2$, $1/m_3$ should be constructible.

It can be shown that the parity condition stated above leads to the equation:

$$\begin{pmatrix} 1 & x^{m_1} & x^{2m_1} & \cdots & x^{(k-1)m_1} \\ 1 & x^{m_2} & x^{2m_2} & \cdots & x^{(k-1)m_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x^{m_r} & x^{2m_r} & \cdots & x^{(k-1)m_r} \end{pmatrix} \begin{pmatrix} d_1(x) \\ d_2(x) \\ \vdots \\ d_k(x) \end{pmatrix} = \begin{pmatrix} x^{-m_1} & x^{-2m_1} & x^{2m_1} & \cdots & x^{-rm_1} \\ x^{-m_2} & x^{-2m_2} & x^{2m_2} & \cdots & x^{-rm_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x^{-m_r} & x^{-2m_r} & x^{2m_r} & \cdots & x^{-rm_r} \end{pmatrix} \begin{pmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_r(x) \end{pmatrix}$$

The above equations are in the ring of polynomials modulo x^w+1. Because this is not a field, matrix inversion is not possible; however, the structure of the matrix does permit a solution, which is expressible as:

$$c_i(x) = \sum_{i=1}^{k} d_i(x) q_{ij}(x) \qquad mod\ x^w + 1$$

where the $\{q_{ij}(x)\}$ are polynomial that depend upon the slopes. The above derermination of the parity polynomial shows that the jth parity bit vector is the sum (mod 2) of the outputs of the k tapped cyclic shift registers Imp. lementation is, therefore, straightforward [1].

In this code, the parity symbols are functions of the data symbols only. That is, the entries of a parity line do not affect the entries of another. Hence, the order of calculating parity symbols is not important. We decided to calculate the three parity lines simultaneously.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

To encode a block of data, we read the data from the data file. Then we insert an all-zero block before the data block and another all zero block after it, as shown in Figure-1. The encoding of the new block is done, as shown in Figure-2. Here we start with the first symbol in the first data line $d_i(\lambda+1)$, and calculate all parity symbols affecting it. Then we calculate the parity symbols affecting the second symbol in the first parity line, an so on until we encode all the data symbols, as shown in Figure-3. For Example, for the ith symbol in the first data line the equation to calculate the parity symbols affecting it are;

$$p_1(i-3) = -[d_1(i) + d_2(i-1) + d_3(i-2)]$$
$$p_2(i-8) = -[d_1(i) + d_2(i-2) + d_3(i-4)]$$
$$p_1(i-35) = -[d_1(i) + d_2(i-7) + d_3(i-14)]$$

Note that as shown in Figure-1, we need $h+\lambda$ lines of ba lock, and λ is the length of the all-zero block. For the particular code discussed in this paper $\lambda = 21$. Therefore, the parity lines are longer than the data lines. This difference in length affects the rate of the code. To minimize this effect on the code rate, we choose $h \gg \lambda$.

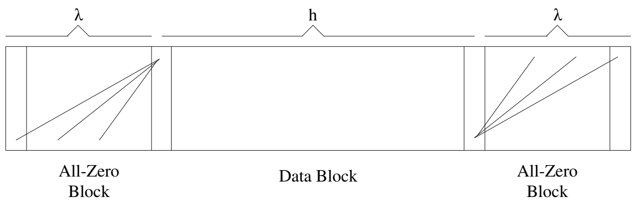

Figure-1. Zero Augmentation of Data Block.

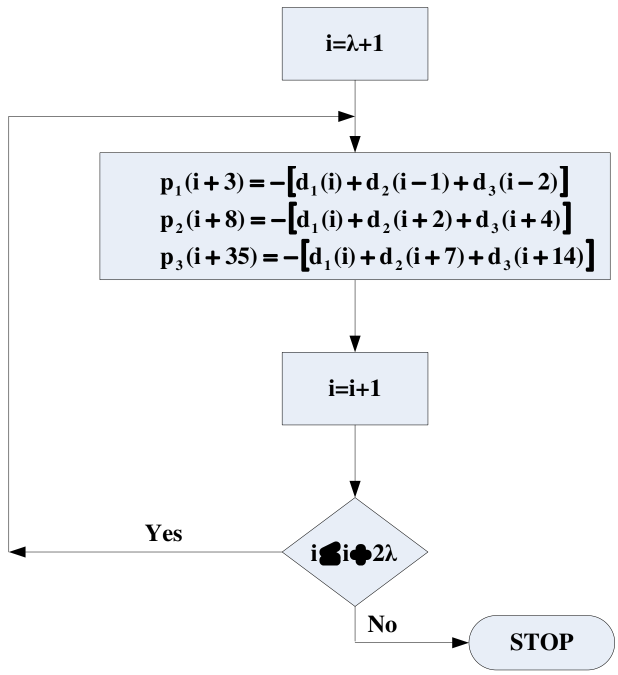

Figure-2. Flow Diagram for the Encoder of rate ½ BGC with 3-data lines.

The decoder consists of two major parts, one for error detection and the other for error correction. For error detection, we start with initializing the error pattern matrix to all-zeros. The error pattern matrix (EP) is a matrix of size (k+r, h), whose entries represent the number of times each symbol in the transmitted block is thought to be in error.

As in the encoding process, we start with the first data symbol of the first data line, $d_i(\lambda+1)$, and check whether errors have occurred on the line of slope 1, starting at $d_i(\lambda+1)$. Then we check if errors have occurred on the line of slope one starting at the second symbol in the first data line, $d_i(\lambda+2)$, and so on. In other words, we detect errors using the first parity line first. To determine whether an error did occur on a line of slope one passing through the ith symbol in the first data line, we check if the sum modulo N of the data symbols along that line and the corresponding parity symbol is equal to zero or not. Here N is the size of the alphabet used, $N = 2^m$, where m is the number of bits per symbol. As shown in Figure-3 if the sum is equal to zero, we assume no errors have occurred. While if the sum is not zero, we assume that at least one error did occur along the line in question. When errors are detected the appropriate positions in the error pater matrix are incremented. This procedure was repeated for all other slopes as shown in Figure-3.

Since every symbol was encoded by r parity lines in this particular example r=3, an error in a particular symbol will be detected by r slopes. Hence its position in the EP matrix would have a value of r. This would be true only if no other errors that would cancel the effect of this error did occur along any of the lines associated with the symbol in question.

To correct errors, we start by inspecting the entries of the Ep matrix. When an entry that has a value of r is found, the corresponding data symbol is assumed to be in error. The error is then found by finding the value that satisfies the most decoding equations shown in Figure-4. For example, if the EP entry corresponding to the ith symbol in the first data line has the value of r. Then the error value in this example would have to satisfy the following equations:

$$d_1(i) + d_2(i-1) + d_3(i-2) + p_1(i-3) + E = 0$$
$$d_1(i) + d_2(i-2) + d_3(i-4) + p_1(i-8) + E = 0$$
$$d_1(i) + d_2(i-7) + d_3(i-14) + p_1(i-21) + E = 0$$

Since a solution may not exist, we search for a value of E that would satisfy the most possible number of equations. Therefore, we assume that every equation might be satisfied by a different value for E say E_1, E_2, E_3, which leads to the following equations:

$$E_1 = -[d_1(i) + d_2(i-1) + d_3(i-2) + p_1(i-3)]$$
$$E_2 = -[d_1(i) + d_2(i-2) + d_3(i-4) + p_1(i-8)]$$
$$E_3 = -[d_1(i) + d_2(i-7) + d_3(i-14) + p_1(i-21)]$$

Then we compare say E_1, E_2, E_3. If they all are equal, then the error is assumed to equal to them as well. If

VOL. 14, NO. 19, OCTOBER 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

they are not equal, then we compare them two at a time. If any two are equal, then the error is assumed to equal to them. Note that in the general case we would compare the outcomes of r equations if they are not all equal, we then compare $r-1$ elements at a time, then $r-2$ and so on.

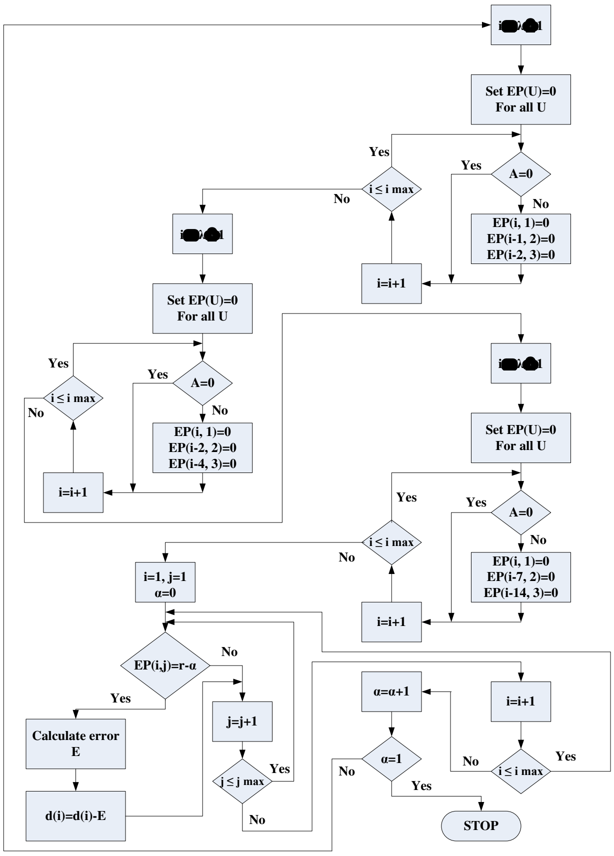

Figure-3. Flow Diagram for the Decoder of the Rate ½ BGC with 3-Data Lines and 3-Parity Lines.

3. SIMULATION RESULTS FOR THE BASIC GEOMETRIC CODE (BGC)

We simulated many codes using this procedure. In this section, we will present some of the simulation results obtained.

Figure-4 shows the results for the binary, 8-ary, and 256-ary BGC rate ½ with 3-data lines and 3-parity lines. The figure shows that the 8-ary code represents a significant improvement over the binary code. The 256-ary code introduces even more improvement. As discussed earlier, additional improvements would be introduced by increasing the number of bits per symbol. For by increasing the number of bit per symbol, the chance of two symbols canceling will be less. Therefore, the probability of symbol error out would approach zero as m the number of bits per symbol approaches infinity. Large values of m are undesirable, as discussed in [1]. We did not simulate any codes of $m > 8$ to limit the run time of the simulation programs.

Figure-5 makes the same comparison for the rate of$3/4$ codes. Hence, showing the error rate improves with increasing m, no matter the rate of the code used.

Figure-6 compares the results obtained for the 8-ary rate $1/2$ code, one using 3-parity lines and other using 4-parity lines. The code using 4-parity lines shows some improvement over the code using 3-parity lines. This improvement is the result of using more estimators per symbol, and therefore, an error will be detected by more slopes. Hence it will be easier to locate. Note that the codes with more than 4-parity lines were not simulated for the reason explained previously.

Figures 7 and 8 compare the are $1/2$ to the rate $3/4$ Basic Geometric Code. Figure-8 compares the 8-are codes, while Figure-9 compares the 256-ary codes. In both cases, the rate $1/2$ codes show considerable improvement over the rate $3/4$ codes which is expected if we fix r, the number of parity lines, k the number of data lines will be less for the rate $1/2$ code. Therefore the code will have fewer errors.

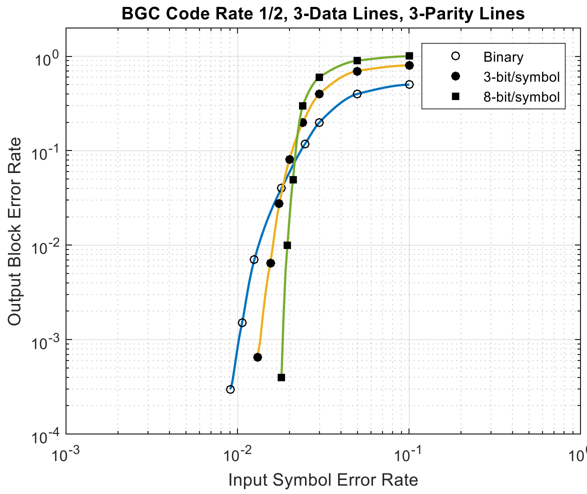

Figure-4.

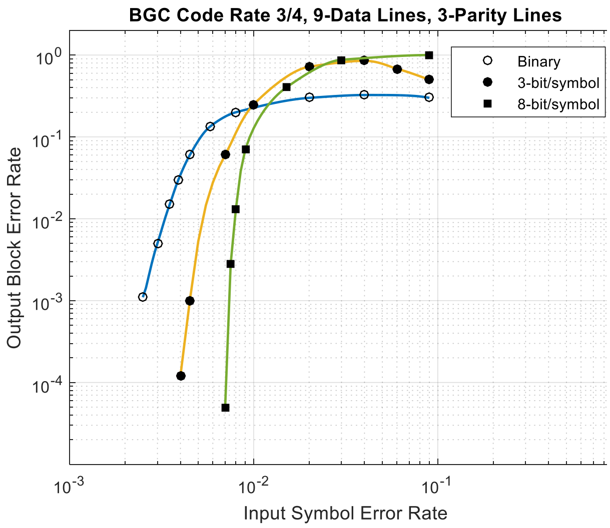

Figure-5.

VOL. 14, NO. 19, OCTOBER 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

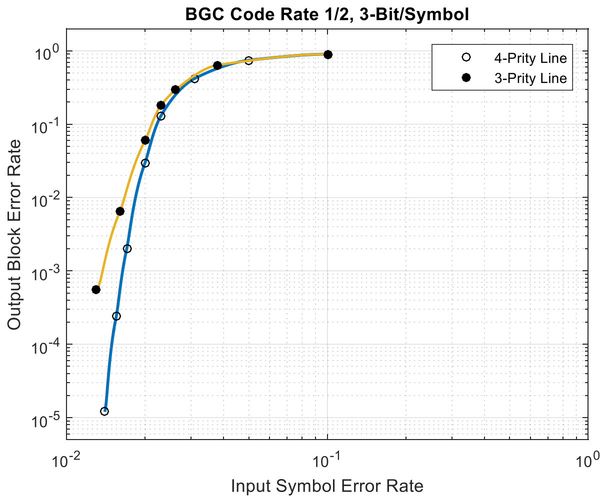

Figure-6.

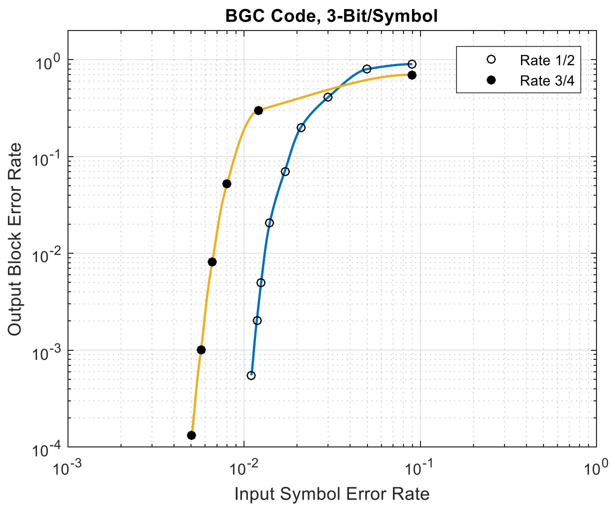

Figure-7.

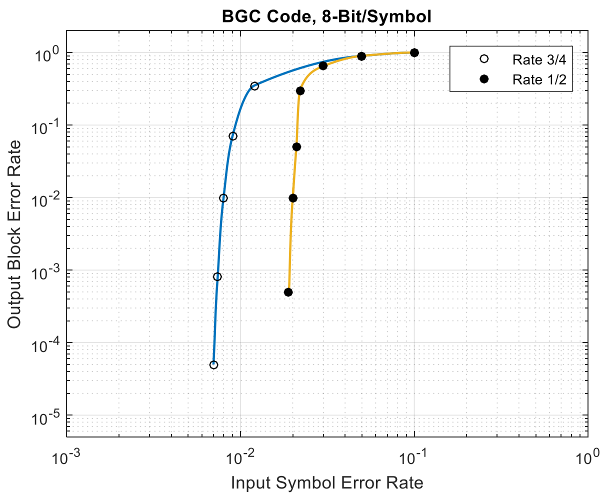

Figure-8.

4. CONCLUSIONS

In this paper, we presented a group of codes called the Geometric Codes; we presented the encoding and decoding algorithms as well as possible hardware implementations. We studied the performance of the Basic Geometric Code over a noisy channel. We then presented the simulation results for this code and have shown that the code performance does improve with the increase of the number of bits per symbol and with the coding rate.

It was shown that though geometric codes are not optimal, they outperform many codes, particularly over noisy or fading channels (input error rates of 10^{-2} to 10^{-3}).

It was also shown that these codes offer simple coding and decoding implementations. Because of their flexibility, geometric codes can be tailored for use in many communication systems.

ACKNOWLEDGMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research project.

REFERENCES

[1] Nasser Abdellatif. 1991. Performance of Non-Binary Projection Codes. The City University of New York.

[2] Maryam Akho-Zahiehand Nasser Abdellatif. 2014. Performance analysis of wavelet packet multirate multicarrier multicode CDMA wireless multimedia communications. International Journal of Electronics and Communications. 68(4): 268-277.

[3] Maryam Akho-Zahiehand Nasser Abdellatif. 2012. Narrow-band interference suppression in wavelet packets based multicarrier multicode CDMA overlay system. Wireless Communications and Mobile Computing. DOI: 10.1002/wcm.2325.

[4] Maryam Akho-Zahiehand Nasser Abdellatif. 2013. Effect of jamming and diversity on the performance of a wavelet packet based multicarrier multicode CDMA communication system. International Journal of Electronics and Communications. 67(2): 157-166.

[5] Maryam Akho-Zahieh, Khalil Saleh and Nasser Abdellatif. 2013. Outage Probability Performance of Wavelet Packet Multirate Multicarrier Multicode CDMA System. The Jordan Journal of Applied Science. 11(1).

[6] Nasser Abdellatifand Maryam Akho-Zahieh. 2013. Effect of Narrowband Interference on the Performance of Wavelet Packets Based Multicarrier CDMA System With Decorrelating Multiuser Detector. The Jordan Journal of Applied Science. 11(1).

[7] Maryam Akho-Zahieh, Khalil Saleh and Nasser Abdellatif. 2013. Performance of Wavelet Packets Based Multicarrier Multicode CDMA Communication System in the Presence of Narrowband Jamming, 1st International Conference on Communications, Signal Processing and their Applications, Sharjah, UAE, 2013 1st International Conference on Communications, Signal Processing and their Applications, Sharjah, UAE, Digital Object Identifier: 10.1109/ICCSPA.2013.6487223, IEEE Conference Publications.

[8] Maryam Akho-Zahieh and Nasser Abdellatif. 2015. Effect of Diversity and Filtering on the Performance of Wavelet Packets Based Multicarrier Multicode CDMA System. Journal of Signal and Information Processing. 6, 165-179.