ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2005 (1379-1383)
VOLUME : 5
NUMBER : 2

ITERATIVE DECODING OF GEOMETRIC PRODUCT CODES

[1]Gokmen ALTAY **[2]Osman N. UCAN**

[1]Bahçeşehir University, Department of Electrical & Electronics Engineering,
34538, Bahçeşehir, Istanbul, Turkey

[2] İstanbul University, Department of Electrical & Electronics Engineering,
34320, Avcılar, İstanbul, Turkey

[1]E-mail: galtay@bahcesehir.edu.tr [2]E-mail: uosman@istanbul.edu.tr

ABSTRACT

We present iterative decoding process of Geometric Product codes that is a newly proposed linear block code construction technique. We also report the bit error performances for one of the constructed code of the technique utilising our iterative decoding approach.

Keywords: *Iteratıve Codes, Geometric Product*

1. INTRODUCTION

A new construction scheme of decomposable codes that generates wide range of binary linear block codes has been proposed recently and named as Geometric Product (GP) Codes [1]. It has attractive generator matrix that allows easy encoding implementation with almost arbitrary code length. Basically, this construction employs a general single parity check matrix and another two or more component generator matrices to geometrically construct the final product generator matrix of the constructed code. In this paper, iterative soft decision decoding of GP codes have been performed using the log-likelihood algebra as Hagenhauer [2] described, but we use a similar notation as in [3].

2. CODE CONSTRUCTION

A decomposable general product code generator matrix, G_y , can be obtained by the *Kronecker* product of matrices G_1 and G_2, as $G_y = G_2 \otimes G_1$, where G_2 is taken as a single parity check matrix, as below,

$$G_2 = \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 1 \\ & & \ddots & & \vdots \\ & & & 1 & 1 \end{bmatrix}_{k_2 \times n_2} \qquad (1)$$

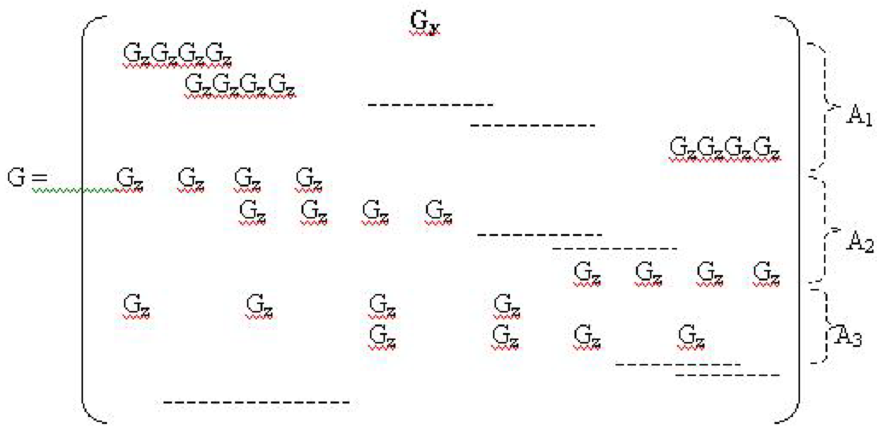
and,

$$G_y = \begin{bmatrix} G_1 & & & & G_1 \\ & G_1 & & & G_1 \\ & & \ddots & & \vdots \\ & & & G_1 & G_1 \end{bmatrix}_{k_y \times n_y} \qquad (2).$$

GP code construction uses G_y defined in (2) as a base matrix and augments another component generator matrix, G_z, in the specified geometrical manner to construct the ultimate generator matrix G as,

Received Date : 13.08.2004
Accepted Date: 11.03.2005

Utilizing the proposed GP code construction of (3), we can construct almost all the optimal Hamming distance-4 even codes in the size of (4) by specifying the component codes as C_1 = (2, 1, 2) and C_z = (2, 1, 1) with generator matrices G_1 = [1 1] and G_z = [1 0], respectively. The generator matrix of the component code, C_2, is a single parity check matrix and the length, n_2, can take any even value greater than or equal to four.



(3)

$$C = (n, k, d) = (s, s - \lfloor \log_2 s + 1 \rfloor, 4) \quad (4)$$

where s is an even integer number and greater than or equal to eight. As an example, C = (n, k. d) = (12, 7, 4) GP code can be constructed as in (5), where n, k, d are code length, dimension and minimum Hamming distance, respectively.

$$G = \begin{pmatrix} 110000000011 \\ 001100000011 \\ 000011000011 \\ 000000110011 \\ 000000001111 \\ 101010100000 \\ 000010101010 \end{pmatrix} \quad (5)$$

3. SYSTEM MODEL AND ITERATIVE DECODING PROCESS

In our transmission system model, we consider Additive White Gaussian Noise (AWGN) channel environment with zero mean and the variance σ^2 as $N_0/2$, where N_0 is single-sided noise spectral density. Binary Phase Shift Keying (BPSK) is used to map coded bits 1 and 0 as +1's and -1's at the output of the GP encoder, respectively. At the receiver, it is assumed that the ideal channel state information is available. Under these assumptions, our system model is designed as in Fig.1.

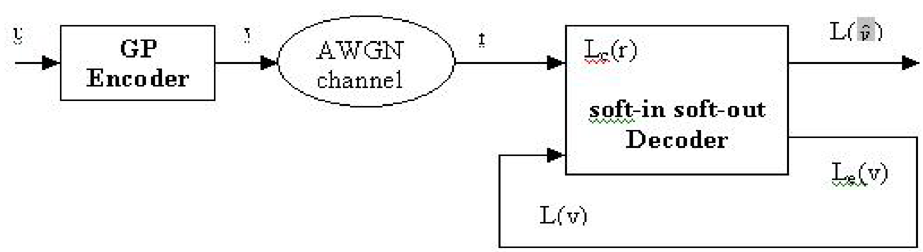


Fig.1 Transmission system model for GP encoding and iterative decoding

In this transmission system model, information bits u_i (i = 1,2,…,k) are encoded by GP encoder and coded bits v_j (j = 1,2,…,n) are transmitted over AWGN channel. The receiver side receives the received symbols $r_j = v_j + n_j$, where n is independent and identically distributed Gaussian noise. The decoding process starts by calculating a useful metric called the log-likelihood ratio (LLR) for each received bit.

$$L(v|r) = \log\left[\frac{P(v=+1|r)}{P(v=-1|r)}\right] = \log\left[\frac{P(r|v=+1)P(v=+1)}{P(r|v=-1)P(v=-1)}\right]$$

$$= \log\left[\frac{P(r|v=+1)}{P(r|v=-1)}\right] + \log\left[\frac{P(v=+1)}{P(v=-1)}\right] \quad (6)$$

$$L(v|r) = L(r|v) + L(v) \quad (7)$$

where $L_C(r)$ is the LLR of the channel measurements of r and $L(v)$ is the a priori LLR of the transmitted bit v, which is zero initially as all the transmitted bits are equally likely. The simplified notation of (7) can be written as in (8):

$$L'(\hat{v}) = L_C(r) + L(v) \quad (8)$$

The channel measurement of a received signal r_j, $L_C(r)$, can be written as LLR under Gaussian noise as follows,

$$L_C(r_j) = \log\left[\frac{P(r_j|v_j=+1)}{P(r_j|v_j=-1)}\right]$$

$$= \log_e\left[\frac{\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{r_j-1}{\sigma}\right)^2\right]}{\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{r_j+1}{\sigma}\right)^2\right]}\right]$$

$$= -\frac{1}{2}\left(\frac{r_j-1}{\sigma}\right)^2 + \frac{1}{2}\left(\frac{r_j+1}{\sigma}\right)^2 = \frac{2}{\sigma^2}r_j \quad (9)$$

After defining the channel measurement and a priori values, we introduce extrinsic LLR, $L_e(v)$, that is obtained from the decoding process at the output of the decoder. The soft decision is made at the output of the decoder regarding $L(\hat{v})$, which is a real number that provides a hard decision.

$$L(\hat{v}) = L_C(r) + L(v) + L_e(v) \quad (10)$$

The sign of $L(\hat{v})$ denotes the hard decision like +1 for positive values and -1 for negative values. The magnitude of $L(\hat{v})$ is the reliability of that decision. In order to perform decoding regarding (10), we must show how to obtain the extrinsic information $L_e(v)$. It can be derived utilizing the generator matrix, G, of the GP encoder. The relation between each coded bits of v provides a sort of parity check information and so extrinsic information at the decoder. This relation can be specified from G by a careful observation. Each bit of v should be represented with other bits of v. We give an example over the generator matrix of GP (12, 7, 4) code as shown in (5).

$$\begin{aligned}
v_1 &= v_2 \oplus v_3 \oplus v_4 \\
&= v_2 \oplus v_5 \oplus v_6 \oplus v_9 \oplus v_{10} \\
&= v_3 \oplus v_5 \oplus v_7 \oplus v_9 \oplus v_{11} \\
&\vdots \\
v_2 &= v_1 \oplus v_3 \oplus v_4 \\
&= v_1 \oplus v_5 \oplus v_6 \oplus v_9 \oplus v_{10} \\
&= v_4 \oplus v_6 \oplus v_8 \oplus v_{10} \oplus v_{12} \\
&\vdots \\
&\vdots \\
v_{12} &= v_{11} \oplus v_9 \oplus v_{10} \\
&\vdots
\end{aligned}$$

Fig. 2 Relationships among the bits of v for (12,7,4) GP code

In Fig. 2, '$\oplus$' denotes modulo-2 addition. The log-likelihood algebra [2] is used to define the sum of LLRs for a statistically independent v as follows:

$$L(v_1) \boxplus L(v_2) = L(v_1 \oplus v_2) = \log_e \frac{e^{L(v_1)} + e^{L(v_2)}}{1 + e^{L(v_1)}e^{L(v_2)}}$$
$$= (-1) \times sign[L(v_1)] \times sign[L(v_2)] \times \min(|L(v_1)|, |L(v_2)|) \quad (11)$$

$$L_e(v_1) = [(L_c(r_2) + L(v_2)) \boxplus (L_c(r_3) + L(v_3)) \boxplus (L_c(r_4) + L(v_4))\ ] \quad (12).$$

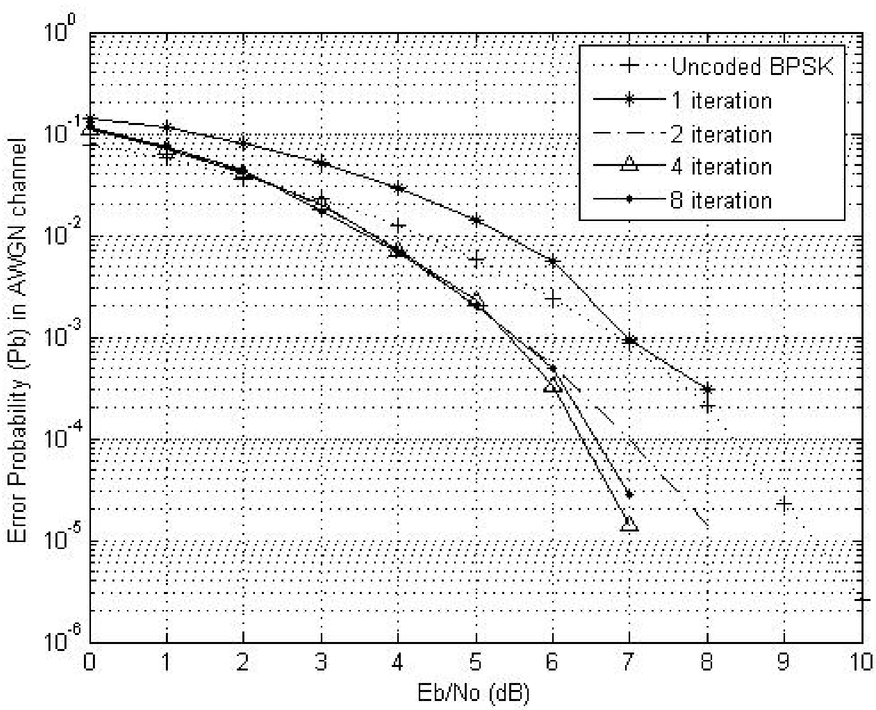
Here, '⊞' denotes log-likelihood addition that gives the LLR of the module-2 sum of the involved bits. As an example, for the GP encoder, the soft output LLR value $L(\hat{v}_1)$ is obtained utilizing (10), where one of the extrinsic LLR, $L_e(v)$, is obtained as shown in (12).

Extrinsic LLR values, $L_e(v)$, are provided from the parity checks of a bit in v. As shown in Fig. 2, there are many parity check equivalents of a bit in v. In order to obtain the final extrinsic LLR value, we must perform the operations similar to (12) for all the possible parity check equivalents of a bit in v repetitively. In our decoder, we perform similar operations as in (12) for all the bits of v with respect to their first parity check equivalents and then decode again for the second parity check equivalents and so on. The process ends when all the parity check combinations are used. As it can be realized that $L_e(v)$ is refined in each process of (12). The iterative decoding algorithm for a GP code proceeds as follows:

1. Initialize the a priori information $L(v) = 0$.
2. Find the $L_e(v)$ using all the possible parity check equivalents of a bit of v. Utilize from Fig. 2 and (12).
3. Set new $L(v) = L_e(v)$.
4. If iteration is necessary for more reliable decision go to step 2, otherwise go to step 5.
5. The soft output LLR value is:

$$L(\hat{v}) = L_C(r) + L_e(v) \qquad (13).$$

If $L(\hat{v}) > 0$ then v = 1, otherwise v = 0.



4. CONCLUSION

We have analyzed the iterative decoding process of GP codes and showed a way for iteratively decoding GP codes. We used similar iterative decoding process as described in [2] and [3] by using log-likelihood algebra for computational simplicity. Simulations have been performed for the (12, 7, 4) GP code over AWGN channel and observed that BER curve converges to the Viterbi decoding results of [1]. This paper is the first step for iterative decoding of GP codes. The future research includes efficient and practical probability decoding methods of GP codes for large code lengths.

REFERENCES

[1] G. Altay, O. N. Ucan, “A New Construction and Trellis Decoding of Decomposable Codes,” submitted to *IEE Proc. Communications*.

[2] J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, March 1996.

[3] B. Sklar, "A Primer on Turbo Code Concepts," *IEEE Comm. Magazine*, pp. 94-102, Dec. 1997.