REDUCED COMPLEXITY BELIEF PROPAGATION DECODING FOR REED MULLER BLOCK CODED SYSTEM

Mahesh L. Davkare[1]
[1]Department of Electronics and Telecommunication, JSPM NTC,
Savitribai Phule Pune University, Pune, India.

Abstract : Digital communications can perform unreliably and inaccurately in the presence of noise and interference along a channel. A technique has been developed not only to detect these inaccuracies, but also correct them when decoded. Here we are implementing a block coded system consisting of Reed Muller (RM) code for error detection and correction. RM codes are a family of linear error-correcting codes used in communications. RM codes are among the oldest known codes and have found widespread applications. A useful log domain belief propagation algorithm is used for decoding purpose. The algorithm works by passing real valued functions called messages along the edges between the bit node and check nodes. The performance will be evaluated and validated under Additive White Gaussian Noise channel environment (AWGN). A MATLAB simulation results were tested to achieve bit error rate of 10-5. Simulations are carried out with variable code rate and with 10 iterations. It is observed that bit error rate performance of system depends on code rate, block length and number of iterations.

IndexTerms - Block Codes. Reed Muller Code, Log Domain Belief Propagation Algorithm, Bit Error Rate

Introduction

In digital communication, the critical area is how to send the message from the source to the destination with high accuracy. One of the most used techniques and also the most convenient is the adoption of error-correcting codes (ECC). Indeed the codes are used to improve the reliability of data transmission over communication channels susceptible to noise. The coding techniques are based on the following principle: add the redundancy to the transmitted message to obtain a vector called "code word". Decoding techniques are based on the algorithms witch try to find the most likely transmitted code word related to the received one. Decoding algorithms are classified into two categories: hard-decision and soft-decision algorithms. Hard-decision algorithms work on a binary form of the received information. In contrast, soft decision algorithms work directly on the received symbols.

According to the manner in which redundancy is added to messages, ECC can be divided into two classes: block and convolution. Block codes implement a one-to-one mapping of a set of k information symbols on to a set of n code symbols. Convolution codes offer an approach to error control substantially different from that of block codes. A convolution encoder converts the entire data stream, regardless of its

length, into a single codeword. Both types of coding schemes have found practical applications. Block codes have been the subject of a considerable amount of research in information theory. General coding theorems and sophisticated implementation techniques have been developed for these codes.

RM codes are a family of linear error-correcting codes used in communications. RM codes were introduced in 1954, first by Muller and shortly after by Reed who oldest and simplest codes to construct; the codewords are the evaluation vectors of all multivariate polynomials of a given degree bound. RM codes have been extremely influential in the theory of computation, playing a central role in some important developments in several areas. In cryptography, they have been used e.g. in secret sharing schemes, instance hiding constructions and private information retrieval. The conjecture that RM codes achieve capacity has been experimentally confirmed in simulations [1,2]. Moreover, despite being extremely old, new interest in it resurged a few years ago with the advent of polar codes [3].

The main iterative belief propagation (BP) decoding algorithms for RM code include soft-decision such as Sum Product (SP) algorithm [4] and hard-decision such as Bit flipping. In iterative decoding, a critical trade-off between complexity and performance is required. Based on these two issues, algorithms may be classified as optimal, sub-optimal or quasi-optimal. The optimal iterative decoding is performed by the Sum- Product algorithm at the price of an increased complexity, computation instability, and dependence on thermal noise estimation errors. The Min-Sum algorithm [5] performs a suboptimal iterative decoding, less complex than the Sum-Product decoding. The sub-optimality of the Min-Sum decoding comes from the overestimation of check-node messages, which leads to performance loss with respect to the Sum-Product decoding. Several correction methods were proposed in the literatures in order to recover the performance loss of the Min-Sum decoding with respect to the Sum-Product decoding which are called quasi-optimal algorithms. An example is Normalized min-sum algorithm proposed by Chen and Fossorier.

This paper is organised as, in section 2 we will discuss basic dimensions of RM code. The generation of RM code is also illustrated. In section 3 gives the insight about log domain BP decoding algorithm. In section 4 we have presented the simulated result for bit error rate (BER) performance of system.

Reed Muller Code

A RM code of order r and m is denoted as *RM (r,m)* or *RM (N,K).* For $m \geq 0$, *and* $0 \leq K \leq N$

Table 1 : Dimensions of RM Code

Mathematical Expression	**Dimensional Quantity**
$N = 2^m$	N – Block Length
$K = \sum_{i=0}^{m} \binom{m}{i}$	K – Input Msg. Length
$d = 2^{m-r}$	d = minimum hamming distance of code
$R = K/N$	R – Code Rate

The RM code of order *(r,m)* can be generated mainly by two methods.

Recursive Construction:

$$G_{RM(r,m)} = \begin{bmatrix} G_{RM(r,m-1)} & G_{RM(r,m-1)} \\ 0 & G_{RM(r-1,m-1)} \end{bmatrix} \quad \textbf{(2.1)}$$

Where $G_{RM\,(r,m)}$ is generator matrix gor RM (r,m)

$$G_{RM\,(m,m)} = I_2^m \quad \& \quad G_{RM\,(0,m)} = [1\ 1\ .....1]_{m\ times}$$

Plotkin Construction:

Let $G_{RM\,(m,m)}$ denote denote the generator matrix of an mth order RM code. Using the well-known Plotkin construction for RM codes, we may take

$$G_{RM\,(m,m)} = G^1_{(2,2)} \otimes G^2_{(2,2)} \otimes ..... \otimes G^m_{(2,2)} \qquad \textbf{(2.2)}$$

Where $G_{(2,2)} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

$G^1_{(2,2)} \otimes G^2_{(2,2)}$ is Krownecker Product

The rth order RM code *RM (r, m)* can then be defined as the linear code with generator matrix $G_{RM\ (r,\ m)}$ which is obtained by taking the rows of $G_{RM\ (m,\ m)}$ with Hamming weights $\geq 2m-r$. For example, $G_{RM(3,\ 3)}$ is given by and $G_{RM\ (1,3)}$ for *RM(1,3)* is given by

The parity check matrix H_{RM} for iterative decoding can be derived using the duality property of of RM code. Parity check matrix for *RM (r,m)* is the generator matrix of RM code of order *(m-r-1,m).*

$$H_{RM\ (r,m)} = G_{RM\ (m-r-1,m)} \qquad \textbf{(2.3)}$$

$$G_{RM\,(3,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad G_{RM(1,3)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Belief Propagation Decoding

By Arikan [] RM codes can be regarded as codes on graphs, and, hence, decoded by BP decoders. Parity check matrix H_{RM} for *RM (1,3)* is given as below

$$H_{RM\,(1,3)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Log Domain BP Decoding Algorithm Steps

Step 1: Initialization

$$L(P_{ij}) = L(C_j) = \frac{2r_j}{\sigma^2}$$

Step 2:Message Passing

From check nodes to bit nodes: For each check node i with an edge to bit node j: Update L(Q ij) as:

$$L(Q_{ij}) = \prod_{j'} \alpha_{ij'} \phi \sum_{j'} \phi(B_{ij'}) \quad ($$

Where $\alpha_{ij'} \overset{\Delta}{=} sign[L(P_{ij})]$ and $B_{ij} \overset{\Delta}{=} |L(P_{ij})|$

The ϕ function is defined as:

$$\phi(x) = -\ln[\tanh(x/2)] = \ln\left[\frac{e^x + 1}{e^x - 1}\right]$$

Step 3: Probability Passing

From bit nodes to check nodes: For each bit node j with an edge to check node i:

Update L(Pij) as:

$$L(P_{ij}) = L(C_j) + \sum_{i'} L(Q_{i'j})$$

Step 4: Extrinsic Probability Calculation and Decoding

Decoding and soft outputs: For j = 1,2,...., n

Update L(Pj) as:

$$L(P_j) = L(C_j) + \sum_{i} L(Q_{ij})$$

$$C_i = \begin{cases} 1 & if \quad L(P_j) < 0 \\ 0 & else \end{cases}$$

If c × H T = 0 or the number of iterations reaches the maximum limit, stop; otherwise, go to step 2.

Simulation Results

The Bit Erroor Rate (BER) performance of RM code for various code rate at a block length of N=64 is tabulated as follow.

N=64 Code words=5000 No. Iterations = 10.

Table 2 : BER Values for different RM Code Rate

Eb_No (dB)	RM (1,6) R =0.109	RM (2,6) R =0.343	RM (3,6) R =0.656	RM (4,6) R = 0.890
-6	0.104691	0.240707	0.239594	0.240725
-4	0.030682	0.155635	0.186197	0.187888
-2	0.005997	0.059047	0.130485	0.131788
0	0.00076	0.007794	0.077	0.07931
2	0	0.000494	0.034675	0.037919
4	0	0.000044	0.009554	0.012685
4.5	0	0	0.006272	0.009016

5	0	0	0.003935	0.0061
5.5	0	0	0.002279	0.003919
6	0	0	0.001229	0.002485
7	0	0	0.000263	0.000772
8	0	0	0.00005	0.000166

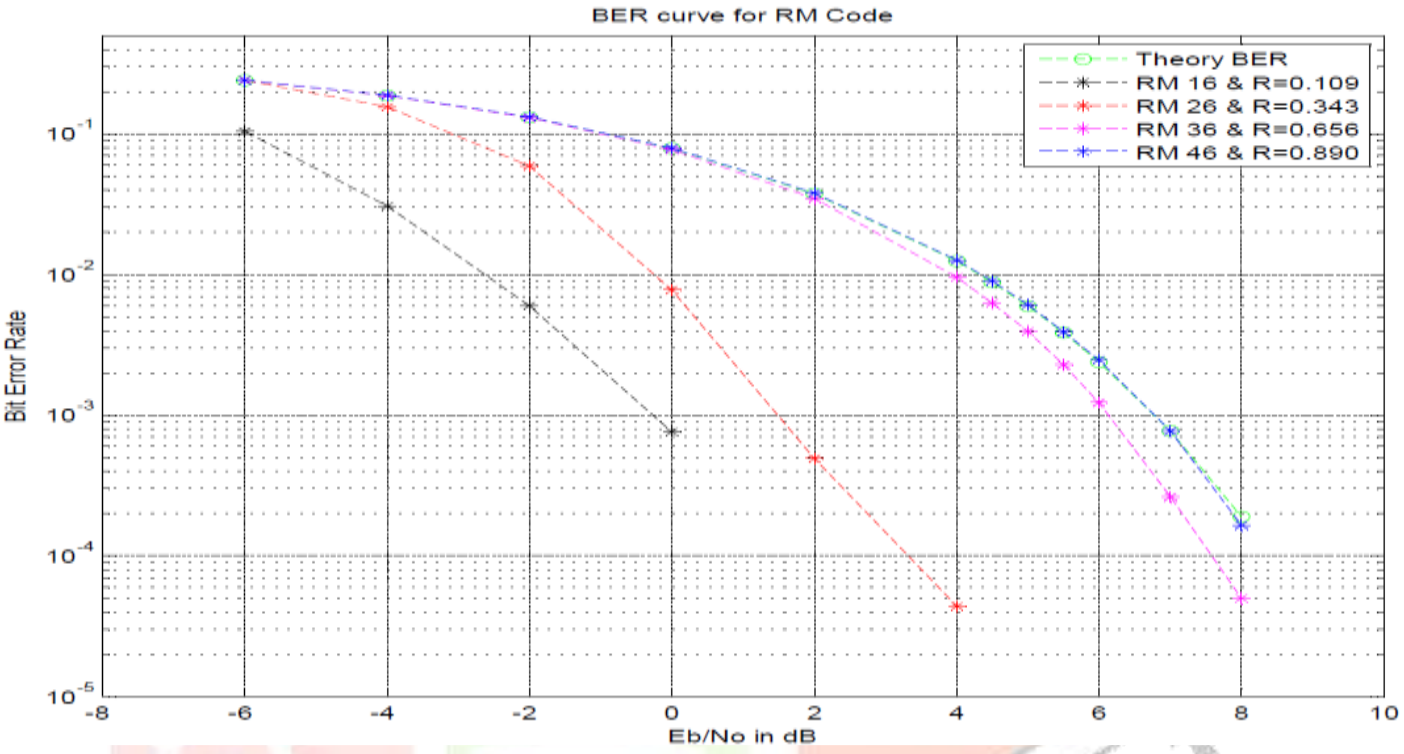

Fig.2 : BER Curve for RM Code

Conclusion

Reed Muller code has very simple code construction and also performs well under iterative decoding belief propagation algorithm. BER performance of Reed Muller code is presented for 64 bit block length. We observe good coding gain of 5db at 10-4 BER at 0.343 code rate.

REFERENCES

[1] E. Arikan, "A Performance Comparison of Polar Codes and Reed-Muller Codes", *Communications Letters, IEEE*, no. 6, 2008.

[2] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From Polar to Reed-Muller Codes: A Technique to Improve the Finite-Length Performance", *Communications, IEEE Transactions on* , no. 9, 2014.

[3] E. Arikan, "Channel polarization: A Method for Constructing Capacity Achieving Codes for Symmetric Binary Input Memoryless Channels", *Information Theory, IEEE Transactions on,* no. 7, 2008.

[4] Frank R. Kschischang, Brendan J. Frey, Hans-Andrea Loel, "Factor Graphs and the Sum-Product Algorithm", *Information Theory, IEEE Transactions On*, Vol. 47, Issue: 2, February 2001.

[5] M.P.C.Fossorier, M. Mihaljevic, and H.lmai, "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation", *IEEE Trans. on Comm*, vol. 47, no. 5, pp. 673-680, May 1999,

[6] Han, Wei, Huang, Jianguo, Fangfei Wu, "A Modified Min-Sum Algorithm for Low-Density Parity-Check Codes", *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on* , 25-27 June 2010