MATLAB Simulation of BCH Codes

Nikola Janjušević, EE '19 *The Cooper Union*

***Abstract*—Three BCH codes were simulated to obtain bit error rate (BER) vs. EbNo curves.**

I. Introduction

For this assignment, three BCH codes were simulated over an AWGN channel in Matlab. These three codes were chosen to be 1, 2, and 3 error correcting codes of the same order ($m = 5$). A BCH code is a cyclic block code that is identified by two integers, $m \quad (\geq 3)$ and $t \quad (< 2^{m-1})$. These two integers determine a BCH code with the following properties: [1]

- Code length: $n = 2^m - 1$
- Number of information symbols: $k \geq n - mt$
- Minimum distance: $d \geq 2t + 1$

The implemented encoding and decoding was written in a manner that allows for an arbitrary choice of m and t (within the constraints given above).

II. Implemented Encoding

Before encoding is performed, the generator matrix, G, and parity check matrix, H are found. The procedure involves finding the minimum polynomials of $2t$ powers of primitive element $\alpha \in GF(2^m)$, $\phi_1(X), \phi_2(X), ..., \phi_{2t}(X)$. The generator polynomial $g(X)$ is then given by the least common multiple of each of the minimum polynomials.

Though a message may be readily encoded by multiplying the message polynomial with the generator polynomial, it is often convenient to obtain a generator matrix $G \in M^{k \times n}$ for quick batch-encoding. In the case of a cyclic block code, G may be obtained by cyclical shifting the n-tuple vector version of $g(X)$, k times (k is obtained by the difference between the codeword length, n, and the order of the generator polynomial). Furthermore, the matrix may be put in reduced-row echelon form to obtain a block decomposition with the $k \times k$ identity matrix for quick separation of the message bits from the decoded block. This matrix manipulation does not alter the integrity of the code.

The parity check matrix, $H \in M^{n-k \times n}$, is obtained by arrangements of powers of α. See [1] pg. 201 for details.

III. Implemented Decoding

Though H is not directly used for decoding, it may be used to check for valid code words and skip over them during decoding (via checking if $vH^T = 0$ for received code word v). The decoding algorithm implemented is the Peterson's Direct-Solution Decoding Algorithm, which is easy enough to follow with some careful attention to details. See [1] for an overview of the algorithm, and perhaps consult another source to find in-depth details and examples.

IV. Simulation

Simulations of the three different t-error correcting codes of order 5 were simulated in Matlab. The simulations encoded 10,000 BPSK symbols (with sufficent padding to match the requirements of the block encoding) averaged over 10 iterations. Each iteration simulated the AWGN channel with several different signal to noise ratios, and thus different EbNos (as seen in Figure 1). The three curves show nicely how the coding gain of BCH increases with its error correcting capabilities. Uncoded BPSK is provided as a curve of reference in the figure.

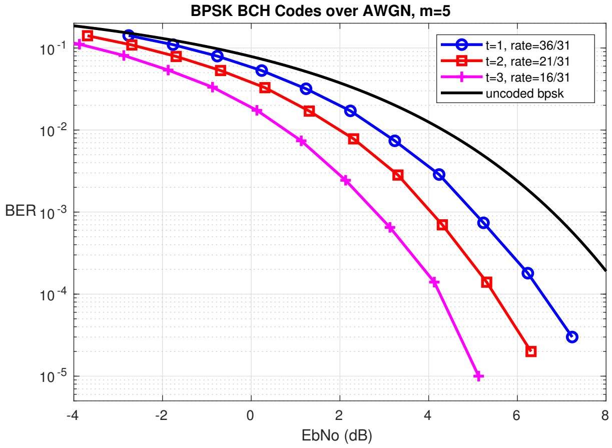

Fig. 1. Simulated bit error rates of BCH codes of order 5 with different error correcting capabilities, using BPSK over an AWGN channel

References

[1] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.