Error Control Coding Simulation Project

ECE 300-Communication Theory

Professor Frost

Bonny Wang, Allister Liu, Amy Leong

Table of Contents

Introduction

This project sets out to explore error control coding. In this project, a framework of linear block codes is developed for error control coding through vector and matrix arithmetic over Galois field 2 (F_2) in MATLAB to evaluate performance in a noisy environment.

A block code, aka codebook, (n, k) is a collection of $M=2^k$ binary sequences called codewords, where n is the length of the codewords. When transmitting information through communication channels, the data is broken into fixed size messages, where k is the length of messages. The rate of a block code is defined as $R = k/n$, which is the amount of actual message per transmitted block, and the rate R tells us the efficiency of the code block. A block code is said to be linear if the sum (element-wise modulo 2 addition) of any two codewords is itself a codeword.

This project specifically uses Hamming Code. For Hamming code, there is code with block length $n=2^m-1$ and message length $k=2^m-m-1$ for some integer $m\geq3$. The minimum Hamming distance was also used in this project to find the distance between two codewords. When correcting the codewords, the codewords that are greater than the Hamming distance would be lost.

For linear code, a generator matrix is a matrix whose codewords are all of the linear combinations of the rows of this matrix, that is, the linear code is the row space of its generator matrix. A special condition of this is the systematic code, which each codeword begins with the data it represents. In other words, For a systematic linear code, the generator matrix G, can always be written as $G=[I_k|P]$, where I_k is the identity matrix of size k.

Methods

There were two functions for the Galois field 2 arithmetic: f2matrixMultp and f2matrixAdd. None of these functions used a for loop. For f2matrixAdd, it takes in an M x N matrix and an N x K matrix with only 0s and 1s, adds the two matrices and takes modulus 2. For f2matrixMultp, it returns the product over F_2 as an M x K matrix with only 0s and 1s by taking the product of the two input matrices and taking the modulus 2. Since matrix multiplication is multiplication (which already follows the rule of Galois field) followed by addition (can be performed as previously stated).

For the binary symmetric channel function, it takes in a string of bits and a bit error probability p, and then outputs a corrupted string of bits without using loops. This was done by generating a random bit error matrix. The bits are supposed to be flipped with a probability error p. After getting the bit error matrix, f2matrixAdd is called so that it is added to the string of bits that were passed into the binary symmetric channel function.

To detect the maximum number of correctable errors, the minimum Hamming distance between the two codewords was found. It is equal to the minimum weight. Then, that value was subtracted by 1 and the difference is divided by 2. This follows the formula for the maximum number of correctable errors.

If the number of bit errors is less than or equal to the maximum correctable errors, then the errors are detectable and correctable. If the number of bit errors is greater than the maximum correctable errors, then none are correctable. Also, the error is detectable if it does not map to another codeword. Larger errors are not correctable because there may not be a codeword that is close and unique. There could be another wrong codeword that has a smaller Hamming distance value that it can be mistaken for. If this happens, it is not detectable.

To generate the truncated syndrome array, an array of all possible error positions is generated. The truncated syndrome is obtained by multiplying that array with the transform of the parity matrix.

Since the max number of errors for G1 is 1, all possible locations are the same as an identity matrix plus no error, which is filled with 0s.

For G2, the maximum number of errors is two, all possible locations are generated by adding 0s at the beginning and moving the first position after appending with an identity matrix(represent the second possible location). It also includes the situation where there is only one error and no error.

The truncated syndrome is like a pattern dictionary of all errors after parity check. Since messages with no error will have a value of 0s after parity check. The pattern (which will only be affected by the error) obtained by parity-check can be used to locate the error by comparing it with the syndrome. When the index is equal to 0, there are no correctable errors and the corresponding index is in the last row of the possible error position matrix which is filled with NaN.

After finding the position of error, Galois field 2 addition is performed to eliminate the error since 1+1 = 0.

Results

In a linear code, the minimum Hamming distance is equal to the minimum Hamming weight: $d_{mun} = w_{min}$. And the maximum number of correctable errors is defined as $e_c = ½(d_{min}-1)$. Therefore, for G1, $e_c = 1$; and for G2, $e_c = 2$ (as noted in line 42 and line 43).

When using a standard array, which contains all of the possible 2^k n-bit strings, the size of the array is given by $2^{n-k} \times 2^k$. In this project's context, the standard two arrays would have 16 rows and 256 rows. Alternatively, when choosing truncated syndrome arrays, the two truncated standard arrays would only have 9 rows and 79 rows (as noted in line 69 and line 71).

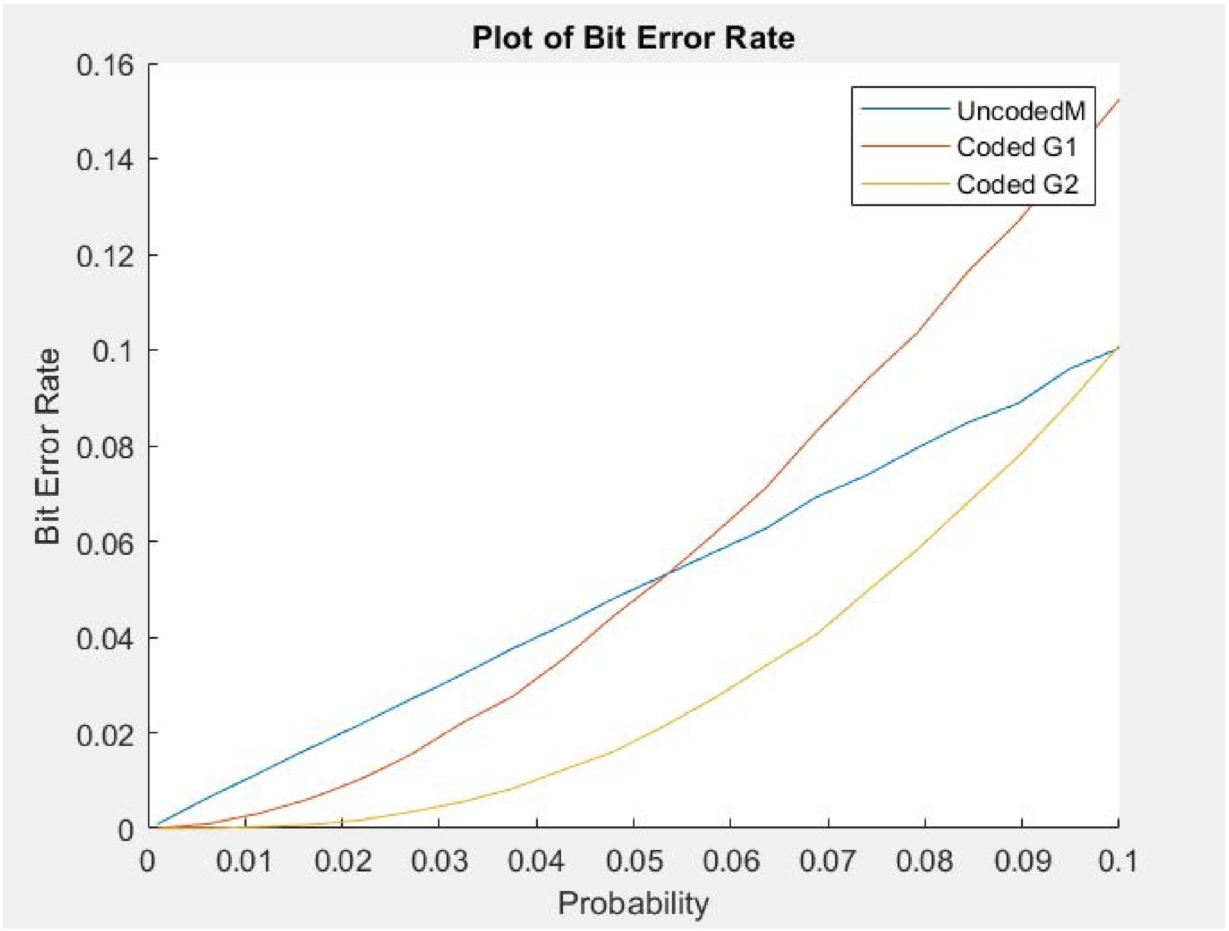

Figure 1: Plot for bit error rate and bit error probability of error for uncoded, G_1, and G_2

Discussion

As shown in Figure1 in the Results section, the bit error probability and the bit error rate for uncoded appear to be linear. When comparing the two different error correction schemes by looking at the error bit rate vs bit error probability graph, G2 has a lower bit error rate than G1, indicating G2 has better performance. Starting from about 0.055 bit error probability, G1 started to perform worse than the uncoded word, while encoding with G2 is better than the uncoded word for most of the bit error probability range. This is noted by the intersecting lines. The reason for this performance may be because a detectable and uncorrectable error results in the entire word being tossed out, whereas for uncoded, only the bits that are wrong are counted.

For decoding the linear code, a truncated syndrome array was used. Syndrome decoding uses minimum distance decoding with a reduced lookup table, which makes it an efficient method of decoding linear code in noisy environments. A truncated array was used to decode instead of a true syndrome array because a truncated array tends to be more efficient in both time and space complexity. When dealing with a larger generator matrix, a truncated syndrome array is much smaller than a standard array, especially when the matrix has a longer block length, n. Some of the words are tossed such as when it is not possible to detect which correction is more correct. This can occur when the Hamming distance is equal between two codewords. When using a standard array, uncorrectable errors may appear, but a truncated syndrome array only includes correctable errors. However, with a truncated array, some of the information on the possible errors are lost since some of the information is tossed.

In practice, smaller errors might only be corrected his way because of the limited redundancy that Hamming codes add to the data. They can only detect and correct errors when the error rate is low. In fact, Hamming codes can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors.

Practically, the tossed information can get requested again or it can be ignored. This is dependent on the situation. In situations where it is important to get all the information such as a downloading service or sending emails, the tossed information should get requested again until the word is no longer corrupted. However, in situations such as a phone call, there may be tossed information because of the momentarily spotty cell service, it may be acceptable to ignore the tossed information.