Assignment 1

Molodtsov Vladislav

1 Introduction

In this assignment, data transmission through AWGN channel using error-correcting codes is considered. In particular, a Gaussian codebook is used as error-correcting code, while decoding is performed according to the principle of maximum likelihood.

This report has the following structure. Section 2 covers implementation details of error-correcting code and maximum likelihood decoder, while Section 3 demonstrates simulation results.

2 Implementation details

The model of the considered AWGN channel is the following:

$$y_k = x_k + z_k, \quad z_k \sim \mathcal{CN}(0,\, \sigma^2) \tag{1}$$

where x_k – complex valued transmitted symbol, y_k – complex valued received symbol, z_k – complex valued additive noise.

To transmit the data over the channel (1), one needs to implement an error-correcting code. From the information theory, one knows that a Gaussian codebook achieves the capacity of the AWGN channel.

The idea of a Gaussian codebook is described below. Consider M random vectors $\mathbf{v}^i$, $i = \overline{1, M}$ in n-dimensional complex space. Components of these vectors v_j^i are independent and drawn from complex standard normal distribution: $v_j^i \sim \mathcal{CN}(0,\, 1)$, $j = \overline{1, n}$. Vectors $\mathbf{v}^i$ are codewords and they form so-called Gaussian codebook. Encoding process consists of mapping input bits to codewords:

$$0\ldots00 \to \mathbf{v}^0, \;\; 0\ldots01 \to \mathbf{v}^1, \;\; \ldots, \;\; 1\ldots11 \to \mathbf{v}^M \tag{2}$$

In order to encode k information bits into one codeword, $M = 2^k$ codewords are required. If coding rate $R = \frac{1}{2}$ is desired then space size $n = \frac{k}{R} = 2k$ and each codeword has $2k$ components. Thus, generating codebook is equivalent to generating matrix $\mathbf{C}$ with size $2^k \times 2k$, where each element is sampled from $\mathcal{CN}(0,\, 1)$, while encoding is equivalent to choosing a row of this matrix $\mathbf{C}$.

Transmission of codeword $\mathbf{v}^i$ over the channel (1) requires n channel uses, one per each codeword's component. In other words, $x_j = v_j^i$, $j = \overline{1, n}$. Important note is that the power of vector $\mathbf{v}^i$ should be normalized to n^2:

$$\left\|\mathbf{v}^i\right\|^2 = \sum_{j=1}^{n} |v_j^i|^2 = n^2 \tag{3}$$

With such normalization, each transmitted sample $x_j = v_j^i$ has average power $|x_j|^2 = 1$ and Signal-to-Noise Ratio (SNR) is equal to $\frac{1}{\sigma^2}$.

After going through the channel (1), received vector of samples $\mathbf{y}$ consists of noisy components $y_j = v_j^i + z_j$. The goal of maximum likelihood decoder is to find the codeword $\hat{\mathbf{v}^i}$ from codebook $\mathbf{C}$ which is the closest one to the received vector $\mathbf{y}$ in terms of Euclidean norm:

$$\hat{\mathbf{v}^i} = argmin \left\|\mathbf{v}^i - \mathbf{y}\right\|^2 = \sum_{j=1}^{n} |v_j^i - y_j|^2 \tag{4}$$

If decoded word $\hat{\mathbf{v}^i}$ differs from transmitted codeword $\mathbf{v}^i$, then it is considered that an error has occured in the block of transmitted samples. After carring out a set of such experiments for each SNR, it is possible to calculate the ratio of these errors to the total number of experiments or Block Error Rate (BER). Thus, it is possible to evaluate the performance of the designed error-correcting code by finding dependence BER on SNR.

3 Results

Figure 1 demonstrates the results of the simulation. Different curves correspond to the various k, number of information bits.

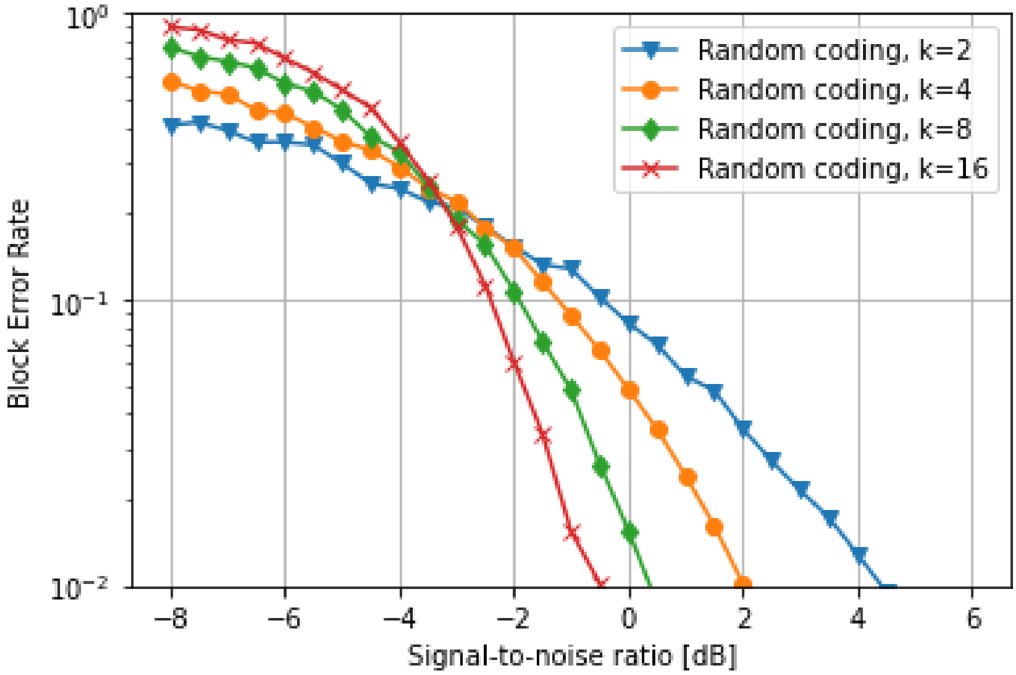

Figure 1: Simulation results

From the obtained results, it is clear that with increasing the number of information bits k, the error-correcting code has better performance in a sense that it achieves desired BER, for instance, 10^{-2}, with lower SNR. It happens because with increasing k, code space dimension $n = 2k$ increases as well, therefore, codewords are spread further apart from each other and the error probability decreases.