Communications Laboratory Design Project

Digital Data Transmission over Noisy Channel: Encoding and Decoding of Binary Image Data

(Fall 2017)

Prepared by Prof. Hyunggon Park
Department of Electronic and Electrical Engineering
Ewha Womans University

1 General Instruction

The goal of this project is to show you how several communications concepts are actually implemented in different applications such as image transmission. In this project, you may use MATLAB.

General information:

1. Deadline: the project due is on **December 13, 2017 (Wednesday)**, in class.
2. The final report must include:
   - **answers to all questions with your reasoning**
   - **part of MATLAB codes of your implementation if necessary,**
   - **MATLAB results/output (plots, figures, etc.) if necessary.**
3. Final report submission: turn in your **electronic copy** of your final report and separately upload all MATLAB source codes in the Cyber Campus by the due date. However, **DO NOT** include all your MATLAB source codes in your reports (if necessary, you may include part of source codes). Rather, turn in your MATLAB source codes separately with appropriate file names (e.g., `problem1.m`).

Note that the project is a **team project** which means that **all the members in your team** would have the same credit for the project. Hence, try to collaborate with each other in your team, such that your team will successfully complete the project. The contribution of this project to the final grade is 25% (Completeness 15%, Reports 5%, Presentation 5%).

2 Project Description: Data Transmission over Noisy Channel

We consider a conventional communication system, where an image data is sent through the noisy channel, and a receiver would like to recover the original image based on the decoding process. The conceptual system is shown in Figure 1.

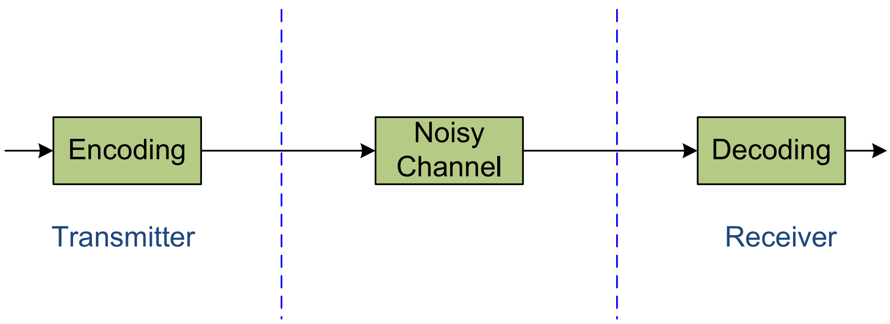

Figure 1: Overall system model.

In this project, you will actually implement the communication system for image transmission using MATLAB and study the impact of channel noise on the communication performance in terms of bit error rates (BER).

2.1 Image Source Data

An image source data is given to each team, where the image includes an alphabet character (e.g., A, B, etc.) and its size is given by 100×100 matrix. Since the image is black and white, each pixel can be expressed as either 0 (black) or 1 (white). Figure 2 shows an example of source image data for character 'E', which can be considered as a matrix with '0's and '1's.

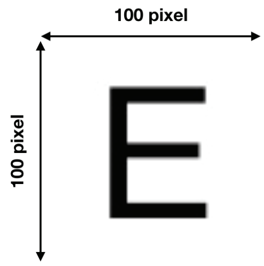

Figure 2: An example for source image data.

Note that image data is serialized as a vector constructed by concatenating row vectors. Let M be a matrix for the image source data, expressed as

$$M = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_{100}^T \end{bmatrix}.$$

The serialized vector $\mathbf{v}$ is constructed as

$$\mathbf{v} = [\mathbf{m}_1^T \ \mathbf{m}_2^T \ \cdots \ \mathbf{m}_{100}^T].$$

2.2 Encoding

For simplicity, each data is first encoded by NRZ (Non Return to Zero) as a baseband modulation, i.e., bit '0' of the image is mapped into '1' and bit '1' of the image is mapped into '-1', which is referred to as $d_i \in \{-1, 1\}$. Then, we use the BPSK (Binary Phase Shift Keying) for passband modulation, which can be expressed for a binary data d_i as

$$s_i(t) = d_i \times \cos(2\pi f t)$$

where $f = 10$ in this project.

2.3 Decoding

Now, it is your turn to design a decoding process. The decoding algorithm should be designed as a *synchronous* manner. The final decoding output should be either '0' or '1', such that you can compare with the original image source data.

2.4 Noise

Zero mean Gaussian random values can be included in order to emulate the AWGN channel noise.

3 Project Problems

3.1 Synchronous Demodulation in Noise-free Channels

As discussed above, each bit of the image source data is encoded by NRZ, serialized and is modulated by BPSK. This is depicted in Figure 3.

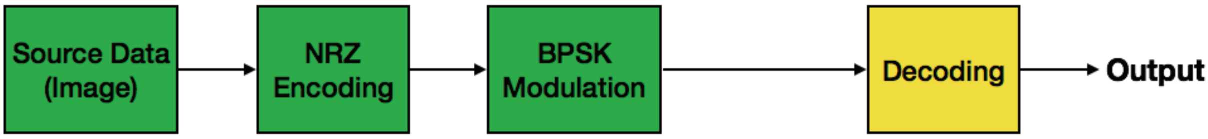

Figure 3: Digital data transmission in a noise-free channel.

The encoded, serialized and modulated data is stored in `mod_data` in the given MATLAB code (by executing `load('encoded_data.mat')`).

Project Problem 1
Design the decoding algorithm in the MATLAB code given as `project_code.m`. In the MATLAB code, default encoding parameters are already given.

1. Write codes for demodulation such that the final output is a matrix with the size of 100×100 having elements either '0' or '1'.
2. Clearly explain how your codes work.
3. Present your results as an image using `imshow` command in MATLAB.

3.2 Impact of Channel Noise on Decoding Performance

We now consider the data transmission through a noisy channel. To emulate this, we may add independently generated zero mean AWGN values to the data stream. This scenario is depicted in This is depicted in Figure 4.

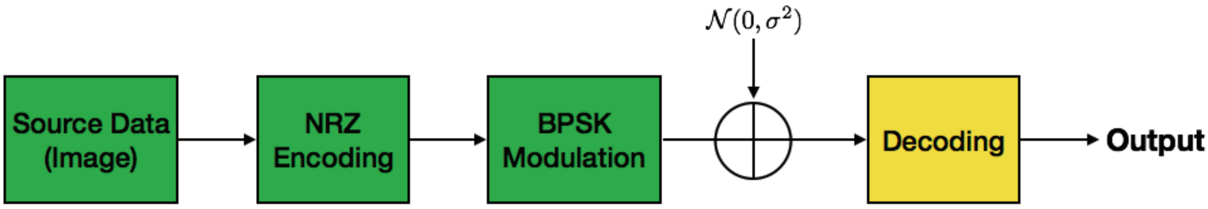

Figure 4: Digital data transmission in a noisy channel.

In MATLAB, Gaussian noise can be generated using MATLAB built-in function, `normrnd`, which is already implemented in the MATLAB code given as `project_code_noisy.m` with $\sigma^2 = 4$ for example.

Project Problem 2
Use the same decoding algorithm implemented for **Project Problem 1**. Define BER (P_b) as

$$P_b = \frac{\text{number of total errors}}{\text{total transmitted data}}. \tag{1}$$

In our project, the number of errors can be computed by comparing between the matrices decoded in noise-free and noisy channels.

1. With the additive noise given in the MATLAB code (i.e., $\sigma^2 = 4$), present your results as an image. What is the impact of the noise in the output image? Why?
2. Compute BERs given in (1) for $\sigma^2 = 1, 2, \ldots, 10$. Present plots for BERs as a function of $\log_{10}(1/\sigma^2)$.