Communication Systems Laboratory
Lab 6: Digital Modulation

(Report Due: 23:30, December 8, 2021)

1 Overview

In real world, we often transmit continuous-time signals that carry the information bits (coming from the source encoder or the channel encoder) through the physical medium[1]. To that end, you would need *symbol mapping*, *pulse shaping*, and *up/down conversion* as taught in class.

When transmitting signals through the waveform channel, noise could happen since the channel usually suffers from a variety of impairments. In this lab, we will study the effect of the additive white Gaussian noise (AWGN) adding the communication system, which might cause symbol error or bit error. In the lecture, we modeled the AWGN in the passband as a random process with several statistical properties. We have shown that under suitable assumptions, this model can be simplified as adding circularly symmetric complex Gaussian random variables to the symbol sequence on the constellation plan which are spanned by certain basis waveform. With this property, we can only focus on the constellation plan with induced Gaussian vectors as noise. We call this the *vector channel model*. In this Lab, the goal is to simulate different signaling schemes such as the Pulse Amplitude Modulation (PAM), the Phase-Shift Keying (PSK), and the Quadrature Amplitude Modulation (QAM) with the vector channel model. Use the *minimum distance rule* to decide the received symbol and calculate the symbol error rate (SER) or the bit error rate (BER). As we analyzed theoretically in the lecture, the error probability is closely related to the Q-function. We will compare our simulated results with the theoretical results to validate our simulation program.

2 Experiments

1. **(10 points)** Implement a `Matlab` function to map the input binary sequence to the symbol sequence in the following format:

   ```
   sym_seq = symbol_mapper(bin_seq, M, d, name)
   ```

   The input and output arguments are defined as

   - `sym_seq` is the symbol sequence consisting of the symbols in the constellation set.
   - `bin_seq` is the input binary sequence.

[1]Here we only consider classical communication. As for quantum communication, quantum photons are good options. We did not touch this part in this course. The interested readers can refer to the journal *Nature Photonics* by search keywords such as "quantum communication" or "quantum information transfer using photons" (otherwise, you would need quantum teleportation as we studied in Lab 2).

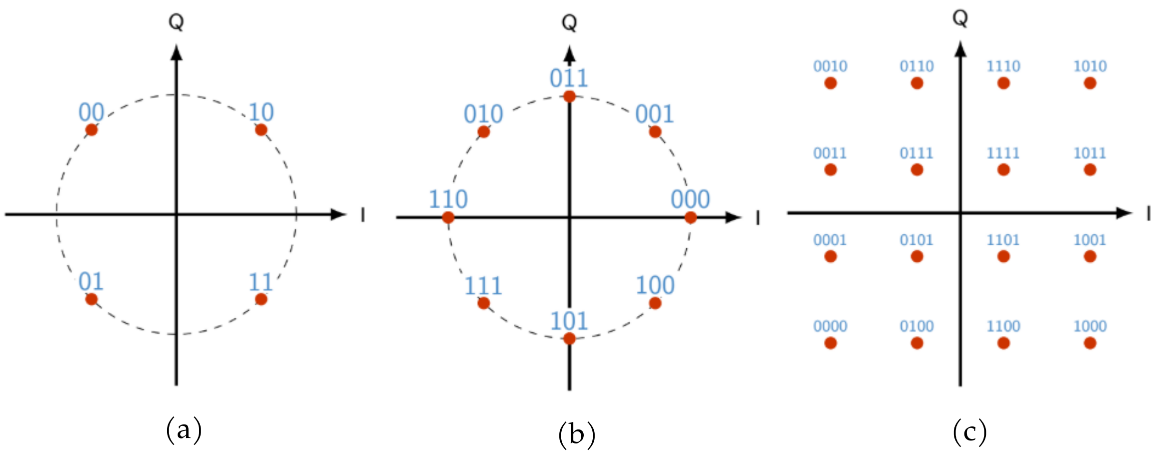

Figure 1: The gray-coded mapping for (a) QPSK, (b) 8PSK, and (c) 16QAM

- `M` is the number of points in the signal constellation set. For PAM and PSK, `M = 2,4,8,16`. For QAM, `M = 4,16,64`.
- `d` is a parameter related to the *minimum distance* among the constellation points.
- `name` is the name of the modulation. It can be `'PAM'`, `'PSK'` or `'QAM'`.

For the gray-coded mapping for PAM and PSK, assign $00\cdots0$ to the rightmost constellation point. Some examples of the gray-coded mapping for PSK and QAM are shown in Figure 1. (Note that the 4-PSK and 4-QAM are equivalent to the QPSK).

2. **(20 points) Decision:** In this part we will have to demodulate the received noisy signal. The performance of the optimal decision rule will be investigated.
Following the analysis as taught in class, additive white Gaussian noise in the passband is equivalent to a complex circularly-symmetric Gaussian random variable (zero-mean with variance $\frac{N_0}{2}$) to the symbol sequence on the constellation plane.

Signaling scheme	PAM	PSK	QAM
Minimum distance d_{min}	$\sqrt{\frac{12\log_2 M}{M^2-1}E_\text{b}}$	$2\sqrt{\log_2 M\sin^2(\frac{\pi}{M})E_\text{b}}$	$\sqrt{\frac{6\log_2 M}{M-1}E_\text{b}}$

(a) **(5 points)** Generate a binary sequence $\{b_i\}$ of length 10^4 where each bit is drawn from a Bernoulli distribution with $p = 1/2$ (i.e. the bits are equiprobable). Next we generate a QPSK sequence $\{u_i\}$ from $\{b_i\}$. The minimum distance $d = 1$. Plot the two-dimensional histogram of the symbol sequence $\{u_n\}$. You can use the `Matlab` command `histogram2` and `symbol_mapper` in Problem 1.

At the receiver's side, the received symbol sequence $\{r_i\}_i$ is given by $r_i = u_i + n$, where n is a circularly-symmetric complex Gaussian random variable with mean 0 and variance N_0. Plot the three 2-D histogram of $\{r_i\}$ for $E_b/N_0 = 0$, 10, and 20dB. Depict the decision regions on the histogram.

(b) **(15 points)** Implement a `Matlab` function `MD_symbol_demapper` to demap the symbol sequence from the binary sequence. We use the *minimum distance decision rule*.

```
bin_seq = MD_symbol_demapper(sym_seq, M, d, name)
```

The input and output arguments are defined as

- `bin_seq` is the output binary sequence.
- `sym_seq` is the received symbol sequence.
- `M` is the number of points in the signal constellation set. For PAM and PSK, `M = 2,4,8,16`. For QAM, `M = 4, 16, 64`.
- `d` is a parameter related to the minimum distance among the constellation points.
- `name` is the name of the modulation. It can be `'PAM'`, `'PSK'` or `'QAM'`.

Validate your implementation of `MD_symbol_demapper` by setting the symbol sequence to be $\{u_i\}$ in Problem 2a.

Compute simulated symbol error rate (SER) from the received symbol sequence $\{r_i\}_i$ in Problem 2a.

3. **(50 points) Error Probability and SNR:** In the previous problem, we learn to simulate the error probability using Monte-Carlo simulations. In the present problem, we will simulate the relationship between the error probability and the signal-to-noise ratio (SNR), compare the results with the theoretical upper bounds given in Table 1 below (you can use the `Matlab` function `qfunc`).

Signaling scheme	PAM	PSK	QAM
Theoretical bound	$2Q\left(\sqrt{\frac{6\log_2 M}{M^2-1}\frac{E_b}{N_0}}\right)$	$2Q\left(\sqrt{2\log_2 M\sin^2(\frac{\pi}{M})\frac{E_b}{N_0}}\right)$	$4Q\left(\sqrt{\frac{3\log_2 M}{M-1}\frac{E_b}{N_0}}\right)$

Table 1: Upper bounds for the error probability for the PAM, PSK, QAM signaling schemes.

(a) **(15 points)** In this question, we aim to simulate the SER with different SNRs E_b/N_0 for PAM, `M=2,4,8,16`. For each PAM signaling, randomly generate a sequence of equiprobable bits and transmit the sequence over such modulation scheme with different SNRs E_b/N_0. Please plot the SER on the vertical axis over the E_b/N_0 on the horizontal axis. The vertical axis is in the log-scale (10^0, 10^{-1}, 10^{-2}, $\dots$), for which you might use the `Matlab` function `semilogy`. Experiments will be conducted for `SNR` = $\{0, 1, \dots, 10\}$ in dB. The simulated SER should be at least 10^{-6}, which means that you need to generate large amounts of bits at the very beginning for simulation.

Furthermore, **plot the theoretical curves of SER in these figures**. Note that you will generate all the simulated curves and the theoretical curves on the same figure. It is recommended to use colors, markers, or legends to present your simulation results; see Figure 2 for example.

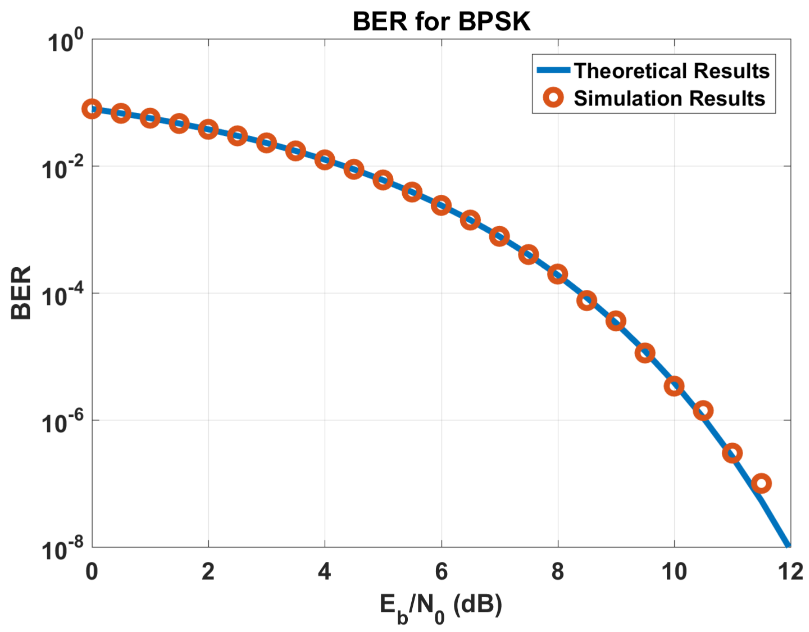

Figure 2: An example figure of the BER vs. SNR using the BPSK signaling; this demonstrates how your simulation figure should look like. If you simulate multiple signaling schemes, make sure the simulation curves are all presented in the same figure (using `hold on`), and use the `legend` indicates all of the curves.

(b) (**15 points**) Simulate the SER with different SNRs E_b/N_0 for PSK, `M=2,4,8,16` like you did in Problem 3a.

(c) (**15 points**) Simulate the SER with different SNRs E_b/N_0 for QAM, `M=4,16,64` like you did in Problem 3a.

(d) (**5 points**) Comment on the above results as much as possible

4. (**20 points**) **Convolutional codes + BPSK soft decoding** In this problem, you will simulate the convolutional codes (with the impulse generators $\mathbf{g1} = [101], \mathbf{g2} = [111]$) with BPSK signaling. At the output of the convolutional codes encoder,the BPSK maps coded bit c_i to $u_i = \sqrt{E_\text{b}}(-1)^{c_i}$ (you may or may not need `symbol_mapper` in Problem 1), i.e.

$$\begin{cases} 0 \mapsto \sqrt{E_\text{b}}; \\ 1 \mapsto -\sqrt{E_\text{b}}. \end{cases} \tag{1}$$

(Here, the minimum distance is $d = 2\sqrt{E_\text{b}}$.)

Send the signals through a vector channel ($r_i = u_i + n$) as you did in Problem 2. In the following, you will implement the *hard decision* and the *soft decision* to extract the information bits $\{b_i\}_i$.

(a) (**5 points**) Use the `MD_symbol_demapper` in Problem 2 to obtain the corrupted coded bits $\{\hat{c}_i\}$. Use the Viterbi algorithm in Lab 5 to decode $\{\hat{c}_i\}$ to get the estimated information bits

$\{\hat{b}_i\}_i$. We call this scheme the *hard decision* since we send coded bits (which are either 0 or 1) to the convolutional decoder (with the minimum Hamming distance rule).

Plot the BER curves with different SNRs all together on one figure. The curve of this setup should start from $E_b/N_0 = 0$ dB all the way to the E_b/N_0 that generates the BER of 10^{-5}. You should have no less than 10 simulation points spreading evenly on each curve.

(b) **(15 points)** Without using `MD_symbol_demapper`, send the received symbols $\{r_i\}_i$ (we called the soft information) directly to the convolutional decoder, and use the Viterbi algorithm (with the minimum Euclidean distance rule) to decode the information bits. We call this the *soft decoding*.

Simulate the performance for the soft decoding as you did in Problem 4a, and compare its performance to that of the hard decision decoding (on the same figure).

3 Lab Report

There is no format/typesetting requirements for your lab report, but you have to make your report decent and looking nice. In the report, you should address the results of the exercises mentioned above. You should also include your simulation program in the appendix of the report. Include whatever discussions about the new findings during the lab exercise, or the problems encountered and how are those solved. Please properly cite the literature if you referred to. Do not limit yourself to the exercises specified here.

So far, you have implemented source coding (the Huffman coding) for classical data compression, channel coding (the convolutional codes) for protecting the information bits, digital modulation which groups bits into symbols, and the optimal decision for deciding the received noisy symbols from the vector channel back to the information bits. The TA will teach you how to use the `USRP` Software Defined Radio and `LabVIEW` to transmit continuous signals through real-world communication channels. I believe you have had basic knowledge and understanding of the fundamental blocks of a digital communication system, and hence I would like say congratulations! This is indeed quite a journey (I know its never easy). And also congratulations to your quantum journey from Lab 1 to Lab 3. Besides, I hope you have realized the importance of simulation and how it is compared with theoretical analysis—both are essential for academic research and industrial development, and how fun by implementing simulation programs for what have been taught in class. However, learning is a never ending process. If you are interested or inspired by what you learned from the class/Labs, go ahead to join more related classes such as *Principles of Communication*, or the *Quantum Information and Computation*. Further, I highly encourage you to dig more in the literature survey and the final project. Make it part of your lifework. Hopefully, you can extend it to be your college seminar projects or even your Master/Ph.D. research projects. In the end, thank you for joining my course of *Communication Systems Laboratory*. I cordially wish you all the best and happiness in your future endeavors.