浙江大學

本科生课程报告

学年、学期： 2021 — 2022 学年 春夏 学期

课程名称： 通信原理

任课教师： 刘安

学生姓名： 黄嘉欣

学　　号： 3190102060

目录

Project：Signal Constellations and A Priori Probabilities

3190102060 黄嘉欣 信工 1903 班

1、Objective

As part of the assessment of the course, the project aims at testing students' understanding on the course materials, and providing students a chance to realize the theories have been taught in the class. In this project, the performance of a simple communication system will be simulated and compared with theoretical analysis.

2、System Simulation Model

Consider the following four 8-ary signal constellations in Figure 2.1. Because the last digit of my student ID is 0, I shall choose the Constellation (0) like this:

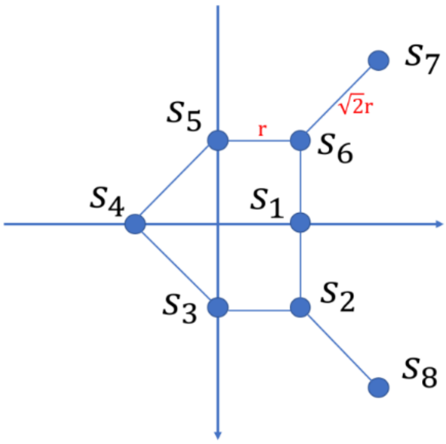

Figure 2.1 the Constellation (0)

Furthermore, a prior probability for each signal as follows will be applied in this project, which means the signals are not equally likely.

$$P(\boldsymbol{S}_1) = 0.2,\ \ P(\boldsymbol{S}_2) = 0.3,\ \ P(\boldsymbol{S}_3) = 0.1,\ \ P(\boldsymbol{S}_4) = 0.1,$$

$$P(\boldsymbol{S}_5) = 0.12,\ \ P(\boldsymbol{S}_6) = 0.08,\ \ P(\boldsymbol{S}_7) = 0.05,\ \ P(\boldsymbol{S}_8) = 0.05$$

In the system, the data source will generate 8-ary symbols according to the specified set of a priori probabilities. The channel model is AWGN and the performance of this 8-ary modulation scheme should be evaluated by computer simulation.

The tasks and their corresponding solutions can be found in the following parts of the report.

3、Task 0 – Minimum Distance Detector

① Receiver Block Diagram

As we have learned in the class, the minimum distance detector aims to classify the received signal to its nearest neighbor. In this procedure, we need to convert the time domain received signal $y(t)$ to the vector $\boldsymbol{y}$ first, and then compute distances between the vector and the points $\{\boldsymbol{S}_1, \dots \dots, \boldsymbol{S}_8\}$, followed with a distance selector. Apparently, because the dimension of the signal space is 2, we will only need 2 correlators. The diagram is shown as follows:

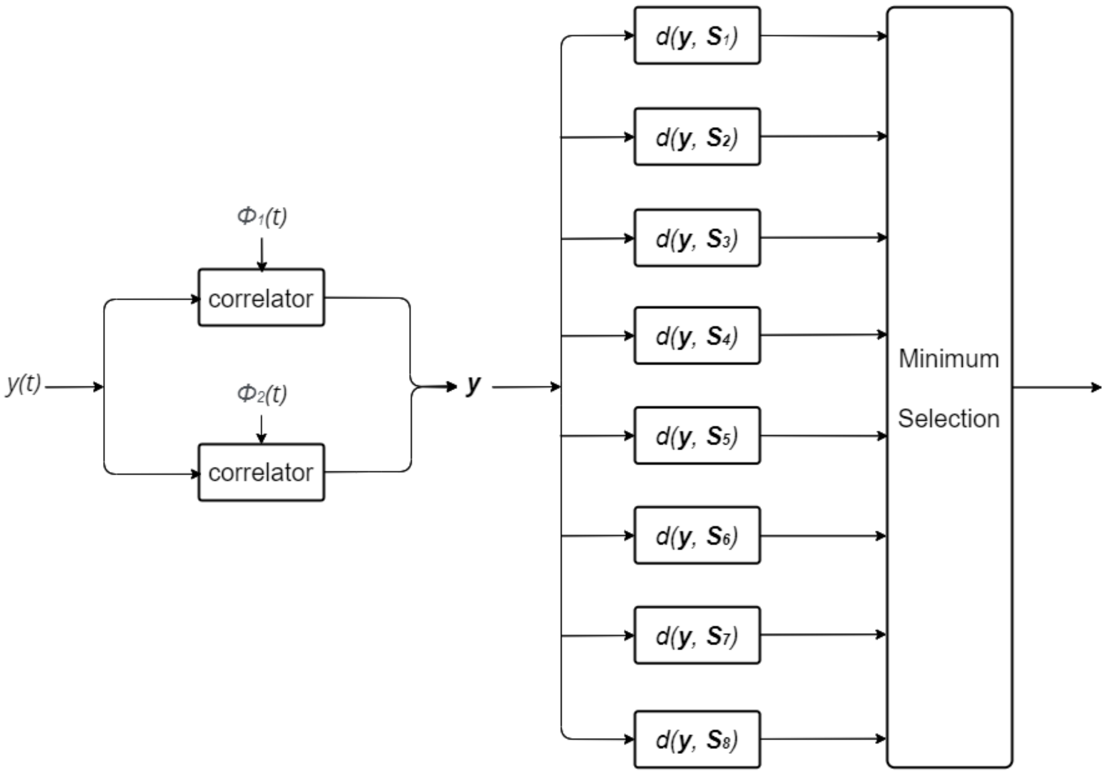

Figure 3.1 Diagram of the Minimum Distance Detector

② Decision Region

As we all know, boundaries of decision regions are perpendicular bisectors of the lines joining the original transmitted signals. Therefore, we are able to obtain the regions on the basis of Figure 2.1 by taking the intersection of the line's bisectors for each signal, which is sketched in Figure 3.2, and a result plotted by MATLAB has also been represented in the figure.

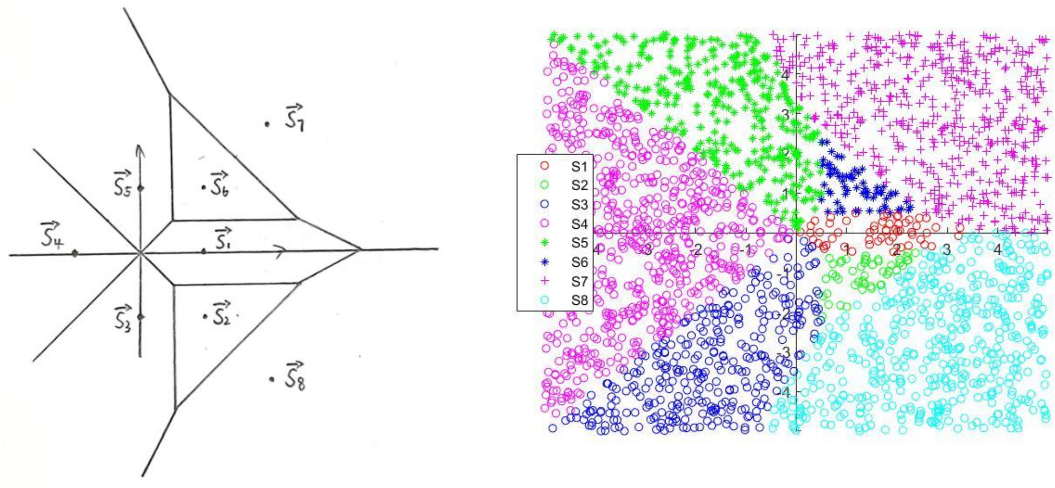

Figure 3.2 Decision Region

③ MATLAB Performance Simulation

On the basis of the analysis above, we will now use MATLAB to compute the SER when minimum distance detector is used. Here we assume that the data source generates symbols according to the a priori probabilities, as stated in part 2. The theoretical nearest neighbor union bound can be derived as follows:

Proof:

According to the Constellation (0), the average energy of symbols is:

$$E_s = 0.2r^2 + 0.6r^2 + 0.1r^2 + 0.1r^2 + 0.12r^2 + 0.16r^2 + 0.4r^2 + 0.4r^2 = 2.08r^2$$

$$\therefore r^2 = \frac{E_s}{2.08}$$

On the other hand, the error probabilities conditioned on each signal are:

$$\boldsymbol{S}_1:\ P_{e1} = P_e(\boldsymbol{S}_1 \to \boldsymbol{S}_2) + P_e(\boldsymbol{S}_1 \to \boldsymbol{S}_6) = 2Q\left(\sqrt{\frac{r^2}{2N_0}}\right) = 2Q(\sqrt{\frac{E_s}{4.16N_0}})$$

$$\boldsymbol{S}_2:\ P_{e2} = P_e(\boldsymbol{S}_2 \to \boldsymbol{S}_1) + P_e(\boldsymbol{S}_2 \to \boldsymbol{S}_3) = 2Q\left(\sqrt{\frac{r^2}{2N_0}}\right) = 2Q(\sqrt{\frac{E_s}{4.16N_0}})$$

$$\boldsymbol{S}_3:\ P_{e3} = P_e(\boldsymbol{S}_3 \to \boldsymbol{S}_2) = Q\left(\sqrt{\frac{r^2}{2N_0}}\right) = Q(\sqrt{\frac{E_s}{4.16N_0}})$$

$$\boldsymbol{S}_4:\ P_{e4} = P_e(\boldsymbol{S}_4 \to \boldsymbol{S}_3) + P_e(\boldsymbol{S}_4 \to \boldsymbol{S}_5) = 2Q\left(\sqrt{\frac{2r^2}{2N_0}}\right) = 2Q(\sqrt{\frac{E_s}{2.08N_0}})$$

$$S_5: \ P_{e5} = P_e(S_5 \to S_6) = Q\left(\sqrt{\frac{r^2}{2N_0}}\right) = Q(\sqrt{\frac{E_s}{4.16N_0}})$$

$$S_6: \ P_{e6} = P_e(S_6 \to S_1) + P_e(S_6 \to S_5) = 2Q\left(\sqrt{\frac{r^2}{2N_0}}\right) = 2Q(\sqrt{\frac{E_s}{4.16N_0}})$$

$$S_7: \ P_{e7} = P_e(S_7 \to S_6) = Q\left(\sqrt{\frac{r^2}{N_0}}\right) = Q(\sqrt{\frac{E_s}{2.08N_0}})$$

$$S_8: \ P_{e8} = P_e(S_8 \to S_2) = Q\left(\sqrt{\frac{r^2}{N_0}}\right) = Q(\sqrt{\frac{E_s}{2.08N_0}})$$

Therefore, the nearest neighbor union bound for SER is:

$$P_e = \sum_{i=1}^{8} P(S_i)P_{ei} = 1.38Q\left(\sqrt{\frac{E_s}{4.16N_0}}\right) + 0.3Q(\sqrt{\frac{E_s}{2.08N_0}})$$

Using the results we derived above, we can plot the simulation results and the nearest neighbor union bound of the Minimum Distance Detector as follows:

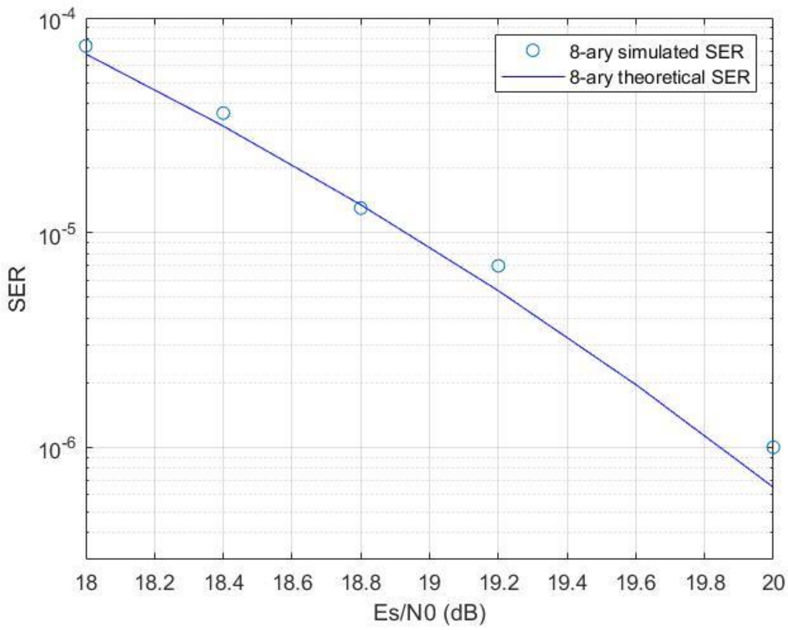

Figure 3.3 SER vs E_s/N_0

It is apparent that the simulated results of the SER versus $\frac{E_s}{N_0}$ are generally consistent with the theoretical upper bound, though the simulated points slightly fluctuate probably due to the randomness of the transmitted signals. Source code of this task can be found in file *src/Task_0.m*. However, because of the unequal a priori probability of each symbol, the Minimum Distance Detector is not the optimal receiver. We will further explore this question in the rest of the report.

4、Task 1 – SER Optimal Detector

① Optimal Detector Diagram

As we have learned in the class, the Optimal Detector should be Maximum a Posterior Detector, that is, for a received vector $\boldsymbol{y}$, we need to find the symbol $\boldsymbol{S}$ in $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3, \boldsymbol{S}_4, \boldsymbol{S}_5, \boldsymbol{S}_6, \boldsymbol{S}_7, \boldsymbol{S}_8\}$, so that the posterior probability $P(\boldsymbol{S}|\boldsymbol{y})$ is maximum. Using Bayes Rule, we can know that:

$$P(\boldsymbol{S}|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|\boldsymbol{S})P(\boldsymbol{S})}{P(\boldsymbol{y})}$$

then let's derive the Optimal Detector here.

Proof:

Because $P(\boldsymbol{y})$ is independent from the sent signals, the problem of maximizing $P(\boldsymbol{S}|\boldsymbol{y})$ is equivalent to the problem of maximizing $P(\boldsymbol{y}|\boldsymbol{S})P(\boldsymbol{S})$. As we all know, $\boldsymbol{y} = \boldsymbol{S} + \boldsymbol{n}$ and the noise vector is iid Gaussion random vector, whose variance is σ_n^2, therefore, $\boldsymbol{y}$ is a Gaussion random vector with a mean of $\boldsymbol{S}$ and variance $\sigma_n^2 = \frac{N_0}{2}$. In this case, we can obtain that:

$$P(\boldsymbol{y}|\boldsymbol{S}) = \frac{1}{(\sqrt{2\pi\frac{N_0}{2}})^2}\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{S}\|^2}{2\frac{N_0}{2}}\right) = \frac{1}{\pi N_0}\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{S}\|^2}{N_0}\right)$$

So $P(\boldsymbol{y}|\boldsymbol{S})P(\boldsymbol{S}) = \frac{1}{\pi N_0}\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{S}\|^2}{N_0}\right)P(\boldsymbol{S})$. Let's logarithm this formula and will have:

$$\ln[P(\boldsymbol{y}|\boldsymbol{S})P(\boldsymbol{S})] = -\ln(\pi N_0) - \frac{\|\boldsymbol{y}-\boldsymbol{S}\|^2}{N_0} + \ln\,[P(\boldsymbol{S})]$$

Apparently, the first term in the formula above is constant. In order to maximize $\ln[P(\boldsymbol{y}|\boldsymbol{S})P(\boldsymbol{S})]$, we are supposed to maximize the sum of the last two terms, which is equivalent to

$$\arg\min\,\{\|\boldsymbol{y}-\boldsymbol{S}\|^2 - N_0\ln[P(\boldsymbol{S})]\}\ \text{ for } \boldsymbol{S} \text{ in } \{\boldsymbol{S}_1, \dots\dots, \boldsymbol{S}_8\}.$$

Above all, the optimal detector needs to handle the following steps: 1) convert the time domain received signal $y(t)$ to vector $\boldsymbol{y}$. 2) compute distances between $\boldsymbol{y}$

and the points $\{\boldsymbol{S}_1, \dots \dots, \boldsymbol{S}_8\}$. 3) compute $\|\boldsymbol{y} - \boldsymbol{S}_i\|^2 - N_0 \ln{[P(\boldsymbol{S}_i)]}$ for each $\boldsymbol{S}_i$. 4) classify $\boldsymbol{y}$ to $\boldsymbol{S}_m$ that corresponds to the minimum value of the formula in 3). The block diagram should be sketched as follows:

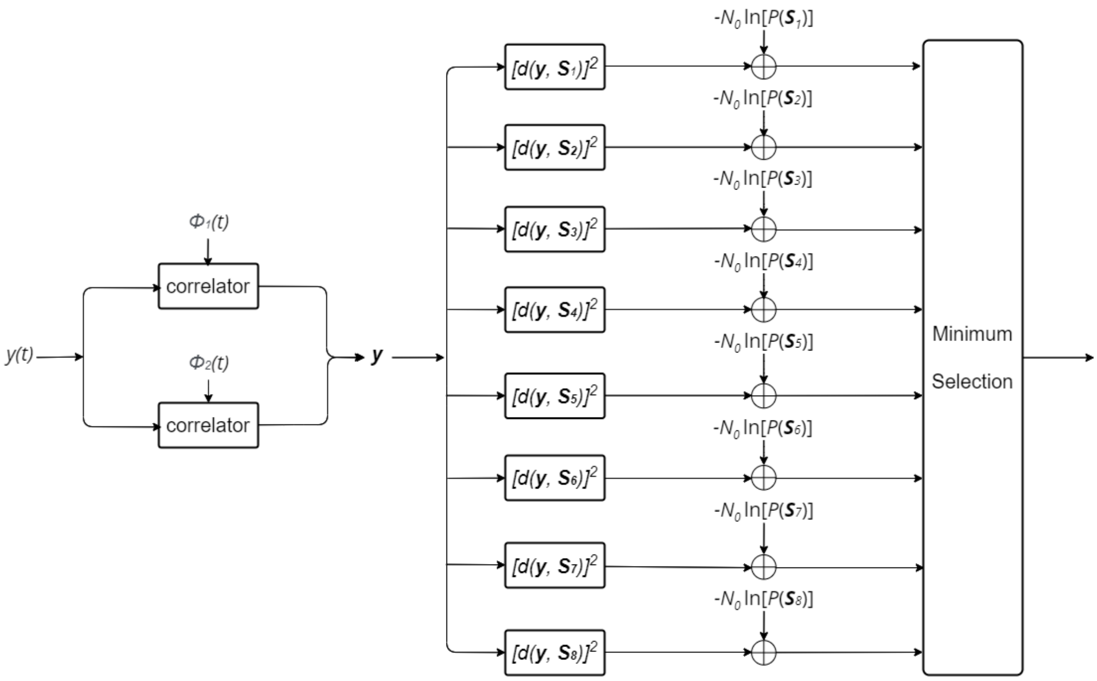

Figure 4.1 Diagram of the Optimal Detector

② MATLAB Decision Region Simulation

As we can see, the Optimal Detector is associated with the noise power. In this case, we will show the decision region under the following conditions, and the source code can be found in *src/Task_1* folder.

1) $N_0 = 0.0330r^2$, where $r = 1$

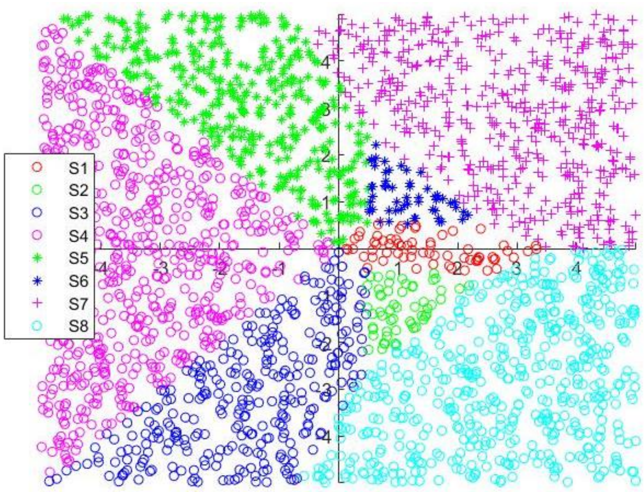

Figure 4.2.2 Decision Region when $N_0 = 0.0330r^2 = 0.0330$

2) $N_0 = r^2$, where $r = 1$

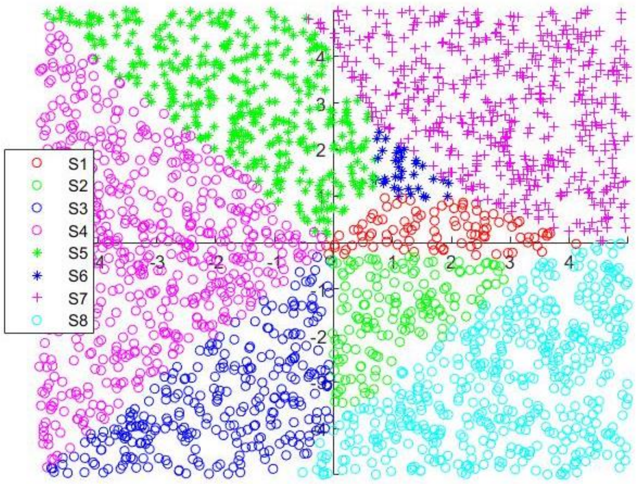

Figure 4.2.2 Decision Region when $N_0 = r^2 = 1$

It is obvious that the decision regions change more, compared with the Minimum Distance Detector, as N_0 increases, which meets our expectations. When N_0 is small, $N_0 \ln[P(\boldsymbol{S})]$ will also be small, making this addend less distinguished between the signals and the Optimal Detector similar with the Minimum Distance Detector to some extent. However, when N_0 gets bigger, the differences between branches will be more differentiated, which means the unequal a priori probabilities are able to make more contributions to rearranging the decision boundary of each signal, making the regions for signals with less probability relatively smaller, as the simulation results indicated, e.g. decision region for $\boldsymbol{S}_6$.

5、Task 2 – SER Simulation using Optimal Detector

① MATLAB Performance Simulation

To evaluate the performance of the Optimal Detector, we should first compute the theoretical SER, that is, the union bound in this problem.

Proof:

Since

$$P_e = \sum_{i=1}^{8} P(\boldsymbol{S}_i) \sum_{j \neq i} P(\boldsymbol{S}_i \to \boldsymbol{S}_j)$$

we will derive the general formula $P(\boldsymbol{S}_i \to \boldsymbol{S}_j)$ for the Optimal Detector now.

Without the loss of generality, let's choose $\boldsymbol{S}_1$ as the transmitted signal and use $\boldsymbol{y}$ to represent the received signal, where $\boldsymbol{y} = \boldsymbol{S}_1 + \boldsymbol{n}$ and $\boldsymbol{n}$ is the noise. We have:

$$P(\boldsymbol{S}_1 \rightarrow \boldsymbol{S}_j) = P\left(\|\boldsymbol{y} - \boldsymbol{S}_1\|^2 - N_0 \ln[P(\boldsymbol{S}_1)] \geq \|\boldsymbol{y} - \boldsymbol{S}_j\|^2 - N_0 \ln[P(\boldsymbol{S}_j)]\right)$$

$$= P(\|\boldsymbol{y} - \boldsymbol{S}_1\|^2 \geq \|\boldsymbol{y} - \boldsymbol{S}_j\|^2 + N_0 \ln [\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}])$$

In the time domain, the terms in the bracket above is equivalent to:

$$\int_0^{T_s} [y(t) - S_1(t)]^2 dt \geq \int_0^{T_s} [y(t) - S_j(t)]^2 dt + N_0 \ln [\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}]$$

Let's substitute $y(t) = S_1(t) + n(t)$ into this formula and simplify it:

$$\int_0^{T_s} -2n(t)[S_1(t) - S_j(t)] dt \geq \int_0^{T_s} [S_1(t) - S_j(t)]^2 dt + N_0 \ln [\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}]$$

Therefore,

$$\int_0^{T_s} n(t)[S_j(t) - S_1(t)] dt \geq \frac{1}{2}\int_0^{T_s} [S_1(t) - S_j(t)]^2 dt + \frac{N_0}{2} \ln [\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}]$$

We can use a random variable Z_j to denote the term on the left-hand side of the inequality, that is,

$$Z_j = \int_0^{T_s} n(t)[S_j(t) - S_1(t)] dt$$

Since $n(t)$ is a white Gaussion random process, Z_j is a Gaussion random variable with a mean of zero and a variance of

$$\sigma_z^2 = \frac{N_0}{2}\int_0^{T_s} [S_j(t) - S_1(t)]^2 dt = \frac{N_0}{2} \|\boldsymbol{S}_j - \boldsymbol{S}_1\|^2$$

Therefore, $Z_j \sim N(0, \sigma_z^2)$ and:

$$P\left(\|\boldsymbol{y} - \boldsymbol{S}_1\|^2 \geq \|\boldsymbol{y} - \boldsymbol{S}_j\|^2 + N_0 \ln \left[\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}\right]\right)$$

$$= P\left(Z_j \geq \frac{1}{2} \|\boldsymbol{S}_j - \boldsymbol{S}_1\|^2 + \frac{N_0}{2} \ln \left[\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}\right]\right)$$

$$= Q\left(\sqrt{\frac{\|\boldsymbol{S}_j - \boldsymbol{S}_1\|^2}{2N_0}} + \sqrt{\frac{N_0}{2\,\|\boldsymbol{S}_j - \boldsymbol{S}_1\|^2}}\ln\left[\frac{P(\boldsymbol{S}_1)}{P(\boldsymbol{S}_j)}\right]\right)$$

In this case, we are now able to compute the standard union upper bound on the basis of following results:

$$\sum_{j\neq 1} P(\boldsymbol{S}_1 \to \boldsymbol{S}_j) = Q\left(\sqrt{\frac{r^2}{2N_0}} + \sqrt{\frac{N_0}{2r^2}}\ln\left(\frac{0.2}{0.3}\right)\right) + Q\left(\sqrt{\frac{r^2}{2N_0}} + \sqrt{\frac{N_0}{2r^2}}\ln\left(\frac{0.2}{0.08}\right)\right) +$$
$$Q\left(\sqrt{\frac{r^2}{N_0}} + \sqrt{\frac{N_0}{4r^2}}\ln\left(\frac{0.2}{0.1}\right)\right) + Q\left(\sqrt{\frac{r^2}{N_0}} + \sqrt{\frac{N_0}{4r^2}}\ln\left(\frac{0.2}{0.12}\right)\right) +$$
$$Q\left(\sqrt{\frac{2r^2}{N_0}} + \sqrt{\frac{N_0}{8r^2}}\ln\left(\frac{0.2}{0.1}\right)\right) + 2Q\left(\sqrt{\frac{5r^2}{2N_0}} + \sqrt{\frac{N_0}{10r^2}}\ln\left(\frac{0.2}{0.05}\right)\right)$$

……

Combine the information that $r^2 = \frac{E_s}{2.08}$, we will obtain the standard union upper bound for SER. However, the expression of $P_{e\_standard}$ is somehow complex, we are not going show it here. Instead, the detailed formula has been implemented in the source file *src/Task_2.m*. Please feel free to check it in the file.

Similarly, we can figure the nearest neighbor's union upper bound out as:

$$P_{e\_nn} = \sum_{i=1}^{8} P(\boldsymbol{S}_i) \sum_{j\neq i, \min \|\boldsymbol{S}_j - \boldsymbol{S}_1\|} P(\boldsymbol{S}_i \to \boldsymbol{S}_j)$$
$$= \sum_{i=1}^{8} P(\boldsymbol{S}_i) P_{ei}$$

where:

$$P_{e1} = Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{3}\sqrt{\frac{1.04N_0}{E_s}}\right) + Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{5}{2}\sqrt{\frac{1.04N_0}{E_s}}\right)$$
$$P_{e2} = Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{3}{2}\sqrt{\frac{1.04N_0}{E_s}}\right) + Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln 3\sqrt{\frac{1.04N_0}{E_s}}\right)$$
$$P_{e3} = Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{1}{3}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{e4} = Q\left(\sqrt{\frac{E_s}{2.08N_0}}\right) + Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{5}{6}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

$$P_{e5} = Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{3}{2}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{e6} = Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{5}\sqrt{\frac{1.04N_0}{E_s}}\right) + Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{3}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{e7} = Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{5}{8}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

$$P_{e8} = Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{1}{6}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

Now we are able to use MATLAB to simulate the SER performance of our Optimal Detector, and the graph is plotted as follows:

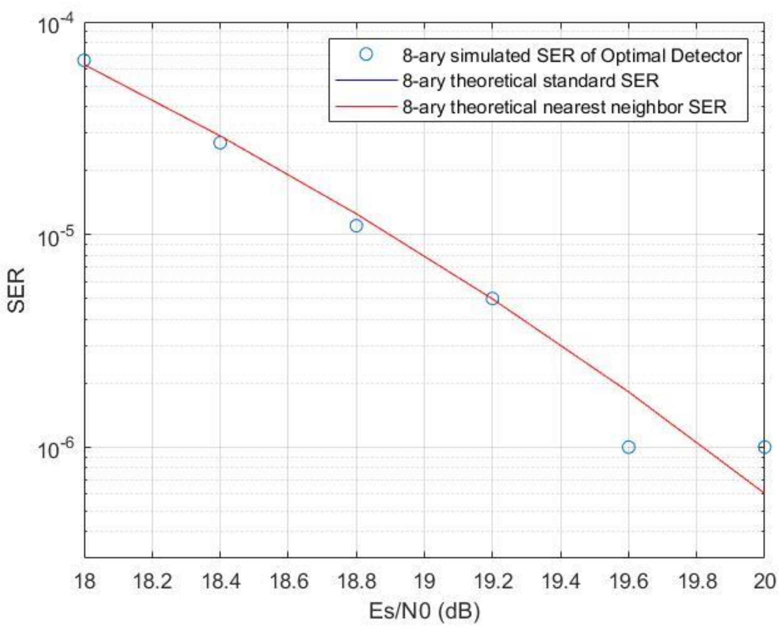

Figure 5.1.1 Performance of the Optimal Detector

Apparently, the simulation results are in general consistent with the theoretical union upper bound we have just computed, which proves the correctness of our designed system. In Figure 5.1.1, it is hard to tell the standard SER union bound from the nearest neighbor's. In this case, let's take a closer look at the figure, as shown in Figure 5.1.2.

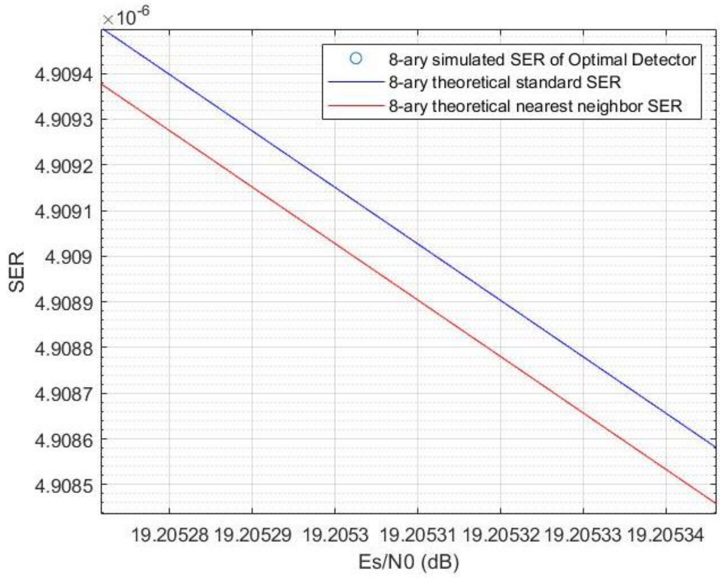

Figure 5.1.2 Comparison of the Union Bounds

As we can see, although the difference between these two bounds is little, the nearest neighbor union bound is smaller than standard union bound, which satisfies our expectations.

② Detectors Comparison

To distinguish the Minimum Distance Detector and the Optimal Detector under the condition of unequal a priori probability of each signal, we compared the performance of these two detectors, and the results are plotted in Figure 5.2.

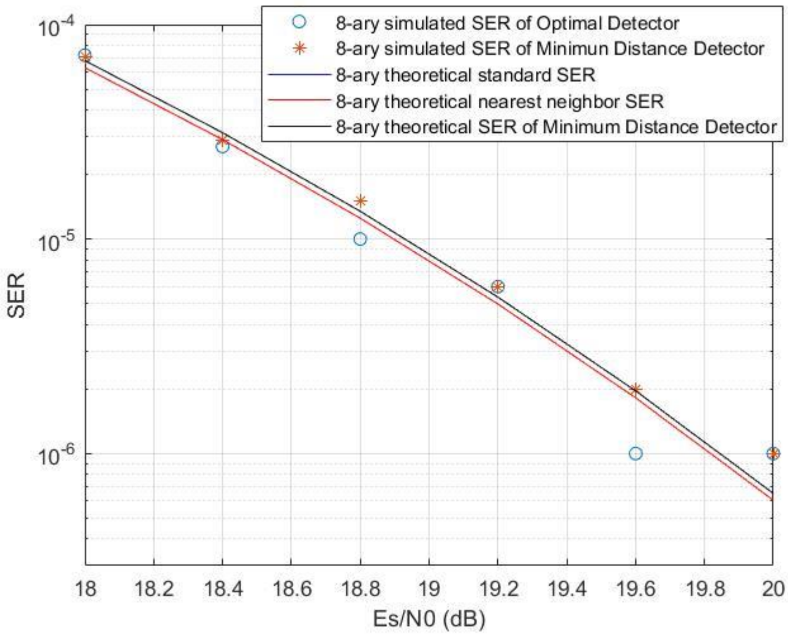

Figure 5.2 Comparison of the Detectors

Firstly, it is obvious that the SER union bounds for Optimal Detector is tighter than that for the Minimum Distance Detector. In this case, the Optimal Detector will have a higher accuracy and thus becoming a better choice for the design of the system. Secondly, the simulated SERs of the Optimal Detector are generally smaller than the Minimum Distance Detector, which means its practical performance is relatively better. To sum up, our Optimal Detector is designed correctly. For the case where the a priori probability for transmitted signal varies from each other, we should not simply use the Minimum Distance Detector to act as the receiver, but taking the probabilities into consideration, as the Optimal Detector does, to obtain a higher accuracy and better performance of the system.

6、Task 3 – Bit Mapping Design

① Proof of the Best Mapping Scheme

As is known to us, BER is associated with the SER to some extent. Because the performance of the nearest neighbor union upper bound has little difference with the standard one, here we will use it to derive the expression the BER.

Proof:

As we have computed,

$$P_{e\_nn} = \sum_{i=1}^{8} P(\boldsymbol{S}_i) \sum_{j\neq i, \min \|\boldsymbol{S}_j - \boldsymbol{S}_1\|} P(\boldsymbol{S}_i \to \boldsymbol{S}_j)$$

$$= \sum_{i=1}^{8} P(\boldsymbol{S}_i) P_{ei}$$

When a symbol error occurs, n out of the $\log_2(M) = 3$ bits ($1 \leq n \leq 3$) will change correspondingly, that means a symbol error will be accompanied by n bit errors. Denote the number of changed bits from $\boldsymbol{S}_i$ to $\boldsymbol{S}_j$ as n_{ij}, then the bit error probability for each signal P_{bi} should be like:

$$P_{b1} = \frac{n_{12}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{3}\sqrt{\frac{1.04N_0}{E_s}}\right) + \frac{n_{16}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{5}{2}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{b2} = \frac{n_{21}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{3}{2}\sqrt{\frac{1.04N_0}{E_s}}\right) + \frac{n_{23}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln 3\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{b3} = \frac{n_{32}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{1}{3}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{b4} = \frac{n_{43}}{3} Q\left(\sqrt{\frac{E_s}{2.08N_0}}\right) + \frac{n_{45}}{3} Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{5}{6}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

$$P_{b5} = \frac{n_{56}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{3}{2}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{b6} = \frac{n_{61}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{5}\sqrt{\frac{1.04N_0}{E_s}}\right) + \frac{n_{65}}{3} Q\left(\sqrt{\frac{E_s}{4.16N_0}} + \ln\frac{2}{3}\sqrt{\frac{1.04N_0}{E_s}}\right)$$

$$P_{\boldsymbol{b}7} = \frac{n_{76}}{3} Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{5}{8}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

$$P_{b8} = \frac{n_{82}}{3} Q\left(\sqrt{\frac{E_s}{2.08N_0}} + \ln\frac{1}{6}\sqrt{\frac{0.52N_0}{E_s}}\right)$$

where $n_{ij} = n_{ji}$. To minimize the BER:

$$P_{b\_nn} = \sum_{i=1}^{8} P(\boldsymbol{S}_i) P_{bi}$$

we are supposed to minimize n_{ij}, the coefficients of the Q functions. Since $1 \leq n_{ij} \leq 3$, it is the best choice to use the Gray code to ensure that only one bit out of the 3 bits changes.

On the other hand, it should be noted that coefficients in the formulas above are asymmetric. For example, P_{b8} contains n_{82} while P_{b2} ignores the bit changes from $\boldsymbol{S}_2$ to $\boldsymbol{S}_8$ for their relatively further distance. Although the code length 3 means that one code will have 3 neighbors, we should know that they are shared and the constellation is somehow irregular. Therefore, it is impossible to find a scheme so that all n_{ij} in the formulas are 1. To be more specific, assume we map $\boldsymbol{S}_i$ to a sequence $[\widehat{\boldsymbol{b}}_i]$, where $i = 1, \dots, 8$. Because $\boldsymbol{S}_2$ and $\boldsymbol{S}_6$ are both adjacent to $\boldsymbol{S}_1$, 2 bits of their corresponding bit sequence will be different. Therefore, the bit sequence of either $\boldsymbol{S}_3$ or $\boldsymbol{S}_8$, which are close to $\boldsymbol{S}_2$, will have a 3-bits change from $\boldsymbol{S}_6$ and the left one will have a 1-bit difference. If so, there need to exist 4 sequences that only have one bit change from $\boldsymbol{S}_6$, which is apparently impossible.

Above all, we should compare the values of the Q functions in the formulas and choose the labeling scheme to obtain an overall minimum. Using the online tool for drawing graphs of functions, we can easily get the order of the terms for $\frac{E_s}{N_0} \in [17,20]$. Denote the jth term in P_{bi} as Q_{ij}, the value's order is:

$$Q_{31} > Q_{61} > Q_{11} = Q_{62} > Q_{21} = Q_{51} > Q_{12} > Q_{22} > Q_{81} > Q_{71} > Q_{42} > Q_{41}$$

The bigger Q_{ij} is, the smaller its corresponding n shall be, so the ideal situation is:

$$n_{32} \le n_{61} \le n_{12} \le n_{65} \le n_{82} \le n_{76} \le n_{45} \le n_{43}$$

However, through the analysis, the best distribution for n is:

$$n_{32} = n_{61} = n_{12} = n_{65} = n_{82} = n_{45} = n_{43} = 1, n_{76} = 2$$

and $n_{ij} = n_{ji}$. Note that we can't assign $n_{45} = 2$ or $n_{43} = 2$ while the others are 1. For instance, if $n_{43} = 2$, $\boldsymbol{S}_6$ and $\boldsymbol{S}_3$ will both have 2-bits difference from $\boldsymbol{S}_4$, which means $\boldsymbol{S}_6$ and $\boldsymbol{S}_3$ are 2-bits different too. However, $\boldsymbol{S}_1$ should change 2 bits on the basis of $\boldsymbol{S}_3$, conflicting with the requirement $n_{61} = 1$. Similarly, if $n_{45} = n_{43} = 2$, $\boldsymbol{S}_1$、$\boldsymbol{S}_4$、$\boldsymbol{S}_5$、$\boldsymbol{S}_8$ will all have 2-bits difference from $\boldsymbol{S}_3$, which is obviously impossible.

Lastly, one possible labeling scheme using the distribution mentioned above could be as follows:

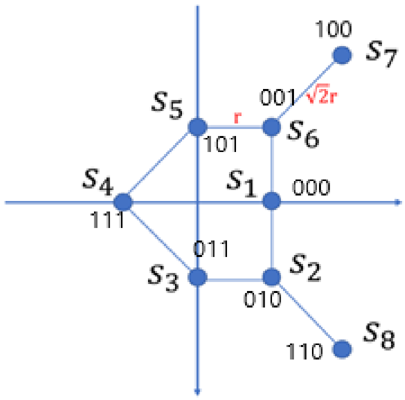

Figure 6.1 the Best Bit Mapping Scheme

that is, $000 \to \boldsymbol{S}_1, 010 \to \boldsymbol{S}_2, 011 \to \boldsymbol{S}_3, 111 \to \boldsymbol{S}_4, 101 \to \boldsymbol{S}_5, 001 \to \boldsymbol{S}_6, 100 \to \boldsymbol{S}_7$, which will be simulated by MATLAB later.

② MATLAB Performance Simulation

Using the bit mapping designed above, we can figure the theoretical BER out by substituting each n_{ij} into the expression of P_{b_nn} It should be noted that $P_{b\_nn} \approx \frac{1}{3} P_{e\_nn}$. For brevity, the detailed formula will not be shown here, but can be checked freely in the source file *src/Task_3.m*. To compare the performance of the design, a random mapping scheme will be also simulated, which is plotted as follows:

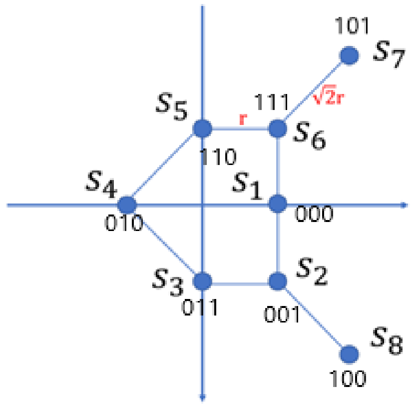

Figure 6.2.1 a Random Bit Mapping Scheme

In this case, we have:

$$n_{32} = n_{12} = n_{65} = n_{76} = n_{45} = n_{43} = 1, n_{82} = 2, n_{61} = 3$$

and the corresponding BER union bound can be found in the source file as well.

Here are the simulation results, representing the simulated BER and theoretical union bound versus $\frac{E_S}{N_0}$ for both the best labeling scheme and the random one.

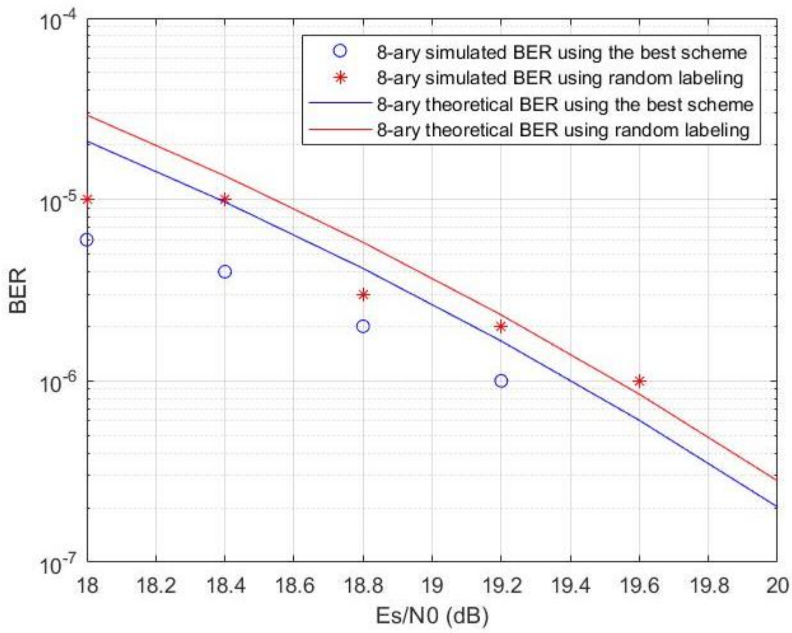

Figure 6.2.2 Comparison of the Two Schemes

Note that the simulation result using the best scheme is 0 when $\frac{E_s}{N_0} = 19.6 \text{ dB}$, which is not drawn in Figure 6.2.2. It is obvious that the simulated BERs are generally consistent with the theoretical union bound. Furthermore, we can see that the simulated BERs and the upper bound using the best scheme are both smaller than the random labeling, which satisfies our requirement and approves the scheme's correctness.