Communications Lab 4 Report B05901092 歐瀚墨

1. Preface

My code is uploaded to the following link:

https://github.com/ouhanmo/Comm_Lab

The main program is lab4.m

To run the program, please set the exper variable to the experiment number (1-3).

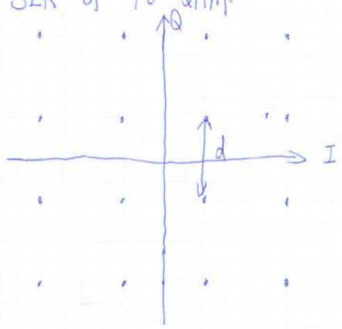
2. Experiment Results

I. BER and SER

Handwriting Part

Communications Lab4 B05901092 歐瀚墨

SER of 16-QAM:



$$E_{avg} = 2(0.5^2+0.5^2+1.5^2+1.5^2)d^2 = 10d^2$$

Perr of I phase:

$$P_{err} = \frac{1}{4}\left(2Q\left(\frac{\frac{d}{2}}{\sqrt{\frac{N_0}{2}}}\right)\underbrace{\times 2}_{\text{Middle}} + Q\left(\frac{\frac{d}{2}}{\sqrt{\frac{N_0}{2}}}\right)\underbrace{\times 2}_{\text{Side}}\right)$$

$$= \frac{3}{2}Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = \frac{3}{4}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\frac{d^2}{N_0}}\right)$$

Perr of Q phase is similar.

$$\Rightarrow P_{Symbol\ Err} = 1-(1-P_{err})^2 = 2P_{err} - P_{err}^2$$

while $P_{err} = \frac{3}{4}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\frac{d^2}{N_0}}\right) = \frac{3}{4}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\frac{E_{avg}}{10N_0}}\right) = \frac{3}{4}\mathrm{erfc}\left(\sqrt{\frac{S}{40N}}\right)$

Upper Bound: (Union Bound)

$$\frac{1}{16}(2\times4+3\times8+4\times4)\,Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = 3Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = \frac{3}{2}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\frac{d^2}{N_0}}\right)$$

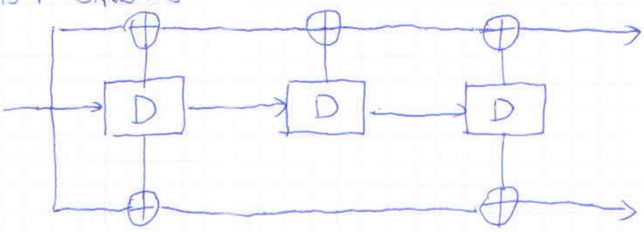
Lower Bound (Only one direction is considered)

$$\frac{1}{16}(1\times16)\,Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = Q\left(\sqrt{\frac{d^2}{2N_0}}\right) = \frac{1}{2}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\frac{A^2}{N_0}}\right)$$

Convolutional Code:

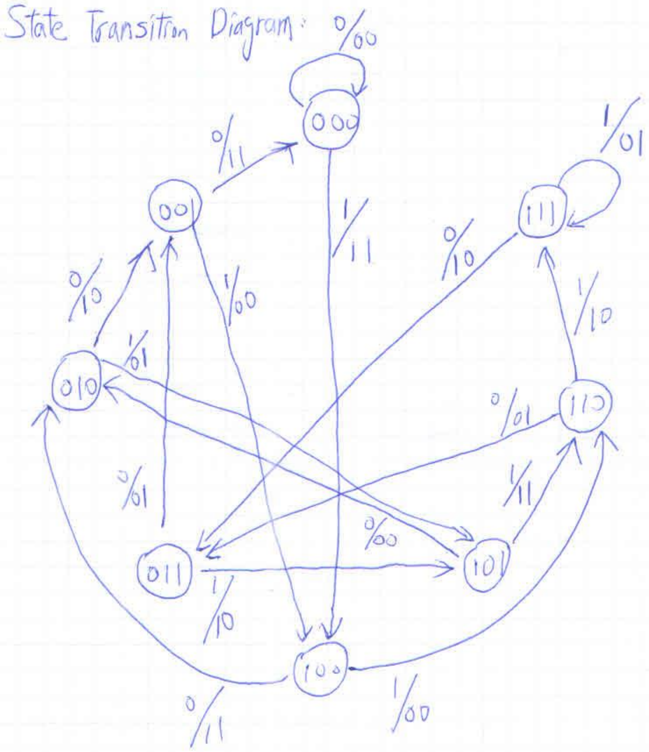
$h^{(1)} = [1\ 1\ 1\ 1] \quad h^{(2)} = [1\ 1\ 0\ 1]$

Register Structure:

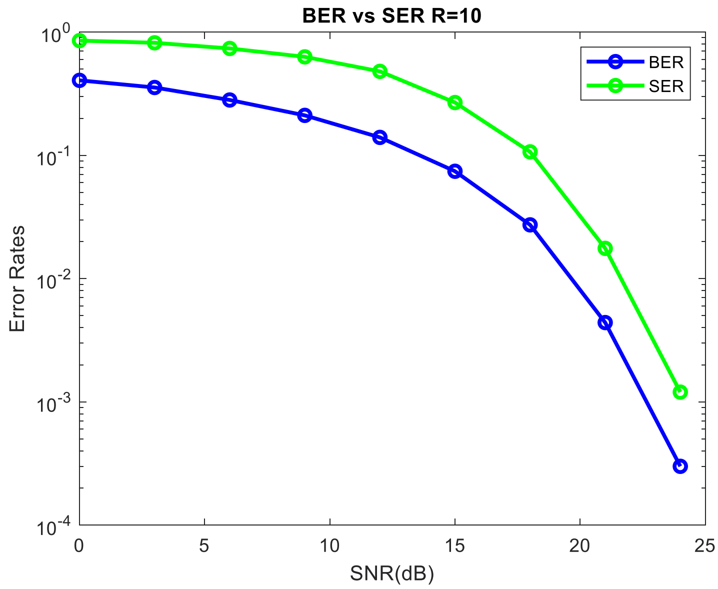


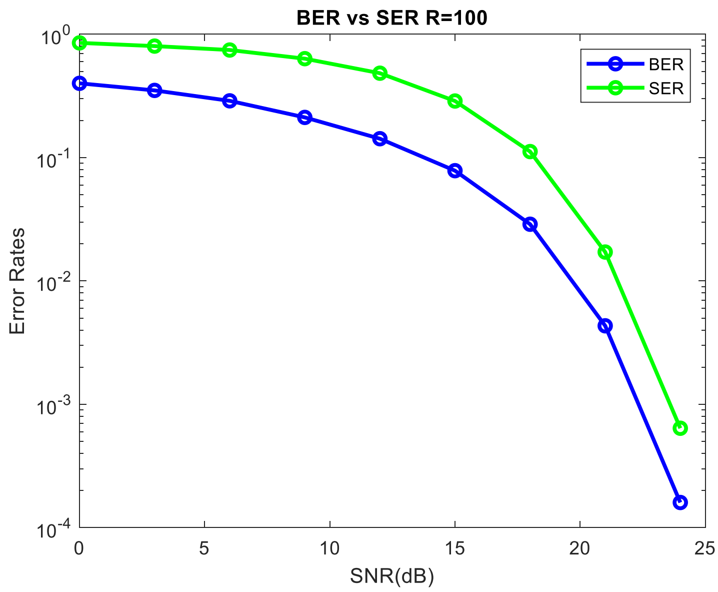
Code Rate $= \frac{1}{2}$

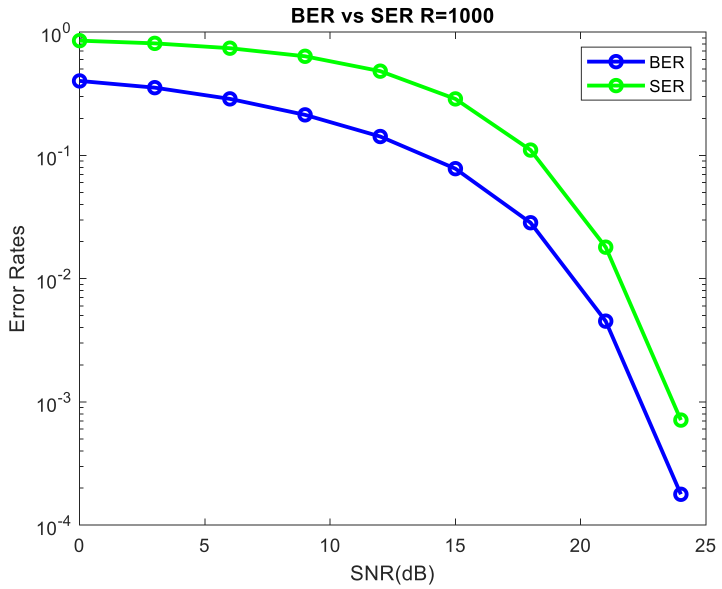
State Transition Diagram:



Figures for different R are below :

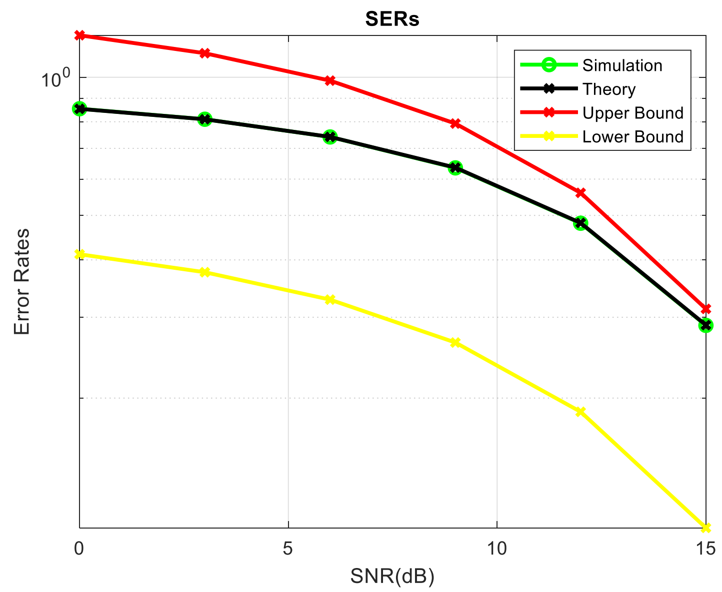

BER vs SER R=10
BER
SER
Error Rates
SNR(dB)
0
5
10
15
20
25


BER vs SER R=100
BER
SER
Error Rates
SNR(dB)
0
5
10
15
20
25



We can see that when in low SNRs, the parameter R does not affect the result, this is because that the number of errors is large enough. While in high SNRs, since errors seldom occur, more repetitions are required to get an accurate result.

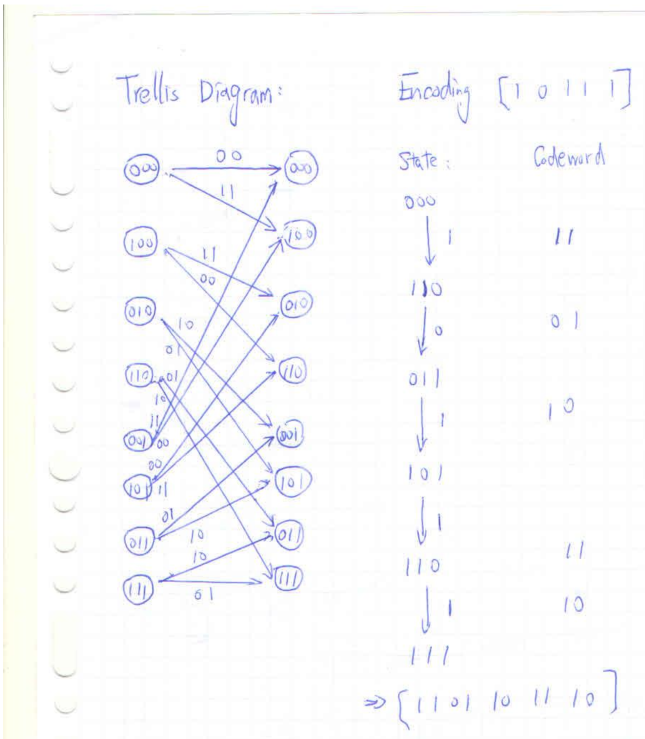
The simulation and theories are compared in the following figure.



The simulation clearly matches the theory, verifying the correctness of itself. While the distance of upper and lower bounds decrease as SNR increases.

II. Convolutional Code

Handwriting Part:



Coding:

During encoding, what I did was simply call the convolution function in matlab, and then apply module-2 on the result. This approach is easier then registers.

The decoding phase is much complicated. By using the de2bi and bi2de functions, I was able to iterate though all possibilities and calculate the Huffman distance. For each node on the trellis diagram, the program computes the shortest distance of both 0 and 1 as inputs, then it carries on the remaining Viterbi algorithm.

My algorithm is provided in the code on Github. Tried testing with random binary sequence. After encoding and decoding, the sequence remains unchanged, which therefore verifies the correctness.

III. Handel

After adding noise, with the help of convolutional code, the noise can be reduced to the amount that it can not be distinguished with the noise caused by quantization.

3. Remarks:

Quite an easy task. The main challenge was the Viterbi Decoding part. With help of de2bi function, the task was finished with relish.