Beijing-Dublin International College

Assignment 2

Simulation of a Hamming-coded 16-QAM System

Name: ChengLitao	**UCD ID:** **17206018**

Honesty Pledge:

I clearly understand the Academic Rules of Beijing University of Technology and University College Dublin and I am aware of the punishment for violating the Rules of Beijing University of Technology and University College Dublin. I hereby promise to abide by the relevant rules and promise the originality of my work. If found violating the rules, I would accept the punishment thereof.

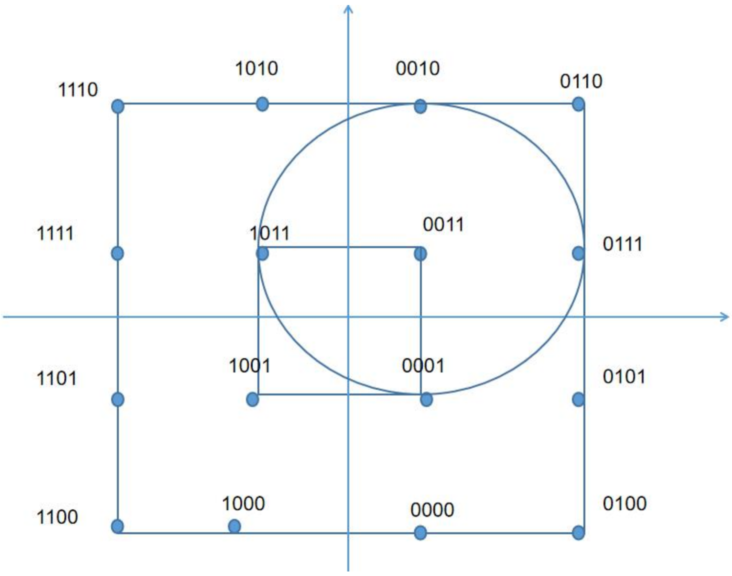
Date: 18/11/2020

1 Introduction

The mechanism of the Hamming-coded：
hamming code inserts the verification code in the transmitted message stream. when the computer stores or transfer data, it can check the check bits to detect and correct single bit errors.

•Block length: $n = 2^m - 1$
•Number of data bits: $k = 2^m - m - 1$
•Number of check bits: $n - k = m$

In this assignment, we use the matrix representation to get encoded information and sent them by the 16-QAM modulation.
The 16 QAM is composed of two independent orthogonal 4-ASK, and the 4-ASK is obtained by using multilevel signal to key the carrier. which is a generalization of 2 ask modulation. Compared with 2 ask, this modulation has the advantage of high information transmission rate.



In this constellation, it use the 4-bit gray code, so each closest message change only one bit.

In sender part, the generation matrix can be described as below(Ik is the k order identity matrix):

$$G = \left(\, P \mid I_k \, \right) = \begin{pmatrix} \boxed{\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}} & \boxed{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}} \end{pmatrix}.$$

G is used to generate the code word **c** from the initial code u:

$$c = uG$$

In this (7,4) linear block code case, the first three code of code word **c** is the check bit of redundancy.

The received vector may be written as **r = c + e** , where **c** denotes the transmitted code word and **e** the error vector happened in the transmission.

In receiver part, the parity-check matrix is given by:

$$H = \left(\, I_{n-k} \mid P^T \, \right).$$

parity-check matrix is used to generated the syndrome **s** which used to detect error and correct one-bit-error **:**

$$s = rH^T = eH^T.$$

In general, we can use the syndrome by following 3 rules:
1. If the syndrome contains all zeros, no error
2. If syndrome contains only one bit set to 1, then there was an error in the check bits and no correction is needed
3.If syndrome has more than one bit set to 1, then the numerical value of the syndrome indicates the position of the data bit in error. We invert the data bit to correct it.

2 Test for Hamming code performance

Assume the initial code for transmission is u=[1100];

Case1: one-bit error:

```
u =

     1     1     0     0


G =

     1     1     0     1     0     0     0
     0     1     1     0     1     0     0
     1     1     1     0     0     1     0
     1     0     1     0     0     0     1


c =

     1     0     1     1     1     0     0


c1 =

     1     0     0     1     1     0     0


s =

     0     0     0


s1 =

     0     0     1
```

If the error e=[0010000] (means that the third bit reverse), then the syndrome s will change from [000] to [001].
The corresponding relation between syndrome s to the error location is shown in the table form:

Table 1: Decoding table for the (7, 4) Hamming code.

Syndrome s	Error Vector e
000	0000000
001	0010000
010	0100000
011	0000100
100	1000000
101	0000001
110	0001000
111	0000010

Case2: two-bit error:

If the error e=[0110000] (there are two bits error in the transmission):

```
c =

     1     0     1     1     1     0     0


c1 =

     1     1     0     1     1     0     0


s =

     0     0     0


s1 =

     0     1     1
```

we can find that it can not correct the error. On the contrary, it will reverse the 5^{th} bit, and create more bit error, that is why there is a cross point between the BER curve and the theoretical curve.

Case3: three-bits or more error:

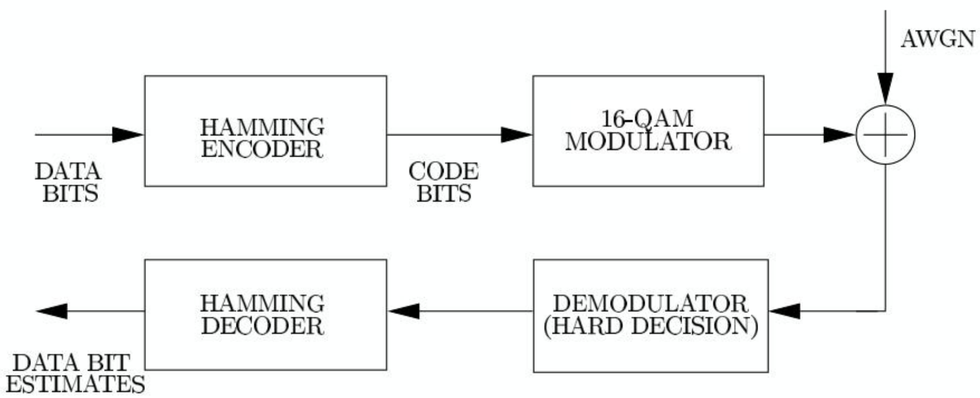
If the error e=[0110100] (there are three bits error in the transmission):

```
c =

     1     0     1     1     1     0     0


c1 =

     1     1     0     1     0     0     0


s =

     0     0     0


s1 =

     0     0     0
```

We can find that s1=[000] which indicate that there is no error in transmission. So it can not detect the bits error more than two bits.

In conclusion, (7,4) Hamming code have one-bit correction and two-bits error detection ability.

3 explanation for simulation code



```
sigma
%1.CreatbitSignal
data_bits_in=rand(symbolNum,4)<0.5;
%2.Hamming encode
code_bits_in=mod(data_bits_in*G,2);
bit_in=(reshape(code_bits_in,[4,7*groupNum]))';


```

The data matrix size is symbolNum*4， and use hamming code to generate symbolNum*7.

```
groupNum=150000;
symbolNum=4*groupNum;
```

Then I design symbolNum=4*groupNum;
So we can reshape hamming code matrix to 4*(7*groupNum), which is suitable for 16-QAM modulation.

Then the main index is designed by:

```
%7calculate SER & BER
a1= (symbol_in'==symbol_out);
SER=[SER,1-(sum(a1(:)==1)/(7*groupNum))];

a3=bit_in==bits_out;
BER=[BER,1-(sum(a3(:)==1)/(4*7*groupNum))];

a4= (code_bits_in(:,4:7)==revised_bits(:,4:7));
BER_revised=[BER_revised,1-(sum(a4(:)==1)/(4*4*groupNum))];

a5= a4(:,1)&a4(:,2)&a4(:,3)&a4(:,4);
SER_revised=[SER_revised,1-(sum(a5(:)==1)/(4*groupNum))];
```

4 Derivation and plot the SER curve

$$E_s = \frac{1}{16}\sum_{i=1}^{16} s_x^2 + s_y^2$$

$$= \frac{1}{4}\sum_{i=1}^{4} s_x^2 + s_y^2$$

$$= \frac{1}{4}\left[(0.5d)^2 + (0.5d)^2 + [(0.5d)^2 + (1.5d)^2]\times 2 + (1.5d)^2 + (1.5d)^2\right]$$

$$= \frac{1}{4} \times 10d^2$$

$$= 2.5d^2$$

$$P(\text{error})$$

$$= 1 - P(\text{correct})$$

$$= 1 - P_x(\text{correct}) \cdot P_y(\text{correct})$$

$$= 1 - \left[1 - \frac{2(4-1)}{4} Q(d/\sqrt{2N_0})\right]^2$$

$$= -\frac{9}{4} Q^2(d/\sqrt{2N_0}) + 3Q(d/\sqrt{2N_0})$$

Form the book supported by dear professor, I found a derivation of effective gain given by :

$$gain = \frac{(2^\rho - 1) * d^2 * d_{min}}{6E_s};$$

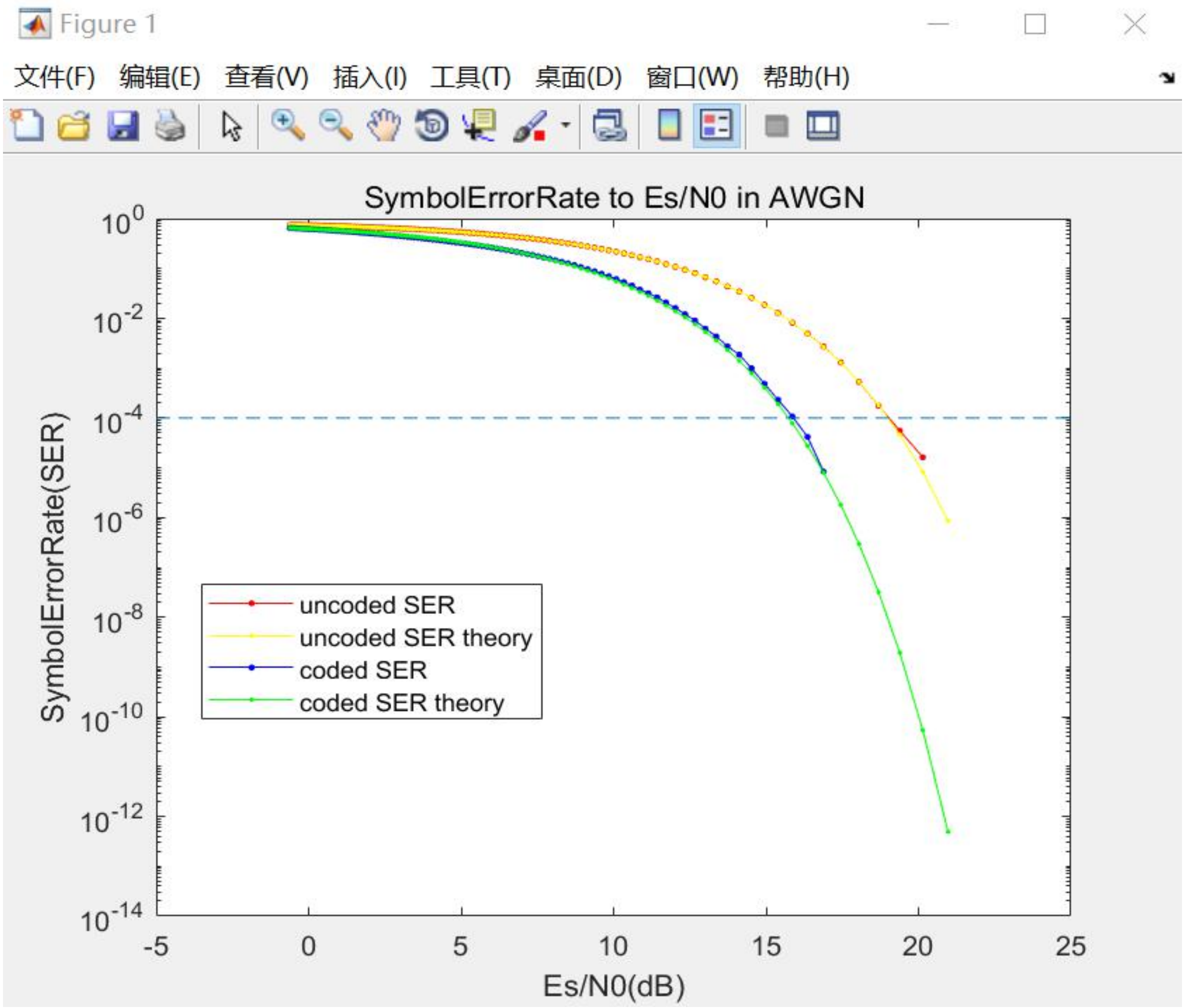
Since the 16-QAM can be divided the For 4-ASK case:

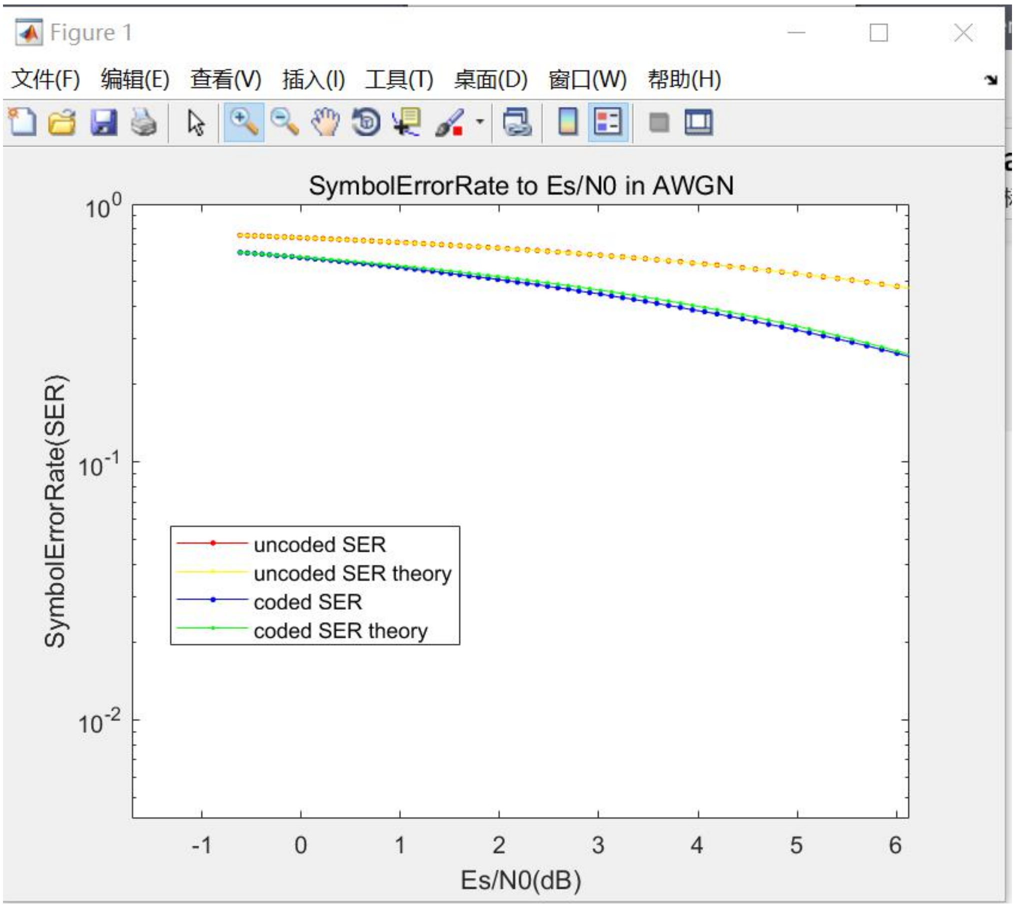
$$SER_{M-ASK} = \frac{2(M-1)}{M} Q(\sqrt{\frac{gain * d^2}{2N0}})$$

P(correct)=1-P(error), and the SER_theory can be derived by:

$$coded - SER - theory = 1 - (1 - SER_{M-ASK})^2$$

Then get the picture of SER to Es/N0(in dB):





It can be noticed that simulation is close to theory value, and there is no SER cross point.

5 Derivation and plot the BER curve

Since we use the Gray code, and most error happen in the closest symbol, and bit error in the most case is only one , so under this assumption, the approximate theory value can be given by:

$$P[bit\ error] = \frac{1}{\lambda}P[symbol\ error]$$

And the λ =log2(M), so the BER-MASK can be given by:

$$BER - theory = \frac{1}{\lambda}SER_{M-ASK} = \frac{1}{log_2 M}SER_{M-ASK}$$

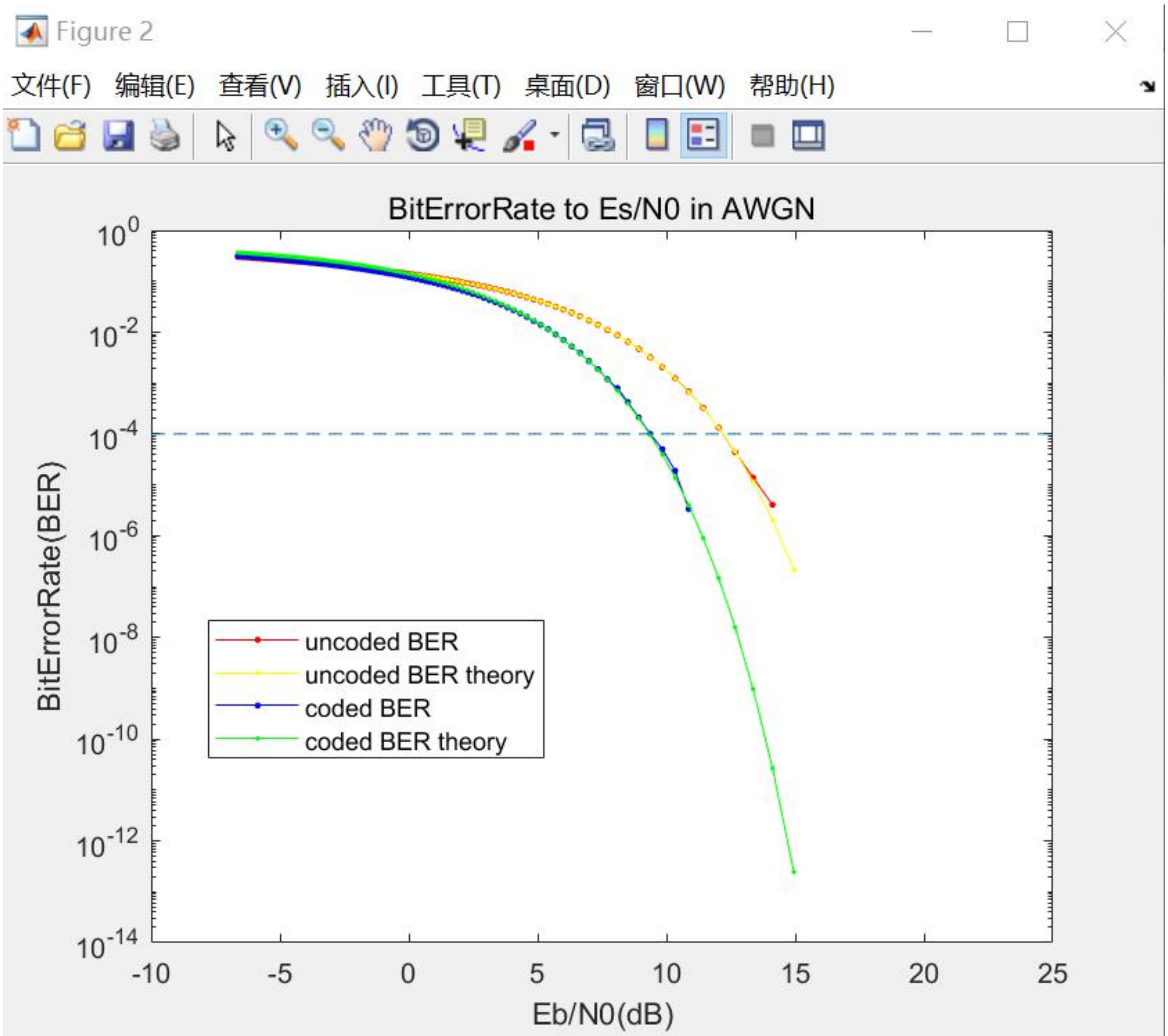
The P(correct)=1-P(error), So the coded_BER_theory can be given by:

$$coded - BER - theory = 1 - (1 - BER_{M-ASK})^2$$

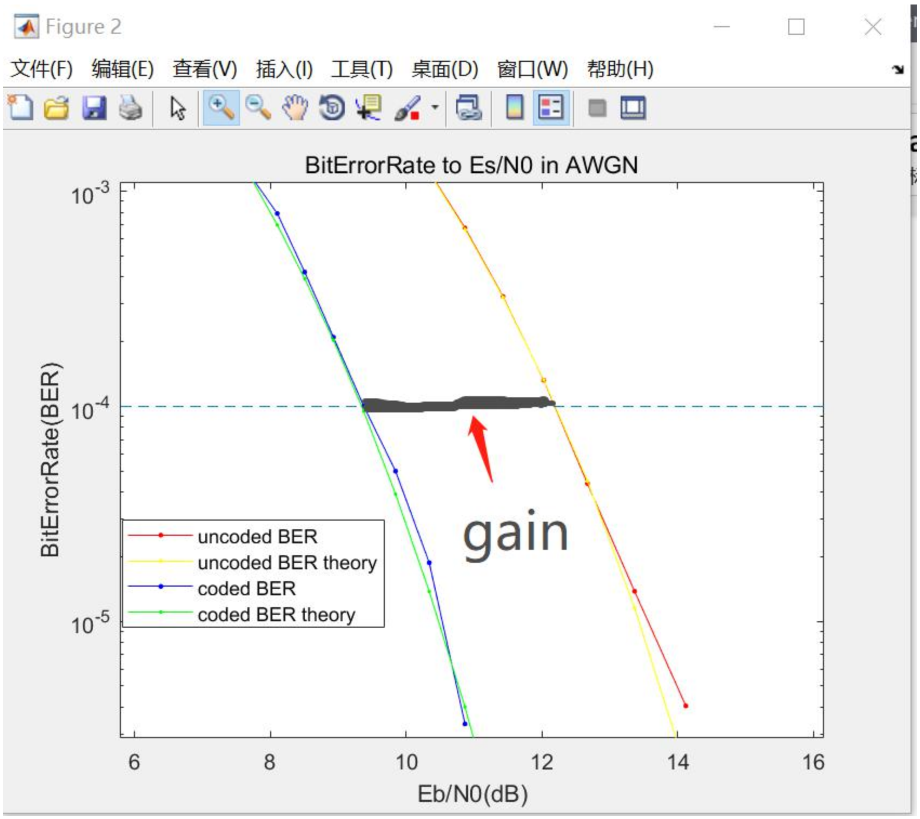
From another paper[1], I found a formula of P[bit error], which performs better to the result of simulation:

$$P_b \cong \frac{\sqrt{M}-1}{\sqrt{M}\log_2\sqrt{M}}\operatorname{erfc}\left(\sqrt{\frac{3\log_2 M\cdot\gamma}{2(M-1)}}\right) + \frac{\sqrt{M}-2}{\sqrt{M}\log_2\sqrt{M}}\operatorname{erfc}\left(3\sqrt{\frac{3\log_2 M\cdot\gamma}{2(M-1)}}\right)$$

Finally we get the picture of BER to Eb/N0(in dB):

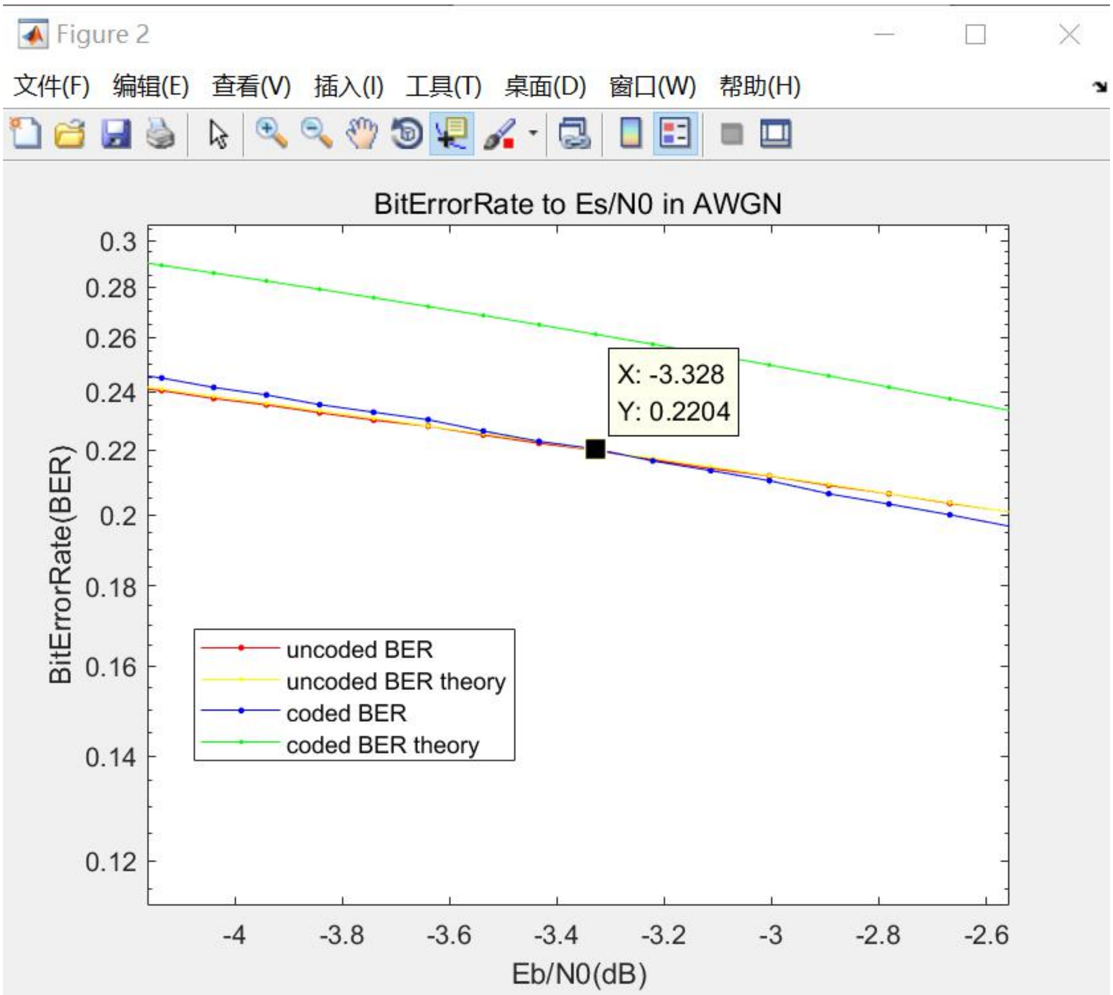


Amplify Pe=10^4 part:



We can calculate the code gain under Pe=10^-4 is 12.3-9.3 =3dB

Then we can amplify the cross part:



There is a cross point in BER cross point(red and blue lines), which Eb/N0=-3.328dB and BER=0.2204 in this simulation.

What is more, it can be found that coded BER value is not accurate in low Eb/N0(green and bule lines).

6 Conclusion

In this assignment, the mechanism of the hamming code is explored, and I am more familiar with the matrix form calculation method. What is more, it is a good practice for us to improve our ability of searching relevant literature.

Hamming code can effectively decrease the bit error rate in 16-QAM communication with not too high noise situation. However, the correction capability can lead to bad performance under too high noise(actually this kind channel is close to useless).

7 Reference:

1 Kyongkuk Cho and Dongweon Yoon, "On the general BER expression of one- and

two-dimensional amplitude modulations," in IEEE Transactions on Communications, vol. 50, no. 7, pp. 1074-1080, July 2002, doi: 10.1109/TCOMM.2002.800818.