

ELEC 5360 Homework 3

Hongzi Mao 2002 5238

1. (a)

The signal set is sketched in *Fig 1*, x and y axes are decision boundaries making the corresponding ML decision regions for receiving signal v .

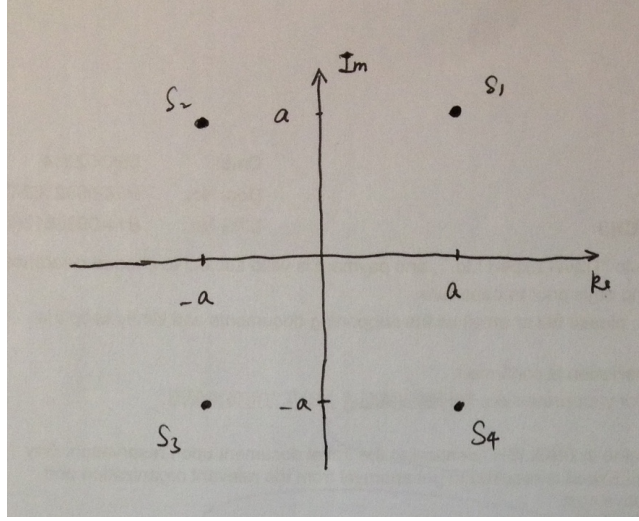


Fig 1.

Without loss of generality, we assume the transmitting signal is s_1 , the probability for decision error is calculated as follows:

$$\begin{aligned}
 P_e &= Pr(Re_Err \cup Im_Err) \\
 &= Pr(Re_Err) + Pr(Im_Err) - Pr(Re_Err \cap Im_Err) \\
 &= Q\left(\sqrt{\frac{d^2(s_1, s_2)}{2N_0}}\right) + Q\left(\sqrt{\frac{d^2(s_1, s_4)}{2N_0}}\right) - Q\left(\sqrt{\frac{d^2(s_1, s_2)}{2N_0}}\right) \times Q\left(\sqrt{\frac{d^2(s_1, s_4)}{2N_0}}\right) \\
 &= 2Q\left(\sqrt{\frac{2a^2}{N_0}}\right) - Q\left(\sqrt{\frac{2a^2}{N_0}}\right)^2
 \end{aligned}$$

Notice that $Pr(Re_Err \cap Im_Err) \neq Q\left(\sqrt{\frac{d^2(s_1, s_3)}{2N_0}}\right)$, because here we require **both** real and

imaginary part get screwed up.

(b)

The distance between $Re(u)$ and $Im(u)$ symbols on 2-PAM system is $2a$, therefore

$$P_e(Re(v)) = P_e(Im(v)) = Q\left(\sqrt{\frac{2a^2}{N_0}}\right)$$

(c)

The difference is that the error in part (a) is **symbol error** whereas in part (b) the error for a particular bit.

(d)

Derive the 4-QAM symbol error probability from coupling 2-PAM system:

$$\begin{aligned} P_e(4QAM) &= 1 - P_c(4QAM) \\ &= 1 - (P_c(2PAM) \times P_c(2PAM)) \\ &= 1 - ((1 - P_e(2PAM)) \times (1 - P_e(2PAM))) \\ &= 1 - ((1 - Q(\sqrt{\frac{2a^2}{N_0}})) \times (1 - Q(\sqrt{\frac{2a^2}{N_0}}))) \\ &= 2Q(\sqrt{\frac{2a^2}{N_0}}) - Q(\sqrt{\frac{2a^2}{N_0}})^2 \end{aligned}$$

2. (a)

The detection region separated by the boundaries in dashed line (and x, y axes) is shown in *Fig 2*, the optimal detector is associated with the detection region.

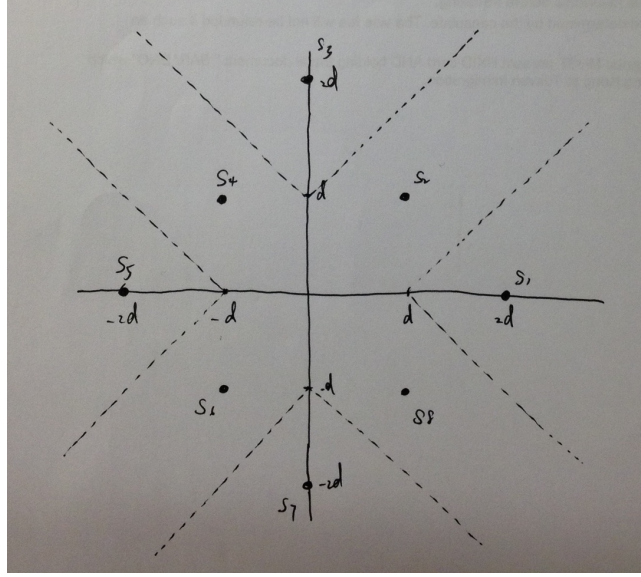


Fig 2.

(b)

The **pairwise** probability is calculated as bellows:

$$P_e(s_1|s_2) = Q\left(\sqrt{\frac{d^2(s_2, s_1)}{2N_0}}\right) = Q\left(\sqrt{\frac{d^2}{N_0}}\right)$$

And

$$P_e(s_6|s_2) = Q\left(\sqrt{\frac{d^2(s_2, s_6)}{2N_0}}\right) = Q\left(\sqrt{\frac{4d^2}{N_0}}\right)$$

(c)

The **union bound** for error probability is as follows

$$Q(error|s_2) \leq \sum_{i \in \{1,3,4,5,\dots,8\}} P(s_i|s_2)$$

Notice that

$$P(s_1|s_2) = P(s_3|s_2) = Q(\sqrt{\frac{d^2}{N_0}})$$

$$P(s_4|s_2) = P(s_8|s_2) = Q(\sqrt{\frac{2d^2}{N_0}})$$

$$P(s_5|s_2) = P(s_7|s_2) = Q(\sqrt{\frac{5d^2}{N_0}})$$

$$P(s_6|s_2) = Q(\sqrt{\frac{4d^2}{N_0}})$$

Hence

$$Q(error|s_2) \leq 2Q(\sqrt{\frac{d^2}{N_0}}) + 2Q(\sqrt{\frac{2d^2}{N_0}}) + 2Q(\sqrt{\frac{5d^2}{N_0}}) + Q(\sqrt{\frac{4d^2}{N_0}})$$

3.

The source code is available at

<https://github.com/Tony-Mao/DigCommPractice/tree/master/HW3>

A copy of the main code is also enclosed below.

(a) The theoretical calculation for symbol error is as follows

$$P_e(M - PAM) = 2(1 - \frac{1}{M})Q(\sqrt{\frac{6\log_2(M)}{M^2 - 1} \frac{E_b}{N_0}})$$

And

$$P_e(M - QAM) = 4(1 - \frac{1}{\sqrt{M}})Q(\sqrt{\frac{3\log_2(M)}{M - 1} \frac{E_b}{N_0}}) \times (1 - (1 - \frac{1}{\sqrt{M}})Q(\sqrt{\frac{3\log_2(M)}{M - 1} \frac{E_b}{N_0}}))$$

The simulation result for symbol error is shown as below, for each simulation point, 10^6 experiments are performed.

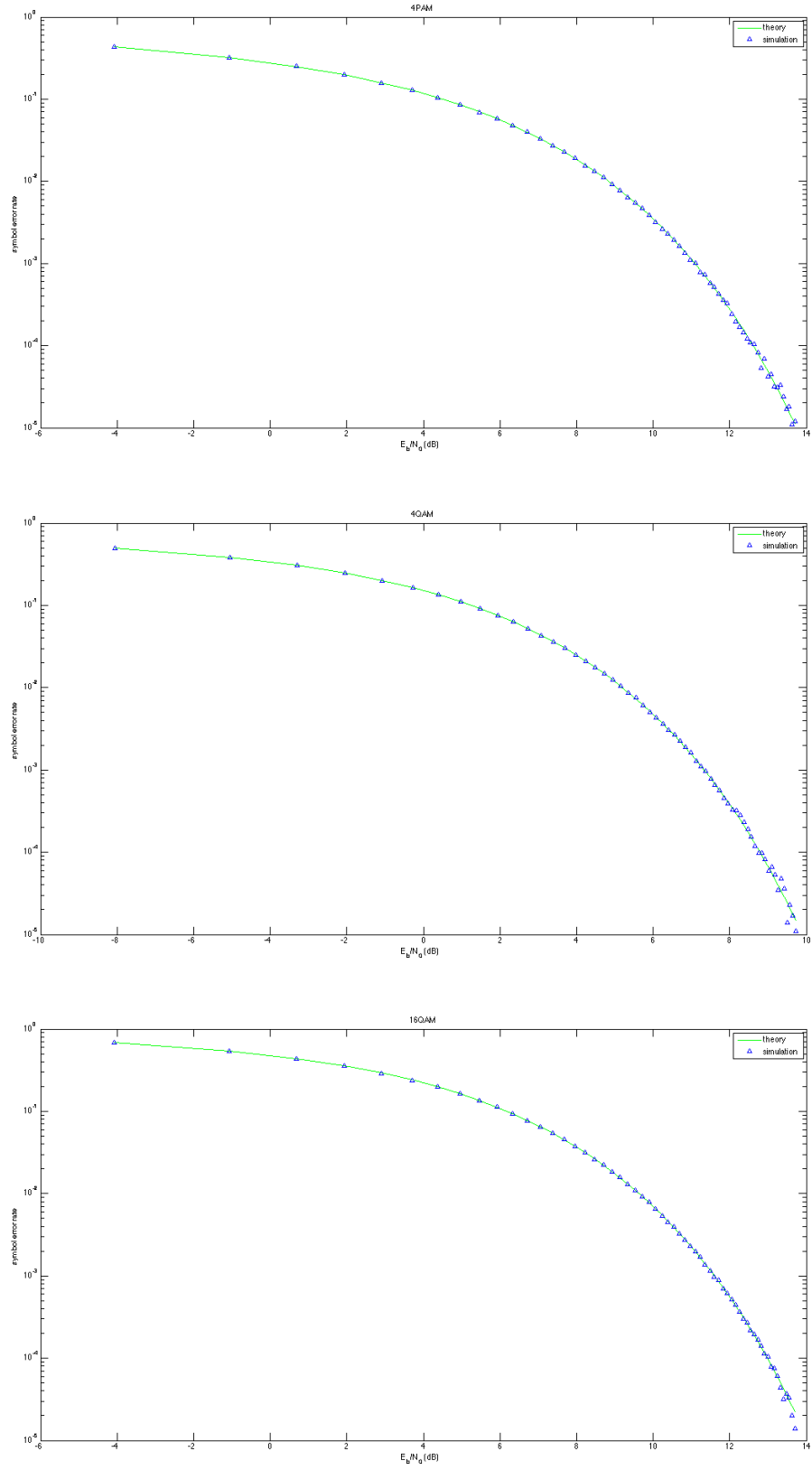
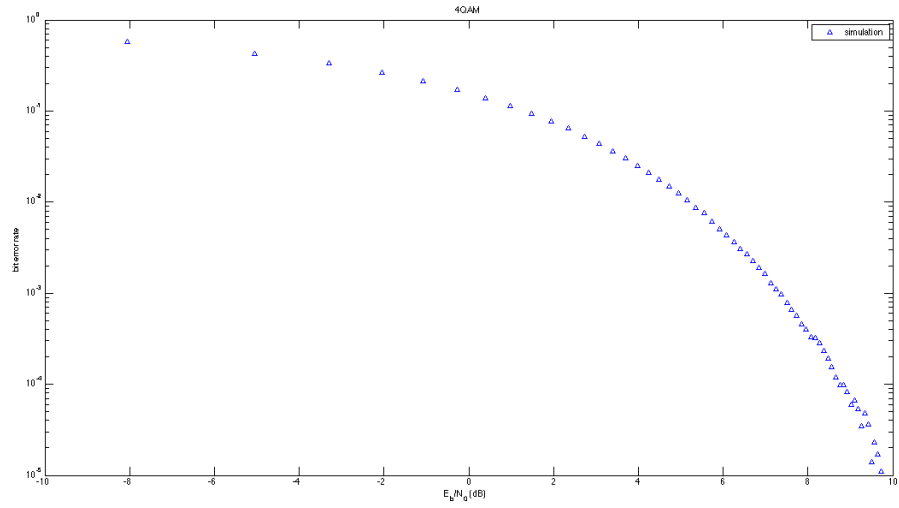
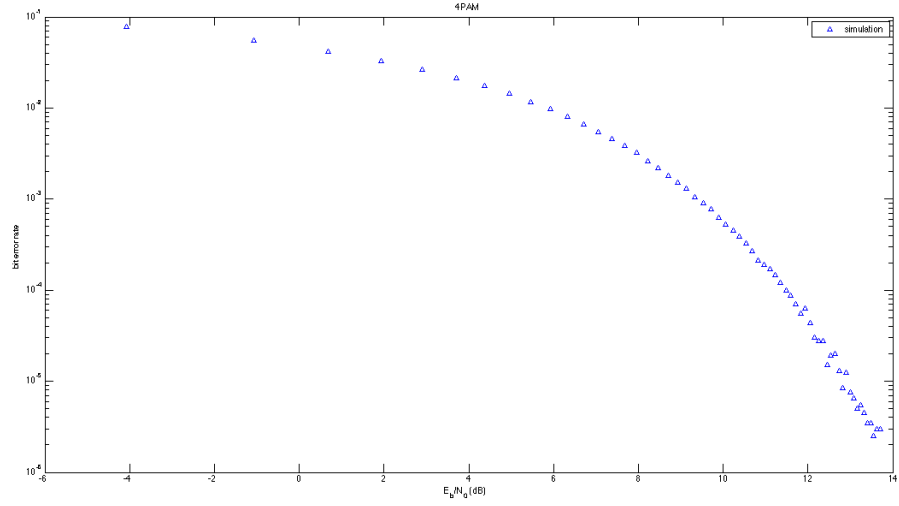


Fig 3.

(b) By definition, the bit error rate is number of bits being altered divided by total number of bits. Using signal constellation defined in the problem, the simulation result for bit error is shown as below , for each simulation point, 10^6 experiments are performed.



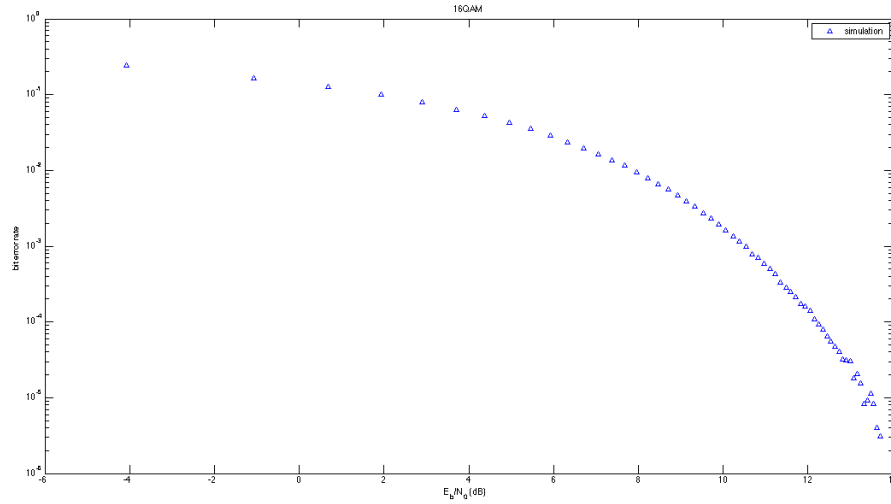


Fig 4.

Part of *16QAM* simulation code is listed below, for complete details please check the GitHub page. <https://github.com/Tony-Mao/DigCommPractice/tree/master/HW3>

```
function [ErrRate, bitErrRate] = sim16QAM(d, NO, iterNum, show)

errNum = 0;
bitErrNum = 0;
if show == true
    figure;
    hold on;
end

QAM_x = [d, 2*d, 3*d, 4*d];
QAM_y = [d, 2*d, 3*d, 4*d];

codeword_x = [[0 0],[0 1],[1 1],[1 0]];
codeword_y = [[1 0],[1 1],[0 1],[0 0]];

if show == true
    scatter(repmat(QAM_x,1,4), reshape(repmat(QAM_x',1,4)',[],16), 400, 'blacks');
end

for iter = 1:iterNum

symbol_x = 0; % 1,2,3,4 for testing symbol error rate, 0 for overall
symbol_y = 0;

    if symbol_x == 0
        tempArr = randperm(4);
        symbol_x = tempArr(1);
        tempArr = randperm(4);
        symbol_y = tempArr(1);
```

```

end

transmit_x = symbol_x * d;
transmit_y = symbol_y * d;

afterNoise_x = transmit_x + randn(1)*sqrt(N0/2);
afterNoise_y = transmit_y + randn(1)*sqrt(N0/2);

[~, minDis_x] = min(abs(QAM_x - afterNoise_x));
[~, minDis_y] = min(abs(QAM_y - afterNoise_y));

if symbol_x == minDis_x && symbol_y == minDis_y
    if show == true
        scatter(afterNoise_x, afterNoise_y, 'bo');
    end
else
    if show == true
        scatter(afterNoise_x, afterNoise_y, 'ro');
    end
    errNum = errNum + 1;
    bitErrNum = bitErrNum + sum(bitxor(codeword_x(symbol_x),codeword_x(
        minDis_x))) + sum(bitxor(codeword_y(symbol_y),codeword_y(minDis_y)));
end

end

ErrRate = errNum / iterNum;

bitErrRate = bitErrNum / iterNum / 4;

end

```