EECS 455 H9 YUZHAN JIANG

P1:

1. The (15,11) Hamming code has the following parity check matrix.

$$H = \underset{4\times 15}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}}.$$

11 12 13 14 15

The received vector is [000000000000011]. Determine the most likely transmitted codeword.

① First Find the Synrdome $S^T = Hr^T$

$$\underset{4\times 1}{\underset{4\times15}{H}\ \underset{15\times1}{r^T}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

② Find the corresponding coset leader for S

$e = [0000\ 0000\ 0010\ 000]$ it is with the ferrest error

③ Subtract the coset leader from the received vector to get transmitted codeword

$c = r - e$

$c = [0000\ 0000\ 0010\ 011]$

P2:

Error probability without coding for FSK

$$P = \frac{1}{2 + \frac{E_b}{N_0}}$$ where E_b is energy per bit

using Hamming code (7,4) with coding:

4 information bits = 7 code bits

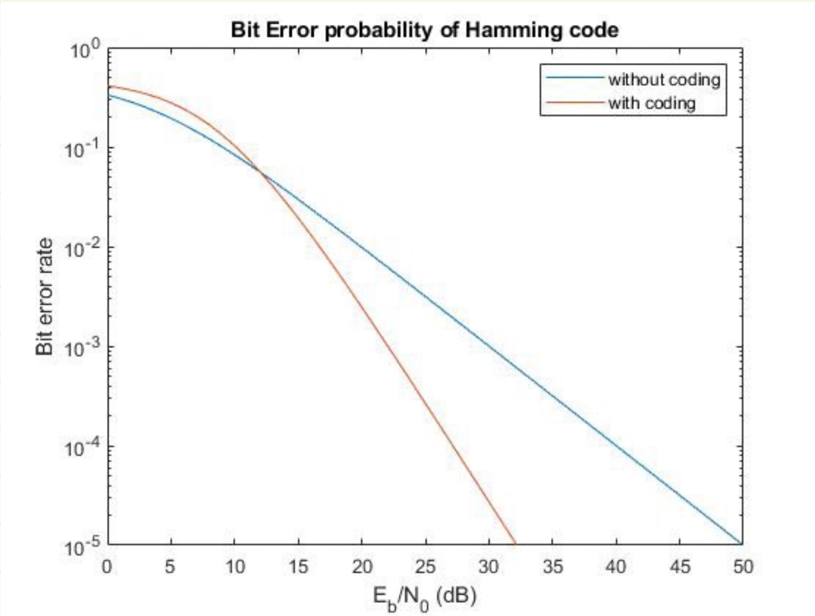
E_b = energy per info bit

E = energy per code bit

$$4E_b = 7E$$

$$E = \frac{4}{7}E_b$$

$$P = \frac{1}{2 + \frac{E_b}{N_0} \cdot \frac{4}{7}}$$




```matlab
clear all;

for m=0:50
%   EbN0dB(m)=-4+(m-1)/2
    EbN0dB(m+1) = m
    EbN0(m+1) =10^(EbN0dB(m+1)/10);
    p = 1 / (2 + EbN0(m+1))
    p_e_without_coding(m+1) = p

    p = 1 / (2 + EbN0(m+1) * 4/7)
    p_e_with_coding(m+1)  = 9 * p^2 * (1-p)^5 + 19 * p^3 * (1-p)^4 + 16 * p^4 * (1-p)^3 + 12 * p^5 *(1-p)^2 + 7*p^6 *(1-p) + p
end

semilogy(EbN0dB, p_e_without_coding);
hold on;

semilogy(EbN0dB, p_e_with_coding);
xlim([0 50])
ylim([10^(-5) 1])
xlabel('E_b/N_0 (dB)')
ylabel('Bit error rate')
legend("without coding", "with coding")
title('Bit Error probability of Hamming code')
```

EbN0	1x51 double
EbN0dB	1x51 double
m	50
p	1.7499e-05
p_e_with_coding	1x51 double
p_e_without_coding	1x51 double

There is $50 - 32 = 18$ dB reduction in energy compared to an uncoded system.

P3:

$m=16, \quad n=8$

(a) The optimum receiver finds and decides which signal is closest to the received signal.

(b) Decoding part:

① Compute the distance from received signals to S

② Find index among S with the minimum distance to received singals.

③ If index $\neq$ transmitted index, then # of symbols error $+1$

bit error $+4$

(The plot is in part (d))

3 (c) I use matlab to compute the pairwise distance (Euclidean)

0	2.0000	2.7229	2.7229	2.0000	2.8284	3.3785	3.3785	2.7229	2.1414	2.8284	2.2741	2.7229	2.1414	2.8284	2.2741
2.0000	0	2.7229	2.7229	2.8284	2.0000	3.3785	3.3785	2.1414	2.7229	2.2741	2.8284	2.1414	2.7229	2.2741	2.8284
2.7229	2.7229	0	2.0000	3.3785	3.3785	2.0000	2.8284	3.2907	2.8284	2.1414	2.7229	3.2907	2.8284	2.1414	2.7229
2.7229	2.7229	2.0000	0	3.3785	3.3785	2.8284	2.0000	2.8284	3.2907	2.7229	2.1414	2.8284	3.2907	2.7229	2.1414
2.0000	2.8284	3.3785	3.3785	0	2.0000	2.7229	2.7229	2.7229	2.1414	2.8284	2.2741	2.7229	2.1414	2.8284	2.2741
2.8284	2.0000	3.3785	3.3785	2.0000	0	2.7229	2.7229	2.1414	2.7229	2.2741	2.8284	2.1414	2.7229	2.2741	2.8284
3.3785	3.3785	2.0000	2.8284	2.7229	2.7229	0	2.0000	3.2907	2.8284	2.1414	2.7229	3.2907	2.8284	2.1414	2.7229
3.3785	3.3785	2.8284	2.0000	2.7229	2.7229	2.0000	0	2.8284	3.2907	2.7229	2.1414	2.8284	3.2907	2.7229	2.1414
2.7229	2.1414	3.2907	2.8284	2.7229	2.1414	3.2907	2.8284	0	2.0000	2.7229	2.7229	2.0000	2.8284	3.3785	3.3785
2.1414	2.7229	2.8284	3.2907	2.1414	2.7229	2.8284	3.2907	2.0000	0	2.7229	2.7229	2.8284	2.0000	3.3785	3.3785
2.8284	2.2741	2.1414	2.7229	2.8284	2.2741	2.1414	2.7229	2.7229	2.7229	0	2.0000	3.3785	3.3785	2.0000	2.8284
2.2741	2.8284	2.7229	2.1414	2.2741	2.8284	2.7229	2.1414	2.7229	2.7229	2.0000	0	3.3785	3.3785	2.8284	2.0000
2.7229	2.1414	3.2907	2.8284	2.7229	2.1414	3.2907	2.8284	2.0000	2.8284	3.3785	3.3785	0	2.0000	2.7229	2.7229
2.1414	2.7229	2.8284	3.2907	2.1414	2.7229	2.8284	3.2907	2.8284	2.0000	3.3785	3.3785	2.0000	0	2.7229	2.7229
2.8284	2.2741	2.1414	2.7229	2.8284	2.2741	2.1414	2.7229	3.3785	3.3785	2.0000	2.8284	2.7229	2.7229	0	2.0000
2.2741	2.8284	2.7229	2.1414	2.2741	2.8284	2.7229	2.1414	3.3785	3.3785	2.8284	2.0000	2.7229	2.7229	2.0000	0

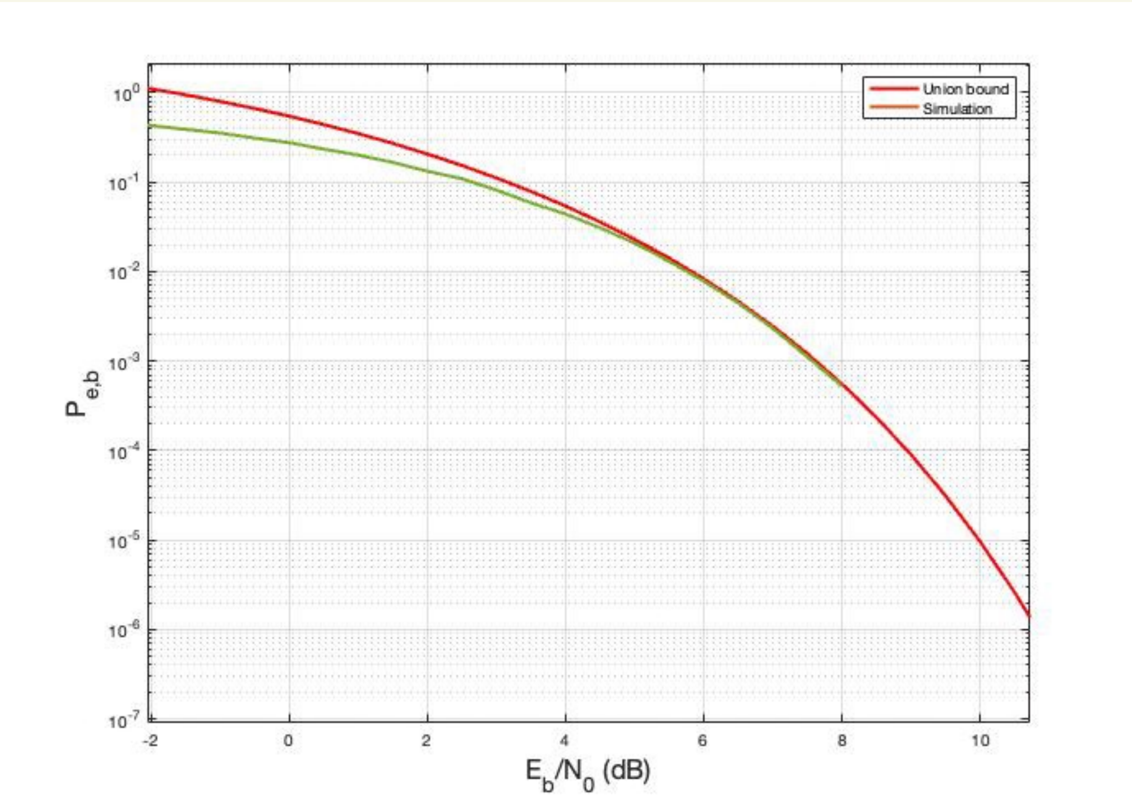
code:

```
clear all;
E=1
Eb=E;                  % Relation between energy per code symbol and bit
s(1,:)  =  sqrt(E)*[ exp(j*2*pi*0/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8) ];
s(2,:)  =  sqrt(E)*[ exp(j*2*pi*0/8), exp(j*2*pi*4/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8) ];
s(3,:)  =  sqrt(E)*[ exp(j*2*pi*0/8), exp(j*2*pi*2/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8) ];
s(4,:)  =  sqrt(E)*[ exp(j*2*pi*0/8), exp(j*2*pi*6/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8) ];
s(5,:)  =  sqrt(E)*[ exp(j*2*pi*4/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8) ];
s(6,:)  =  sqrt(E)*[ exp(j*2*pi*4/8), exp(j*2*pi*4/8), exp(j*2*pi*0/8), exp(j*2*pi*0/8) ];
s(7,:)  =  sqrt(E)*[ exp(j*2*pi*4/8), exp(j*2*pi*2/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8) ];
s(8,:)  =  sqrt(E)*[ exp(j*2*pi*4/8), exp(j*2*pi*6/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8) ];
s(9,:)  =  sqrt(E)*[ exp(j*2*pi*2/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8), exp(j*2*pi*0/8) ];
s(10,:)  =  sqrt(E)*[ exp(j*2*pi*2/8), exp(j*2*pi*1/8), exp(j*2*pi*2/8), exp(j*2*pi*0/8) ];
s(11,:)  =  sqrt(E)*[ exp(j*2*pi*2/8), exp(j*2*pi*3/8), exp(j*2*pi*7/8), exp(j*2*pi*2/8) ];
s(12,:)  =  sqrt(E)*[ exp(j*2*pi*2/8), exp(j*2*pi*7/8), exp(j*2*pi*7/8), exp(j*2*pi*2/8) ];
s(13,:)  =  sqrt(E)*[ exp(j*2*pi*6/8), exp(j*2*pi*5/8), exp(j*2*pi*2/8), exp(j*2*pi*0/8) ];
s(14,:)  =  sqrt(E)*[ exp(j*2*pi*6/8), exp(j*2*pi*1/8), exp(j*2*pi*2/8), exp(j*2*pi*0/8) ];
s(15,:)  =  sqrt(E)*[ exp(j*2*pi*6/8), exp(j*2*pi*3/8), exp(j*2*pi*7/8), exp(j*2*pi*2/8) ];
s(16,:)  =  sqrt(E)*[ exp(j*2*pi*6/8), exp(j*2*pi*7/8), exp(j*2*pi*7/8), exp(j*2*pi*2/8) ];

for j=  1:16
    for i = 1:16
        a(j,i) = norm(s(j,:) - s(i,:))
    end
end
```

P3: (d)

Union Bound (for $m = 25$ dB)

```
>> pes_ub

pes_ub =

  Columns 1 through 16

    1.8288    1.6359    1.4494    1.2708    1.1017    0.9434    0.7972    0.6641    0.5447    0.4394    0.3481    0.2705    0.2058    0.1531    0.1112    0.0786

  Columns 17 through 32

    0.0539    0.0358    0.0230    0.0141    0.0083    0.0046    0.0024    0.0012    0.0006    0.0002    0.0001    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 33 through 35

    0.0000    0.0000    0.0000
```



We can see that union bound is above the simulation which is exactly what we expect

```
%======================================================================%
%                                                                      %
%   YOUR CODE FOR UNION BOUND GOES HERE                                %
%                                                                      %
%======================================================================%
    pes_ub(m)=0
    for i=1:16
        pe(i)=0;
        for l=1:16
            d(i,l)=sqrt(sum((abs(s(i,:)-s(l,:))).^2));
            if (l ~= i)
                pe(i)=pe(i)+qfunc(d(i,l)/(2*sigma));
            end
        end
        pes_ub(m)=pes_ub(m)+pe(i)/16;
    end
%   pes_ub(m)= .... this is your result for the union bound

end
semilogy(EbN0dB,pes_ub,'r','LineWidth',2)    % PLOT THE RESULTS
hold on
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                       %
%           Decode the received signal              %
%                                                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r = [ rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd;rcvd];
vec_distance = vecnorm(r - s, 2, 2);
[min_distance, dec_index] = min(vec_distance);
%
% min_distance = 999;
% dec_index = 0;
% for i = 1:16
%
%      distance = norm(s(i,:) - rcvd )
%      distance_vec(i) = distance
%     if(distance < min_distance)
%         min_distance = distance
%         dec_index = i
%     end
% end

if(dec_index ~= index)
    nserrors = nserrors +1;
end
nbsim = nbsim + 4;
nsim = nsim + 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function Total_Q = pij(si, S,Sigma)
    Total_Q = 0
    for i =  1:16
        if i == si
            continue;
        else
            distance = norm(S(i,:) - S(si,:))
            disp("distance in " + si +" "+ i +"   : "+ distance)
            y = qfunc(distance / (2* Sigma));
            Total_Q =  Total_Q + y;
            disp("P2 for "+ si + " " + i +" is: "  + y)
        end
    end
     disp("Pei for " + i + " is :" + Total_Q)
end
```

P4:

$M=4 \quad n=6$

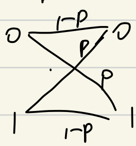
000 000

000 111

111 000

111 111

BSC



(a) The minium Hamming distance $d_{H,min}$ is 3

The minium squared Euclidean distance is $4 \times 3 = 12$

(b) Since the minimum distance of this code 3,
So the optimum decoder can always correct $\frac{3-1}{2} = 1$ errors

(c) The number of vectors in the decoding region of a codeword is

$$\sum_{k=0}^{1} \binom{6}{k} = \binom{6}{0} + \binom{6}{1}$$
$$= 1+6$$
$$= 7$$

(d) Based on c, the total number of vectors in all the decoding regions is

$$4 \times 7 = 28$$

There are total $2^6 = 64$ vectors

$$\frac{28}{64} = \frac{7}{16}$$

$\therefore$ There are $\frac{7}{16}$ of total received vectors are in the decoding region

(e) consider 101 101 that is not in the decoding region of any codeword.

The optimum decoder would choose 111 111 codeword for this received vector

(f)

000 000 :

000 111

111 000

111 111

For 000000: If there are two bits errors, it could be (110 000), (011000)
(101 000), (000 110) (000 101) (000 011)

If there are three bits errors, it could be (111 000), (110100) ~
$\binom{6}{3} p^3 (1-p)^3$

Similarly for 4, 5, 6 bits errors occur,

For 000000, the $P_{e,c} = 6 p^2 (1-p)^4 + \binom{6}{3} p^3 (1-p)^3 + \binom{6}{4} p^4 (1-p)^2 + \binom{6}{5} p^5 (1-p) + p^6$

For 000111: ① Two bits errors, it could be (110 111), (011 111), (101 111),
(000 001), (000 010), (000 100)

For 000111, the $P_{e,c} = 6 p^2 (1-p)^4 + \binom{6}{3} p^3 (1-p)^3 + \binom{6}{4} p^4 (1-p)^2 + \binom{6}{5} p^5 (1-p) + p^6$

Similarly for codeword 111000, 111111

Overall, the $P_e = 24 p^2 (1-p)^4 + 80 p^3 (1-p)^3 + 60 p^4 (1-p)^2 + 24 p^5 (1-p) + 4 p^6$

(g) Given the codeword 000 000 was transmitted

The received vectors in the decoding region of the codeword (000111) are

Zero bits error: (000111)

One bits error: (100 111), (010 111), (001 111), (000 011), (000 101) (000 110)

~~Two bits error: (100 011), (100 101) (100 011), (010 011) (010 101), (010 110),~~
~~(001 011), (001 101), (001 110)~~

So given 000000, the probability is:

$$p^3(1-p)^3 + 3p^4(1-p)^2 + 3p^2(1-p)^4$$

(h) ① Probability of correct (Bounded Distance decode)

$000000 \Rightarrow 100000 \quad 010000 \quad 001000 \quad 000100 \quad 000010 \quad 000001 \quad 6p(1-p)^5 + (1-p)^6$

$p(1-p)^5 \quad 000111 \Rightarrow 100111 \quad 010111 \quad 001111 \quad 000011 \quad 000101 \quad 000110 \quad 6p(1-p)^5 + (1-p)^6$

$111000 \Rightarrow 011000 \quad 101000 \quad 110000 \quad 111100 \quad 111010 \quad 111001 \quad 6p(1-p)^5 + (1-p)^6$

$111111 \Rightarrow 011111 \quad 101111 \quad 011111 \quad 111011 \quad 111101 \quad 111110 \quad 6p(1-p)^5 + (1-p)^6$

$\therefore P_{correct} = 24p(1-p)^5 + 4(1-p)^6$

② Probability of decoding error:

$000000 \Rightarrow 000111$ decoding region: $P_{e,c} = p^3(1-p)^3 + 3p^4(1-p)^2 + 3p^2(1-p)^4$

$000000 \Rightarrow 111000$: $P_{e,c} = p^3(1-p)^3 + 3p^4(1-p)^2 + 3p^2(1-p)^4$

$000000 \Rightarrow 111111$: $P_{e,c} = 6p^5(1-p) + p^6$

$000111 \Rightarrow 000000$ decoding region: $= p^3(1-p)^3 + 3p^4(1-p)^2 + 3p^2(1-p)^4$

$000111 \Rightarrow 111000$ $P_{e,c} = p^6 + 3p^5(1-p) + 3p^5(1-p)$

$000111 \Rightarrow 111111$ $P_{e,c} = p^3(1-p^3) + 3p^2(1-p)^4 + 3p^4(1-p)^2$

Symmetricly:

$P_{e,000000} = p^6 + 2p^3(1-p)^3 + 6p^4(1-p)^2 + 6p^2(1-p)^4 + 6p^5(1-p)$

$P_{e,000111} = p^6 + 6p^2(1-p)^4 + 2p^3(1-p)^3 + 6p^4(1-p)^2 + 6p^5(1-p)$

Overall, $P_e = 4p^6 + 24p^2(1-p)^4 + 8p^3(1-p)^3 + 24p^4(1-p)^2 + 24p^5(1-p)$

③ Probability of decoding failure:

$P_{failure} = 4 - P_{correct} - P_e$

④ Probability of error (optimum Decoding)

By part (f)

$$P_e = 24p^2(1-p)^4 + 80p^3(1-p)^3 + 60p^4(1-p)^2 + 24p^5(1-p) + 4p^6$$

⑤ Union Bound

$$P_{e,i} \le \sum_{d=d_{min}}^{n} A_d P_2(d)$$

For 000000, 111111

d	Ad
3	2
6	1

$$P_{e,i} \le \sum_{d=3}^{7} A_d P_2(d)$$

$$\le 2 \cdot P_2(3) + P_2(6)$$

$$P_2(d) \begin{cases} \sum_{i=2}^{3} \binom{3}{i} \cdot p^i (1-p)^{3-i} & d=3 \quad = 3p^2(1-p)^4 + p^3(1-p)^3 \\ \sum_{i=3}^{6} \binom{6}{i} p^i (1-p)^{6-i} + \frac{1}{2}\binom{6}{3} p^3 (1-p)^3 & d=6 \quad = 20p^3(1-p)^3 + 15p^4(1-p)^2 + 6p^5(1-p) + p^6 + 10p^3(1-p)^3 \end{cases}$$

For 000111, 111000

it is same above

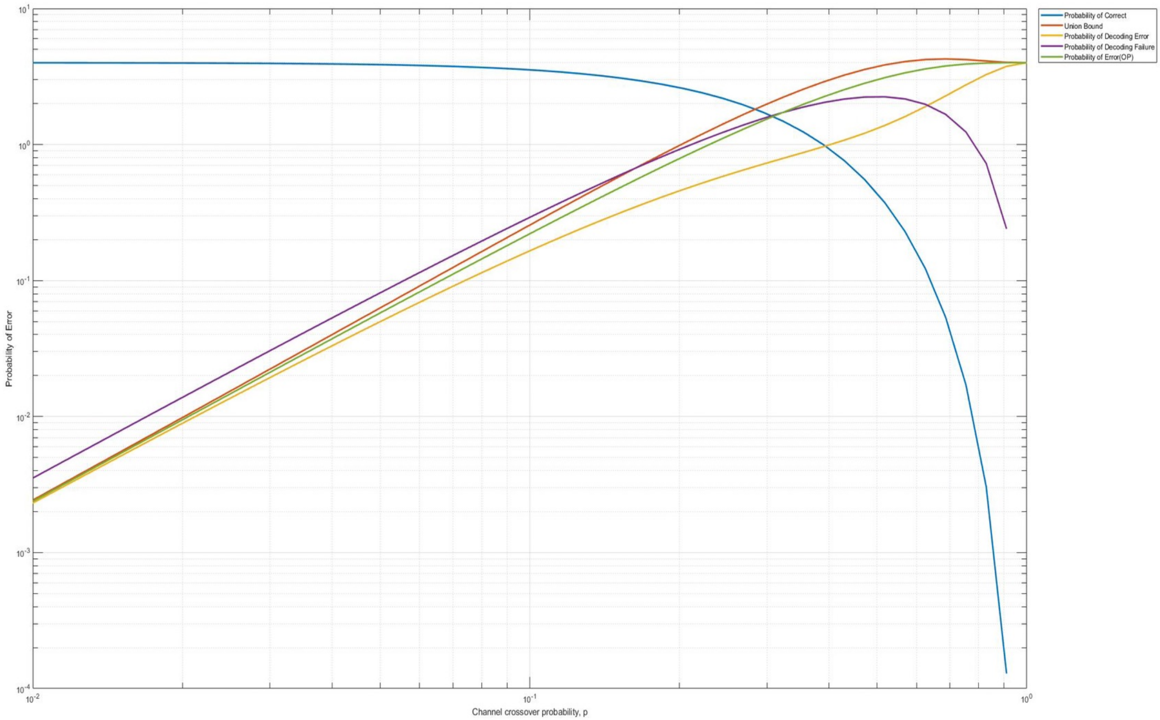
d	Ad
3	2
6	1

$$P_e = 4 \; P_{e,i}$$

$$= 4 \cdot \left[2\left(3p^2(1-p)^4 + p^3(1-p)^3\right) + 30p^3(1-p)^3 + 15p^4(1-p)^2 + 6p^5(1-p) + p^6 \right]$$

$$= 4\left[6p^2(1-p)^4 + 32 + 15 + 6 + 1 \right]$$

$$= 24p^2(1-p)^4 + 128p^3(1-p)^3 + 60p^4(1-p)^2 + 24p^5(1-p) + 4p^6$$



HW9P3.m | problem4.m

```
clear all;
p = logspace(-2,0)

a = p.*(1-p).^5
b = p.^2 .*(1-p).^4
c = p.^3 .* (1-p).^3
d = p.^4 .* (1-p).^2
e = p.^5 .* (1-p)
f = p.^6
g =(1-p).^6

p_correct = 24 *a + 4 *g
p_decode_error = 4 *f + 24 * b+ 8*c + 24 *d +24 * e
p_ub = 24 *b  + 128 * c + 60 *d + 24 * e + 4*f;
p_decode_f = 4 - p_correct - p_decode_error
p_optimum_error = 24 *b + 80 *c + 60 * d + 24* e +4 *f

loglog(p, p_correct,LineWidth  = 2)
grid on
hold on
loglog(p, p_ub,LineWidth  = 2)
loglog(p, p_decode_error, LineWidth  = 2)
loglog(p, p_decode_f,LineWidth  = 2)
loglog(p, p_optimum_error,LineWidth  = 2)
 legend('Location','northeastoutside')
legend('Probability of Correct','Union Bound','Probability of Decoding Error','Probability of Decoding Failure','Probability of Error(OP)')
ylabel('Probability of Error')
xlabel('Channel crossover probability, p')
```