

# Communication Engineering Term Project

– QAM(Quadrature Amplitude Modulation) with matlab -

Student Name: Jeongbong Han [20105097]

**Abstract—** In this Term project, you can materialize the communication system step by step using Matlab program and basic knowledge of communication engineering. In the process you can learn about QAM mapping, modulation, channel properties, demodulation and returning from QAM values to quantized values and original bit sequence. Also you can accumulate a store of basic knowledge of MATLAB programming.

## **1. Introduction.**

### **- Communication system**

When we communicate with signals we use transmitter, channel and receiver to send and receive the signals. In this term project, we can learn more about the details of transmitter, channel and receiver especially for modulation and demodulation technic.

### **-QAM(Quadrature Amplitude Modulation)**

Quadrature amplitude modulation (QAM) is both an analog and a digital modulation scheme. It conveys two analog message signals, or two digital bit streams, by changing (modulating) the amplitudes of two carrier waves. The two carrier waves, usually sinusoids, are out of phase with each other by  $90^\circ$  and are thus called quadrature carriers or quadrature components. The modulated waves are summed, and the final waveform is a combination of both phase-shift keying (PSK) and amplitude-shift keying (ASK), or (in the analog case) of phase modulation (PM) and amplitude modulation.

### **-Transmitter**

When we get the signal for sending with random bits. Then map it to the binary symbol and send QAM values. To avoid ISI, we could put the cyclic prefix that paste the last part to the head of the signals in time domain.

### **-Channel**

We can compute the channel using convolution in time domain or multiplication in frequency domain. We should make the carrier fit to the channel by handling the carrier frequency. So the signals can pass through the channel minimizing the signal errors. In this project we are using rectangular tx rx matched filter.

### **-Modulation and Demodulation**

\*Modulate the QAM signals passed by rectangular filter with carriers like cosine and sine that are orthogonal to each other. (quadrature) Now it is ready for the transmission.

\*Demodulate the arrived signal to the receiver by using same carriers (cosine and sine). This mean we are multiplying signals to cosine and sine each.

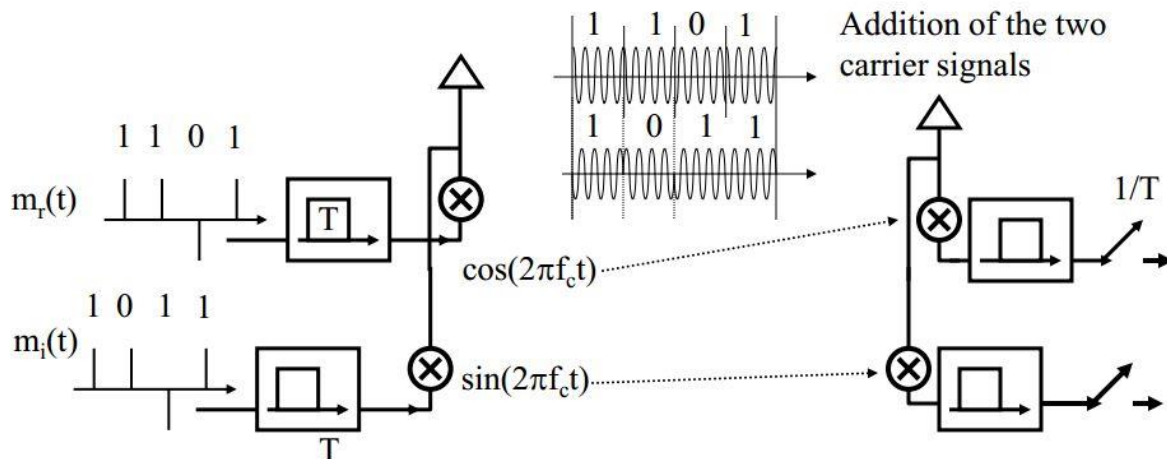
### -Receiver

Now the signals arrives the receiver. Then demodulate the received signal by using same carriers. Then use same matched filter to pass the signal (convolution). During the convolution, due to the orthogonal condition of cosine and sine, we can separate the two input signals.  $\int \cos(2\pi f_c t) * \sin(2\pi f_c t) = 0$  . Then we sample the signal by

### -AWGN(Additive White Gaussian Noise)

Basic noise model used in Information theory to mimic the effect of many random processes that occur in nature. The modifiers denote specific characteristics. White means same strength throughout every frequency and the Gaussian means it is Gaussian distributed random variable with the mean is 0 and variance is  $\frac{N_0}{2}$ . The power spectral density of AWGN noise is  $\frac{N_0}{2}$ .

## 2. System Description In MATLAB



**Transmitter -  $S(t) = S_1(t) + S_2(t)$**

1. Generate the two sets of random bits sequence  $\langle m_{r1}(t) \text{ and } m_{r2}(t) \rangle$  (First do it for smaller example to see if it works)
2. Convert the bits signals to symbols( 0 1 to -1 1)
3. Make signals to Pass the Rectangular filter (Tx)

4. Modulate each signals by multiplying the cosine and sine pulse-  $\cos(2\pi f_c t), \sin(2\pi f_c t)$  with same frequency  $f_c$ . Then add two carrier signals  $S_1(t)$  and  $S_2(t)$  to transmit.

### AWGN - W(t)

5. Add white Gaussian noise by select the suitable SNR dB.

$$-\text{SNR} = \frac{P}{N} = \frac{E_b * R}{N_0 * w} \quad , \quad E_{Tx} = \int_0^{10^{-3}} A^2 = \sqrt{2}, A(\text{amplitude of rec}) = 2^{1/4} * 10^{3/2}$$

$$-E_b = \int_0^{10^{-3}} A^2 \cos^2(2\pi f_c t) + A^2 \sin^2(2\pi f_c t) = \sqrt{2}$$

$$-R = 2\text{bits} / 10^{-3}\text{sec} \quad ,$$

$$-W = 0.5 \times 10^3 \text{Hz} \quad (\text{rec} * \text{rec} = \text{tri} - \text{twice the channel bandwidth})$$

$$-\text{SNR} = \frac{\sqrt{2} \text{energy/bit} \times 2\text{bits} / 10^{-3}\text{sec}}{0.5 \times 10^{-3}\text{sec}} \times \frac{1}{N_0} = 4\sqrt{2} \times \frac{1}{N_0}$$

$$\text{Where } \frac{1}{N_0} \text{ is dB scale. So SNR} = 4\sqrt{2} \times 10^{\frac{1}{N_0} \times \frac{1}{10}} \quad (\text{because } 10 \log_{10} 10^{\frac{1}{N_0} \times \frac{1}{10}} = \frac{1}{N_0})$$

### Receiver- R(t)=S(t)+W(t)

6. Demodulate the received signal by multiplying the cosine and sine pulse and pass the matched filter (rectangular filter). In this process due to the matched filter response we are doing convolution. Because of the property of orthogonal property below.

$$\int \cos(2\pi f_c t) * \sin(2\pi f_c t) = 0$$

We can separate the added two QAM signals.

7. Sampling two signals with interval T (=1m sec=number of samples  $\times$  number of function) (passed the matched filter and demodulated).
8. Compare the sampled receiving sequence and the random generated sequence.

### Find P(e) in Simulation

9. Determine the probability of making bit errors in simulation

$P(e)$  = the number of bit decision errors / the total number of bits = error rate

The number of bit decision errors =  $\sum (\text{Random bit array} - \text{received array})^2$

10. Draw the curve  $P(e)$  vs  $\frac{1}{N_0}$  for  $\frac{1}{N_0} = 1\text{dB} \sim 7\text{dB}$

In MATLAB, we are inserting SNR dB value to AWGN function. So we should represent the SNR as a function of  $\frac{1}{N_0}$  with dB scale.

$$\text{SNR} = 4\sqrt{2} \times 10^{\frac{1}{N_0} \times \frac{1}{10}} \quad (\text{I calculated step 5})$$

$$\text{SNR\_dB} = 10 \log_{10} 4\sqrt{2} \times 10^{\frac{1}{N_0} \times \frac{1}{10}}$$

11. Using 'for' to plot the P(e) vs  $\frac{1}{N_0}$

I found that, P(e) is always 0 during the  $\frac{1}{N_0}$ 's domain 1dB to 7dB

So I check the P(e) against the  $\frac{1}{N_0}$  for the domain (-100dB to 100dB)

### 3. Process and Results with MATLAB

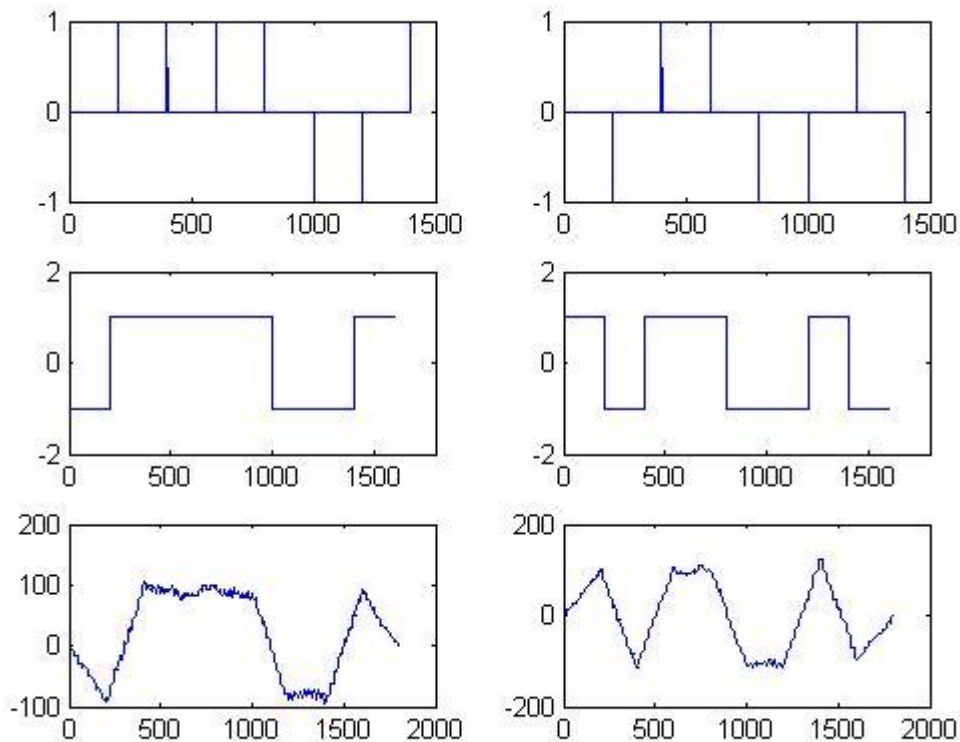
1) Check the system with the small bit example. (Do it for 3bit)

A.  $m_{r1} = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$   $m_{r2} = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$

B. (1,1) =  $m_{r1}(t)$  converted to symbol (1,2) =  $m_{r2}(t)$  converted to symbol

(2,1) = Rec filter response of  $m_{r1}(t)$  (2,2) = Rec filter response of  $m_{r2}(t)$

(3,1) = Receive signal  $R_1$  before sampling (3,2) = Receive signal  $R_2$  before sampling

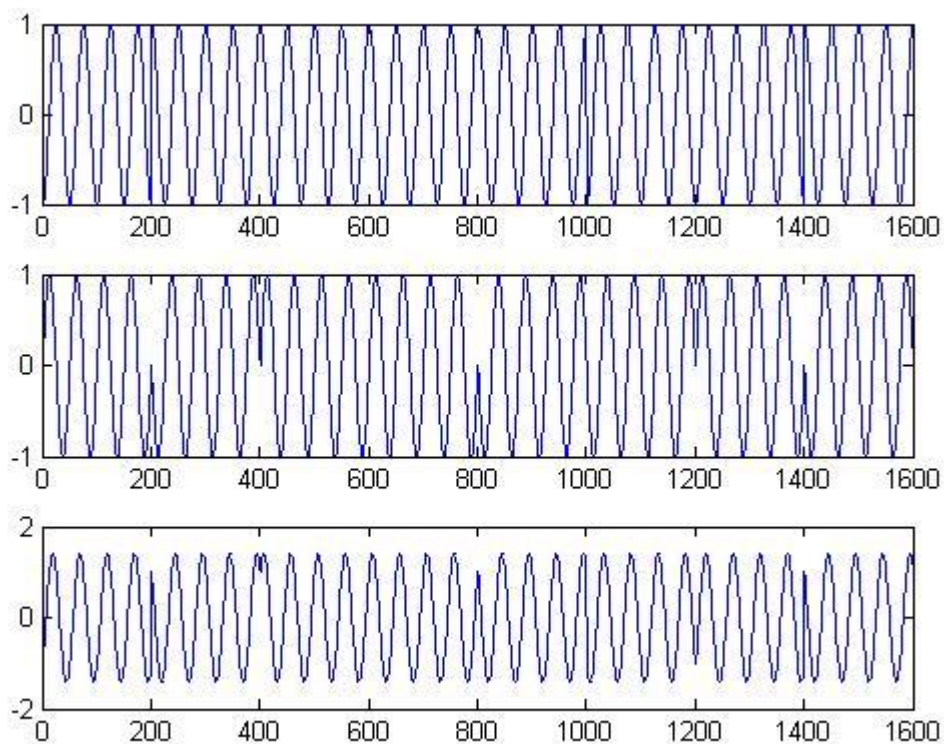


### C. Modulation results

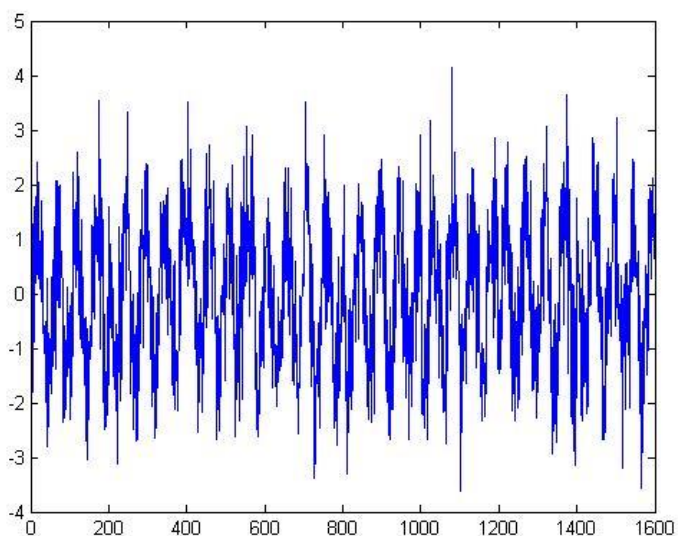
$$(1,1) S_1(t) = m_{r1}(t) * \cos(2\pi f_c t)$$

$$(1,2) S_2(t) = m_{r2}(t) * \sin(2\pi f_c t)$$

$$(1,3) S(t) = S_1(t) * \cos(2\pi f_c t) + S_2(t) * \sin(2\pi f_c t)$$



### D. AWGN – Noise to $S(t) < S(t) + W(t) >$ (I used the SNR =1)



E. Overall result Comparing the input and out put response

Random bit signal  $m_{r1} = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$   $m_{r2} = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$

Sampled received signal  $R_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$   $R_2 = [1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$

Error1 =0, Error2 =0

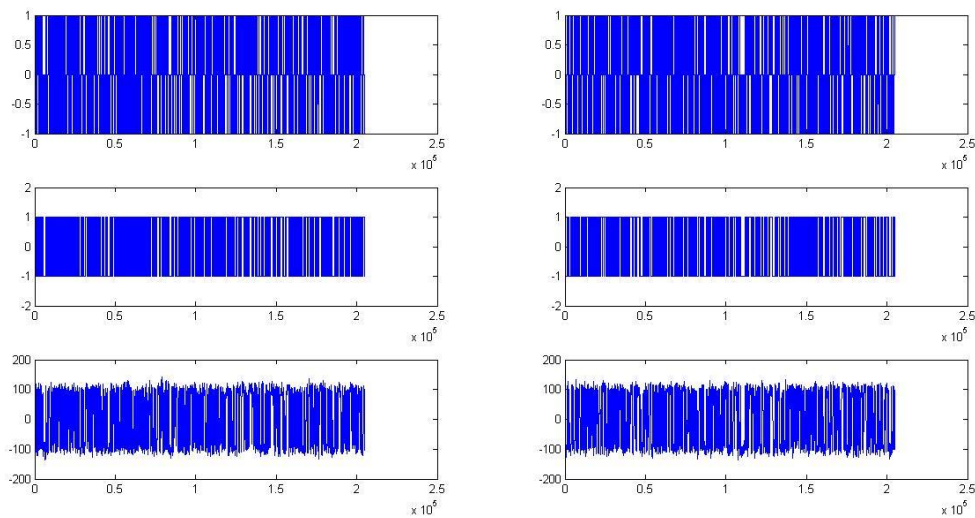
Error\_rate1 = 0, Error\_rate2 =0 .

**I checked that the overall MATLAB coding system works!!**

**2) Do the Same process for 10bit system.**

**I'm attaching the MATLAB file because it is hard to see the picture.**

- A. (1,1) =  $m_{r1}(t)$  converted to symbol (1,2)=  $m_{r2}(t)$  converted to symbol  
(2,1) =Rec filter response of  $m_{r1}(t)$  (2,2) =Rec filter response of  $m_{r2}(t)$   
(3,1) =Receive signal R1 before sampling (3,2) =Receive signal R2 before sampling

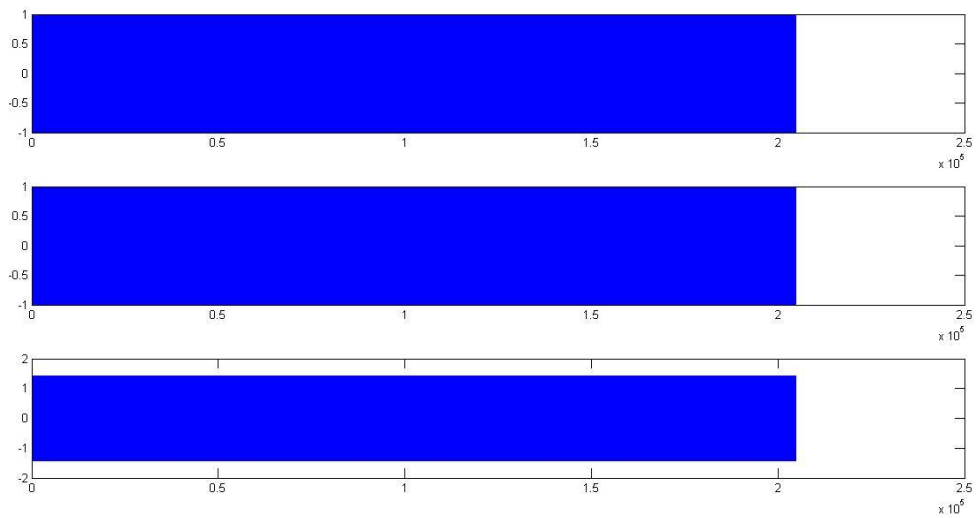


B. Modulation results

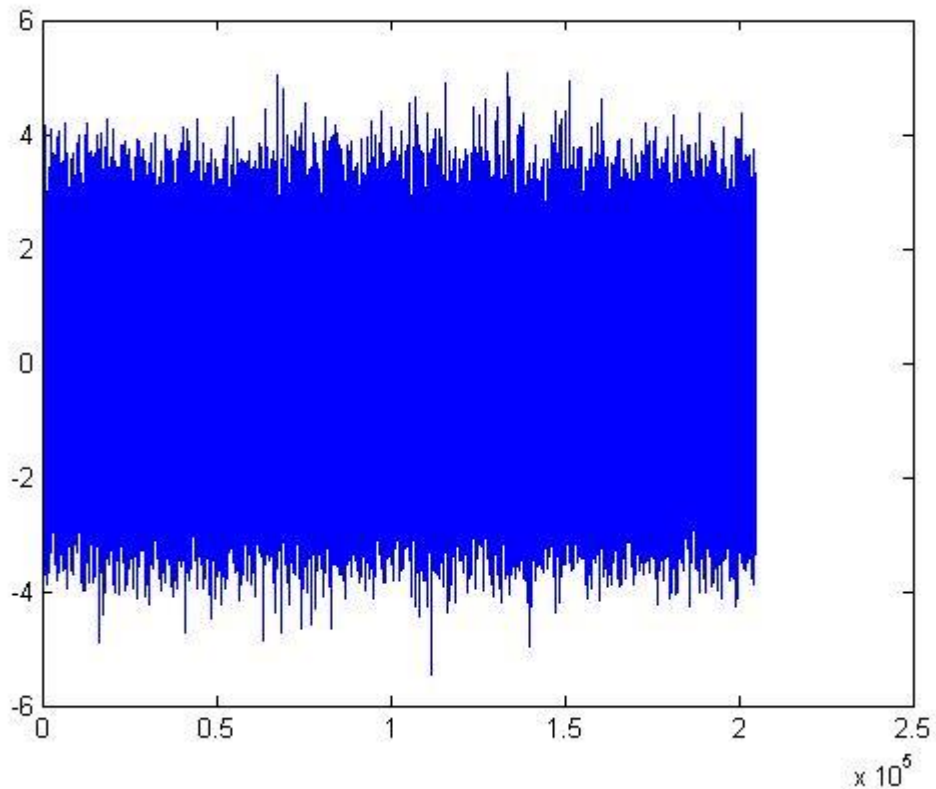
$$(1,1) S_1(t) = m_{r1}(t) * \cos(2\pi f_c t)$$

$$(1,2) S_2(t) = m_{r2}(t) * \sin(2\pi f_c t)$$

$$(1,3) S(t) = S_1(t) * \cos(2\pi f_c t) + S_2(t) * \sin(2\pi f_c t)$$



C. AWGN – Noise to  $S(t) < S(t) + W(t) >$  (I used the SNR =1)



D. Overall result Comparing the input and out put response

Random bit signal  $m_{r1}, m_{r1}$  - They are too long so I attach the separate text file.

Sampled received signal  $R_1, R_2$  – They are too long so I attach the separate text file.

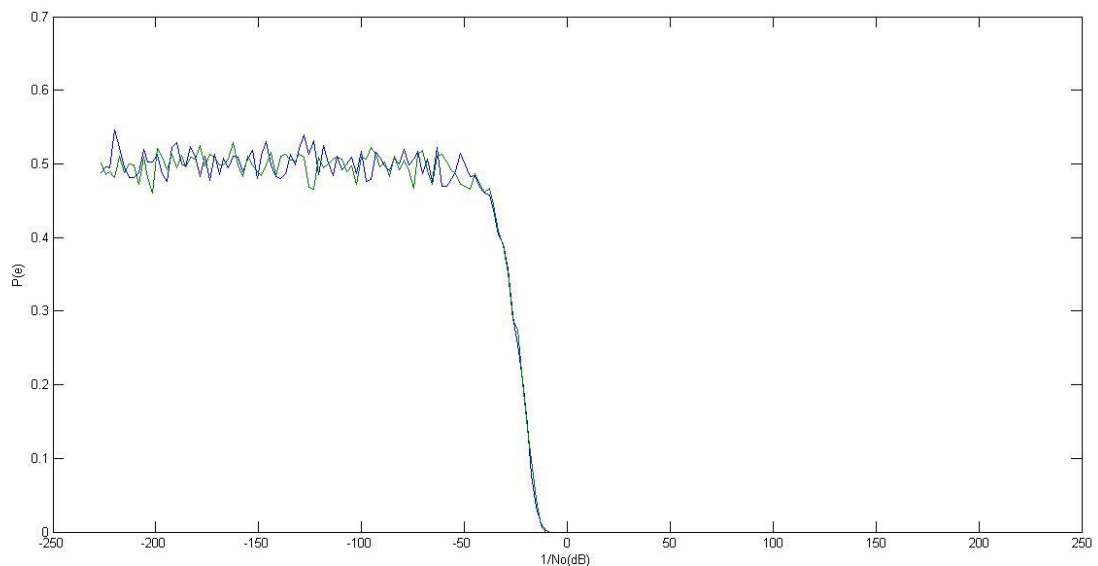
Error1 =0, Error2 =0

Error Rate =  $P_1(e) = 0, P_2(e) = 0$  .

**It doesn't make any error in SNR = 1. Let's plot the P(e) vs 1/N<sub>0</sub>**

### **3) Plot P(e) vs 1/N<sub>0</sub> for 1/N<sub>0</sub> ~ -100dB to 100dB**

According to graph below, P(e) saturated around the 0.5 and the error rate begins to collapse to zero when the dB of 1/N<sub>0</sub> becomes around -20dB



## **4. Conclusion**

I could see the result of each process by MATLAB that I can get the same signal from the receiver even if the noise ratio is not so much big like  $\frac{1}{N_0}$  is about -20dB. This results thanks to the role of modulation and demodulation with sinusoid as a QAM(Quadrature Amplitude Modification) system.

But there must be improvements in coding and channel decision to make better communication system. Though it was challenging term project, I could understand much better about the details of transmitter, receiver, AWGN, modulation, demodulation and MATLAB coding.