

# Design of Digital Simulation system for QAM Modulation and Demodulation using MATLAB

**Donghoon Ryu (Student ID: 20115032)**

**Lecturer: Professor Heung-No Lee**

Senior in Electrical Engineering and Computer Sciences

Abstract: In this report, we implement the digital simulation system using MATLAB to illustrate the Quadrature Amplitude Modulation (QAM). QAM is a modulation scheme to convey two digital bit streams which can be converted to symbol signals by modulating the amplitudes and the phase of two carrier waves. First, we transmit a signal composed of two different 1024 bit streams and modulate them using QAM. Next, at the receiver, we demodulate them using an orthogonality of two carriers and matched filters. We also conduct the error analysis of two sampled signals in order to observe how the probability of making bit errors changes depending on the inverse of Power Spectral Density (PSD) of Additive White Gaussian Noise (AWGN). Finally, we compare the error analysis in simulation with theoretical analysis.

## 1. Introduction

Modulation of signals has been playing an important role in the field of communication engineering. In order to create desired signals which work for the communication system that we want to use and transmit multiple signals simultaneously over a single channel, the modulation of signals is in high demand. Basically, there are three types of modulating in practice: 1. Analog modulation 2. Digital modulation, and 3. Spread modulation. Amongst a wide variety of modulation methods, the QAM is practically important in the point that it can be utilized for Amplitude Modulation (AM) and Frequency Modulation (FM) and it has an advantage on communicating signals within the limited frequency spectrum.

In QAM, two orthogonal carriers, commonly referred as to in-phase and quadrature, can modulate their amplitude and frequency. Suppose that you send signals  $r_1(t)$  and  $r_2(t)$  which are converted from bit signals in each channel. Then, the signal at the end of receiver  $s(t)$  before transmitting to the receiver, based on QAM can be expressed as follows.

$$s(t) = \text{Re}\{(r_1(t) + r_2(t)) * g\left(\frac{t}{T}\right) \exp(j2\pi f_c t)\} - (1)$$

where  $g(t/T)$  is a transmit filter to shape the symbol signals,  $*$  is convolution operator and  $f_c$  is a modulation frequency.

More concisely,  $s(t) = r_1(t) \cos(2\pi f_c t) + r_2 \sin(2\pi f_c t) - (2)$ .

This signal at the end of the receiver and as the signal we aim to transmit is expressed using two orthogonal basis (in-phase and quadrature) as you can see equation (2). By doing so, we can simply express signals and at the receiver, we can retrieve  $r_1(t)$  and  $r_2(t)$  using

orthogonality of sinusoidal carriers and the matched filter at the receiver. Generally, even with extremely low SNR, we can always retrieve the original information in the ideal system. However, in negative SNR, we observe the error that input bits and output bits are mismatched.

## 2. Design of QAM Digital Simulation System

### 2.1 System Schematics of the System

Basically, we can categorize the QAM system in this project into “three major blocks” : 1. Modulation, 2. Addition of Additive White Gaussian Noise (AWGN) and 3. Demodulation. In the figure 1, the left part illustrates the modulation part and the signals with sinusoidal carriers are drawn in the middle which AWGN is added to (AWGN part is not described in the figure 1.) Then, the right part describes the demodulation part.

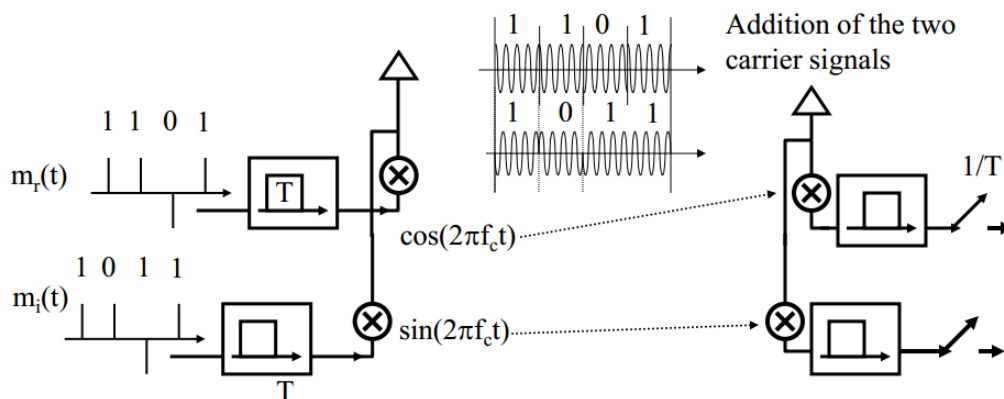


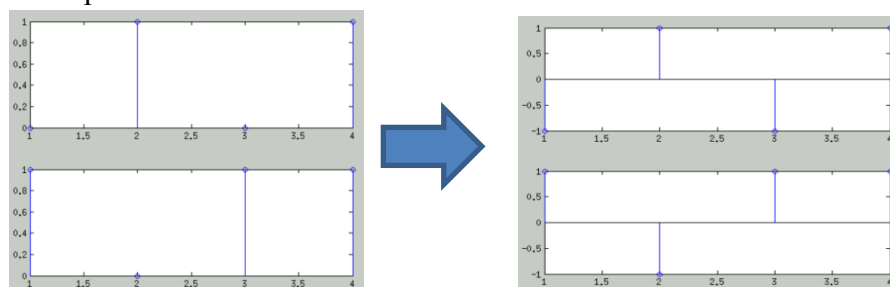
Figure 1: The QAM system is illustrated above. This figure illustrates how two signals of bit stream is modulated and demodulated at the transmitter and receiver, respectively. (Reuse of figure 1 from the lecture notes)

To make a better understanding, I would like to explain each part in the QAM system in step by step fashion from the following sections

### 2.2 Modulation part

#### A. Bit streams to Symbols

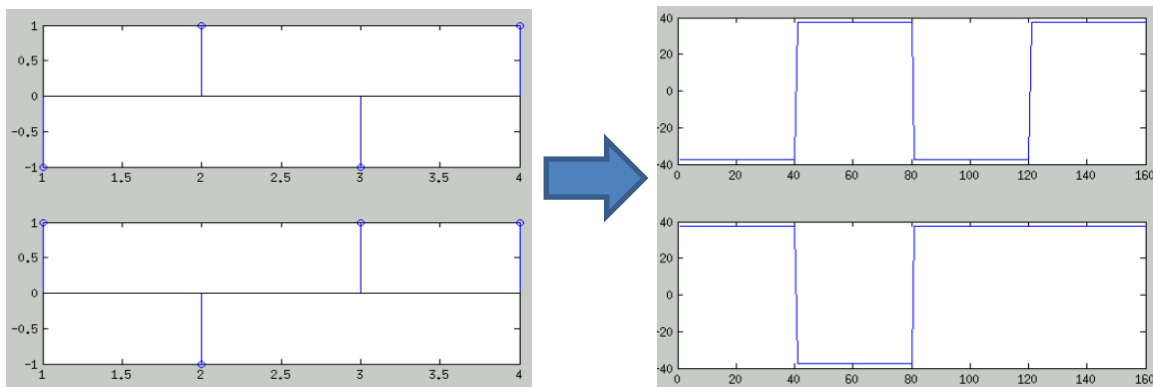
First, we convert signals of bit streams (1 or 0) into symbol signals. In this project, one – bit is exactly expressed as one symbol. For example, “bit 1” corresponds to the symbol level 1 and “bit 0” corresponds to the symbol level -1. In this project, 1024 random bits are generated for the experiment.



As an example, I will use 4 bit- signal transmission case. This explains how bit stream is converted into symbol signals. (0101) at the left-top corresponds to the (-1 1 -1 1) at the right-top.

### B. Symbol signals to pulse shaped signals

In order to multiply the signals with sinusoidal carriers, we pass the signals through the rectangular shape filter. After passing through the filter, the symbol signals are pulse shaped signals as you can see the below figure.



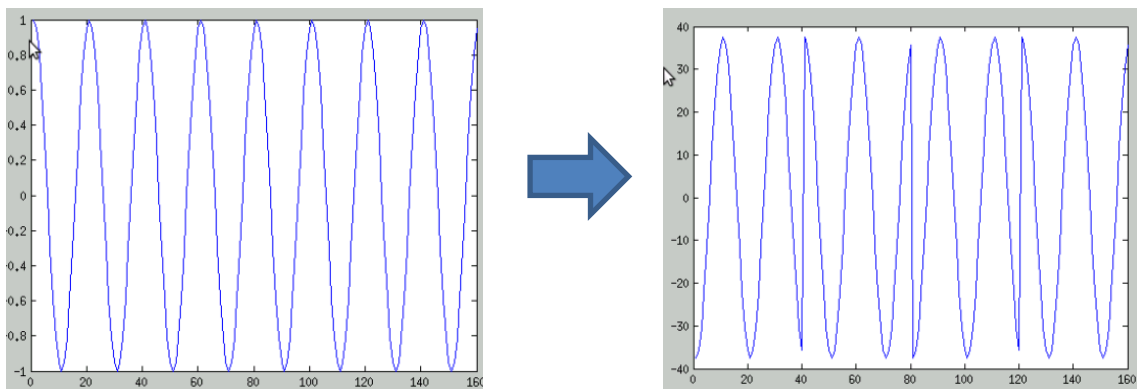
In order to keep the energy of the filter to be equal to  $\sqrt{2}$  with filter width in time domain,  $T=1\text{msec}$ ,

$$\int_0^{1\text{msec}} A^2 dt = \sqrt{2} \Rightarrow A=2^{1/4}10^{3/2}$$

The pulse shaped signal at the right=  $r(t) * g\left(\frac{t}{T}\right)$  ( $g\left(\frac{t}{T}\right)$  is rectangular filter)

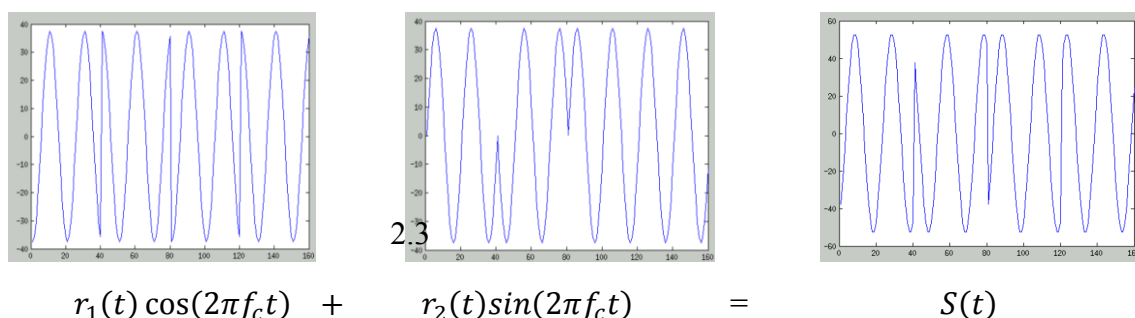
### C. Multiplication with carriers and sum of two modulated signals

The essential part of QAM is to use two independent basis carriers. Depending on the signal level (whether it is 1 or -1) the phase of sinusoidal is differently shaped in the each pulse.



Sinusoidal function is inverted in level -1 pulse and same in level 1 pulse.

At this point, two signals in two channels are summed up before transmitting to the receiver. Transmitted signal  $S(t) = r_1(t) \cos(2\pi f_c t) + r_2(t) \sin(2\pi f_c t)$ .



Also, modulation frequency should be 20 times larger than the baud according to the specification from the book. So,  $f_c$  should be larger than  $2 * 10^4$ Hz. I pick  $f_c$  of  $4 * 10^4$ Hz throughout this project. This means that 40 sinusoidal functions exist within one-rectangular pulse.

### 2.3 Adding AWGN

AWGN  $N(t)$  is added in to the transmitted signal  $S(t)$  in this step. Hence, the received signal  $R(t)$  is following as.

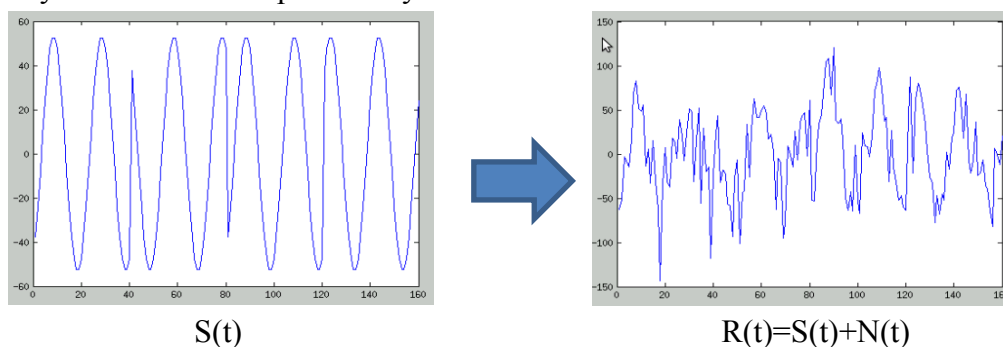
$$R(t) = S(t) + N(t)$$

And Signal-to-Noise Ratio determines the level of AWGN, which influences on the signal.

$$\begin{aligned}
 \text{SNR} &= \frac{E_b * R}{W * N_o} = \frac{\int_0^{1m} A^2 (\cos(2\pi f_c t) + \sin(2\pi f_c t))^2 dt * \frac{2\text{bit}}{10^{-3}\text{sec}}}{\left(\frac{1}{2}\right) * 10^3 \text{Hz}} * \left(\frac{1}{N_o}\right) \\
 &= 4\sqrt{2} \left(\frac{1}{N_o}\right) - (3)
 \end{aligned}$$

Where  $E_b$  is energy per bit,  $R$  is a bit rate,  $W$  is a bandwidth of the overall system, and  $N_o$  is PSD of AWGN.

Thus, SNR is directly proportional to  $\left(\frac{1}{N_o}\right)$ , which states that PSD determines SNR and may eventually influence on the probability errors. We will discuss about it on later section.



### 2.4 Demodulation part

#### A. Multiplication with sinusoidal and use of filter

Demodulation of QAM signals can be easily achieved by multiplying carrier again and passing them into identical filter that we used in modulation part (Matched filter)

In other words, two retrieved signals, I(t) and Q(t) are

$$I(t) = (R(t)\cos(2\pi f_c t)) * g\left(\frac{t}{T}\right)$$

$$Q(t) = (R(t)\sin(2\pi f_c t)) * g\left(\frac{t}{T}\right)$$

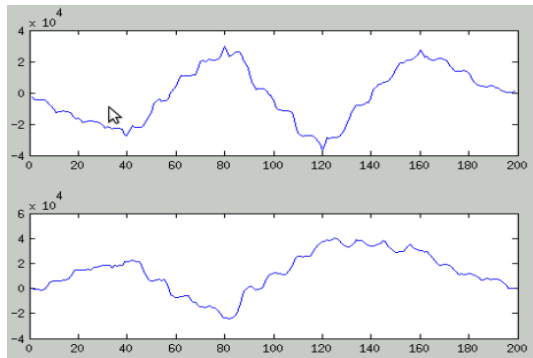
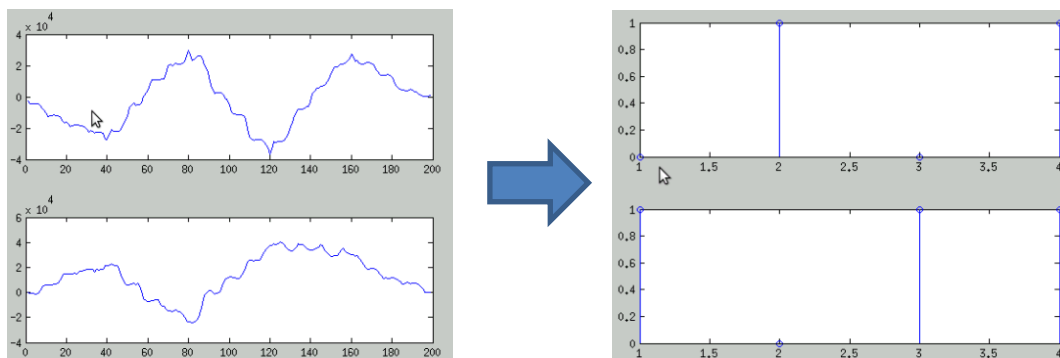


Figure 2: These correspond to I(t) and Q(t), respectively. Upper one is I(t) and lower one is Q(t). At this point, we can guess that the peak value above at the 0 indicates “level 1” and the peak value below at the 0 indicates “level -1”.

### B. Sampling of demodulated function and conversion to the bit stream

As I explained about the threshold of signals, the function value above 0 will be level 1 and the function value below 0 will be level 0. At every T sec (which is the width of rectangular shape filter), the function values are determined based on the aforementioned standard. Thus, figure 2 can be converted into symbol level and finally converted into the bit stream as follows.



Finally, the bit streams are retrieved. And this is exactly same with the input original bit streams even if AWGN is added during the transmission.

### 3. Results with randomly generated 1024 bits and error analysis

In the previous section, we go through whole procedure of QAM modulation and demodulation step by step with the example of 4 bit stream for better visualization and

understanding. And we confirm that bit streams are successfully retrieved even with AWGN.

In this part, since the 1024 bit information is hard to be visualized, we would like to analyze the probability of making bit errors as a function of  $\left(\frac{1}{N_0}\right)$ . We also would like to note that all of parameters used in the following except the change of number of bits (bit number: 1024) are identical to the previously discussed experiment. From the equation (3), SNR is directly related to  $\left(\frac{1}{N_0}\right)$  term. That is, the PSD of AWGN changes SNR and accordingly changed SNR may or may not influence on the probability of making bit errors  $P(e)$  ( $P(e)$  is defined as the number of bit decision errors/the total number of bits)

In positive SNR, even with significantly low SNR,  $P(e)$  is always zero. By contrast, In negative SNR, there may exist the probability of making errors. Hence, we somewhat expect that in negative  $\left(\frac{1}{N_0}\right)$ , there will be, to certain degree, errors whereas  $P(e)$  is zero in positive  $\left(\frac{1}{N_0}\right)$ .

I set the range of  $\left(\frac{1}{N_0}\right)$  from -1000dB to 100dB and plot the corresponding  $P(e)$  as follows.

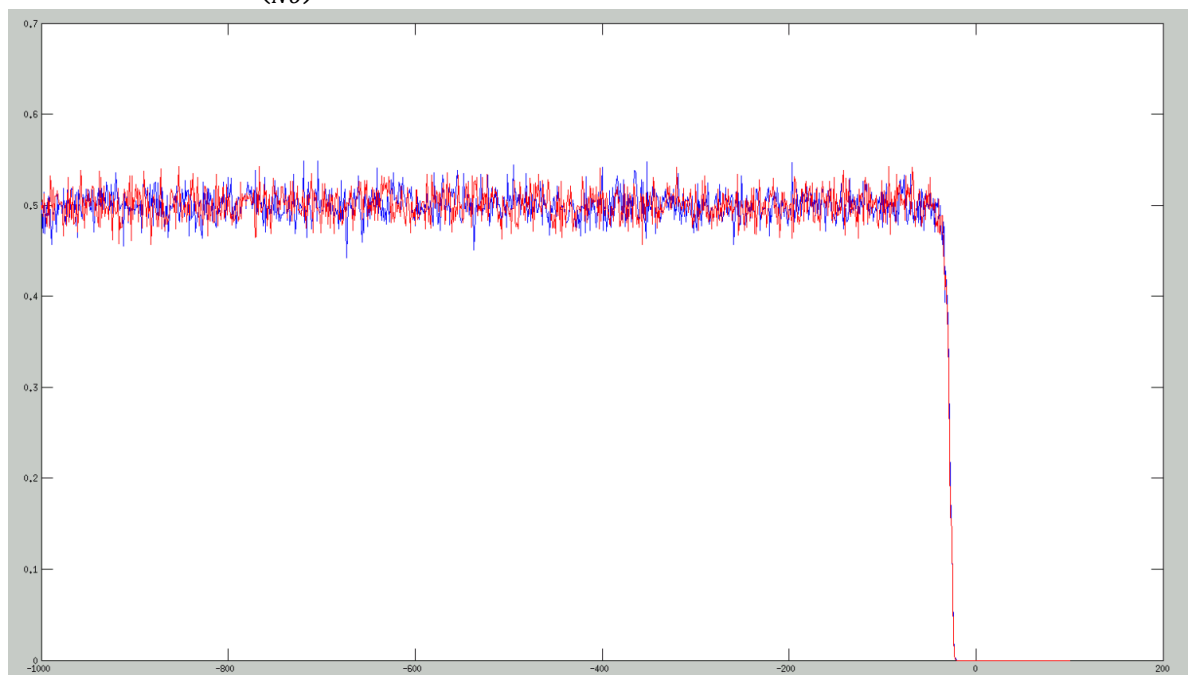


Figure 3:  $P(e)$  is the probability of making errors as a function of  $\left(\frac{1}{N_0}\right)$ . x-axis is  $\left(\frac{1}{N_0}\right)$  in dB scale and y-axis is  $P(e)$ . Red graph indicates  $P(e)$  for the “I- bits stream” and Blue graph indicates  $P(e)$  for the “Q-bits stream”. In both case, for the positive region,  $P(e)$  is completely zero. As the  $\left(\frac{1}{N_0}\right)$  goes to the negative realm,  $P(e)$  goes up. Finally, it seems that  $P(e)$  below certain value it rapidly fluctuates within a small range at the center of 0.5.

We can argue that if SNR is too low, it is difficult for us to distinguish noise and signal. Thus, the point that the probability of making error  $P(e)$  approaches to 0.5 seems reasonable.

Theoretically, the way we can calculate the  $P(e)$  is to obtain the ratio of  $(1/2) \times$  overlapped area of two Gaussian functions representing each noise distribution on each bit to the total area of one Gaussian function.

$P(e) = Q\left(\sqrt{\frac{E_d}{2N_0}}\right)$ . This equation tells us that as  $\left(\frac{1}{N_0}\right)$  negatively decreases in dB scale, the overlapped portion of two distributions of  $s_1$  and  $s_2$  in the below figure increases. Eventually, the overlapped region becomes approximately half of the total region.

$P(e) \sim 0.5$ . We can confirm that theoretical result and experimental result for  $P(e)$  show high consistency.

