Contents

List of Tables

List of figures

Chapter 1

Introduction

A modulation is a process by which an information signal is converted to a sinusoid waveform for digital communication, such a sinusoid of duration T is referred to as a digital symbol. The sinusoid has just three features that can be used to distinguish it from other sinusoids: amplitude, frequency, and phase. Therefore, the modulation can be defined as a process whereby the amplitude, frequency, or phase of the RF carrier, or a combination of them, is varied in accordance with the information to be transmitted. The general form of the carrier wave, *s(i),* is as follows:

$$s(t) = A(t)\cos[\omega_0 + \varphi(t)]$$

Where A(t) is the time-varying amplitude, ω_0 is the radian frequency of the carrier and O(t) is the *phase,* When the frequency, *f,* is used instead of *ω,* the relationship between them is $\omega+2\pi f$

The basic digital modulation/demodulation types as follows:

- Phase Shift Keying (PSK)
- Frequency Shift Keying (FSK)
- Amplitude Shift Keying (ASK)
- Hybrids

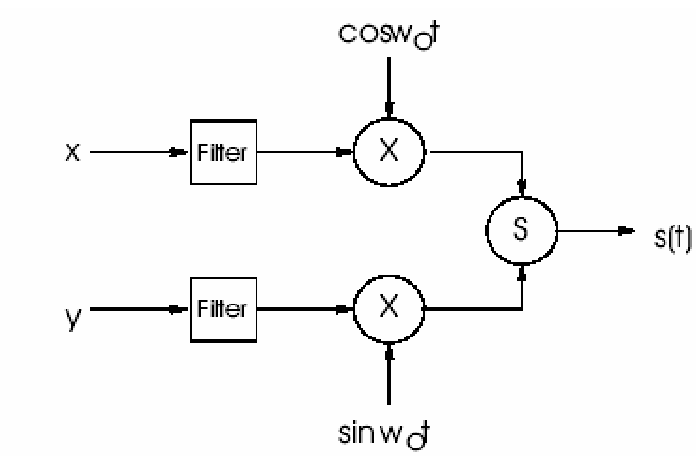
Quadrature Amplitude Modulation (QAM) is generally the combination of ASK and PSK (APK). The general expression in equation 1.2 illustrates the indexing of both the signal amplitude term and the phase term.

$$s_i(t) = \sqrt{\frac{2Ei(t)}{T}}\cos[\omega_0 + \varphi_i(t)] \qquad \begin{matrix} 0 < t < T \\ i = \end{matrix} \qquad (1.2)$$

M is the number of symbols of the information signal. When the set of M symbols in the two-dimensional signal space are arranged in a rectangular constellation, the signaling is referred to as quadrature amplitude modulation (QAM).

1.1 Modulation

A quadrature-amplitude-modulated (QAM) signal uses two quadrature, carriers, cosine function ($\cos 2\pi f_0 t$), and sine function ($\sin 2\pi f_0 t$), each of which is modulated by an independent sequence of information bits. The simplest block diagram of the QAM modulator is shown in Figure 1.1



Picture 1.1: Canonical QAM Modulator

The transmitted signa1 waveforms have the form given in Equation 1.3

$$Si(t) \quad A_{0i}\, g(t) \cos 2\pi f_0 t + A_{si} g(t), \sin 2\pi f_0 t \qquad i = 1, 2; 3, \ldots, M \qquad (1.3)$$

Where A_{0i} and A_{si} are the of amplitude level-, that are obtained by mapping k-bit sequence into signal amplitudes, For example, Figure 1.2 shows a 16-QAM signal constellation that is obtained by amplitude modulating each quadrature carrier by M = 4 *Pulse Amplitude Modulation* (PAM). Generally, rectangular signal constellations result when two quadrattire carriers are each modulated by PAM.

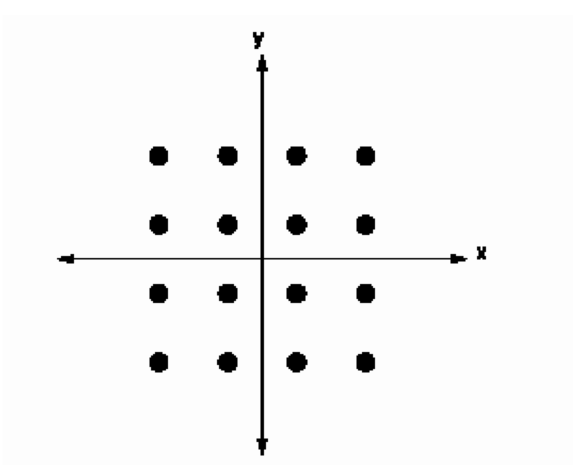


Figure 1.2: 16-ary Signal Space for QAM Modulation

More generally, QAM may be viewed as a form of combined digital amplitude and digital-phase modulation. Therefore the transmitted QAM signal waveform may be expressed as:

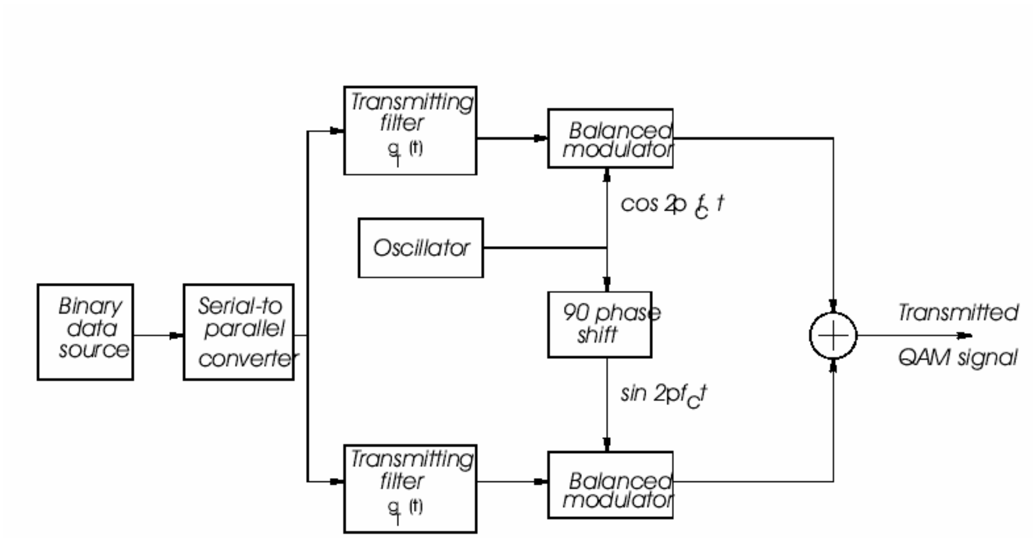


Figure 1.3: Functional Block Diagram Of Modulator for QAM

:

$S_{ij}(t) = A_i g(t) \cos(2\pi f_0 t + \varphi_j)$ (1.4) i= 1,2,….,M

J= 1,2,….,M

If $M_1 = 2^{k1}$ and $M_2 == 2^{k2}$, the combined amplitude and phase-modulation method results in the simultaneous transmission of $k_1 + k_2 = log_2 M_1 M_2$ binary digits occurring at a symbol rate $R_b=(k_1+ k_2)$. Figure 1.3 shows the functional block diagram of a QAM modulator [3].

It is clear that the geometric signal representation of the signals give by 1.3 and 1.4 is in terms of two-dimension signal vectors of the form

$$Si = (\sqrt{E_s A_{0i}} \sqrt{E_s A_{si}}) \qquad m=1,2,\ldots,M$$

There are many types of signal space constellations. Figure 1.4 shows th samples of rectangular and circular signal space constellations [3].

1.2 Noise in Communications

Noise in communications can cause distortion of the transmitted signal. There are a variety of noise sources, such as galaxy noise, terrestrial noise amplifier noise, and unwanted signals from other sources [4]. An unavoidable cause of noise is the thermal motion of electrons in any conducting media. This motion produces thermal noise in amplifiers and circuits which corrupt the signal in an additive fashion; that is, the received signal, r(t), is the sum of the transmitted signal, s(t), and the thermal noise, n(t). The statistics o thermal noise have been developed using quantum mechanics and are well known [1].

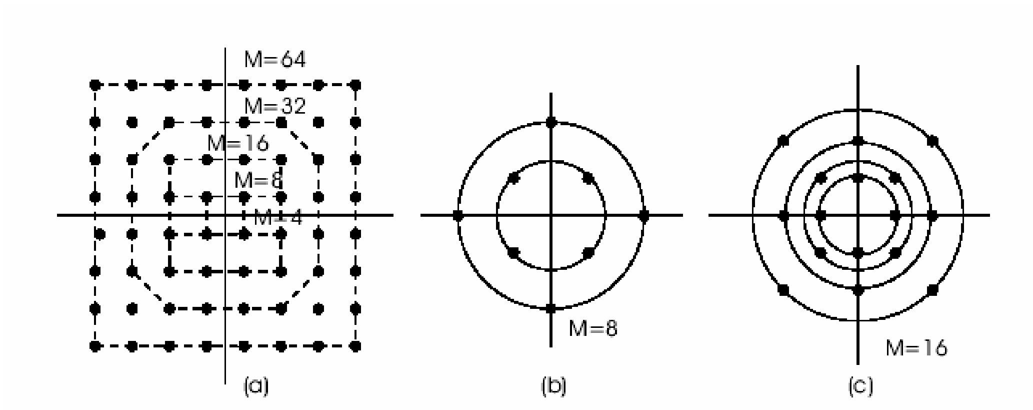


Figure 1.4: (a) Rectangular and (b), (c) Circular QAM Signal Constellation

The main statistical characteristic of thermal noise is that the noise amplitudes .are distributed according to a normal or Gaussian distribution. The, probability density function (pdf), p(n.), of the zero-mean noise voltage is shown as

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2\right] \quad (1.6)$$

Where σ^2 is the noise variance.

We will often represent a random signal as the sum of a Gaussian noise random variable and a dc signal:

$$z = a + n \quad (1.7)$$

Where z is the random signal, a. is the de component, and n is the Gatissian noise random variable. The pdf, p(z) is shown as

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{z-a}{\sigma}\right)^2\right] \quad (1.8)$$

Where σ^2 is the noise variance of n.

The primal, spectral characteristic of thermal noise is that its power spectral density is the same for all frequencies of interest in most communication systems. Therefore, a simple model for thermal noise assumes that its power spectral density $G_n(f)$ is flat for all frequencies, as shown in Figure 1.5 [4], and is denoted as follows:

$$G_n(f) = \frac{N_0}{2} \qquad \text{watts/hertz}$$

Where the factor of 2 is included to indicate that $G_n(f)$ is a two-sided power spectral density. When the noise power has such a reform spectral density, we refer to it as white noise. The adjective "white" means that white light contains equal amounts of all frequencies within the visible band of electromagnetic radiation.

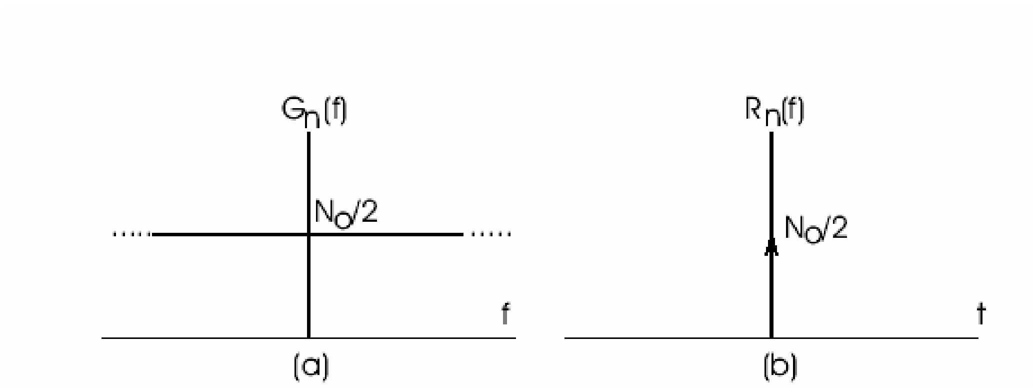


Figure 1.5: (a) Power spectral density of white noise (b) Autocorrelation function of white noise

From the characteristics of white noise, any two different samples of a White noise process are uncorrelated. Since thermal noise is a Gaussian process and the samples are uncorrelated, the noise samples are also independent [2]. Therefore, the effect on the detection process of a channel with additiv*e* white Caussian noise (AWGN) is that the noise affects each transmitted symbol independently. The term "additive" means that it is simply superimposed or added to the signal [4].

Since thermal noise is present in all communication systems and is the prominent noise source for most systems, the thermal noise characteristics, additive, white and Gaussian, are most often used to model the noise in communication systems.

1.3 Demodulation and Detection

The received signal is transmitted over the channel. We assume that the transmitted signal is transmitted by Additive white Gaussian Noise (AWGN). In addition, we assume that a carrier-phase offset is introduced in the transmission of the signal through the channel. Therefore the received signal, r(t), may be expressed as

$$r(t) = A_{0i}g(t)cos(2\pi f_0 t + \varphi) + A_{si}g(t),sin(2\pi f_0 t + \varphi) + n(t)$$
(1.10)

Where, n (t) is the AWGN signal, and φ is the carrier phase offset.

We express the simple AWGN channel model in Figure 1.6.
In addition, the AWGN, n (t) can be expressed as follows.

$$n(t) = n_o(t)cos2\pi f_0 t - n_s(t)sin2\pi f_0 t$$

(1.11)

The received signal is correlated with the, two phase-shift basis functions

Ψ_1 (t) g(t)cos($2\pi f_0 t + \varphi$)

$\psi_2(t)$ g(t)sin($2\pi f_0 t + \varphi$)

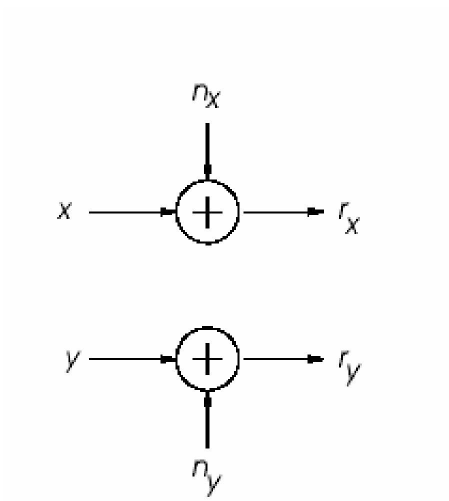


Figure 1.6: QAM Noise Channel Model

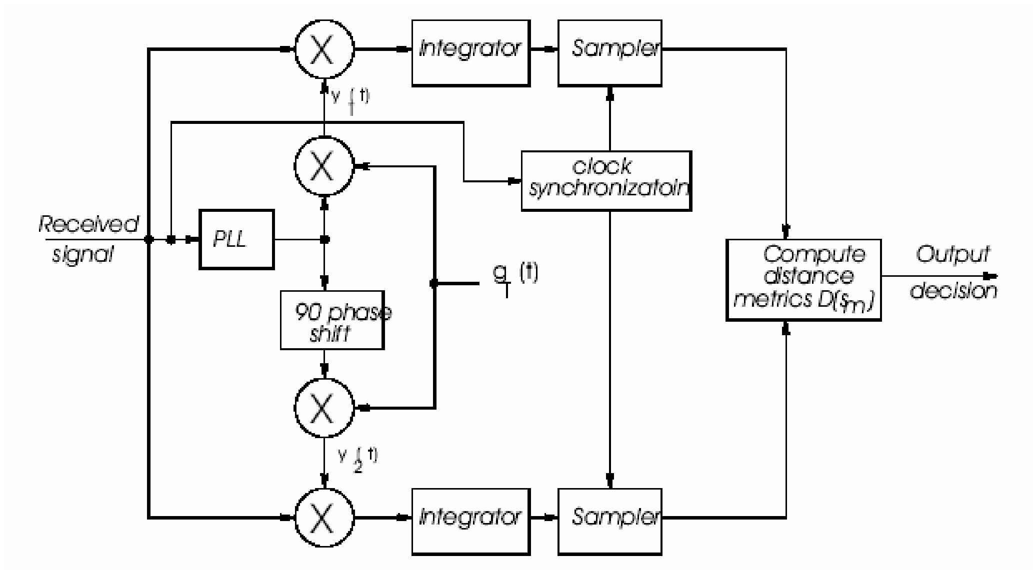


Figure 1.7: Demodulation and Detection of QAM signals [3]

As shown in Figure 1.7, and the outputs of the correlators are sampled and passed to the detector.
The phase-locked loop (PLL) illustrated in Figure 1.7 estimates the carrier phase offset φ of the received signal and compensates for this phase offset by phase shifting $\psi_1(t)$ and $\psi_2(t)$ as indicated in Equation 1.12. The clock shown in Figure 1.7 is assumed to be synchronized to the received signal so that the correlator outputs are sampled at the proper instant in time. Under these assumptions, the outputs from the two correlators are:

$$r_1 = A_{0i} + n_{cos}\cos\varphi - n_{sin}\sin\varphi$$

$$r_2 =. A_{si} + n_{cos}\cos\varphi - n_{sin}\sin\varphi \qquad (1.13)$$

Where :

$$n_{\cos} = \frac{1}{2}\int_0^T n_c(t)\,g(t)\,dt$$

$$n_{\sin} = \frac{1}{2}\int_0^T n_s(t)\,g(t)\,dt$$

(1.14)

The noise components are zero-mean, uncorrelated Gaussian random variables with variance No/2.
The decision process of the detector depends on the distance between the received signal and the transmitted signal. The distance metrics are:

$$D(T, S_m) = |r - s_m|^2 \qquad (1.15)$$
$$\text{where} \quad r^t = [r1, r2] \quad \text{and} \quad s_m \text{ is given by Equation } 1.5$$

1.4 Probability of Error in the AWGN Channel

In this section we consider the performance of QAM systems that employ rectangular signal constellations. Rectangular QAM signal constellations have the distinct advantage of being easily generated as two PAM signals improved on phase quadrature carriers. In addition, they axe easily demodulated.

For rectangular signal constellations in which $M = 2^k$, where k is even, the QAM signal constellation is equivalent to two PAM signals on quadrature carriers, each having $\sqrt{M} = 2^{k/2}$ signal points. Because the signals in the phase quadrature components are perfectly separated by coherent detection, the probability of error for QAM is easily determined from the probability of error for PAM. Specifically, the probability of a correct decision for the M-ary QAM system is [3]

$$P_c = \left| 1 - P_{\sqrt{m}} \right|^2$$

Where $P_{\sqrt{m}}$ is the probability of error of a $\sqrt{M - ay}$ PAM average power in each quadrature signal of the equivalent QAM system, By appropriately modifying the probability of error for M-ary PAM, we obtain:

$$P_{\sqrt{m}} = 2\left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\sqrt{\frac{3}{M-1}\frac{E_{av}}{N_0}}\right) \tag{1.17}$$

Where E_{av}/N_0 is the average SNR per symbol [3]. Thus, the probability of a symbol error for the M-ary QAM can be expressed as [3].

$$P_M = 1 - \left| 1 - P_{\sqrt{M}} \right|^2 \tag{1.18}$$

We note that this result is exact for $M=2^k$ when k is even. Otherwise, when k is odd, there is no equivalent. $\sqrt{M}$ -ary PAM system. This is no problem, however, because it is quite easy to determine the error rate for a rectangular signal set. If we employ the optimum detector that base.-, its decisions on the optimum distance metrics given by Equation 1.15, it relatively straightforward to show that the, symbol-error probability is tightly tipper-bound as

$$
\begin{aligned}
P_M &\le 1-\left[1-2Q\left(\sqrt{\frac{3E_{av}}{(M-1)N_0}}\right)\right]^2 \\
&\le 4Q\left(\sqrt{\frac{3kE_b}{(M-1)N_0}}\right)
\end{aligned}
\tag{1.19}
$$

, for any k $\geq$ 1, where E_b/N_0 is the average, SNR per bit [3]. The probability of a symbol error is plotted in **Figure** 1.8 as a function of the average SNR per bit.

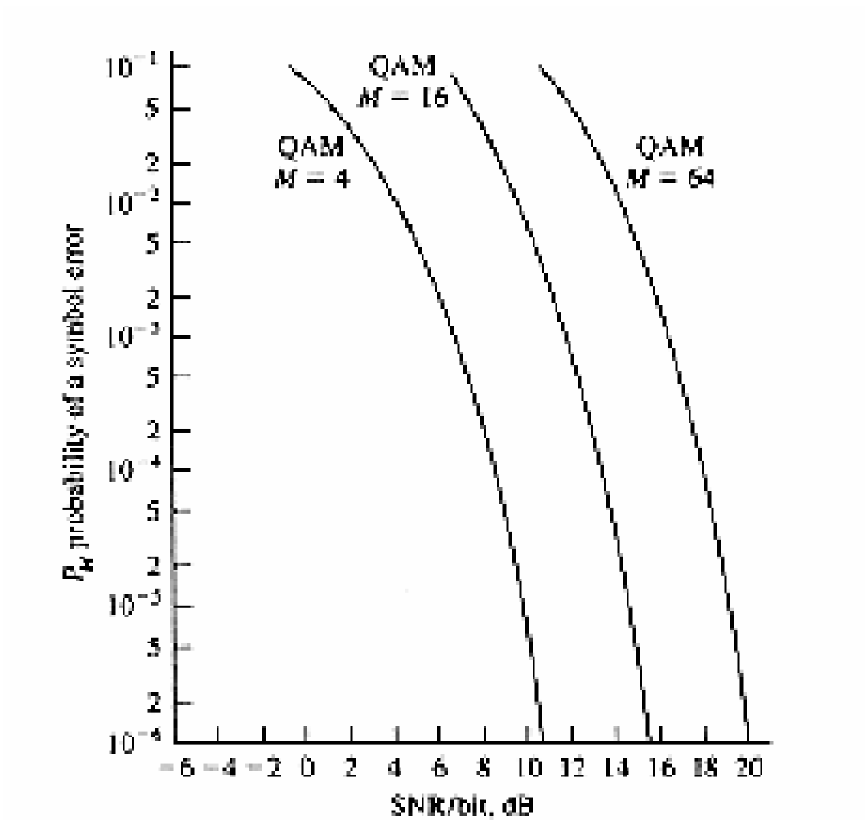


Figure 1.8 : Proobability of a symbol error for QAM [3]

Chapter 2

Implementation

In this chapter, we discuss about the implementation methods of the 16-QAM communication system. The following section describes %n implementation of simulation method.

2.1 Simulation of the 16-QAM Communication System

Our simulator of the 16-QAM communication system is described as follows. First, we create the source information using the random member generator (RNG). Then the signal constellation space is created and the source data is mapped into the signal constellation to produce the QAM signal to be transmitted over the signal channel. In addition, we generate the 4-AWGN channel over which the source signal is to be transmitted. At the receiver side, we receive the corrupted signal and compute the distance between the received signal and the transmitted signal using the characteristic, of signal space. After the simulator completes the minimum distance between the received signal and the transmitted signal, it makes a decision from which the source signal is transmitted. If the received signal is not the same as the transmitted signal, we accumulate the number of errors. Finally, we calculate the probability of error from the total number of errors over the number of samples.

2.1.1 Random Number Generator

We generate the random number using MATLAB command “rated” to generate uniformly distributed numbers. The distribution is on the interval of 1.0. Then, we multiply the random number with 16, and round it up. Therefore, the data source we have generates the random number between 1 and 16. Totally we generate 10.000 samples.

2.1.2 Constellation

For 16-QAM communication systems, the constellation space consist-, of 16 signal points in the signal space, There are many patterns of 16-point signal constellation, but we select the square pattern as shown in Figure 1.2. We select the signal space between signal point to be one (d - 1).

2.1.3 Modulator

After we have the data source from the random number generator and the signal constellation, we modulate the data source to the signal constellation. The modulation is done by mapping data source number (1-16) to the signal point number (1-16) in the constellation space. Then, we have the 16-QAM signal to be transmitted over the channel.

2.1.4 Additive White Gaussian Noise (AWGN) Channel

We create the function generating the two independent Gaussian random variables with the mean and standard deviation (σ) as given. We first generate the AWGN by using the command "rand" to generate the uniformly distributed random number between 0.0 and 1.0. The probability distribution function F(X) of the Gaussian pdf *f (X)* is the area under the Gaussian pdf function over the range (-oo, X). Unfortunately, the integral of *F(C)* is not easy to be expressed in the simple functions. In addition, the inverse mapping is also difficult to do. However, the solution is found in using the Rayleigh distribution. From the probability theory it is known that Ravleigh distributed random variable, R, with the probability distribution function:

$$F(R) = \begin{cases} 0 & R < 0 \\ 1-\exp\left(\frac{-R^2}{2\sigma^2}\right) & R \geq 0 \end{cases} \tag{2.1}$$

, is related to a pair of Gaussian random variables N_x and N_y through the transformation.

$$\begin{aligned} N_x &= R\cos(2\pi v) \\ N_y &= R\sin(2\pi v) \end{aligned} \qquad (2.2),(2.3)$$

Therefore, we first generate the Rayleigh distributed random number from the uniformly distributed random number as shown in Equation 2.4.

$$x = \sigma\left(\sqrt{2\ln\frac{1}{1-u}}\right) \tag{2.4}$$

Note that z is the Rayleigh distributed random variable and u is the uniformly distributed random variable.

After that, we create two Gaussian random variable numbers as the inphase component and quadrature component, by using the Equation 2.5.

$$\begin{aligned} N_x &= \mu + z\cos(2\pi u) \\ N_y &= \mu + z\sin(2\pi u) \end{aligned} \tag{2.5}$$

After we have two components of AWGN, we simply add the noise to the transmitted 16-QAM signal. The result we get is the received signal, which is corrupted by AWGN. Subsequently, the received signal will be demodulated at the receiver side as described in Section 2.1.5.

2.1.5 Demodulator and Detector

At the receiver side, the signal corrupted by noise is passed to the 16-QAM demodulator. First, we calculate the distance between the received signal point and all constellation signal points in the signal space as shown in Equation 2.6.

$$d = |r_x - c_x|^2 + |r_y - c_y|^2 \tag{2.6}$$

Note that d is the distance, r is the received signal and c is constellation signal.

We then compare all 16-distance values to find the minimum one which will be the decision value that we extracted from the noise signal. After we get the decision value, we compare it to the actual signal value which is transmitted from the source. If the, transmitted data is not the same as the detected data, then we accumulate the number of errors.

After we detect all the data from the source, we calculate the probability of error by simply dividing the member of errors with the total member of data sent.

2.1.6 16-QAM Communication System

The simulation of the 16-QAM communication system is shown as the block diagram in Figure 2.1.

First, the information source is generated from the uniform RNG block. Then, the information is modulated to the 16-QAM signal. The constellation of signal space is square as shown in Figure 1.2.

After the source signal is modulated with the 16-QAM signal, it is transmitted through the AWGN channel. The transmitted signal is corrupted by noise, which is generated by the Gaussian random number generator (RNG). At the block of Demodulator and Detector, the received signal is demodulated and detected the source data. The result from the signal detection is compared to the original signal generated from the Uniform RNG block. The number of errors is accumulated after it finds that the result is not the same as the transmitted signal.

All of the above implementation is built using MATLAB software. All the MATLAB scripts are shown in Appendix.

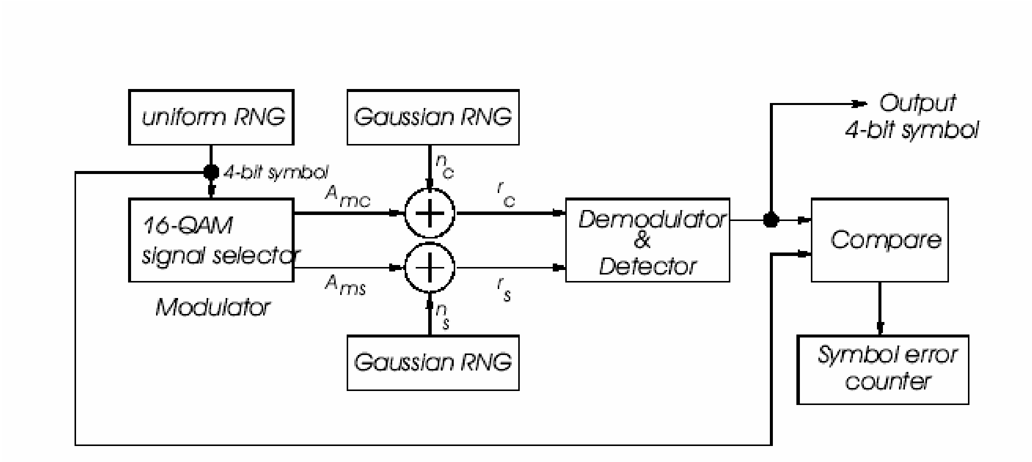


Figure 2.1: Block diagram of an 16-QAM communication system for the simulation

Chapter 3

Experimentation

In this section, we describe the experiment for the simulation. The simulation consists of two parts, the simulated result and the theoretical result. We compare the performance metric as described in Section 3.1 of the simulated results to that of the theoretical results. The simulation model is defined in Section 3.2.

3.1 Performance Metrics

The result of simulation is represented in two parts, simulated result and theoretical result. These results are compared with the probability of symbol error (P_S) and the signal energy to noise ratio (SNR).

The definitions of these performance metrics are described below:

- Probability of a symbol error (P_S) is the probability that the detector makes an incorrect decision at the symbol level.

- SNR is the ratio of the average modulating signal power to average noise power.

3.2 Simulation Model

In our experimentation, we set parameters in our simulation model as follows:

- For the theoretical experiment, the signal to noise ratio (SNR,) of the input is ranged from 0 dB to 15 dB which is in steps of 0.1 dB.

- For the simulated experiment, the signal to noise ratio (SNR) of the input, is ranged from 0 dB to 15 dB which is in steps of 2 dB.

- Since **our** system is 16-QAM communication, we have 16 symbols which have 4 bits per symbol.

- The number of samples of the source information is 10000 samplers.

- The minimum between symbols in the constellation plot is set to 1 unit (d=1)

- Energy per symbol, E_S, is $10*d^2=10$

- Zero-mean noise and its variance is $\sqrt{E_S / 8SNR}$

Chapter 4

Results

This section shows the results of the simulation and the theoretical calculation.

4.1 Symbol Error Probability

The plot in Figure 4.1 shows the probability of symbol error of the system according to its SNR values. From figure 4.1, at the SNR, equal to 10 dB, the probability of symbol error of the simulated result is 0.0062. However, in the theoretical results, the probability of symbol error is 0.009355 at the same SNR value.

The difference between the simulated results and the theoretical results is shown in ratio form as follows.

$$\text{Difference} = \frac{\left| P_{s\ simulated} - P_{s\ theoritical} \right|}{P_{s\ theoritical}}$$

Table 4.1 shows the simulated and the theoretical results, and the difference between the two results at the different values of SNR.

SNR (dB)	P, from Simulated results	P, from Theoretical result	Difference
0	0.47.33	0.742187	0.36
2	0.3502	0.520316	0.32
4	0.2234	0.312632	0.28
6	0.1143	0.148647	0.23
8	0.0119	0.049318	0.35
10	0.0062	0.009355	0.34
12	0.0008	0.000739	0.08
14	0.0001	0.000015	5.67

Table 4.1: The Probability of Symbol Error from the simulated and theoretical results

The values of the results of simulation and theoretical calculation are nearly the same. The results from the simulation tend to be less than those from the theoretical calculation. Furthermore, beyond the point that SNR is equal to 14 dB, the result from the simulation tends to be much more different than that of thoritical calculation. We could say that this simulation experiment gives as a reasonable result when the SNR is not over 12 db.

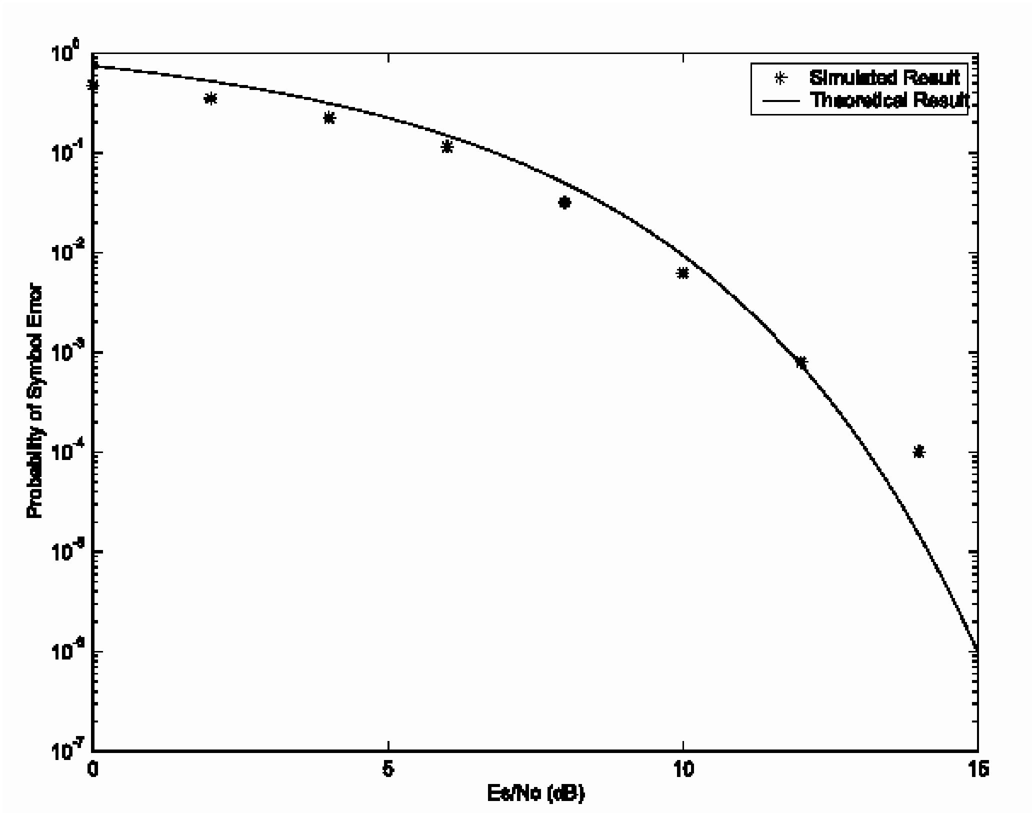


Figure 4.1: Probability of symbol error and SNR of the simulated and theoretically calculated results

Chapter 5

Conclusions

Quadrature Amplitude Modulation (QAM) is a hybrid modulation, which modulates the signal by changing the amplitude and phase of the carrier signal. The QAM modulation techniques has an advantage over PSK and ASK modulation techniques. It consumes less bandwidth than the PSK modulation technique and has less probability of bit error than the ASK modulation technique at the same number of symbols.

The modulation of QAM technique uses two quadrature carriers, cosine and sine functions, each of which is modulated by an independent sequence of information bits. The signal space of all the carrier signals is shown in the form of the signal space constellation. After the signal is transmitted over the channel, it will be corrupted by noise. As in our experiment, the noise is assumed to be only AWGN. Therefore, the received signal is the sum of the transmitted signal and the noise signal. The received signal is filtered, and then passed to the detector. The detector decides which symbol is transmitted. The decision is based on the minimum distance of the received signal and the transmitted signal. The wrong decision is kept in the record to calculate the probability of symbol error.

We use MATLAB Software to run the simulation and also to calculate the result of the same experiment in theory. The results of the simulation and theoretical calculation are compared. We have shown that the result from the simulation is very close to the one from the theoretical calculation. However, the simulation has a limitation that it can be run only with the SNR, level not over 12 dB. Otherwise, the result is unreliable.

For future work, the simulation might be developed to be able to run the simulation at the more precise SNR. We also can extend the simulation to run on the difference of the signal constellation such as hexagonal shape or circular shape.

Chapter 6

Appendix

6.1 Main Simulation Script

```
**************************
% This is the main script for 16-QAM
****************************************

echo on

% Generate the number of SNR points in dB
SNR_dB_1  = 0:  2 : 15 ;
SNR_dB_2  = 0 :  0.1 : 15;

% number of symbols
M = 16 ;

% number of bits
k = log2(M) ;

% find the simulated received signal
for i = 1 : length(SNR_dB_1),

    %%%% call function 'find_prob_err'


        simulated_error_prob(i) = find_prob_err( SNR_dB_l(i)

end ;

% find the signal output in the theory
for i = 1 : length(SNR_dB_2),


    % Calculate SNR from SNR in dB
    SNR = exp( SNR_dB_2(i) * log(10) / 10

    % Calculate the symbol error rate in theory
    %%%% Call function 'Qfunct'


theory_error_prob(i)= 4*Qfunct( sqrt( 3*k*SNR /   (M-1) ) ;

end  ;
```

```
% Plot theoretical and simulated results figure;
semilogy( SNR_dB_1 , simulated_error_prob, '*' ) ;
hold
 semilogy( SNR_dB_2 , theory_error_prob)

xlabel('Es/No (dB)');
 ylabel('Probability of Symbol Error');
 legend('Simulated Result','Theoretical Result');
```

6.2 Scripts of the Simulated 16-QAM munication System

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function for calculating the simulated error probability
%    Name : simulated_error_prob()
%    Input : SNR in dB
%    Output : Probability of Error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [prob] = simulated_error_prob ( SNR_DB)

% Number of source samples
N = 10000 ;

% minimum distance between symbol (in constellation plot)
   d = I ;

% Energy per symbol
Energy_per_symbol = 10 * d ^ 2 ;

% Signal to Noise Ratio per bit
SNR_per_bit = 10^( SNR_DB / 10) ;

% Noise variance
Noise_variance = sqrt(Energy_per_symbol/( 8*SNR_per_bit)) ;

% Number of symbol
M = 16 ;

% Generate the data source
for i = 1 : N,
    random_num= rand
    data_source(i) = 1 + floor( M * random_num);
    end ;

% Create the constellation signal space
```

```
constellation =        -3*d    3*d;     -d 3*d ;      d 3*d;     3*d   3*d  ;
                       -3*d    d    ;  -d d    ;    d d    ;   3*d  d     ;
                       -3*d   -d    ;  -d –d   ;    d –d   ;   3*d –d     ;
                       -3*d   -3*d ;     -d -3*d;      d -3*d;    3*d -3*d] ;

% Modulate the data source to QAM signal constellation
% QAM_signal is the signal to be transmitted over the channel

for i = 1    N,
   QAM_signal( i,:)= constellation( data_source(i),:);
   end ;

%    Receive signal is the QAM_signal plus Noise signal
%    Note that the Noise is the Gaussian Noise

for i = 1 : N,
     %%%% Call function 'generate_gaussian' for Noise
     [noise(l) noise(2)]= generate_gaussian(Noise-variance) ;
   receive_signal(I,: )=  QAM_signal( I,: )  + noise;
end ;

% plot received signal in the matrix version
plotmatrix(receive-signal,',');

% Detection and Error Probability Calculation

number_of_error = 0 ;

for i = 1 : N,
% Metric calculation

   for j = i : M
       metric(j) = (receive_signal(i,1)-constellation(j,1))^2
                 + (receive_signal(i,2)-constellation(j,2))^2
   end ;

   [ min_metric decision]= min( metric);
    if ( decision = data_source(i) ),
    number_of_error = number_of_error + 1;
    end;
end ;


prob  =  number_of_error / N;

%%%% End of simulated_error_prob()  function   %%%%
```

6.3 MATLAB Script of Gaussian Noise Generator

```
function [ gaussian_random_var_1, gaussian_random_var_2]…

                    = generate_gaussian( mean, sigma )

% This function generate the two independent Gaussian
% Random Variable with the mean and standard
% deviation (sigma) as given.
%
% [arg1, arg2]= generate_gaussian( mean, sigma)
% [arg1, arg2]= generate_gaussian( sigma)
% [argi, arg2 I = generate-gaussian( )
%
% - If mean is not provided, mean will be set as zero.
% - If neither mean nor sigma is given, is generate
% two standard Gaussian random variable with mean= 0
% and variance = 1 (sigma =1)

if nargin == 0,
   mean = 0 ; sigma = 1 ;
 end ;

if nargin == 1,
sigma = mean ; mean= 0 ;
end ;

% Uniform random variable in (0,1)


u = rand ;

% Rayleigh distributed random variable
z = sigma * ( sqrt ( 2 * log( 1 / (1 - u))) ;

% Another uniform random variable in (0,1)
u = rand ;

gaussian_random_var_l =  mean + z * cos( 2 * pi * u) ;
gaussian_random_var_2 =  mean + z * sin( 2 * pi * u) ;
```

6.4 MATLAB ScriDt of Q-Function

```
% [y] =Qfunct (x)
% QFUNCT evaluates the Q-function,
%    y = 1/sqrt(2*pi) * integral from x to inf of exp( -t^2/2) dt.
%    y = (1/2) * erfc(x/sqrt(2)).
```

```
function [y]=Qfunct(x)

y=(  1 / 2)* erfc ( x  /sqrt (2)  ) ;
```

Bibliography

[1] H. Nyquist. Thermal agitation of electric charge in conductors. Phys. *Rev.,* 32:110-113, **July** 1928.

[2] A. Papoulis. Probability, Random *Variables, and Stochastic Processed.* McGraw-Hill [1]Book Company, New York, 1965.

[3] J. G. Proakis. Communication System Engineering. Prentice Hall, Now Jersey, 1995.

[41 B. Sklar. Digital Communications: Fundamentals and Applications. Prentice Hall, New Jersey, 1988.