Chapter 7

Lecture Note Developing on Simulation Programming of 16QAM in the AWGN Channel

Author: Tri Budi Santoso

Supervisors: Prof. Hiroshi SUZUKI

Prof. Kazuhiko FUKAWA

Book Reference:

1. Digital Communication by Satellite
2. Digital Communications, John G. Proakis

Topic: 16QAM Transmission System

7.1. Session 1: Review of 16QAM

7.1.1. The Signaling of 16QAM

In *M*-ary PSK the signal amplitude is the same for all signals thus constraining the signal points to a circular constellation. If this constraint is removed and the in-phase (I) and quadrature (Q) signal components are allowed to vary independently we have a scheme called *quadrature amplitude modulation* (QAM). In this case we can have any signal constellation we choose. QAM waveforms are a combination of PAM and PSK. The basis functions are as for PSK system. The QAM signal waveforms expresses as

$$\begin{aligned} s_m(t) &= \sum_m A_m e^{j\theta_m} g(t - mT) e^{j2\pi f_c t} \\ &= A_m g(t)\left[\cos(2\pi f_c t + \theta_m) + \sin(2\pi f_c t + \theta_m)\right] \end{aligned} \tag{7-1}$$

Where $A_m = (A_I^2 + A_Q^2)^{1/2}$. And A_I and A_Q are information bearing signal amplitudes of the quadrature carriers $g(t)$ is the signal pulse. In the other book reference the parameter $A_m.g(t)$ is simplified as $A_m(t)$, which indicate a modulation amplitude. The parameter θ_m which indicate a phase modulation, has a value:

$$\theta_m = \tan^{-1}(A_Q/A_I) \tag{7-2}$$

We may choose any combination of M_1-level PAM and M_2-level PSK to give an $M=M_1M_2$ level QAM signal constellation. Figure (7.1) shows some rectangular signal constellations for different values of *M*. The optimum signal constellation will be that which requires the least average power for a given minimum distance between signal points. For M=4 this will be when the four points lie on a circle and the resulting signal constellation is the same as that for QPSK.

For M=8 the optimal signal constellation consists of the points lying on two circles as indicated in Figure (7.2).

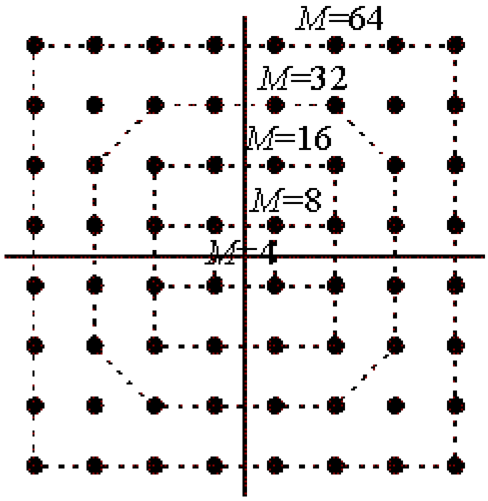


Figure (7.1). Several signal space

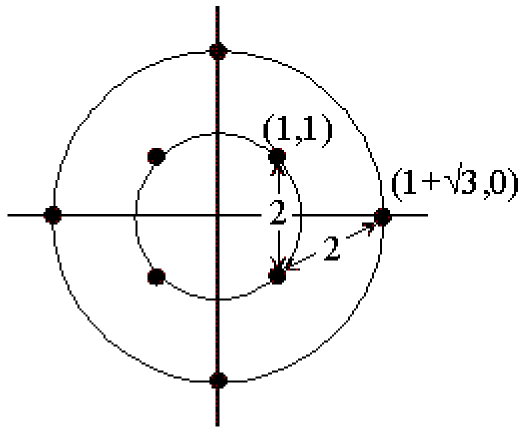
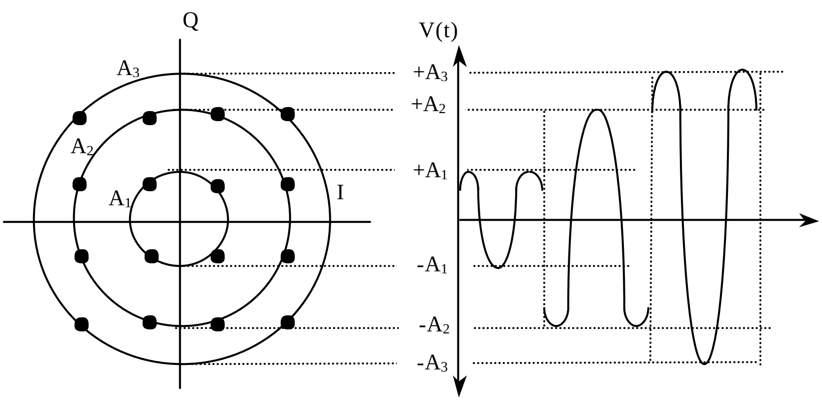


Figure (7.2) Example of PAM-PSK

For M=16 there are many more ways of building up the signal constellation however rectangular signal constellations as shown in Figure (7.3a) have the distinct advantage that they can be generate as two PAM signal impressed on the phase quadrature carriers and they are easily demodulated. We say it as 16QAM by rectangular constellation or in some book call as square constellation. Also for M=16 the average transmitter power required for the rectangular constellation is only slightly worse than the best M-ary QAM signal constellation. The signal output of 16QAM has general form as like Figure (7.3b).



a) 16QAM Rectangular   b) Signal output representative

Figure (7.3). The rectangular constellation of 16 QAM and signal output of 16QAM

The transmitter of 16QAM as the Figure (7.4). By assumption that signal input is a sequence of 4 bits pair, the next process is serial to parallel conversion. Two MSB bits in Q channel, and the others are in I channel. The parallel bit information encoded by using Gray code. The next process is mapping to 16QAM rectangular constellation (more detail in the section 7.1.2). Each channel modulates the amplitude of carrier signal. I channel modulates the sinusoid signal that has initial phase $\pi/2$ radiant ($\cos 2\pi f_c t$), then called as in-phase channel. Q channel modulates the sinusoid signal that has initial phase 0 radiant ($\sin 2\pi f_c t$) and then called as quadrature channel. The carrier-modulated signal are summing and transmitting.

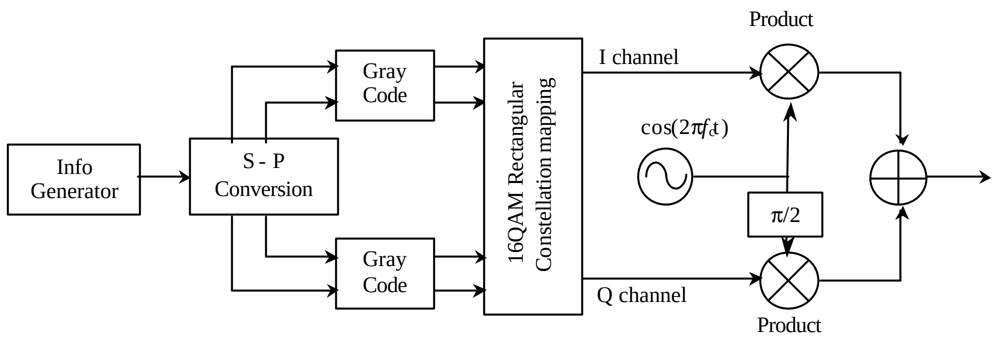
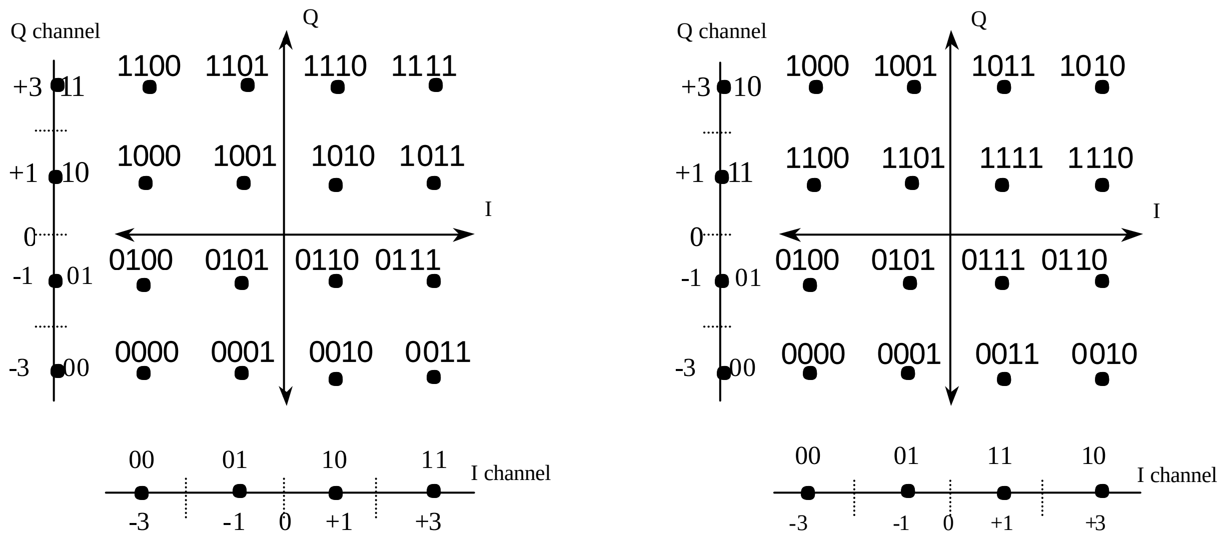


Figure (7.4) 16QAM Transmitter

7.1.2. 16QAM Rectangular Constellation

In this section we will describe how to map the bits input to rectangular constellation. Four ways mapping, will describe are:

- Natural binary code of 16QAM
- 2D gray code 16QAM
- 16QAM as sum of QPSK
- 16QAM as sum of DQPSK



a) Natural binary Code of 16QAM

b) 2D Gray Code of 16QAM

Figure (7.5) The 16 QAM Rectangular Constellation by Natural and 2D Gray Coded

- **Natural binary code of 16QAM**

In the natural binary code of 16QAM, the 2 bits pair at Q channel and I channel are naturally coded. The2 bits pair input, naturally coded and Q channel representative are like in the Table 1. The constellation is like Figure (7.5a). From this figure we see that between two points nearest two bits different is possible. If receiver part makes a mistake in symbol recovery, there will be one or two bit mistake.

If we have a sequence input: 0010, 1000,1111,and 0101. The 2 bits pair outputs of modulator Q channel are 00, 10, 11, and 01. Q channel output is -3sin($2\pi f_c$t), +1sin($2\pi f_c$t), +3sin($2\pi f_c$t), and -1sin($2\pi f_c$t). The 2 bits pair output of I channel are 10, 00, 11, and 01. I channel output are -1cos($2\pi f_c$t), +3cos($2\pi f_c$t), -3cos($2\pi f_c$t), and +1cos($2\pi f_c$t). By using equation (7-1) we will get the output of transmitter is $10^{1/2}$exp($2\pi f_c$t + 247.5^o), $10^{1/2}$exp($2\pi f_c$t + 157.5^o), $18^{1/2}$exp($2\pi f_c$t + 45^o), and $2^{1/2}$exp($2\pi f_c$t + 225^o).

Table 7.1. Comparison of Natural and gray coded channel output

Bits pair input		Naturally coded				Gray Coded			
Q	I	Q	Q channel out put	I	I channel out put	Q	Q channel out put	I	I channel out put
00	00	00	-3 sin($2\pi f_c$t)	00	-3 cos($2\pi f_c$t)	00	-3 sin($2\pi f_c$t)	00	-3 cos($2\pi f_c$t)
01	01	01	-1 sin ($2\pi f_c$t)	01	-1 cos($2\pi f_c$t)	01	-1 sin($2\pi f_c$t)	10	-1 cos($2\pi f_c$t)
10	10	10	+1 sin ($2\pi f_c$t)	10	+1 cos($2\pi f_c$t)	11	+1 sin($2\pi f_c$t)	11	+1 cos($2\pi f_c$t)
11	11	11	+3 sin ($2\pi f_c$t)	11	+3 cos($2\pi f_c$t)	10	+3 sin($2\pi f_c$t)	10	+3 cos($2\pi f_c$t)

- **2D Gray Code 16QAM**

In the 2 dimension (2D) gray code of 16QAM, the gray encoded data of channel Q and channel I then mapped on to 16QAM Rectangular constellation. The 2-bit pair input, gray coded and Q channel representative is like in the Table 1. The constellation is like Figure (1.5b). From this figure we see that between two points nearest is only one bit different. If receiver part make a mistake in symbol recovery, there will only one mistake.

If we have a sequence input: 0010, 1000,1111, and 0101. After gray coding process, the 2 bits pair outputs of modulator Q channel are 00, 11, 10, and 01. Q channel output is -3sin($2\pi f_c$t), +1sin($2\pi f_c$t), +3sin($2\pi f_c$t), and -1sin($2\pi f_c$t). The 2 bits pair outputs of I channel are 11, 00, 10, and 01. I channel outputs are +1cos($2\pi f_c$t), -3cos($2\pi f_c$t), +3cos($2\pi f_c$t), -1cos($2\pi f_c$t). By using equation (7-1) we will get the output of transmitter is $10^{1/2}$exp($2\pi f_c$t + 247.5^o), $10^{1/2}$exp($2\pi f_c$t + 157.5^o), $18^{1/2}$exp($2\pi f_c$t + 45^o), and $2^{1/2}$exp($2\pi f_c$t + 225^o).

- **16QAM as sum of QPSK**

The signal constellation of 16QAM system as sum of QPSK is divided by four quadrants as Figure (7.6). In this system the rule mapping of constellation point in the 16 QAM is determine by the Q bits information:

- If the Q bits information is 00, bits information of I channel are placed same with the QPSK system and the position at quadrant I.
- If the value of Q bits information are 01, I bits information are placed at quadrant II and rotated $\pi/2$ radiant from their QPSK original position.
- If the value of Q bits information are 11, I bits information are placed at quadrant III and rotated π radiant from their QPSK original position.
- If the value of Q bits information are 10, I bits information are placed at quadrant IV and rotated $-\pi/2$ radiant from their QPSK original position.

If we have sequence input: 0010, 1000,1111,0101. The 2 bits pair outputs of Q channel are 00 with the signal point at quadrant I, 11 with the signal point at quadrant III, 10 with the signal point at quadrant IV, and 01 with the signal point at quadrant II. The 2 bits pair outputs of I channel are 11 not rotate from QPSK original position, 00 rotated π radiant from QPSK original position, 10 rotated $-\pi/2$ radiant from original QPSK position, and 01 rotated $\pi/2$ radiant from original QPSK position. The2 bits pair input, naturally coded and Q channel representative are like in the Table 2.

Table 2. Bit input and channel output of 16QAM as sum of QPSK

Bits pair input		**16QAM as sum of QPSK**			
Q	I	Q	Q channel out put	I	I channel out put
00	00	00	$+3\sin(2\pi f_c t)$	00	$+3\cos(2\pi f_c t)$
00	01	00	$+3\sin(2\pi f_c t)$	01	$+1\cos(2\pi f_c t)$
00	10	00	$+1\sin(2\pi f_c t)$	11	$+1\cos(2\pi f_c t)$
00	11	00	$+1\sin(2\pi f_c t)$	10	$+3\cos(2\pi f_c t)$
01	00	01	$+3\sin(2\pi f_c t)$	00	$-3\cos(2\pi f_c t)$
01	01	01	$+1\sin(2\pi f_c t)$	01	$-3\cos(2\pi f_c t)$
01	10	01	$+1\sin(2\pi f_c t)$	11	$-1\cos(2\pi f_c t)$
01	11	01	$+3\sin(2\pi f_c t)$	10	$-1\cos(2\pi f_c t)$
10	00	11	$-3\sin(2\pi f_c t)$	00	$-3\cos(2\pi f_c t)$
10	01	11	$-3\sin(2\pi f_c t)$	01	$-1\cos(2\pi f_c t)$
10	10	11	$-1\sin(2\pi f_c t)$	11	$-1\cos(2\pi f_c t)$
10	11	11	$-1\sin(2\pi f_c t)$	10	$-3\cos(2\pi f_c t)$
11	00	10	$-3\sin(2\pi f_c t)$	00	$+3\cos(2\pi f_c t)$

11	01	10	$-1\ \sin(2\pi f_c t)$	01	$+3\ \cos(2\pi f_c t)$
11	10	10	$-1\ \sin(2\pi f_c t)$	11	$+1\ \cos(2\pi f_c t)$

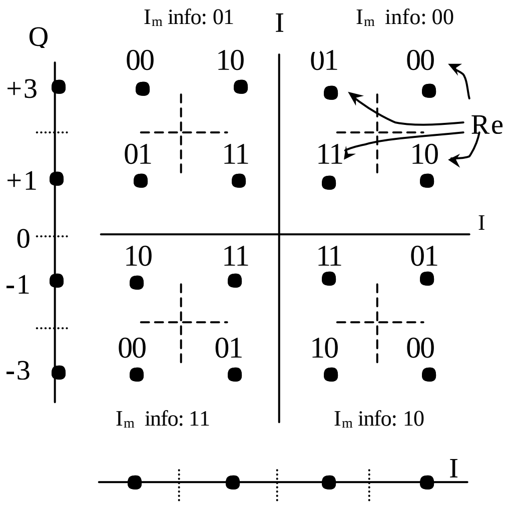


Figure (7.6). The 16QAM sum of QPSK

11	11	10	$-3\ \sin(2\pi f_c t)$	10	$+1\ \cos(2\pi f_c t)$

Based on the constellation signal as Figure (7.6) and Table 2, we will know that the outputs of modulator Q channel are $+1\sin(2\pi f_c t)$ $-3\sin(2\pi f_c t)$, $-3\sin(2\pi f_c t)$, and $+1\sin(2\pi f_c t)$. The outputs of modulator I channel are $+1\cos(2\pi f_c t)$, $-3\cos(2\pi f_c t)$, $+1\cos(2\pi f_c t)$, and $-1\cos(2\pi f_c t)$. By using equation (1-1) we will get the output of transmitter is $2^{1/2}\exp(2\pi f_c t + 45^o)$, $18^{1/2}\exp(2\pi f_c t + 225^o)$, $2^{1/2}\exp(2\pi f_c t - 67.5^o)$, and $2^{1/2}\exp(2\pi f_c t + 112.5^o)$.

- **16QAM as sum of DQPSK**

The signal constellation of 16QAM system as sum of DQPSK is similar with the constellation of 16QAM system as sum of QPSK. But in this system the rule of rotation is based DQPSK system. The signal point at present time is a function of the signal point at the previous time. In this system have two steps the transition point. First step is transition point for Q channel, the second step is transition for I channel.

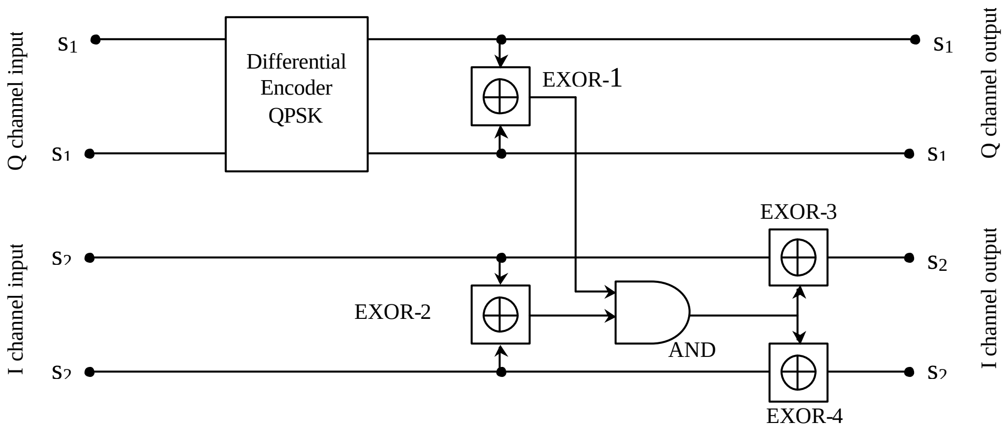


Figure (7.7) Logic circuit mapping of 16QAM as sum of DQPSK

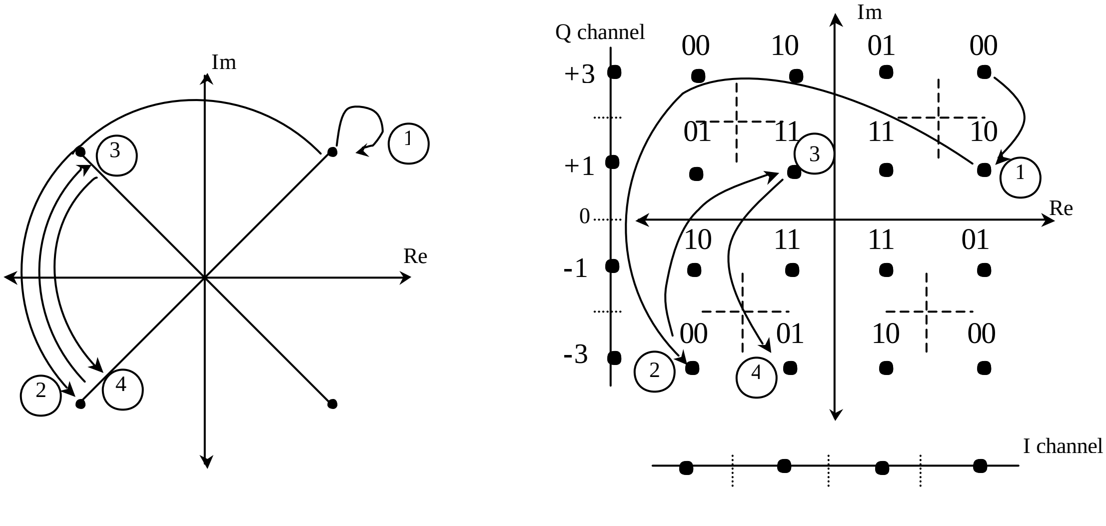
By using the rule transition of DQPSK system, mapping of constellation point in the 16 QAM of Q bits information is:

- If the Q bits information are 00, there is no transition.
- If the value of Q bits information is 01, is rotated $\pi/2$ radiant from previous position.
- If the value of Q bits information is 11, is rotated π radiant from previous position.
- If the value of Q bits information are 10, is rotated - π /2 radiant from previous position.

For initial condition is made by assumption that 2 bits at Q channel are s_{11_0} = 0 and s_{11_0} = 00.

The transition point for I channel based on the logic circuit at Figure (7.7). The output of I channel is a function of the output from Differential Encoding of Q channel and bits information input of I channel.

If we have a sequence input: 0010, 1000,1111,0101. The 2 bits pair inputs of Q channel after gray coding are 00, 11, 10, and 01. The Q channel output for after differential-encoding process is 00, 11, 01, and 11. I channel inputs after gray coding are 10, 00, 11, and 01. Based on Figure (1.7) we will know that the output of I channel are 10, 00, 11, and 01.



a) Rotation process based on DQPSK

b) Transition process at 16QAM sum of DQPSK

Figure (7.8). The 16QAM as sum of DQPSK

Based on the constellation signal as Figure (7.8b), the outputs of modulator Q channel are $+1\sin(2\pi f_c t)$, $-3\sin(2\pi f_c t)$, $+1\sin(2\pi f_c t)$, and $-3\sin(2\pi f_c t)$. The outputs of modulator I channel are $+3\cos(2\pi f_c t)$, $-3\cos(2\pi f_c t)$, $-1\cos(2\pi f_c t)$, and $-1\cos(2\pi f_c t)$. By using equation (1-1) we will get the output of transmitter is $10^{1/2}\exp(2\pi f_c t + 22.5^o)$, $18^{1/2}\exp(2\pi f_c t + 225^o)$, $2^{1/2}\exp(2\pi f_c t + 135^o)$, and $10^{1/2}\exp(2\pi f_c t + 247.5^o)$.

7.1.3. The Receiver of 16QAM

The receiver of 16QAM is similar with the QPSK system, but in this system each channel content of two bits information. Generally the block diagram of 16QAM receiver is like Figure (7.9). Same with transmitter part, the different among them is at de-mapping point of constellation signal. At this section we make an assumption that the local carrier oscillator exactly synchronized, and phase and frequency of local carrier of receiver part is same with which used at transmitter part.

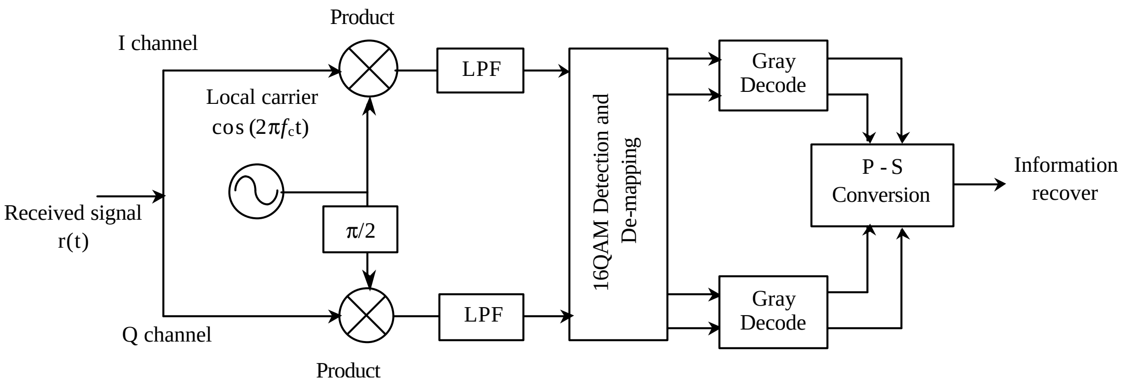


Figure (7.9) The 16QAM Receiver

After filtering process by using LPF, the PAM signal of each channel is detected to make a decision of their levels. The next process is de-mapping, and these stages depend on the mapping system that used by transmitter part. If the mapping system in transmitter part using natural binary code, the de-mapping at receiver must use natural binary decode, so with 2 D gray code, sum of QPSK and sum of DQPSK de-mapping system.

- **Natural Binary Decode**

  In this technique Gray Decode part is not used. The PAM signal output from LPF is decode by using Natural binary code. If the received signal is $10^{1/2}\exp(2\pi f_c t + 247.5^o)$, $10^{1/2}\exp(2\pi f_c t + 157.5^o)$, $18^{1/2}\exp(2\pi f_c t + 45^o)$, and $2^{1/2}\exp(2\pi f_c t + 225^o)$. After demodulation and LPF process the output from Q channel output are -3, +1, +3, and -1. Normalizing the level will change the Q channel output be 0, 2, 3, and 1. By using binary decode we will get the output as 00, 10, 11 and 01. At I channel demodulation and LPF process give the output at I channel as are +1, -3, +3, and -1. Normalizing the level will change the Q channel output be 2, 0, 3, and 1. By using binary decode we will get the output as 10, 00, 11 and 01. By using parallel to serial conversion the output of bit recovery is 0010, 1000,1111, and 0101.

- **2 D gray Decode**

In this technique similar with Natural binary code, but in this technique use Gray Decode. After the process in demodulator and LPF, the PAM signal output is decoded by using gray decode. If the received signal $10^{1/2}\exp(2\pi f_c t + 247.5^o)$, $10^{1/2}\exp(2\pi f_c t + 157.5^o)$, $18^{1/2}\exp(2\pi f_c t + 45^o)$, and $2^{1/2}\exp(2\pi f_c t + 225^o)$.

After demodulation and LPF process the output from Q channel output is -3, +1, +3, and -1. Normalizing the level will change the Q channel output be 0, 2, 3, and 1. De-mapping process by using 2D gray will give the output at Q channel as 00, 11, 10 and 01. By using gray decode we will get the output will be 00, 10, 11, and 01. At I channel demodulation and LPF process give the output at I channel as are +1, -3, +3, and –1. Normalizing the level will change I channel output be 2, 0, 3, and 1. De-mapping process by using 2D gray will give the output at I channel as 11, 00, 10 and 01. By using gray decode we will get the output as 10, 00, 11 and 01. After parallel to serial conversion the output of bit recovery are 0010, 1000,1111, and 0101.

- **Sum of QPSK De-mapping**

In this technique after demodulator and LPF process, de-mapping signal is based on QPSK de-mapping. If the received signal is $2^{1/2}\exp(2\pi f_c t + 45^o)$, $18^{1/2}\exp(2\pi f_c t + 225^o)$, $2^{1/2}\exp(2\pi f_c t - 67.5^o)$, and $2^{1/2}\exp(2\pi f_c t + 112.5^o)$. After demodulation and LPF at channel, the output of Q channel is +1, -3, -3, and +1. At I channel the output is +1, -3, +1, and -1.

Based on the Figure (7.6) we see that these signal are located at quadrant I, III, IV, and II respectively. Based on this position we know that the Q channel output are 00, 11, 10, and 01.. By using gray decode we will get the output of Q channel are 00, 10, 11, and 01. From constellation signal at Figure (7.6) we see that I channel has bit output as 11, 00, 10, and 01. By using gray decode I channel output will be 10, 00, 11, and 01. By using parallel to serial conversion the output of bit recovery are 0010, 1000,1111, and 0101.

- **Sum of DQPSK De-mapping**

In this technique after demodulator and LPF process, de-mapping signal is based on DQPSK de-mapping. If the received signal is $10^{1/2}$exp ($2\pi f_c$t + 22.5°), $18^{1/2}$exp ($2\pi f_c$t + 225°), $2^{1/2}$exp ($2\pi f_c$t + 135°), and $10^{1/2}$exp ($2\pi f_c$t + 247.5°). After demodulation and LPF at channel, the output of Q channel is +1, -3, +1, and -3. At I channel the output is +3, -3, -1, and -1. Based on the Figure (7.8) we see that the bits of Q channel are 00, 11, 01, and 11. And the bits of I channel are 10, 00, 11, and 01. By using differential decoding process as Figure (7.10) and bit 00 as initial condition we will get the Q channel is 00, 11, 10, and 01. After gray decodes process it will be 00, 10, 11, and 01. At I channel we will get the bit output as 11, 00, 10, and 01. After gray decodes process it will be 10, 00, 11, and 01. After parallel to serial conversion the output of bit recovery are 0010, 1000,1111, and 0101.

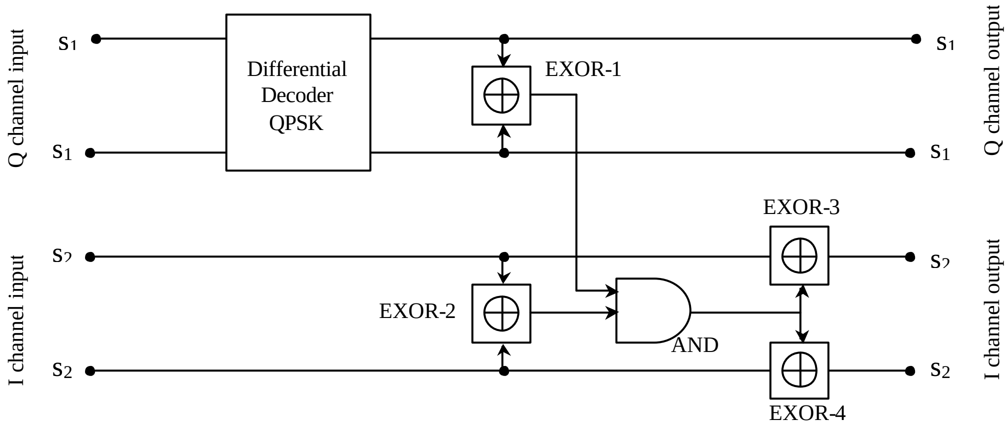


Figure (7.10) Logic circuit de-mapping of 16QAM as sum of DQPSK

7.1.3. 16QAM Transmission in the AWGN Channel

The 16QAM-transmission system is similar with QPSK transmission, but in 16QAM transmission each channel contains of 2 bits or the other hand that energy average each channel of 16QAM l is twice compare than in the QPSK channel. The simplification for it can be described as Figure (7.11).

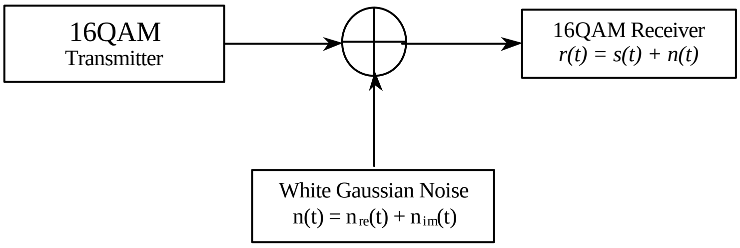


Figure (7.11) 16QAM transmission through AWGN channel

In the AWGN channel, each channel of 16QAM (In-phase and Quadrature or in the some books reference is called as Real and Imaginary) will disturbed by two independent additive noise which Gaussian random.

The signal transmission of 16QAM system is:

$$s(t) = Re(t) + Im(t)$$

Where Re(t) is represent the I channel of 16QAM and Im(t) is represent the Q channel of 16QAM. The noise generate from AWGN channel is:

$$n(t) = n_{re}(t) + n_{im}(t)$$

The addition process in the channel will make the received signal at receiver part as

$$r(t) = s(t) + n(t)$$

$$= [Re(t) + n_{re}(t)] + [Im(t) + n_{im}(t)] \quad (1\text{-}5)$$

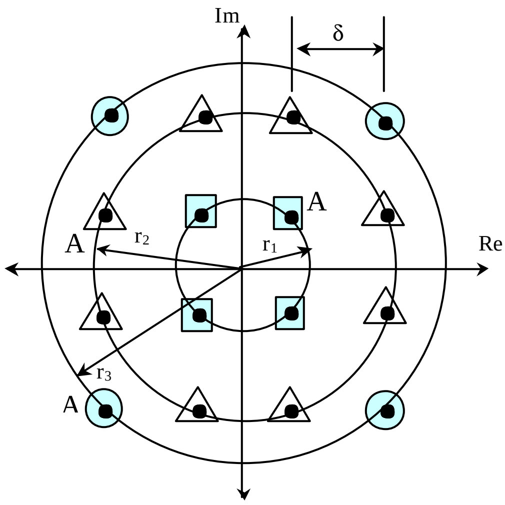


Figure (7.12). Three groups signal of 16QAM rectangular

To evaluate the performance of 16QAM in the AWGN channel, it must consider its constellation signal. By modification Figure (7.2) we will get three groups with different energy average. The distance between two nearest signals is δ, in this case is represent the energy different between two nearest point signals.

- For group A_1 with $r_1 = \sqrt{2}d/2$, the energy average each signal equivalent with $r_1^2 = \delta^2/2$.
- For group A_2 with $r_2 = \sqrt{5/2}d$, the energy average each signal equivalent with $r_2^2 = 5\delta^2/2$.
- For group A_2 with $r_2 = \sqrt{2}(3/2)\ \delta^2$, the energy average each signal equivalent with $r_2^2 =$

$9\delta^2/2$.

The total energy for 16-signal point of 16QAM rectangular constellation is:

$$E_{tot} = (4 \text{ x } \delta^2/2) + (8 \text{ x } 5\delta^2/2) + (4 \text{ x } 9\delta^2/2). \tag{7-6}$$

The average energy of each signal is:

$$E_s = [(4 \text{ x } \delta^2/2) + (8 \text{ x } 5\delta^2/2) + (4 \text{ x } 9\delta^2/2)]/16 = 5\,\delta^2/2 \tag{7-7}$$

The AWGN channel gives noise, which has a variance σ^2. Average energy to noise ratio of the received signal at receiver is:

$$E_s/No = (5/2)\ \delta^2/\sigma^2 \tag{7-8}$$

And bit energy to noise ratio is:

$$E_b/No = (5/8)\ \delta^2/\sigma^2 \tag{7-9}$$

Performance of 16QAM system in the AWGN channel in the case of symbol error can be evaluated as a function of error function of Energy to noise ratio, and expressed as:

$$P_s = 2\frac{\sqrt{M}-1}{\sqrt{M}} erfc\left(\sqrt{\frac{3}{2(M-1)}\frac{E_s}{No}}\right) = 2\frac{\sqrt{16}-1}{\sqrt{16}} erfc\left(\sqrt{\frac{3}{2(16-1)}\frac{4E_b}{No}}\right)$$
$$= \frac{3}{2} erfc\left(\sqrt{\frac{2}{5}\frac{E_b}{No}}\right) \tag{7-10}$$

By include the value in equation (7-9) we will get the value of symbol error probability as:

$$P_s = \frac{3}{2} erfc\left(\sqrt{\frac{2}{5}\frac{5}{8}\frac{d^2}{s^2}}\right) \tag{7-11}$$

And the bit error probability is

$$P_b = \frac{3}{8} erfc\left(\sqrt{\frac{2}{5}\frac{5}{8}\frac{d^2}{s^2}}\right) \tag{7-12}$$

The symbol error probability performance of 16QAM system in the AWGN channel as a function of energy per bit to noise ratio can be described as Figure (7.13).

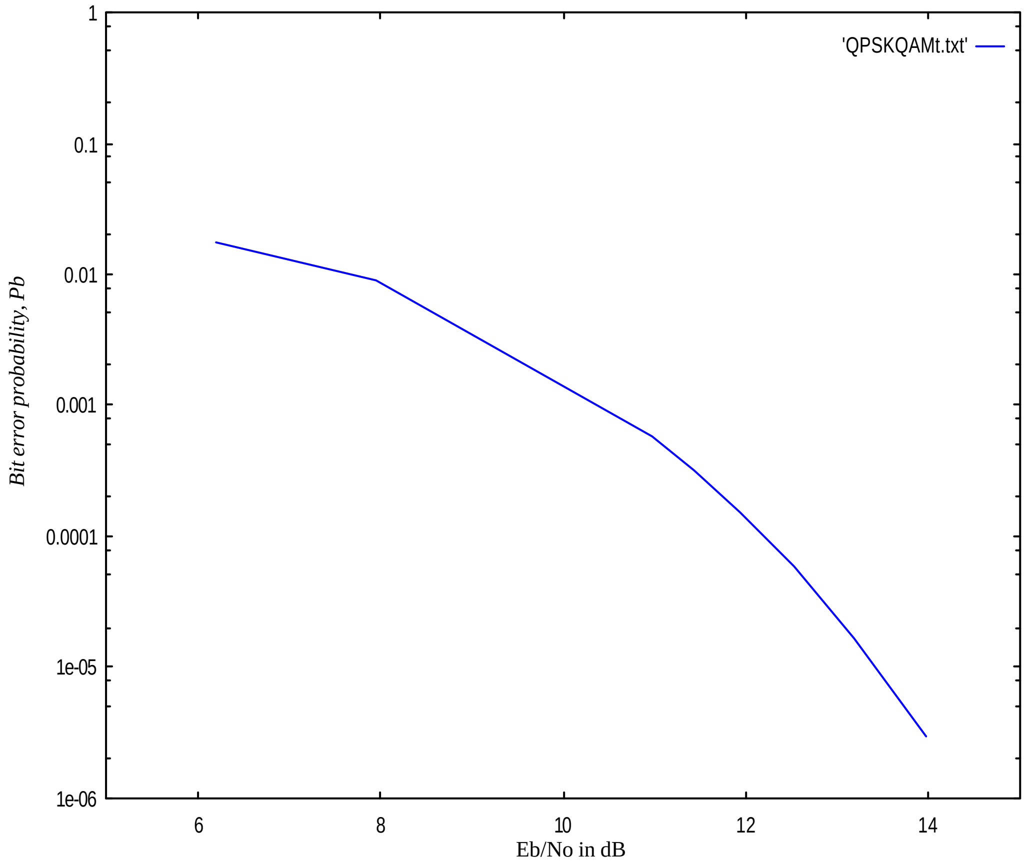


Figure (7.13). Performance of 16 QAM rectangular constellation in the AWWGN channel

7.1.4. Carrier-phase Synchronization of 16QAM System

For the real system the frequency and initial phase of transmitter part is unknown by the receiver part. At the receiver part must be generated a local carrier with initial phase estimated. The system will work after the frequency and phase of receiver part has been synchronized to the frequency and phase of transmitter part. The process synchronization then called as carrier-phase recovery. In this example we use Costas Loop as carrier-phase recovery for synchronization process. For it we denote the frequency carrier at transmitter as f_{cT}, and local carrier frequency generated by receiver as f_{cR}. The initial phase of transmitter signal as θ_i, and the phase estimate at receiver part as θ_{est}.

Carrier-phase recovery by using Costas Loop for 16QAM is a combination from two parallel carrier-phase recoveries, which we have applied it for QPSK system. But in this case we must consider the amplitude of received signal. General descriptions of Costas loop carrier recovery for 16QAM in the base band term as the Figure (7.14). There is two parts in the carrier recovery for 16 QAM, carrier-phase detector and amplitude detector.

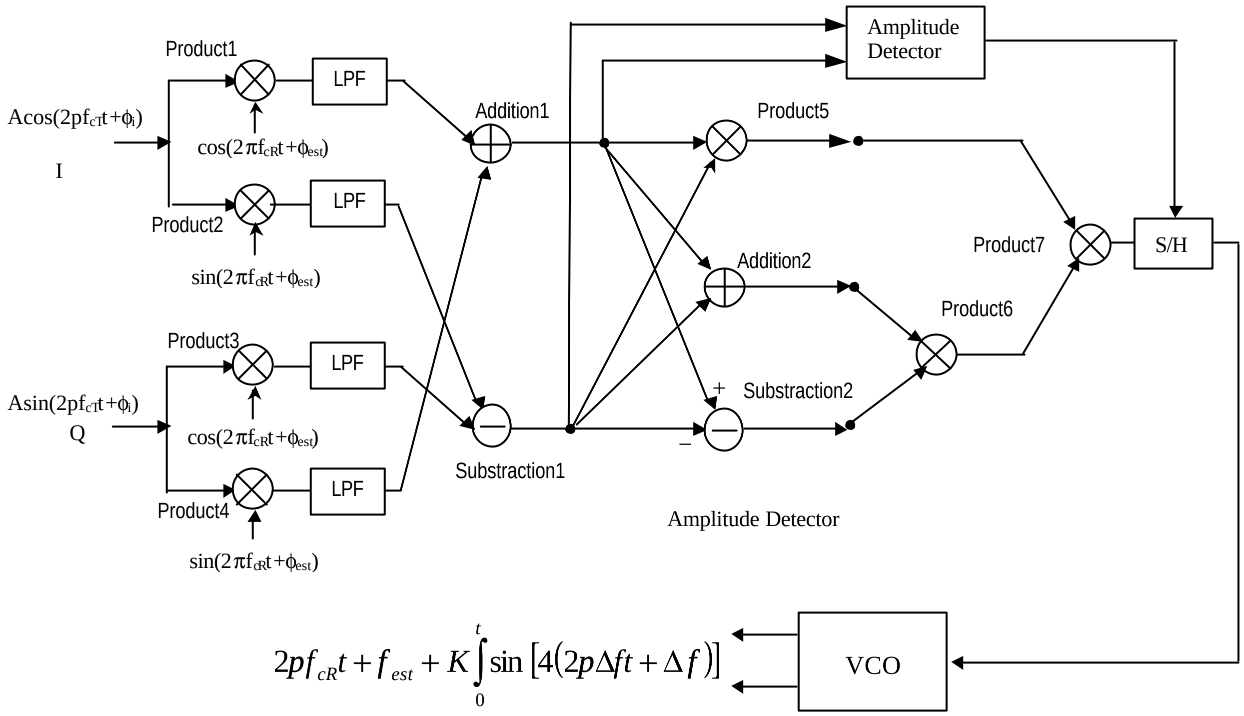


Figure (7.14) Simplification of carrier recovery for QPSK system

- **Carrier-phase detector**

In QPSK system the transmission signal is content of Real (Re) and Imaginary (Im) part:

$$s(t) = A_{Re}\cos(2\pi f_{cT}t + \phi_i) + A_{Im}\cos(2\pi f_{cT}t + \phi_i).$$

In this case the average value of A_{Re} and A_{Im} is 1. The synchronization process by using Costas Loop is as Figure (7.14). In synchronization process for 16QAM each channel is multiply by $\cos(2\pi f_{cR}t + \phi_{est})$ and $\sin(2\pi f_{cR}t + \phi_{est})$

- The output of Product 1 is:

$y_{p1}(t) = \cos(2\pi f_{cT}t + \phi_i)\cos(2\pi f_{cR}t + \phi_{est}) + n(t)_{Re}\cos(2\pi f_{cR}t + \phi_{est})$

$= (1/2)\cos\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} + (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$

$+ n_{Re}(t)\cos(2\pi f_{cR}t + \phi_{est})$

The low pass filter process will give the output as:

$y_{LPF1}(t) = (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\} = (1/2)\cos(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Product 2 is:

$y_{p1}(t) = \cos(2\pi f_{cT}t + \phi_i)\,(\sin(2\pi f_{cR}t + \phi_{est})) + n(t)_{Re}\sin(2\pi f_{cR}t + \phi_{est})$

$= (1/2)\sin\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} - (1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$

$+ n_{Re}(t)\sin(2\pi f_{cR}t + \phi_{est})$

The low pass filter process will give the output as:

$y_{LPF2}(t) = (-1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}= (-1/2)\sin(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Product 3 is:

$y_{p3}(t) = \sin(2\pi f_{cT}t + \phi_i)\cos(2\pi f_{cR}t + \phi_{est}) + n(t)_{Im}\cos(2\pi f_{cR}t + \phi_{est})$

$= (-1/2)\sin\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} + (1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$

$+ n_{Im}(t)\cos(2\pi f_{cR}t + \phi_{est})$

The low pass filter process will give the output as:

$y_{LPF3}(t) = (1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}= (1/2)\sin(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Product 4 is:

$y_{p4}(t) = \sin(2\pi f_{cT}t + \phi_i)\sin(2\pi f_{cR}t + \phi_{est}) + n(t)_{Im}\sin(2\pi f_{cR}t + \phi_{est})$

$= -(1/2)\cos\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} + (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$

$+ n_{Im}(t)\sin(2\pi f_{cR}t + \phi_{est})$

The low pass filter process will give the output as:

$y_{LPF4}(t) = (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}= (1/2)\cos(2\pi\,\Delta f\,t + \Delta\phi_t)$

The next steps are

- The output of Addition1 is:

$add1(t) = (1/2)\cos(2\pi\,\Delta f\,t + \Delta\phi_t)+(1/2)\cos(2\pi\,\Delta f\,t + \Delta\phi_t) = \cos(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Substraction1 is:

$subst1(t) = (1/2)\sin(2\pi\,\Delta f\,t + \Delta\phi_t)-(-1/2)\sin(2\pi\,\Delta f\,t + \Delta\phi_t) = \sin(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Product5 is:

$y_{p5}(t) = \cos(2\pi\,\Delta f\,t + \Delta\phi_t)\sin(2\pi\,\Delta f\,t + \Delta\phi_t) = (1/2)\sin\{2(2\pi\,\Delta f\,t + \Delta\phi_t)\}$

- The output of addition2 is:

$add2(t) = \cos(2\pi\,\Delta f\,t + \Delta\phi_t) + \sin(2\pi\,\Delta f\,t + \Delta\phi_t)$

- The output of Subtraction2 is:

Sub2 = $\cos(2\pi\, \Delta f\, t + \Delta\phi_t) - \sin(2\pi\, \Delta f\, t + \Delta\phi_t)$

- The output of Product6 is:

$y_{p6}(t) = [\cos(2\pi\, \Delta f\, t + \Delta\phi_t) + \sin(2\pi\, \Delta f\, t + \Delta\phi_t)][\cos(2\pi\, \Delta f\, t + \Delta\phi_t) - \sin(2\pi\, \Delta f\, t + \Delta\phi_t)]$

$= \cos^2(2\pi\, \Delta f\, t + \Delta\phi_t) - \sin^2(2\pi\, \Delta f\, t + \Delta\phi_t)$

$= \cos\{2(2\pi\, \Delta f\, t + \Delta\phi_t)\}$

- The output of Product7 or the error loop signal is:

$e(t) = (1/2) \sin\{2(2\pi\, \Delta f\, t + \Delta\phi_t)\}\cos\{2(2\pi\, \Delta f\, t + \Delta\phi_t)\}$

$=(1/4)[2 \sin\{2(2\pi\, \Delta f\, t + \Delta\phi_t)\}\cos\{2(2\pi\, \Delta f\, t + \Delta\phi_t)\}]$

$=(1/4) \sin\{4(2\pi\, \Delta f\, t + \Delta\phi_t)\}$ (7-13)

It is known as error signal into the loop filter of QPSK carrier-phase recovery, and used to drive the VCO.

The new output phase of VCO is given as

$$2pf_{cR}t + f_{est} + \int_0^t K_c e(t)dt$$

or (7-14)

$$2pf_{cR}t + f_{est} + K\int_0^t \sin\left(4(2p\Delta ft + \Delta f)\right)dt$$ where is K = Kc/4.

The loop phase error and its time derivative is

$$q_e(t) = \left(2pf_{cT}t + f_i\right) - \left(2pf_{cR}t + f_{est} + K\int_0^t \sin\left(4(2p\Delta ft + \Delta f)\right)dt\right)$$

$$\frac{d\left(q_e(t)\right)}{dt} = \left(2pf_{cT} - 2pf_{cR}\right) - K\sin\left(4(2p\Delta ft + \Delta f)\right)$$

If $f_{cT} = f_{cR}$ and letting K =1, the time derivative of the loop phase error is given by

$$\frac{d\left(q_e(t)\right)}{dt} = -\sin\left(4\Delta f\right) \qquad (7\text{-}15)$$

The null value will be happen at 0,$\pi/2$, π, $3\pi/2$ radiant, and it's known as four phase ambiguity of phase-carrier recovery at 16QAM system.

- **Amplitude detector**

Consider the constellation signal in Figure (7.12). It can be modified as Figure (7.15). In its figure, there is:

- Group A consists of signals that have amplitude value: amplitude< or amplitude > 14.
- Group B consists of signals that have amplitude value: 6 $\leq$ amplitude $\leq$ 14.

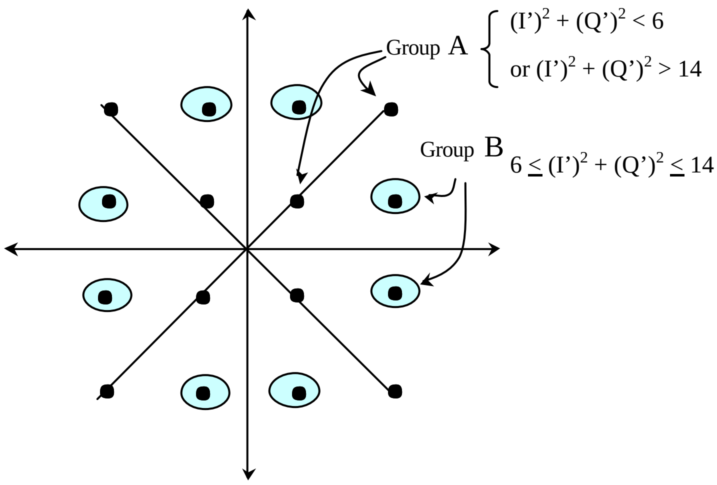


Figure (7.15). Two groups signal

If the received signal is group A, the group detector drive sample and hold (S/H) to make an action for the phase detected by carrier-phase recovery is used and hold its value. In the case where the received signal is group B, the group detector drive S/H to make an action that the phase detected by carrier-phase recovery is unavailable and used the previous phase detected from previous carrier-phase recovery process.

The carrier-phase recovery for 16QAM is only capable to detect the phase of received signal in the group A, but this system can't detect the phase of the received signal in the group B.

7.2. Session 2: Simulation System

We will tray to develop the 16QAM simulation in the base band system and using 2D Gray to develop a rectangular constellation. By using Costas Loop as phase-carrier recovery at the receiver part, we will tray to understand the performances of 16QAM system through the AWGN channel. The description of 16QAM for base band simulation as in the Figure (7.16).

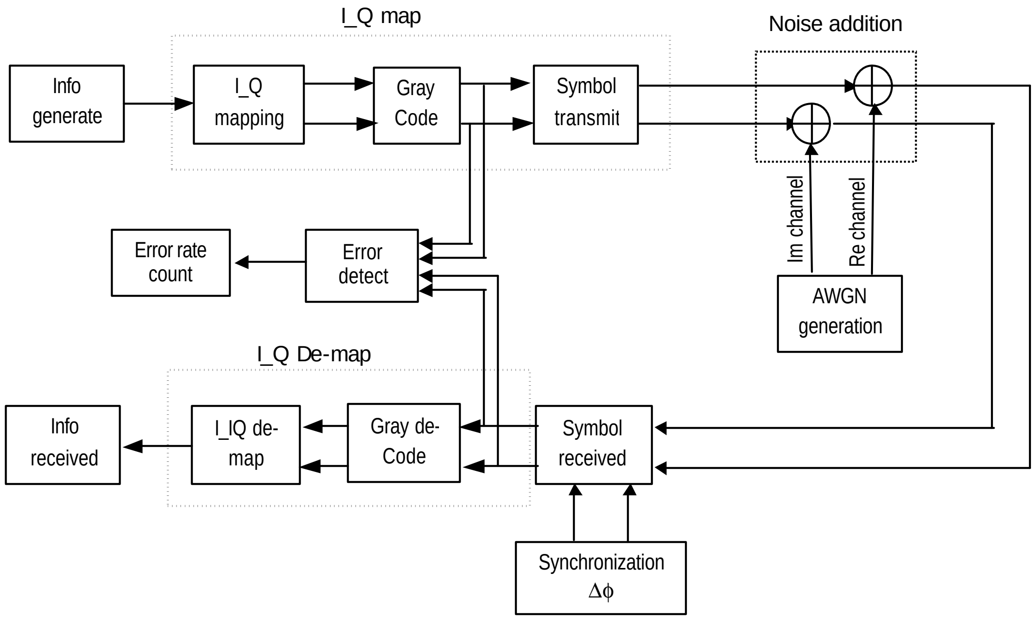


Figure (7.16) Block diagram of 16QAM Base band simulation

7.2.1. Algorithm

The simulation of 16QAM through AWGN channel involving several functions as follows:

- **Info Generator**

Bit information generated randomly by info generator function. The random number generated is set value 0,1,2 ~ 15, as the integer number to represent PAM signal. By using a command *(int)rand%16;* of the Microsoft C, it will generate a number from 0 ~15 uniform randomly.

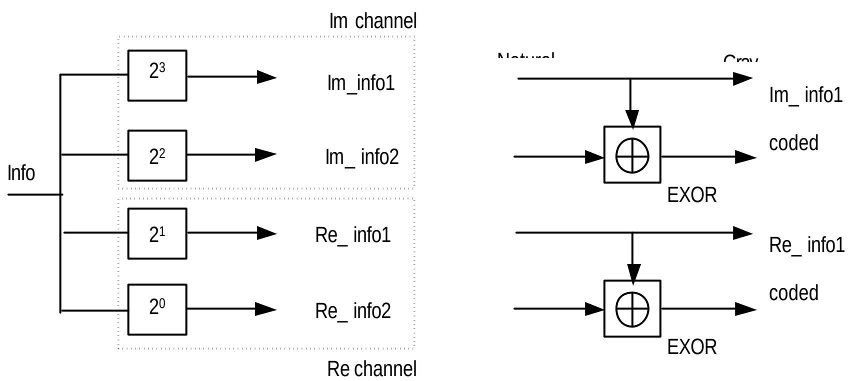
- **I - Q Channel Mapping**

This function consists of three parts in the Figure (7.16): I_Q Map, Gray Code and Symbol Transmit.

- *The I_Q Map*. In this program we denote I channel as Real (Re) channel and Q channel we denote as Imaginary (Im) channel. The information will be mapped into Re and Im channel. The decimal value is converted to binary, the two MSB is mapped into Im channel, and the two LSB is mapped into Re channel. Based on the Figure (7.17a) we will understand that if the information value is 0 the two bits pair generates is 0000, the Im channel will have the value 00, and Re channel will have the value 00. If the information value is 15 the two bits pair generates is 1111, the Im channel will have the value 11, and Re channel will have the value 11. It can do by using this statement:

```
Im_info1[i]=info[i]/8;
Im_info2[i]=(info[i]-Im_info1[i]*8)/4;
Re_info1[i]=(info[i]-Im_info1[i]*8-Im_info2[i]*4)/2;
Re_info2[i]=(info[i]-Im_info1[i]*8-Im_info2[i]*4-Re_info1[i]*2);
```

- *The Gray Code*. Re channel and Im channel info has natural bit value, based on Table 7.1 and the logic as like in the Figure (7.16b) we able to make Gray code logic. If the value of Re channel is 00, it will coded as 00, if the value is 01 it will coded as 01, if the value is 10 it will coded as 11, and if the value is 11 it will coded as 10. It is same for Im channel. From the above case the value 0000 will be 0000 and the value 1111 will be 1010.
- *The Symbol Transmit*. By using the properties in the Table 7.1 wee can develop the symbol representative for each channel. This part it will convert the value 00 as –3, the value 01 as –1, the value 11 as +1, the value 10 as +3.



a) I_Q Map   b) Gray Code

Figure (7.17) Description of channel mapping

- **Synchronization**

In this function we must make sure that receiver part understand exactly the phase of the received signal from transmitter. Unfortunately in the real system, local carrier frequency and its initial phase does not same with the transmitter part. The Costas loop, as carrier recovery is uses to correct the phase different.

As we know that in the synchronization process by using Costas loop for 16QAM system there two phase ambiguity among 0, $\pi/2$, π, and $3\pi/2$ radiant. Using this property the algorithm looks for one of two possible phases. When it matched we must resolve by de-rotate the phase.

- **AWGN**

To generate AWGN channel we must do it by using two steps: uniform random generate and shifting from uniform to Gaussian distribution.

- *First step* is generates two sequences of uniform random generator, for this purpose we can set variable x1 and x2. Both of them have double data type. These two variables have a value between 0 and 1. To do it we can generate the value x1 and x2 from 0 until default value of RAND_MAX or 32767.00. Then we divide x1 and x2 by RAND_MAX.
- *Second step* do by using Box-Muller method. Two variable x1 and x2, which have the uniform distribution, shift by using formulation:

$re = (s^2\ lnx1)^{1/2}\ cos(2*p*x2)$;*real part*

$im = (s^2\ lnx1)^{1/2}\ sin(2*p*x2)$;*imaginary part.*

- **Noise Addition**

The noise generated from the AWGN function are real and Imaginary part, and independent each one another. In this function we add the Re channel with real noise and the Im channel with imaginary noise. In some books Re channel from 16QAM transmitter is called as real channel, and Im channel is called as imaginary channel. AWGN will destroy the amplitude of signal transmission, and will depredate the level of signal. If the value of AWGN is negative it will make the level of signal transmission is down, other wise if the AWGN value is positive is will make the level of signal transmission will up.

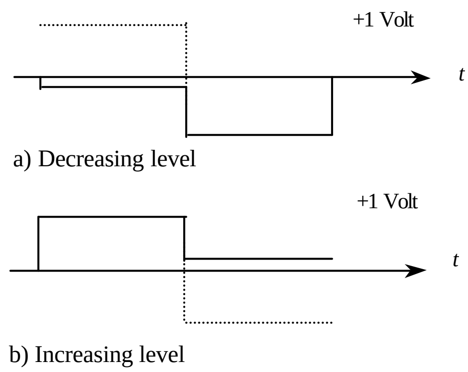


Figure (7.18). The effect AWGN at the level of antipodal signal transmission

The effect of AWGN will change the level of transmitted signal as like in the Figure (7.18).

- **Detection**

This function represents the *Symbol Received* part. Based on the several levels as a threshold, the decision is made:

if signal level < -2.0 the decision is –3.0

if -2.0 ≤ signal level ≤ 0.0 the decision is -1.0

if 0.0 ≤ signal level ≤ 2.0 the decision is +1.0

if signal level ≥ 2.0 the decision is +3.0

This rule for both of Re channel and Im channel signal received, and work independently.

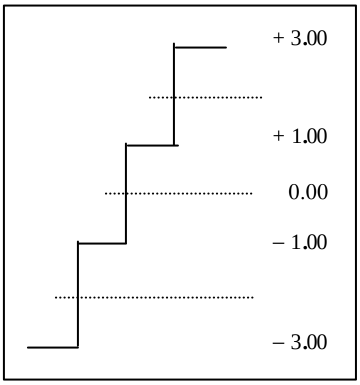
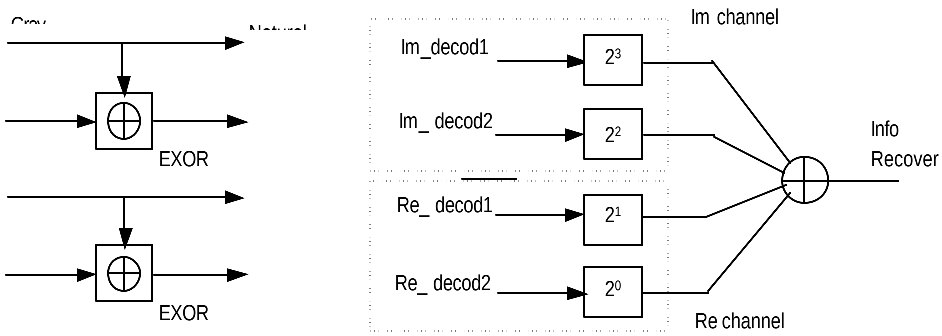


Figure (7.19) Decision symbol for received signal

- **Rx De-map**

This function consists of two parts of block diagram in the Figure (7.20).

  - *Gray de-Code.* First step in this block is converts from antipodal signal to binary signal. By using the mathematical expression opposite with we has used in the symbol transmission. Second step is converts from gray form to natural binary from. By using the XOR logic we get the natural binary result as like Figure (7.20a).
  - *I_Q De-map.* The purpose of this part is get the symbol value by convert the two pair bit (binary) decimal value. By multiply the Im channel by 2^3 and 2^2 and Re channel by 2^1 and 2^0. The next step is addition the multiplication result and we will get the decimal information value.



b) Gray Decode

b) I_Q De-Map

Figure (7.20) Description of Rx De-mapping

- **Error Detect**

By using the output from detection function and compare bit by bit between input signal of symbol transmit to output of symbol receive. If the value between both parts is the error didn't happen, other wise if these value is not same, the error was happening. By combine the result of these result we will get the information of symbol error detection. If one bit or more error happened, the symbol error detect indicate the value 1, if no error both of channel there is no error happened and the value of symbol error detect indicate the value 0.

- **Error Rate**

In this part we sum all of error detected that happened during transmission process. And the result from this summation subtracted by the total bit transmitted. The result is *bit error rate*. If we compare the summation of symbol error detected to total symbol transmitted, the result is *symbol error rate*. This is the real performance resulted by our transmission system.

♦ **Pb_theory**

In this part we calculate the bit error probability of 16QAM system as a function of *energy per bit to noise ratio*. The output is theoretical value of *error probability* performance of 16QAM system. From it we get the information of bit error probability and symbol error probability.

7.2.2. The Listing Program

To develop a program in order the user understand it easily we must set the symbol which represent the parameters in theory:

N = total symbol transmit.

info= information generate by transmitter part.

Im_info1= Imaginary channel 1 information generated at transmitter part

Im_info2= Imaginary channel 2 information generated at transmitter part

Re_info1= Real channel 1 information generated at transmitter part

Re_info2= Real channel 2 information generated at transmitter part

Re_enc1= Real channel 1 information encoded at transmitter part

Re_enc2= Real channel 2 information encoded at transmitter part

Im_enc1= Imaginary channel 1 information encoded at transmitter part

Im_enc2= Imaginary channel 2 information encoded at transmitter part

pi = π radiant.

Re_Tx = symbol generate by transmitter part.

Im_Tx = symbol generate by transmitter part.

pha_Tx = phase generate by transmitter part

phase = phase error of synchronization process

Ave_Pha_err = average phase error in the all the synchronization process

Re_AWGN = real part of noise AWGN

Im_AWGN = imaginary part of noise AWGN

var = noise varians

Re_Rx = Real channel symbol received at receiver part

Im_Rx = Imaginary channel symbol received at receiver part

pha_receiv = phase receive after transmission

Re_1_Rx = Real channel 1 information recovered at receiver part

Re_2_Rx = Real channel 2 information recovered at receiver part

Im_1_Rx = Imaginary channel 1 information recovered at receiver part

Im_2_Rx = Imaginary channel 2 information recovered at receiver part

SNR = Signal to Noise Ratio

BER = bit error rate

Pb = probability of bit error of 16QAM system

Ps = probability of symbol error of 16QAM system

The listing program of 16QAM system is as follows.

```
//Program of 16QAM
//editing by Tri Budi Santoso
//Suzuki Laboratory
//Mobile Communication Group, Tokyo Institute of Technology, Japan

#include<stdio.h>
#include<math.h>
#include<stdlib.h>

#define N 1000000

int nn=N-1,i;
int info[N],Im_info1[N],Im_info2[N],Re_info1[N],Re_info2[N];
int Re_enc1[N],Re_enc2[N],Im_enc1[N],Im_enc2[N];
double Re_Tx[N],Im_Tx[N],pha_Tx[N],Amp[N];
double Im_Rx[N],Re_Rx[N];
double phase[N],Ave_Pha_err,Pha_err,deg_error,pha_i;
int info_Rx[N],Im_1_Rx[N],Im_2_Rx[N],Re_1_Rx[N],Re_2_Rx[N];
double SNR,Ps,Pb,BER,SER,pi=acos(-1.0),var=0.2;
double Re_AWGN[N],Im_AWGN[N];
```

```
void info_generate(int info[],int Im_info1[],int Im_info2[],int Re_info1[],int Re_info2[]);
void encoding(int Im_enc1[],int Im_enc2[],int Re_enc1[],int Re_enc2[]);
void mapping_transmit(double Re_Tx[],double Im_Tx[],double pha_Tx[],double Amp[]);
void costas(double phase[],double *pha_er);
void synchronization(double pha_i,double *deg_er);
void AWGN_channel(double Re_AWGN[],double Im_AWGN[]);
void detection(double Re_Rx[],double Im_Rx[]);
void demapping_recover(int Im_1_Rx[],int Im_2_Rx[],int Re_1_Rx[],int Re_2_Rx[]);
void error_detect(double *ber,double *ser);
void signal_noise(double *snr);
void error_prob(double Ave_Pha_err,double *ps, double *pb);

void main()
{
FILE *fdata;
char filename[]="natural.txt";
fdata = fopen(filename,"a");

 info_generate(info,Im_info1,Im_info2,Re_info1,Re_info2);
 encoding(Im_enc1,Im_enc2,Re_enc1,Re_enc2);
 mapping_transmit(Re_Tx,Im_Tx,pha_Tx,Amp);
 costas(phase,&Ave_Pha_err);printf("\nAve_Pha_err:%f",Ave_Pha_err);
 AWGN_channel(Re_AWGN,Im_AWGN);
 detection(Re_Rx,Im_Rx);
 demapping_recover(Im_1_Rx,Im_2_Rx,Re_1_Rx,Re_2_Rx);
 error_detect(&BER,&SER);
 signal_noise(&SNR);
 error_prob(Ave_Pha_err,&Ps,&Pb);

 printf("\n SNR:%4.2f Pb:%7.6f BER:%7.6f Ps:%7.6f SER:%7.6f",SNR,Pb,BER,Ps,SER);
fprintf(fdata,"%4.2f %7.6f %7.6f",SNR,BER,SER);
fclose(fdata);
}

void info_generate(int info[],int Im_info1[],int Im_info2[],int Re_info1[],int Re_info2[])
{
 for (i=1;i<=nn;++i)
 {
   info[i]=(int)rand()%15;
   Im_info1[i]=info[i]/8;
   Im_info2[i]=(info[i]-Im_info1[i]*8)/4;
   Re_info1[i]=(info[i]-Im_info1[i]*8-Im_info2[i]*4)/2;
   Re_info2[i]=(info[i]-Im_info1[i]*8-Im_info2[i]*4-Re_info1[i]*2);
   //printf("\ninfo[%d]:%d \tIm_info1[i]:%d Im_info2[i]:%d Re_info1[i]:%d
              Re_info2[i]:%d",i,info[i],Im_info1[i],Im_info2[i],Re_info1[i],Re_info2[i]);
 }
}

void encoding(int Im_enc1[],int Im_enc2[],int Re_enc1[],int Re_enc2[])
{
 for (i=1;i<=nn;++i)
 {
   Im_enc1[i]=Im_info1[i];
   Im_enc2[i]=(Im_info1[i] + Im_info2[i])%2;
   Re_enc1[i]=Re_info1[i];
   Re_enc2[i]=(Re_info1[i] + Re_info2[i])%2;
 //printf("\nIm_enc1[i]:%d Im_enc2[i]:%d Re_enc1[i]:%d
```

```
            Re_enc2[i]:%d",Im_enc1[i],Im_enc2[i],Re_enc1[i],Re_enc2[i]);
 }
}

void mapping_transmit(double Re_Tx[],double Im_Tx[],double pha_Tx[],double Amp[])
{
 for (i=1;i<=nn;++i)
 {
  Im_Tx[i]=(Im_info1[i]*2.0+Im_info2[i]*1.0)*2.0-3.0;
  Re_Tx[i]=(Re_info1[i]*2.0+Re_info2[i]*1.0)*2.0-3.0;
  Amp[i] = Re_Tx[i]*Re_Tx[i] + Im_Tx[i]*Im_Tx[i];
  pha_Tx[i]=atan(Im_Tx[i]/Re_Tx[i]);
  if (Re_Tx[i]>0.0 && Im_Tx[i]>0.0)
 {pha_Tx[i]=pha_Tx[i]/pi*180;}
 else if (Re_Tx[i]>0.0 && Im_Tx[i]<0.0)
 {pha_Tx[i]=360.0 + pha_Tx[i]/pi*180;}
 else if (Re_Tx[i]<0.0 && Im_Tx[i]>0.0)
 {pha_Tx[i]=180.0 - pha_Tx[i]/pi*180;}
 else if (Re_Tx[i]<0.0 && Im_Tx[i]<0.0)
 {pha_Tx[i]=180.0 + pha_Tx[i]/pi*180;}
 else
 printf("");
//    printf("\nIm_Tx[%d]:%4.2f Re_Tx[%d]:%4.2f Amp[%d]:%4.2f",i,Im_Tx[i],i,Re_Tx[i],i,Amp[i]);
 }
}

void costas(double phase[],double *pha_er)
{
int input;
 Pha_err=0.0;

 for(i=1;i<=nn;++i)
 {
  pha_i=pha_Tx[i];
  synchronization(pha_i,&deg_error);
  phase[i]=deg_error;
  if(Amp[i]<6.0)
  {input=1;}
  else if(Amp[i]>=6.0 && Amp[i]<=14.0)
  {input=0;}
  else if(Amp[i]>14.0)
  {input=1;}
  else
  printf("");

  if(input==0)
  {phase[i]=phase[i-1];}
  else if(input==1)
  {phase[i]=phase[i];}
  else
  printf("");

  //printf("\n phase[%d]:%f phase[%d]:%f",i-1,phase[i-1],i,phase[i]);

  Pha_err += fabs(phase[i]);
 }
 Ave_Pha_err=Pha_err/nn;
```

```
  *pha_er=Ave_Pha_err; //printf("\nAve_Pha_err:%f",Ave_Pha_err);
}

void synchronization(double pha_i,double *deg_er)
{
 int d=1,i=1,symbol_rate=16000,phase_uni;
 double rad_error,fc=800e+6,T,delta_fc,phase_estim;
 double VCO,K=1.0;

 T=1.0/(double)symbol_rate;
 delta_fc=1e-6 * fc *T;
 do
 {
  phase_uni=(int)rand()%360;
  phase_estim=(double)phase_uni;
  deg_error=(delta_fc + pha_i - phase_estim);
  //printf("\nphase_estim:%f deg_error:%f ",phase_estim,deg_error);
   do
   {
    deg_error=(delta_fc + pha_i - phase_estim);
    rad_error=deg_error/360*2*pi;
    VCO=K*sin(4*rad_error);
    phase_estim = phase_estim + VCO;
    i++;
   }while(i<=10);
  d++;
 } while(fabs(deg_error)>=4.0);
 *deg_er=deg_error;
}

void AWGN_channel(double Re_AWGN[],double Im_AWGN[])
{
 double x1,x2;
 for(i=1;i<=nn;++i)
 {
   x1=(double)rand()/(RAND_MAX+1.0);
   if (x1<=1e-38)
   {x1=1e-38;}
   x2=(double)rand()/(RAND_MAX+1.0);
   Re_AWGN[i]=sqrt(-2.0*var*log(x1))*cos(2*pi*x2);
   Im_AWGN[i]=sqrt(-2.0*var*log(x1))*sin(2*pi*x2);
   //printf("Re_AWGN[i]:%f ,Re_AWGN[i]:%f \n",Re_AWGN[i],Im_AWGN[i]);
 }
}

void detection(double Re_Rx[],double Im_Rx[])
{
double costas_effect=0.0;
   for (i=1;i<=nn;++i)
   {
    costas_effect=cos(phase[i]/360*2*pi);
    Im_Rx[i] = Im_Tx[i]*costas_effect + 1.75*Im_AWGN[i];
   Re_Rx[i] = Re_Tx[i]*costas_effect + 1.75*Re_AWGN[i];
//  printf("\nIm_Rx[i]:%f \tRe_Rx[i]:%f costas_effect:%f",Im_Rx[i],Re_Rx[i],costas_effect);
   }
}
```

```
void demapping_recover(int Im_1_Rx[],int Im_2_Rx[],int Re_1_Rx[],int Re_2_Rx[])
{
   for (i=1;i<=nn;++i)
   {
   if(Re_Rx[i]<-2.0)
  {Re_1_Rx[i]=0;Re_2_Rx[i]=0;}
  else if(Re_Rx[i]>=-2.0 && Re_Rx[i]<0.0)
  {Re_1_Rx[i]=0;Re_2_Rx[i]=1;}
  else if(Re_Rx[i]>=0.0 && Re_Rx[i]<2.0)
  {Re_1_Rx[i]=1;Re_2_Rx[i]=1;}
  else if(Re_Rx[i]>2.0)
  {Re_1_Rx[i]=1;Re_2_Rx[i]=0;}
  else
  printf("");

  if(Im_Rx[i]<-2.0)
  {Im_1_Rx[i]=0;Im_2_Rx[i]=0;}
  else if(Im_Rx[i]>=-2.0 && Im_Rx[i]<0.0)
  {Im_1_Rx[i]=0;Im_2_Rx[i]=1;}
  else if(Im_Rx[i]>=0.0 && Im_Rx[i]<2.0)
  {Im_1_Rx[i]=1;Im_2_Rx[i]=1;}
  else if(Im_Rx[i]>2.0)
  {Im_1_Rx[i]=1;Im_2_Rx[i]=0;}
  else
  printf("");
    // printf("\nIm_1_Rx[i]:%d Im_2_Rx[i]:%d Re_1_Rx[i]:%d Re_2_Rx[i]:%d "
                  ,Im_1_Rx[i],Im_2_Rx[i],Re_1_Rx[i],Re_2_Rx[i]);
   }
}

void error_detect(double *ber,double *ser)
{
  double BER1=0.0,BER2=0.0,BER3=0.0,BER4=0.0,BER_i,SER_i;
   for (i=1;i<=nn;++i)
   {
   // BER_i=0.0;SER_i=0.0;
   if(Re_1_Rx[i]==Re_enc1[i])
   BER1=0.0;
   else
   BER1=1.0;
   if(Re_2_Rx[i]==Re_enc2[i])
   BER2=0.0;
   else
   BER2=1.0;
   if(Im_1_Rx[i]==Im_enc1[i])
   BER3=0.0;
   else
   BER3=1.0;
   if(Im_2_Rx[i]==Im_enc2[i])
   BER4=0.0;
   else
   BER4=1.0;

     BER_i=(BER1 + BER2 + BER3 + BER4);
    BER += BER_i;
    if (BER_i>0.0)
```

```
    SER_i=1.0;
   else
   SER_i=0.0;
   SER +=SER_i;
   }
 BER=BER/(4.0*nn);
 SER=SER/nn;
 *ber=BER;
 *ser=SER;// printf("\nBER:%f SER:%f",BER,SER);
}

void signal_noise(double *snr)
{
 double gama;
 gama=(5./8.0)*2.0/var;
 SNR=10*log10(gama);
 *snr=SNR;
}

void error_prob(double Ave_Pha_err,double *ps, double *pb)
{
double gama,x,term_1,term_3,term_5;
gama=Ave_Pha_err*(5.0/8.0)/var;
x=sqrt(0.4*gama);
term_1=1/(2*pow(x,2));
term_3=(1*3)/(2*2*pow(x,4));
term_5=(1*3*5)/(2*2*2*pow(x,6));
Ps=3.0*(exp(-(x*x))/(x*sqrt(pi)))*(1-term_1+term_3-term_5);
*ps=Ps;
Pb=Ps/4.0;
*pb=Pb;
}
```

7. 2.3. Simulation Result

The simulation result for the 16QAM system is like in the Table 7.3.

Table 7.3. Simulation result of 16QAM in the AWGN channel

Eb/No	Pb	BER
6.20	0.01748865	0.03369800
7.96	0.00878699	0.01263550
10.97	0.00058334	0.00078775
11.43	0.00032042	0.00044950
11.94	0.00015218	0.00022350
12.52	0.00005879	0.00009800
13.19	0.00001669	0.00002800
13.98	0.00000290	0.00001175

From the Table 7.3 we can see that the different value between probability of error and bit error rate (BER) as simulation result. As example for value of Eb/No = 6.02, probability of error value is 0.01748865. But the simulation result is 0.03369800. In this case the bit error rate happen is higher than the theoretical value (probability of error).

For higher Eb/No, the value of bit error rate as simulation result is still higher than the value of probability of error. The example is at Eb/No = 13.98 the value of probability of error is 0.00000290 and the value of bit error rate is 0.00001175.

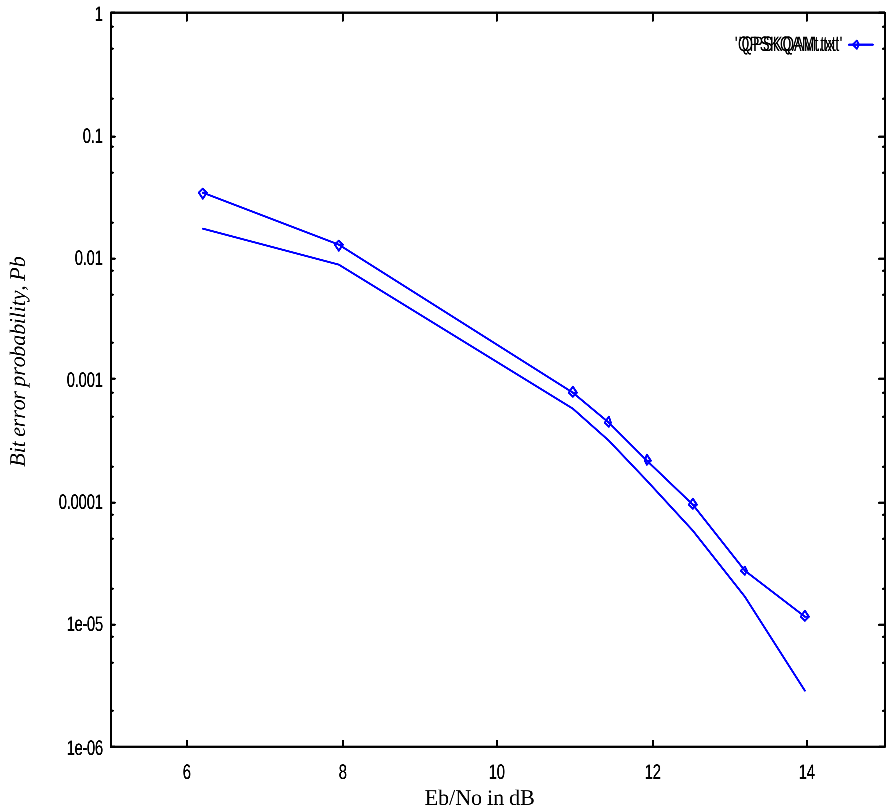


Figure (7.13). The simulation result of 16 QAM rectangular constellation in the AWGN channel