Chapter 5

Lecture Note Developing on Simulation of QPSK System 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: QPSK Transmission System

5.1. Session 1: Review of QPSK

The QPSK is very similar with BPSK, but in the QPSK the signals have one of the four possible phases as a function of the information modulation signals. General description of the QPSK signal can be expressed by the mathematical formulation as:

$$\begin{aligned} s(t) &= \sqrt{E_s/T_s}\cos\left[2\pi f_{cT}t + \phi_{\mathrm{Re}}\right] + \sqrt{E_s/T_s}\sin\left[2\pi f_{cT}t + \phi_{\mathrm{Im}}\right] \\ &= \sqrt{2E_s/T_s}\sin\left[2\pi f_{cT}t + \phi(t)\right] \end{aligned} \tag{5-1}$$

Where:

E_s = is transmitted signal energy per symbol for channel Real and channel Imaginary.

T_s = is the symbol duration.

f_{cT} = is the frequency carrier.

ϕ_{Re} = is initial phase of Real channel

ϕ_{Im} = is initial phase of Imaginary channel

$\phi(t)$ = is the phase as a function of information content, and has the probability value of 45^o, 135^o, 225^o, and 315^o. And the value of $\phi(t)$ is:

$$\phi(t) = a\tan\left[\sin\left[2\pi f_{cT}t + \phi_{\mathrm{Im}}\right] / \cos\left[2\pi f_{cT}t + \phi_{\mathrm{Re}}\right]\right] \tag{5-2}$$

And furthermore by a normalization we set the value of Es/Ts = 1.

5.1.1. Transmitter of QPSK

The QPSK transmission use two input bits at the same time. These input bits could be parallel input channels or be a consecutive even and odd bits form a serial output. The serial data is converted to parallel and will be Real (Re) channel and Imaginary (Im) channel, or some times call as the Real and Imaginer channels. Each bit can be a logical 0 or 1, and there fore four combinations corresponding to four outputs carrier phase are possible. The general form of the QPSK transmitter can be description as like the Figure (5.1).

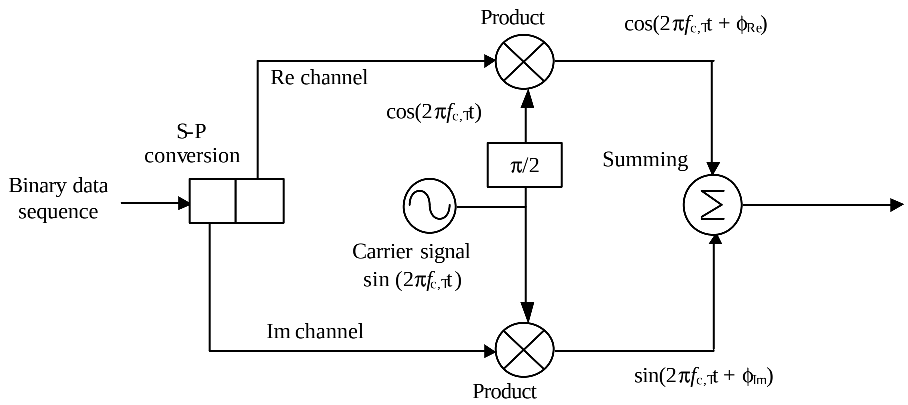


Figure (5.1) The QPSK transmitter

In this part the signal is base band rectangular form. The bit information 0 is represented by a 0 voltage, and the bit 1 is represented by a 1 voltage as like in the Figure (5.2) bellow.

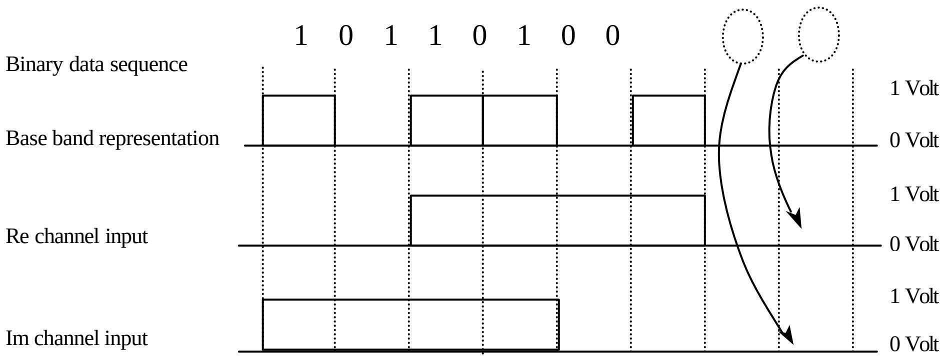


Figure (5.2) Base band input at the Re and Im channel

In the Re channel the bit information will modulate the carrier signal $\cos(2\pi f_{c,T}t)$. The output from this modulator will have a form:

- If bit modulation is 0 the output signal is $\cos(2\pi f_{c,T}t)$.
- If bit modulation is 1 the output signal is $-\cos(2\pi f_{c,T}t)$.

Then the output from Re channel can be represented by the general expression:

$\cos(2\pi f_c t + \phi_{Re}) = \cos(2\pi f_c t + i_{odd}*\pi)$, where is i has a value 0 or 1.

In the Im channel the bit information modulate the carrier signal $\sin(2\pi f_{c,T} t)$. The output from this modulator will have a form:

- If bit modulation is 0 the output signal is $\sin(2\pi f_{c,T} t)$.
- If bit modulation is 1 the output signal is $-\sin(2\pi f_{c,T} t)$.

Then the output from Im channel can be represented by the general expression:

$\sin(2\pi f_{c,T} t + \phi_{Im}) = \sin(2\pi f_{c,T} t + i_{even}\pi)$, where is i has a value 0 or 1.

The next process is a summing point. Re channel output is added to Im channel output, and the result will be:

$$\cos(2\pi f_{c,T} t + i_{odd}\pi) + \sin(2\pi f_{c,T} t + i_{even}\pi) = \sqrt{2}\sin\left(2\pi f_{c,T} t + \tan^{-1}\left[\frac{\sin(2\pi f_{c,T} t + i_{odd}\pi)}{\cos(2\pi f_{c,T} t + i_{even}\pi)}\right]\right) \quad (5\text{-}2)$$

***For example**:*

Sequence of bit input is 1 0 1 1 0 1 0 0 (it is read two bits pair from right to left). We will get the pairs of two bits as: 00, 01, 11, and 10.

From the first pair we have: 00

Re channel input: 0 Re channel output: $\cos(2\pi f_{c,T} t + 0*\pi) = \cos(2\pi f_{c,T} t)$.

Im channel input: 0 Im channel output: $\sin(2\pi f_{c,T} t + 0*\pi) = \sin(2\pi f_{c,T} t)$.

The summing point output is:

$s(t) = \cos(2\pi f_{c,T} t) + \sin(2\pi f_{c,T} t) = \sqrt{2}\sin(2\pi f_{c,T} t + \tan^{-1}[\sin(2\pi f_{c,T} t) / \cos(2\pi f_{c,T} t)])$

$= \sqrt{2}\sin(2\pi f_{c,T} t + 45^o)$

$= \sqrt{2}\sin(2\pi f_{c,T} t + \pi/4)$

From the second pair we have: 01

Re channel input: 1 Re channel output: $\cos(2\pi f_{c,T} t + 1*\pi) = -\cos(2\pi f_{c,T} t)$.

Im channel input: 0 Im channel output: $\sin(2\pi f_{c,T} t + 0*\pi) = \sin(2\pi f_{c,T} t)$.

The summing point output is:

$s(t) = -\cos(2\pi f_{c,T} t) + \sin(2\pi f_{c,T} t) = \sqrt{2}\sin(2\pi f_{c,T} t + \tan^{-1}[\sin(2\pi f_{c,T} t) / -\cos(2\pi f_{c,T} t)])$

$= \sqrt{2}\sin(2\pi f_{c,T} t + 135^o)$

$= \sqrt{2}\sin(2\pi f_{c,T} t + 3\pi/4)$

From the third pair we have: 11

Re channel input: 1 Re channel output: $\cos(2\pi f_{c,T}t + 1*\pi) = -\cos(2\pi f_{c,T}t)$.

Im channel input: 1 Im channel output: $\sin(2\pi f_{c,T}t + 1*\pi) = -\sin(2\pi f_{c,T}t)$.

The summing point output is:

$$s(t) = -\cos(2\pi f_{c,T}t) - \sin(2\pi f_{c,T}t) = \sqrt{2}\sin(2\pi f_{c,T}t + \tan^{-1}[-\sin(2\pi f_{c,T}t) / -\cos(2\pi f_{c,T}t)])$$

$$= \sqrt{2}\sin(2\pi f_{c,T}t + 225^{o})$$

$$= \sqrt{2}\sin(2\pi f_{c,T}t + 5\pi/4)$$

From the third pair we have: 10

Re channel input: 0 Re channel output: $\cos(2\pi f_{c,T}t + 0*\pi) = \cos(2\pi f_{c,T}t)$.

Im channel input: 1 Im channel output: $\sin(2\pi f_{c,T}t + 1*\pi) = -\sin(2\pi f_{c,T}t)$.

The summing point output is:

$$s(t) = \cos(2\pi f_{c,T}t) - \sin(2\pi f_{c,T}t) = \sqrt{2}\sin(2\pi f_{c,T}t + \tan^{-1}[-\sin(2\pi f_{c,T}t) / \cos(2\pi f_{c,T}t)])$$

$$= \sqrt{2}\sin(2\pi f_{c,T}t + 315^{o})$$

$$= \sqrt{2}\sin(2\pi f_{c,T}t + 7\pi/4) = \sqrt{2}\sin(2\pi f_{c,T}t - \pi/4)$$

If will be easier to understand if we use phase or constellation diagram for these signals as like the Figure (5.3) bellow. To understand the constellation diagram is different with the to read the information sequence. In the information sequence we read from right to the left. For the constellation 00, the meaning is information content for Re channel = 0 and Im channel = 0. For the constellation 01, the meaning is information content for Re channel =1, and Im channel = 0. For the constellation 11, the meaning is information content for Re channel = 1 and Im channel = 1. And for the constellation 10, the meaning is information content for Re channel = 0 and Im channel = 1. In this constellation the information value 0 is represented by +1 and the information value 1 is represented by –1.

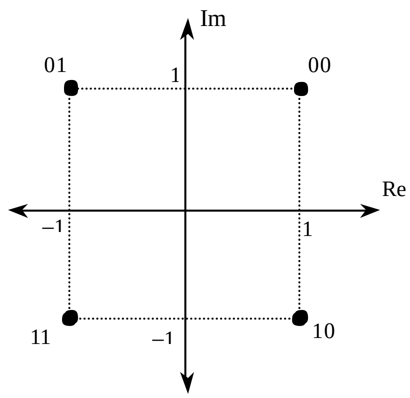


Figure (5.3) QPSK Constellation diagram

We can write it in this constellation 00 = (+1, +1), 01 = (+1, -1), 11 = (-1, -1), and 10 = (-1, +1). In the some book maybe we find the other constellation style, but actually the meaning is same.

From the above example we will have the signal output as follows:

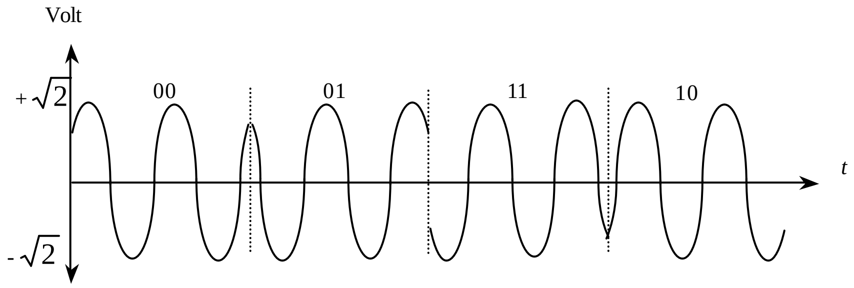


Figure (5.4) The QPSK signal

In this figure, the amplitude signal in the Volt. And this figure described the amplitude signal as a function of time. Every two cycle of waveform represent one output of the QPSK signal.

5.1.2. Receiver of QPSK

In the ideal receiver, the received signal is multiplied by a reference signal phase locked, and then filters the resulting output to the base band pulse waveform. The QPSK receiver can be visualized as consisting of two channels, as like the Figure (5.5).

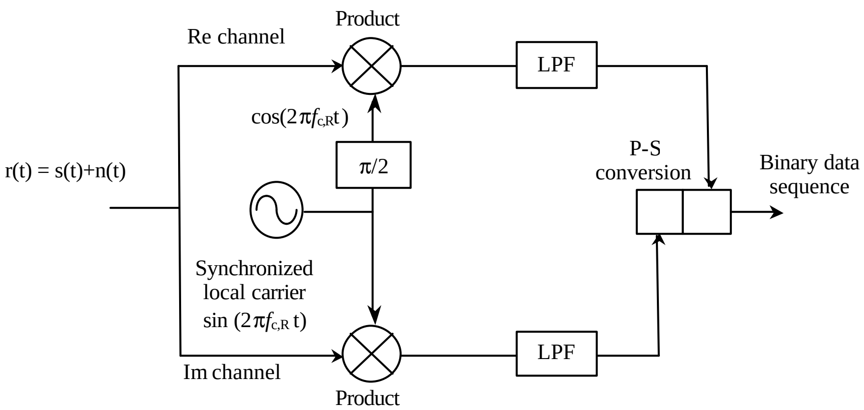


Figure (5.5) The QPSK Receiver

By assumption that between transmitter and receiver part have been synchronized, and no delay along signal propagation. We can decide that in this case $f_{c,R} = f_{c,T} = f_c$. From the above figure and consider this assumption, Re channel id multiplied by using $\cos(2\pi f_c t)$, and Im channel is multiplied by using $\sin(2\pi f_c t)$. The output form each channel filtered by using low pass filter (LPF) and decision is made from each channel. The detection of bits in one channel is independent to the other one (no cross talk effect) if the other channel adds zero voltage to the channel integrator output.

To simplification, here we use the example of the signal output from the transmitter above and by assumption that no noise in the channel, or we use the system by using the ideal channel condition. In this cases we made an assumption that the carrier recovery work perfectly so the error phase estimate is zero, therefore the received signal is same with the transmitted signal, or r(t) = s(t).

In the case received signal is $r(t) = \sqrt{2}\sin(2\pi f_c t + \pi/4)$ This received signal has the equivalent value with

$r(t) = \sqrt{2}\,[\sin(2\pi f_c t)\cos(\pi/4) + \cos(2\pi f_c t)\sin(\pi/4)]$
$\quad = \sin(2\pi f_c t) + \cos(2\pi f_c t)$

After the demodulation process, the output signal of the Re channel will be:

$r(t)*\cos(2\pi f_c t) = [\sin(2\pi f_c t) + \cos(2\pi f_c t)]*\cos(2\pi f_c t)$
$\quad = \sin(2\pi f_c t)\cos(2\pi f_c t) + \cos(2\pi f_c t)\cos(2\pi f_c t)$
$\quad = \sin(2\pi f_c t)\cos(2\pi f_c t) + 0.5*[\cos(2*2\pi f_c t) + 1]$

By the same way, the Im channel will have output:

$r(t)*\sin(2\pi f_c t) = [\sin(2\pi f_c t) + \cos(2\pi f_c t)]*\sin(2\pi f_c t)$
$\quad = \sin^2(2\pi f_c t) + \cos(2\pi f_c t)*\sin(2\pi f_c t)$
$\quad = 0.5*[\cos(2*2\pi f_c t) + 1] + \cos(2\pi f_c t)*\sin(2\pi f_c t)$

Filtering process, by using integration from t = 0 until t = T, the Re channel will be:

$$\int_0^T \left(\sin(2pf_c t)\cos(2pf_c t) + 0.5*\left[\cos(2*2pf_c t) + 1\right]\right)dt$$

$$= \int_0^T \cos(2pf_c t)\, d\left(\frac{\cos(2pf_c t)}{2pf_c}\right) + \int_0^T 0.5*\left[\cos(2*2pf_c t) + 1\right]dt$$

$$= \left.\frac{\cos^2(2pf_c t)}{2pf_c}\right|_0^T + 0.5*\left[\frac{\sin(2*2pf_c t)}{2*2pf_c} + 1\right]_0^T = 0.5$$

By the same way, filtering process of the Im channel, by using integration from t = 0 until t = T:

$$\int_0^T \left(\sin(2pf_c t)\cos(2pf_c t) + 0.5*\left[\cos(2*2pf_c t)+1\right]\right)dt$$

$$= \int_0^T \cos(2pf_c t)\, d\left(\frac{\cos(2pf_c t)}{2pf_c}\right) + \int_0^T 0.5*\left[\cos(2*2pf_c t)+1\right]dt$$

$$= \left.\frac{\cos^2(2pf_c t)}{2pf_c}\right|_0^T + 0.5*\left[\frac{\sin(2*2pf_c t)}{2*2pf_c}+1\right]_0^T = 0.5$$

5.1.3. QPSK Transmission System

The transmission of QPSK system through AWGN channel is similar with the BPSK system in the same channel. But in this case, the signal transmitted is content of real and imaginer component. Therefore in the transmission we must consider the real and imaginer components of the AWGN channel. The simplification form is described in Figure (5.6).

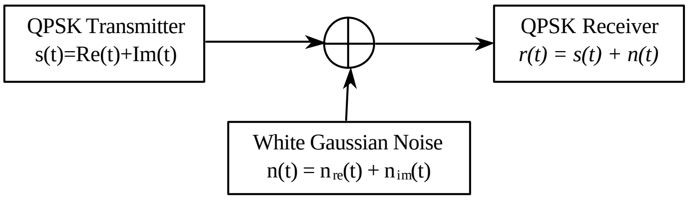


Figure (5.6) QPSK transmission through AWGN channel

In the QPSK system, each channel has independent signal and the noise content at received signal is independent too. By using the transmitted signal in the equation (5-2), we know that the transmitted signal is

$$s(t) = Re(t) + Im(t)$$
$$= \sin(2\pi f_c t + i_{odd}*\pi/4) + \cos(2\pi f_c t + i_{even}*\pi/4)$$

The noise from AWGN channel is

$$n(t) = n_{re}(t) + n_{im}(t) \qquad (5\text{-}3)$$

And the received signal will has the form:

$$r(t) = s(t) + n(t)$$
$$= \sin(2\pi f_c t + i_{odd}*\pi/4) + n_{re}(t) + \cos(2\pi f_c t + i_{even}*\pi/4) + n_{im}(t) \qquad (5\text{-}4)$$

To evaluate the performance of QPSK we can use the BPSK performance as a reference. As we know that each channel of QPSK is independent each one another, so QPSK signal is same with summation of two BPSK signal. The AWGN channel effect for each channel is same with its effect in to the BPSK signal.

The bit error probability of the QPSK signal is: $P_b = \frac{1}{2} erfc\left[\sqrt{E_b / N_o}\right]$ (5-5)

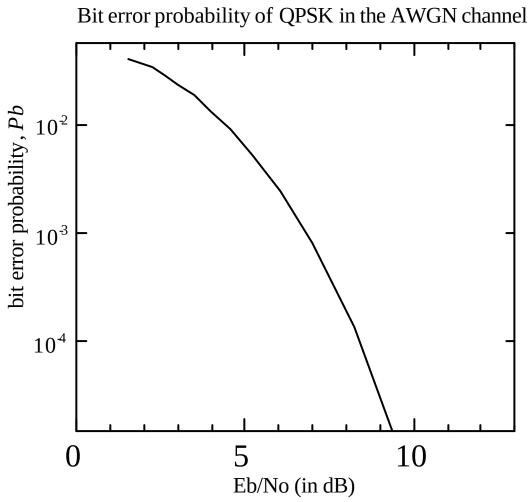


Figure (5.7) Performance of QPSK system in the AWGN channel

By evaluate the bit error rate probability of the QPSK signal we will know the symbol error probability of QPSK signal, because the symbol error probability is a function of bit error probability, and represented by using mathematical formulation:

$$Ps = \log_2(M)Pb \qquad (5\text{-}6)$$

In the QPSK we know that M = 4, so the value of symbol error probability is

$$Ps = \log_2(4)Pb = 2Pb$$

The bit error probability performance of QPSK transmission through AWGN channel as a function of Eb/No is like the Figure (4.7).

5.1.4. Carrier-phase Synchronization of QPSK System

Carrier-phase recovery by using Costas Loop for QPSK is a combination from two parallel carrier-phase recoveries, which we have applied it for BPSK system. General descriptions of Costas loop carrier recovery for QPSK in the base band term as the Figure (5.8).

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 (*.*). In synchronization process for QPSK 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:

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

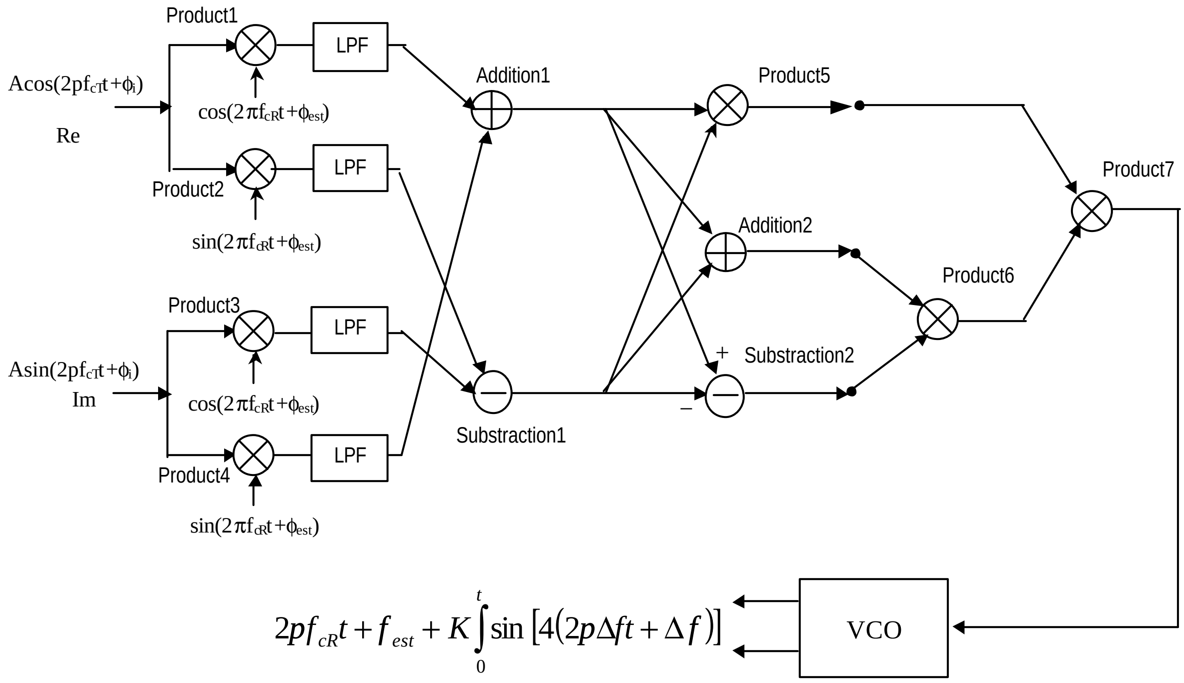


Figure (5.8) Carrier recovery for QPSK system

- 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:

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

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

$$= (1/4)\sin\{4(2\pi\,\Delta f\,t + \Delta\phi_t)\} \qquad (5\text{-}7)$$

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 (5-8)

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

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

$$\frac{d\left(q_e(t)\right)}{dt} = \sin\left(4\Delta f\right) \tag{5-9}$$

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

5.2. Session 2: Question and Answer

Question 1:

What is the advantage if we use the QPSK system compare to the BPSK system?

Answer:

In the BPSK one symbol content of bit, but in the QPSK one symbol content 2 bits. And all symbol use same bandwidth frequency for transmission. BPSK is a *double-sided band suppressed carrier* (DSBSC) system with 1 bit per symbol; the bit efficiency is 0.5 bit/Hz. By using QPSK with 2 bit per symbol; the bit efficiency is 1 bit/Hz. We can say that the bit efficiency of QPSK system is twice comparing to BPSK system.

Question 2:

How we setting the energy per bit to noise ratio on the QPSK transmission systems?

Answer:

QPSK system is consist of two independent BPSK systems, channel I and channel Q or some times is called as Real and Imaginer channels. In the transmission by using AWGN channel, there is generated a complex noise amplitude, real and imaginer noise components, and these component are independent each one another. Each noise component will destruct each channel, real noise destruct real component of transmission signal and the imaginer noise component will destruct the imaginer component of transmission signal. By using this property we will get the result that energy bit per noise ratio for QPSK system is same with the BPSK system, Eb/No. And it will generate the bit error probability of QPSK system will be same with the BPSK system. But for symbol error probability it will be different.

Question 3:

How we setting the limitation error value of the phase estimate for QPSK phase-carrier recovery?

Answer:

In the BPSK every symbol is separated by π radiant or 180^o of phase. And maximum phase deviation without change the symbol value for this is 90^o. The setting of maximum error estimate for phase is limited to 15^o. By this value it will not give the effect for the symbol, because 15^o compare to 90^o is too small.

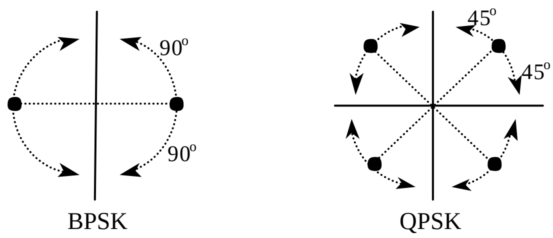


Figure (5.9) Phase deviation of BPSK and QPSK system

In the QPSK very symbol are separated by $\pi/2$ radiant of phase or 90^o. And maximum phase deviation without change the symbol value for this is 45^o. If the setting of maximum error estimate for phase is limited to 15^o, the probability of error happen is large, so it will better if the setting of maximum error estimate is smaller to get the better performance.

5.3. Session 3: Simulation of QPSK

Here we tray to build the QPSK simulation system in the base band system. By using Costas Loop as phase-carrier recovery at the receiver part, we will tray to understand the performances of QPSK system through the AWGN channel. The General Description of QPSK for base band simulation as in the Figure (5.10).

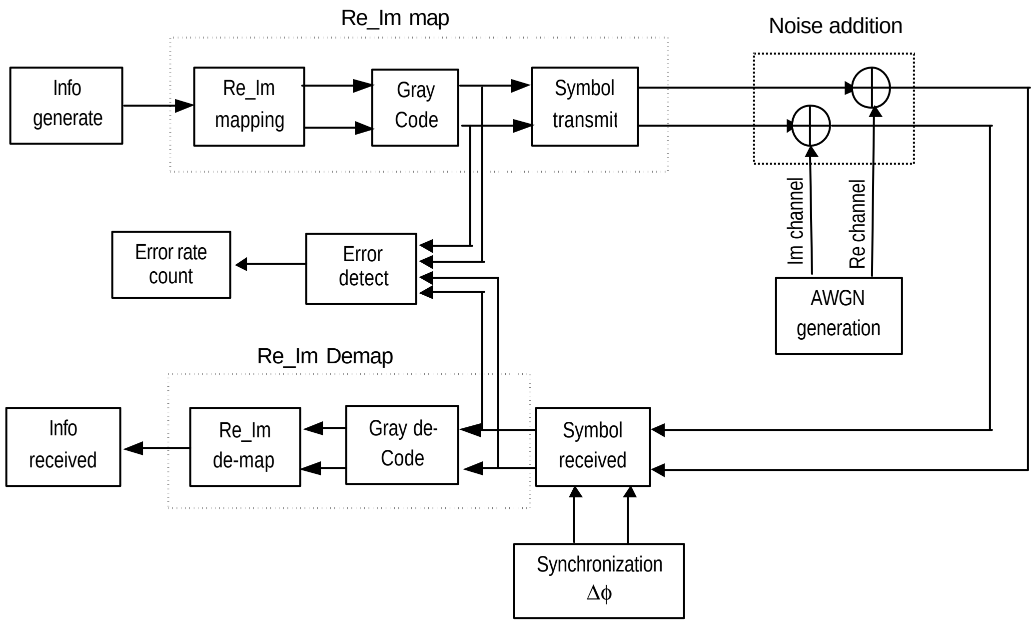


Figure (5.10) Block diagram of QPSK Base band simulation

5.3.1. The Algorithm

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

♦ **Info Generator**

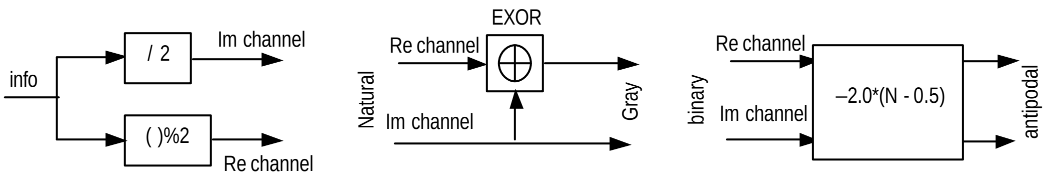
Bit information generated randomly by info generator function. The random number generated is set value 0,1,2 and 3 as the integer number to represent PAM signal. By using Microsoft C++ software, this function will generate a uniform random number.

♦ **Re channel And Im channel Mapping**

This function consists of three parts in the Figure (5.10); Re_Im map, Gray Code and Symbol Transmit.

- *The Re_Im Map*. Integer information generated will mapped to two bit pair. Based on the Figure (5.12a) we will understand that if the integer information value is 0 the two bits pair generate are 00, if the integer information value is 1 the two bits pair are 01, if the integer information is 2 the two bits pair are 01, and if the integer information is 3 the two bits pair are 11. Then we set the LSB value as Re channel info and the MSB value as Im channel info.

- The Gray Code. Re channel and Im channel info has natural bit value, by using the logic as like in the Figure (5.12b) we able to make a gray code logic. If the value of Re and Im is 10, the Im channel value will change to 1. If the value of Re and Im is 11, it will change to 10. The other value is not change.
- *The Symbol Transmit*. Re channel value and Im channel value has "1 or 0" info and have integer variable. By using this part it will change as double variable and have the value –1.00 and +1.00. The bit 1 will be –1.00 and the bit 0 will be +1.00. This is represents an antipodal signals.



a) Re and Im mapping b) Gray Code c) Symbol transmit

Figure (5.12) 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 use to correct the phase different.

As we know that in the synchronization process by using Costas loop for BPSK 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:

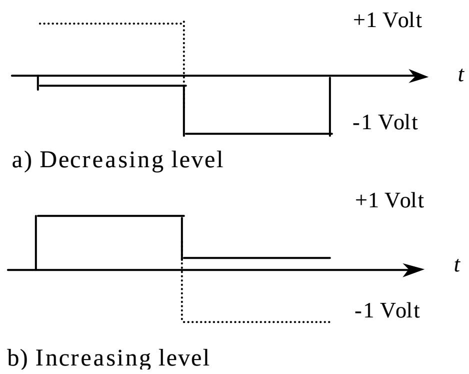
  $re = (s^2\ lnx1)^{1/2}\ cos(2*p*x2)$;*real part*

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

♦ **Noise Addition**

The noise generated from the AWGN function are real and Imaginer part, and independent each one another. In this function we add the Re channel with real noise and the Im channel with imaginer noise. In some books Re channel from QPSK transmitter is called as real channel, and Im channel is called as imaginer channel. AWGN will destruct 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.

The effect of AWGN will change the level of transmitted signal as like in the Figure (5.13).



Figure(5.13). The effect AWGN at the level of antipodal signal transmission

♦ **Detection**

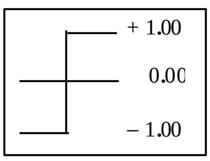


Figure (5.14) Decision symbol for received signal

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

if signal level < 0.0 the decision for symbol received is –1.0

Other wise

if signal level $\geq$ 0.0 the decision for symbol received is 1.0.

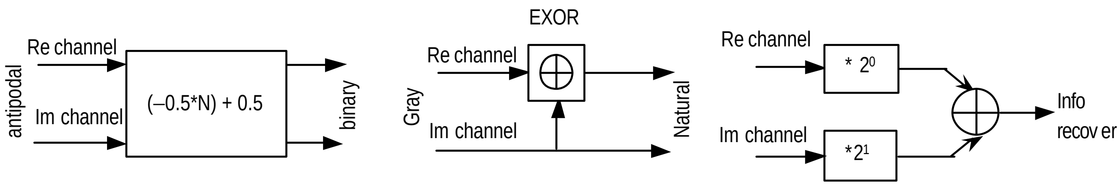
This rule for both of Re channel and Im channel signal received, and work independently.

♦ **Rx Demap**

This function consists of two parts of block diagram in the Figure (5.10).

- *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 block. More easily we can look at Figure (5.15a). Second step is converts from gray form to natural binary from. By using the XOR logic, for Re channel input we get the natural binary result as like Figure (5.15b).

- *RE_IM demap.*The purpose of this part is get the symbol value by convert the two pair bit (binary) decimal value. By multiply the MSB with 2 and LSB with 2^0 or 1, and continue with addition of both results, we will get the decimal information value.



a) Antipodal to binary b) Gray de-Code c) Re and Im de-mapping

Figure (5.15) Description of Rx De-mapping

♦ Error Detect

By using the output from detection function and compare channel by channel to the output of Re_Im map we able detect what the error in the receiving signal happen. If the value of output from detection function is same with the output from Re_Im map, the error didn't happen, other wise if these value is not same, the error was happening. By using channel by channel error detection we gets bit error information of Re and Im channels, and combine the result of these result we will get the information of symbol error detection. If Re channel error or Im channel 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 QPSK system as a function of *energy per bit to noise ratio*. The output is theoretical value of *error probability* performances QPSK system. From it we get the information of bit error probability and symbol error probability.

5. 3.2. 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.

Re_info = information generate by transmitter part.

Im_info = information generate by 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

Re_anti_Rx = Real channel symbol recovered at receiver part in the antipodal format

Im_anti_Rx = Imaginary channel symbol recovered at receiver part in the antipodal format

pha_receiv = phase receive after transmission

Re_info_Rx = Real channel information recovered at receiver part

Im_info_Rx = Imaginary channel information recovered at receiver part

SNR = Signal to Noise Ratio

BER = bit error rate

Pb_QPSK = probability of bit error of QPSK system

Ps_QPSK = probability of symbol error of QPSK system

The listing program of QPSK system is as follows.

```
//Program of QPSK
//editing by Tri Budi Santoso
//Suzuki Laboratory
//Mobile Communication Group, Tokyo Institute of Technology, Japan
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

```
#define N 1000000

int nn=N-1,i;
int Re_info[N],Im_info[N];
double Re_Tx[N],Im_Tx[N],pha_Tx[N];
double phase[N],Ave_Pha_err,Pha_err,deg_error,pha_i;
double pi=acos(-1.0),Re_AWGN[N],Im_AWGN[N],var=0.1;
double Re_Rx[N],Im_Rx[N];
double Re_anti_Rx[N],Im_anti_Rx[N];
int Re_info_Rx[N],Im_info_Rx[N];
double BER;
double SNR;
double Ps_QPSK,Pb_QPSK;


void info_generate(int Re_info[],int Im_info[]);
void symbol_generate(double Re_Tx[],double Im_Tx[],double pha_Tx[]);
void costas(double phase[],double *pha_er);
void synchronization(double pha_i,double *deg_er);
void noise_AWG(double Re_AWGN[],double Im_AWGN[]);
void received_symbol(double Re_Rx[],double Im_Rx[]);
void decission(double Re_anti_Rx[],double Im_anti_Rx[]);
void info_recover(int Re_info_Rx[],int Im_info_Rx[]);
void error_rate(double *ber);
void sig_noise_ratio(double *snr);
void error_prob(double *ps,double *pb);


void main()
{
 srand(1);
 info_generate(Re_info,Im_info);
 symbol_generate(Re_Tx,Im_Tx,pha_Tx);
 costas(phase,&Ave_Pha_err);
 noise_AWG(Re_AWGN,Im_AWGN);
 received_symbol(Re_Rx,Im_Rx);
 decission(Re_anti_Rx,Im_anti_Rx);
 info_recover(Re_info_Rx,Im_info_Rx);
 error_rate(&BER);
 sig_noise_ratio(&SNR);
 error_prob(&Ps_QPSK,&Pb_QPSK);
}

void info_generate(int Re_info[],int Im_info[])
{
 for (i=1;i<=nn;++i)
 {
    Re_info[i]=(int)rand()%2;
    Im_info[i]=(int)rand()%2;
    printf("\nRe_info[%d]:%d Im_info[%d]:%d",i,Re_info[i],i,Im_info[i]);
 }
}
```

```
void symbol_generate(double Re_Tx[],double Im_Tx[],double pha_Tx[])
{
 for (i=1;i<=nn;++i)
 {
     Re_Tx[i]=((double)Re_info[i]-0.5)*(-2.0);
    Im_Tx[i]=((double)Im_info[i]-0.5)*(-2.0);
    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("\nRe_Tx[%d]:%3.2f  Im_Tx[%d]:%3.2f
   pha_Tx[%d]:%4.2f",i,Re_Tx[i],i,Im_Tx[i],i,pha_Tx[i]);
 }
}

void costas(double phase[],double *pha_er)
{
 double pha_i;
 Pha_err=0.0;
 for(i=1;i<=nn;++i)
 {
  pha_i=pha_Tx[i];
  synchronization(pha_i,&deg_error);
  phase[i]=fabs(deg_error);
 // printf("\n phase[i]:%f ",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 + phase_estim;
        i++;
   }while(i<=10);
  d++;
 } while(fabs(deg_error)>=7.5);
 *deg_er=deg_error;
}

void noise_AWG(double Re_AWGN[],double Im_AWGN[])
{
 double u1,u2,rmax=(RAND_MAX+1.0);
 for(i=1;i<=nn;++i)
 {
  u1=(double)rand()/rmax;
  if(u1<=1.0e-38)
  {u1=1.0e-38;}
  u2=(double)rand()/rmax;

  Re_AWGN[i]=sqrt(-2.0*var*log(u1))*cos(2*pi*u2);
  Im_AWGN[i]=sqrt(-2.0*var*log(u1))*sin(2*pi*u2);
 }
}

void received_symbol(double Re_Rx[],double Im_Rx[])
{
 for(i=1;i<=nn;++i)
 {
  Re_Rx[i]=Re_Tx[i]*cos(phase[i]/360*2*pi) + Re_AWGN[i];
  Im_Rx[i]=Im_Tx[i]*cos(phase[i]/360*2*pi) + Im_AWGN[i];
  //printf("\nRe_Rx[%d]:%f Im_Rx[%d]:%f",i,Re_Rx[i],i,Im_Rx[i]);
 }
}


void decission(double Re_anti_Rx[],double Im_anti_Rx[])
{
 for(i=1;i<=nn;++i)
 {
 if(Re_Rx[i]>=0.0)
    Re_anti_Rx[i]=1.0;
 else
    Re_anti_Rx[i]=-1.0;
 if(Im_Rx[i]>=0.0)
    Im_anti_Rx[i]=1.0;
 else
    Im_anti_Rx[i]=-1.0;
 //printf("\nRe_anti_Rx[%d]:%4.2f Im_anti_Rx[%d]:%4.2f",i,Re_anti_Rx[i],i,Im_anti_Rx[i]);
 }
}
```

```
void info_recover(int Re_info_Rx[],int Im_info_Rx[])
{ double re_info_Rx,im_info_Rx;
  for(i=1;i<=nn;++i)
 {
  re_info_Rx = (Re_anti_Rx[i]/(-2.0))+0.5;
  Re_info_Rx[i]=(int)re_info_Rx;
  im_info_Rx = (Im_anti_Rx[i]/(-2.0))+0.5;
  Im_info_Rx[i]=(int)im_info_Rx;
    }
}

void error_rate(double *ber)
{
    double ber_re=0.0,ber_im=0.0;
    BER=0.0;
 for(i=1;i<=nn;++i)
 {
  if(Re_info_Rx[i]==Re_info[i])
     ber_re = ber_re;
  else
   ber_re += 1.0;
  if(Im_info_Rx[i]==Im_info[i])
     ber_im = ber_im;
  else
   ber_im += 1.0;
 }
 BER=(ber_re + ber_im)/nn;
 *ber=BER;
 printf("\n\tBER:%f",BER);
}

void sig_noise_ratio(double *snr)
{
    double gama;
    gama = 1.0/var;
    SNR=10*log10(gama);
    *snr=SNR;
printf("\nSNR:%f",SNR);
}

void error_prob(double *ps,double *pb)
{
    double gama1, x,term_1,term_3,term_5;
    gama1 = 1.0/var*cos(deg_error/360*2*pi);
    x=sqrt(2.0*gama1)*sin(3.1415/sqrt(2.0)/4.0);
    term_1=1.0/(2.0*pow(x,2));
    term_3=1.0*3.0/(2.0*2.0*pow(x,4));
    term_5=1.0*3.0*5.0/(2.0*2.0*2.0*pow(x,6));
    Ps_QPSK=(exp(-x*x)/(x*sqrt(3.1415)))*(1-term_1+term_3-term_5);
    *ps=Ps_QPSK;
    Pb_QPSK=0.5*Ps_QPSK;
    *pb=Pb_QPSK;
printf("\t Pb_QPSK:%f Ps_QPSK:%f",Pb_QPSK,Ps_QPSK);
}
```

5. 3.3. Simulation Result

The simulation result for the QPSK system is like in the Table 1 bellow.

Table 5.1. Costas Loop Effect on the QPSK system

Eb/No	Pb (pure modulation)	Pb(using Costas Loop)
3.01	0.024096	0.025116
3.47	0.018742	0.019647
3.98	0.013642	0.014389
4.56	0.009073	0.009640
5.23	0.005290	0.005674
6.02	0.002509	0.002726
6.99	0.000834	0.000923
8.24	0.000138	0.000157
9.03	0.000033	0.000039
10.00	0.000004	0.000005

From the Table 4.1 we can see that the degradation of probability of error value was happened. As example for value of Eb/No = 3.01, the pure BPSK modulation without using Costas Loop as Carrier recovery the probability of error value is 0.024096. But after we use Costas Loop as carrier recovery, the probability of error is 0.025116. It was happen because the phase different between the received signal and the local carrier oscillator is 0.708^o, and will give the degradation effect by factor cos (0.708^o) or 0.9594 for the Eb/No parameter at the error function in the equation (5-9).

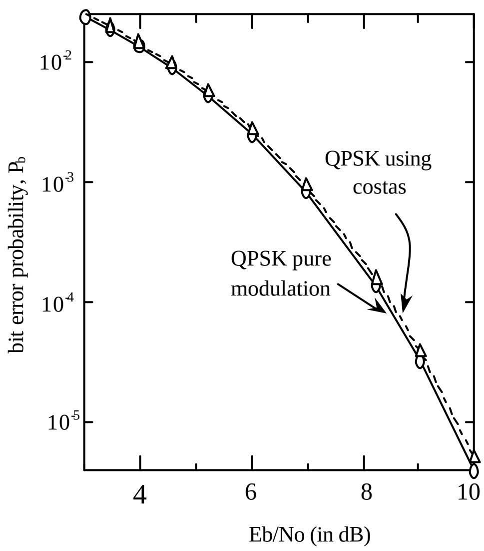


Figure (5.16) Performance of QPSK pure modulation and using carrier–phase recovery

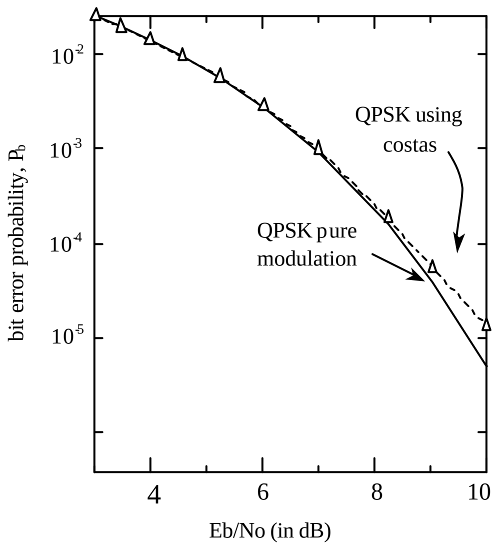


Figure (5.17) Performance evaluation of QPSK using carrier–phase recovery

From the Table 4.1 we can build the graph for degradation performance of the QPSK system after using Costas Loop as like the Figure (5.16). We can compare the performance to the pure QPSK system without Costas Loop.

By this figure we get the result that after using Costas Loop as carrier recovery the average degradation of the QKSP performance is about 0.2 dB.

It will be better if we evaluate our program simulation of Costas Loop of BPSK system. By using the same way as in the BPSK transmission system we get the result as like in the Figure (5.17).

The simulation result showed that our simulation has performance that very closed to the theoretical value for the value of Eb/No 3.01 dB to 8.00dB. But after Eb/No > than 8.00 dB, the simulation result indicated the deviation value from the theory.

For this result we can make a hypothesis, that it will be better if we use more bit information for the small value of Eb/No to evaluate our simulation. By using large data transmission, it we will get the more accurate result of our simulation performance. But it will need more time to run the simulation program.