Lecture Note Developing on Simulation of BPSK System in the AWGN Channel

Author: **Tri Budi Santoso**

Supervisors: **Prof. Hiroshi SUZUKI**

Prof. Kazuhiko FUKAWA

Book Reference: **Digital Communication, John G. Proakis.**

Topic: BPSK Transmission System

3.1. Session 1: Review of BPSK System

3.1.1. BPSK Signaling

In coherent binary phase shift keying (PSK) a phase shift of phase in a carrier-modulated signal is used to distinguish between the binary symbols 0 and 1. The two signals are therefore given by

$$\begin{aligned} s_1(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2pf_c t) \\ s_2(t) &= \sqrt{\frac{2E_b}{T_b}}\cos(2pf_c t + p) \\ &= -\sqrt{\frac{2E_b}{T_b}}\cos(2pf_c t) \end{aligned} \tag{3-1}$$

where f_c is the carrier frequency. The relationship among bit information and waveform signal output and initial phase from modulator is illustrated as Figure 3.1.

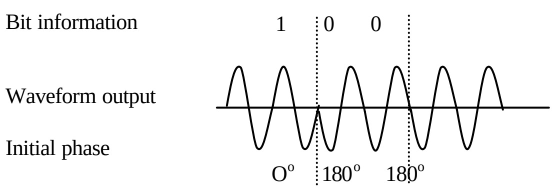

Figure (3.1) BPSK signal

This system may be represented using a single basis function

$$f_1(t) = \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t), \qquad 0 \le t \le T_b \tag{3-2}$$

with the two signal vectors amplitude given by

$$\begin{aligned} s_1 &= \sqrt{E_b} \\ s_2 &= -\sqrt{E_b} \end{aligned} \tag{3-3}$$

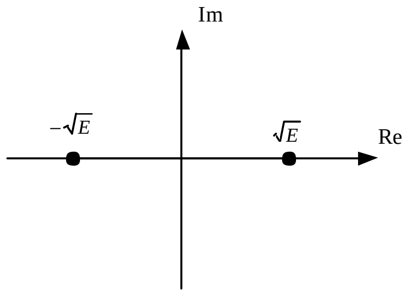

Figure (3.2) Constellation diagram of BPSK

where E_b is the signal energy which is also in this case the energy per bit.

A coherent binary PSK system is therefore characterized by having a signal space that is one dimensional with a signal constellation consisting of two message points of equal and opposite amplitude Such signals with equal energy and a cross correlation coefficient of -1 are called *antipodal.*

This signal space diagram is shown in Figure (3-2). If the symbols are equiprobable then the rule for deciding which symbol was transmitted is to choose the closest message point.
The BPSK transmission system is illustrated as Figure 3.3.

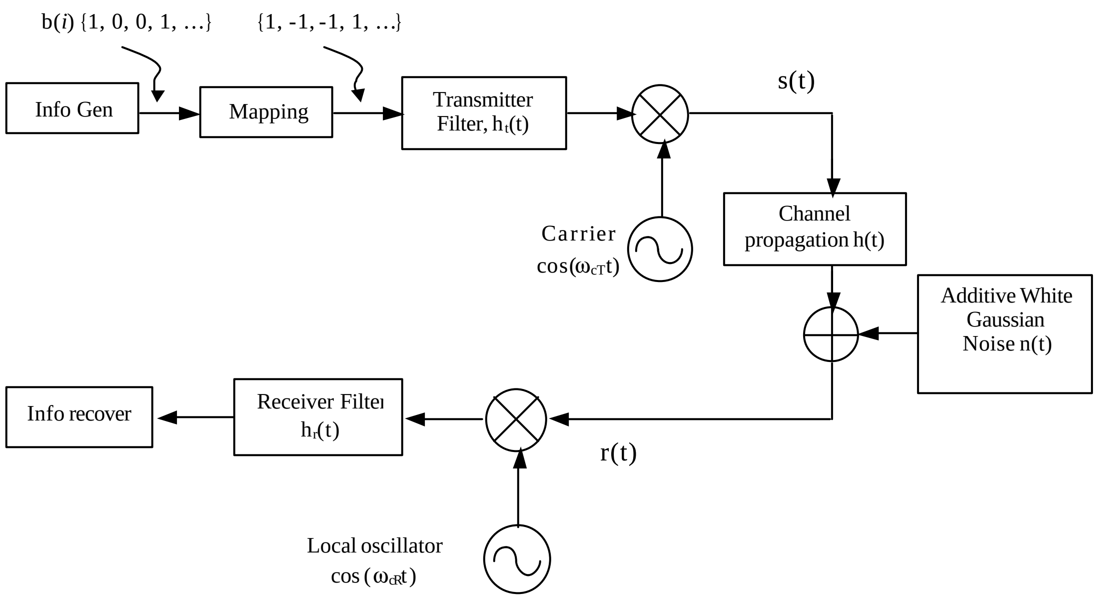

Figure (3.3) Block Diagram BPSK transmission system

From the Figure (3.3) we can describe the BPSK transmission system as 3 parts; transmitter, receiver and AWGN channel.

3.1.2. Transmitter Part

By using the block diagram in Figure (3.3) above, if we generate the binary number randomly we will get the information sequences. Information generated: {b(i)} = 1, 0, 0,……1
This bit information sequences will modulate the carrier frequency, and the phase of the carrier frequency will be shifted as function of binary information. The bit information "1" will not shift the carrier phase, and the bit information "0" will shift the carrier phase 180^o or π radiant. The waveform of the modulated signal can be describes as mathematical formulation as the equation (3-1).

3.1.3. Receiver Part

Basically the receiver part as similar with the transmitter part, but the function is contradictive. After demodulation process by using local signal oscillator and filtering by using LPF, the base band signal will be decided. If the amplitude of the base band signal is <0 it's decides that a binary "0" was send and if the amplitude of the baseband signal is >= 0, the receiver decides that binary "1" was send by the transmitter.
This description is made by assumption that all of the process of modulator, Band Pass Filter, demodulator and filtering is work perfectly and inter symbol inter fervencies (ISI) is not happen.

3.1.4. The AWGN Channel

The signal output from transmitter part is propagated through the channel, which is corrupted by the additive white Gaussian noise, and illustrated in the Figure (3.3). Thus the received signal at the receiver part will has a form:

$$r(t) = s(t) + n(t); \; 0 <= t <= T \qquad (3\text{-}4)$$

where n(t) denotes a sample function of the additive white Gaussian noise (AWGN) process with variants value σ^2 = No/2 Watt/Hertz. Here we made an assumption that the mean value of the AWGN is 0.

The sample of noise output from AWGN channel can described as the Figure (3.4).

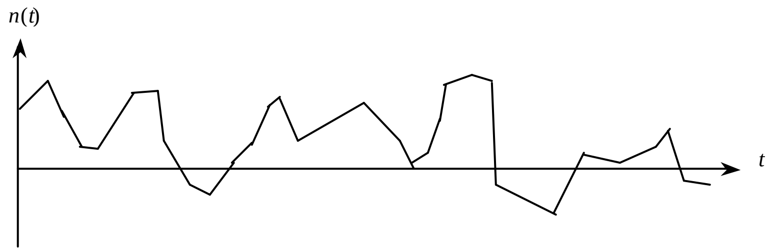

Figure (3.4) Output from AWGN channel

The probability distribution of n (along the y-axis) is given by

$$P(n) = \frac{1}{s\sqrt{2p}} \exp\left[-\frac{n^2}{2s^2}\right] \tag{3-5}$$

From Fourier analysis, notice all frequencies are necessary to create signal $n(t)$. The power spectral density spectrum is shown below.

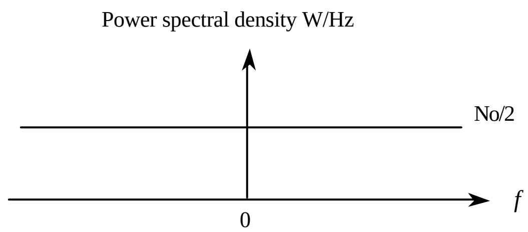

Figure 1.5. Psd of AWGN channel

No/2 is the power spectral density of white noise. This is because all mathematical analysis is (using complex numbers) is done using a two-sided spectrum as shown above. The term "white" is used because the spectrum is similar to white light, which contains equal amounts of all frequencies within the visible electromagnetic waves spectrum.

The effect of the AWGN channel changes the amplitude of the transmission signal. If the addition noise is positive value, the level amplitude of the transmitted signal will increase. Other wise if the addition noise has a negative value; the level amplitude of the transmitted signal will decrease.

The probability of receiver part can make an true decision the received signal can expressed as the probability of the received signal is "s_1" if the "s_1" was transmitted by the transmitter part or $P(s_1 | s_1)$. The probability of the receiver make a fault decision can be expressed if the receiver part receives the bit information "s_2" if the bit "s_1" was transmitted by the transmitter part or $P(s_2|\ s_1)$. This rule if same if the transmitted bit is "s_2" Some time we will find the other description by using s_1=1 and s_2 = 0 or the other. The probability a signal received by error at receiver then can be written by a formulation:

$$P_b = \frac{1}{2} erfc\left[\sqrt{E_b / N_o}\right] \qquad (3\text{-}5)$$

where the is a standard formulation to express the probability of error in the AWGN channel The ratio of E_b/N_o then called as signal to noise ratio (SNR) per bit in dB.

$$\text{SNR per bit} = 10 \log (E_b/N_o) \text{ dB} \qquad (3\text{-}6)$$

3.1.5. Carrier-Phase Recovery of BPSK System

The double-sided band suppressed carrier (DSBSC) technique, as like BPSK and FSK the signals spectral are symmetric with respect to the carrier frequency. In this system the transmission signal doesn't content the discrete carrier frequency. So why in the receiver part, phase-carrier synchronization is required. The carrier regeneration process, then called as *carrier recovery* can be achieved several way. One method for generating a properly phase-carrier for a double-sided band suppressed carrier signal is illustrated by the block diagram shown in Figure (1.2). This scheme is developed by Costas (1956) and is called as Costas Loop. The receiver signal is multiplied by a $\cos(2\pi f_{cR}t + \phi_{est})$ and $\sin(2\pi f_{cR}t + \phi_{est})$ which are the output from the VCO. Where f_{cR} is a local carrier frequency and ϕ_{est} is initial phase generated.

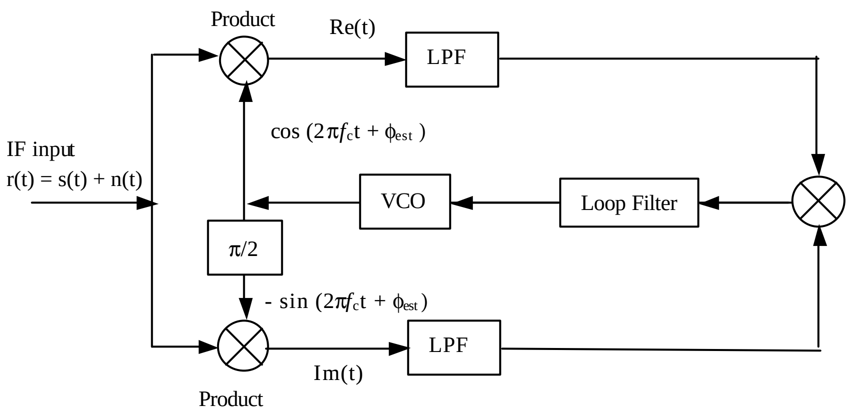

Figure (1.2) Costas loop carrier-phase recovery

In the case of BPSK system s(t) content of $(E_b)^{1/2}\cos(2\pi f_{cT}t + \phi_i)$, where $(E_b)^{1/2}$ is the amplitude of signal transmitted, and in this case we make an assumption that the average value is 1. The f_{cT} is the carrier frequency of transmitter part, and ϕ_i is initial phase of transmitter signal. Further more it can be rewritten as $\cos(2\pi f_{cT}t + \phi_i)$.

There are two channels in the Costas Loop, real (Re) and imaginary (Im).

From the real part, the product output is:

$$y_{Re}(t) = [s(t) + n_{Re}(t)]\cos(2\pi f_{cR}t + \phi_{est})$$
$$= \cos(2\pi f_{cT}t + \phi_i)\cos(2\pi f_{cR}t + \phi_{est}) + n(t)_{Re}\cos(2\pi f_{cR}t + \phi_{est})$$
$$= (1/2)\cos\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} + (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$$
$$+ n_{Re}(t)\cos(2\pi f_{cR}t + \phi_{est})$$

After low pass filter process

$$y_{Re}(t) = (1/2)\cos\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\} + n(t)_{Re}\cos(2\pi f_{cR}t + \phi_{est})$$
$$= (1/2)\cos(2\pi\Delta f t + \Delta\phi) + n_{Re}(t)\cos(2\pi f_{cR}t + \phi_{est})$$

Where the Δf is different value between frequency carrier at transmitter and receiver parts, and $\Delta\phi$ is different value between initial phase at transmitter and receiver parts.

From the imaginary part, the product output is:

$$Y_{Im}(t) = [s(t) + n_{Im}(t)](\sin(2\pi f_{cR}t + \phi_{est}))$$
$$= \cos(2\pi f_{cT}t + \phi_i)(\sin(2\pi f_{cR}t + \phi_{est})) + n(t)_{Im}\sin(2\pi f_{cR}t + \phi_{est})$$
$$= (1/2)\sin\{2\pi(f_{cT} + f_{cR})t + (\phi_i + \phi_{est})\} - (1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\}$$
$$+ n_{Im}(t)\sin(2\pi f_{cR}t + \phi_{est})$$

After low pass filter process

$$y_{Re}(t) = (-1/2)\sin\{2\pi(f_{cT} - f_{cR})t + (\phi_i - \phi_{est})\} + n_{Im}(t)\sin(2\pi f_{cR}t + \phi_{est})$$
$$= (-1/2)\sin(2\pi\Delta f t + \Delta\phi) + n_{Im}(t)\sin(2\pi f_{cR}t + \phi_{est})$$

Where the Δf is different value between frequency carrier at transmitter and receiver parts, and $\Delta\phi$ is different value between initial phase at transmitter and receiver parts.

Multiplying the two outputs of low pass filter generates an error signal. Thus,

$$e(t) = y_{Re}(t)\, y_{Im}(t)$$
$$=[(1/2)\cos(2\pi\Delta f t+\Delta\phi) + n_{Re}(t)\cos(2\pi f_{cR}t+\phi_{est})][(-1/2)\sin(2\pi\Delta f t+\Delta\phi) + n_{Im}(t)\sin(2\pi f_{cR}t+\phi_{est})]$$
$$= (-1/4)\cos(2\pi\Delta f t + \Delta\phi)\sin(2\pi\Delta f t + \Delta\phi) - n(t)_{Re}\cos(2\pi f_{cR}t+\phi_{est})(1/2)\sin(2\pi\Delta f t+\Delta\phi) -$$
$$(1/2)\cos(2\pi\Delta f t+\Delta\phi)\, n_{Im}(t)\sin(2\pi f_{cR}t+\phi_{est}) + n_{Re}(t)\cos(2\pi f_{cR}t+\phi_{est})\, n_{Im}(t)\sin(2\pi f_{cR}t+\phi_{est}).$$

We note that the error signal into the loop filter consists of the desired term (-1/4)cos(2πΔ*f*t +Δϕ) sin(2πΔ*f*t +Δϕ) plus terms that involve *signal* x *noise* and *noise* x *noise*. It is interesting to note that the optimum low pass filter for rejecting the double-frequency terms in the Costas Loop filter is a filter matched to the signal pulse in the information-bearing signal. If matched filters are employed for the low-pass filters, their outputs could be sampled at the bit rate, at the end of each signal interval, and the discrete-time signal samples could be used to drive the loop. The use of the matched filter results in a smaller noise into the loop. And it will give the matched filter condition where is signal >> noise. The signal to drive VCO can be rewritten as:

$$e(t) = (-1/4)\cos(2\pi\Delta f t + \Delta\phi)\sin(2\pi\Delta f t + \Delta\phi)$$
$$= (-1/8)\sin\{2(2\pi\Delta f t + \Delta\phi)\}$$

The new output phase of VCO is given as

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

or

$$2pf_{cR}t + f_{est} + K\int_0^t \sin\left(2(2p\Delta ft + \Delta f)\right)dt$$ where is K = -Kc/8.

The loop phase error and its time derivative is

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

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

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

$$\frac{d\left(q_e(t)\right)}{dt} = -\sin\left(2\Delta f\right)$$

The null value will be happen at 0 and π radiant, and it's known as two phase ambiguity of phase-carrier recovery at BPSK system.

3.2. Session 2: Question and Answer

Question 1

I want to build a simulation program for BPSK system, would should I do? I must build in the pass band system or base band system?

Answer:

A simulation program is not the real process, but it must represent the properties of the real process. To build in pass band system is difficult, it will be better to build the simulation in the base band system, especially for the beginner in programming. We made an assumption that our system is in ideal condition, the meaning is transmitter filter, receiver filter, modulator, and demodulator process do well. The base band system is available to represent the real communication system.

Question 2:

What is the relationship among the transmitter filter, receiver filter and root cosine roll off filter?

Answer:

The output signal from transmitter is a sequence of:

$s(t)$ = (sum of sequence) $\{a(i)\ h_t(t - iT)\}$

Information filter transfer function

For ideal channel condition we can illustrate the block bloc diagram of communication system as follows:

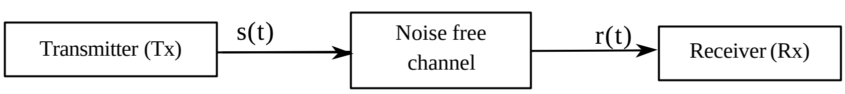

Figure (3.6) block diagram of communication system

The propagation channel has a function h(t), and the additive noise has a function n(t). The received signal will be:

$$r(t) = h_r(t)\ (*)\ h(t)\ (*)\ s(t) + h_r(t)\ (*)\ n(t) \quad (*)\text{:convolution}$$

$$= h(t)\ (*)\{h_r(t)\ (*)s(t)\} + h_r(t)\ (*)\ n(t) \qquad (3\text{-}8)$$

If the channel propagation has no effect to the transmitted signal, and as we know our assumption is no noise, we will set $h_r(t)\ (*)\ n(t) = 0$, and h(t) equal with 1.

Our assumption, between transmitter filter and receiver filter are in matched condition, so the combination of these two filters can represent by a root cosine roll off filter.

$$h_c(t) = h_t(t)\ (*)\ h_r(t) = [\sin(\pi t/T)/(\pi t/T)]\ x\ [\cos(\pi\alpha t/T)/\ (1\text{-}\ (2\pi\alpha t/T)^2)] \qquad (3\text{-}9)$$

where $0 \leq \alpha \leq 1$ is a roll off factor. And t has a value: $t = 0 \rightarrow h_c(0) = 1$

$t = \pm T/(2\alpha) \rightarrow h_c(\pm T/(2\alpha)) = \alpha/2$; where α is not 0.

The simple description from this formulation is as figure (3.7).

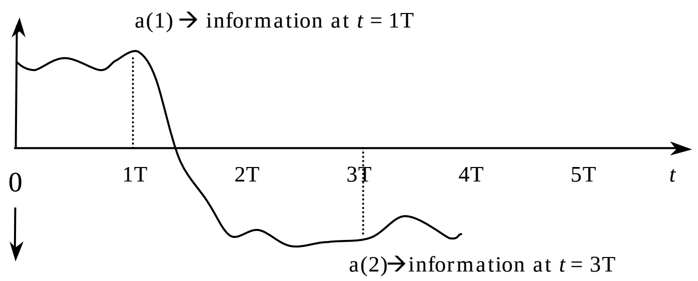

Figure (3.7) the relationship of information sequence and sampling

We can understand that put the sample of received signals at t = 1T or 3T to get the information sequence from transmitter. Remember, this is just for simplification in our image, may be the real system has the different conditions.

Question 3:

Where we set the position of AWGN noise in simulation system?

Answer:

As we know, our system is base band. And AWGN noise occupied all of the frequency spectral. After we filtered the signal at receiver by using LPF (low pass filter), the output is still content of AWGN noise. To set the position of AWGN noise in our simulation, we can set as Figure (3.8).

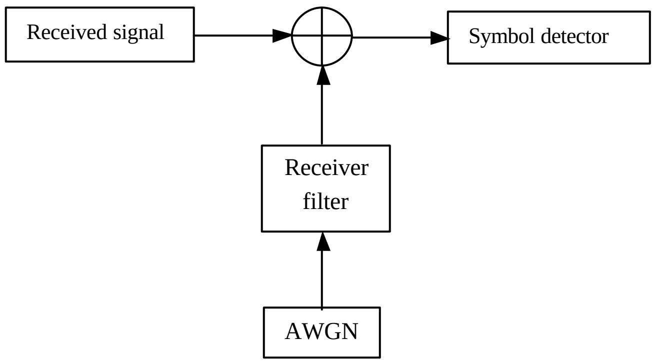

Figure (3.8) Setting the AWGN noise in simulation

The output from AWGN is n(t), the transfer function of receiver Filter is hr(t), and the output of received filter is

$\hat{n}(t) = hr(t) \otimes n(t)$; where is the sign $\otimes$ indicate the convolution operation.

By using transformation fourrier we will get this equation as

$$\hat{N}(f) = Hr(t) \cdot n(t) \qquad (1)$$

From the autocorrelation of noise:

$$\int_{-\infty}^{\infty} \hat{n}(t)\hat{n}^*(t-t)dt = \int df \left|\hat{N}(f)\right|^2 e^{j2pft}\, df \qquad (2)$$

Substituting (1) and (2) we get

$$\begin{aligned} <\hat{n}(t)\hat{n}^*(t-t)> &= \int df \left|H_r(f)\right|^2 \left|N(f)\right|^2 e^{j2pft}\, df \\ &= No\int df \left|H_r(f)\right|^2 e^{j2pft}\, df \\ &= No\, h(t) \end{aligned}$$

Where the $|H_r(f)|^2$ is Fourrier Transformation of Cosine Roll-Off, and h(t) is impulse response of Cosine Roll-off.

The value of h(τ) = h(nT), which has a value “1” for n = 0 and has a value “0” for the other n.

The above equation can be rewritten as

$$<\hat{n}(t)\hat{n}^*(t-nT)> = Noh(nT)$$

If $\hat{n}(t)$ is sampled at the sampling rate 1/T, the sampled filtered noise will be white.

3.3. Session 3: Simulation of BPSK System

The simulation for BPSK transmission system using base band system, and here we use a coherent detection. As we know the real system must use carrier recovery for at because the BPSK using DSBSC, and in the DSBSC carrier frequency is not transmitted so the receiver didn't know the frequency of transmitter signal exactly. But for this topic, we want to start from the lowest level of simulation program for digital modulation.

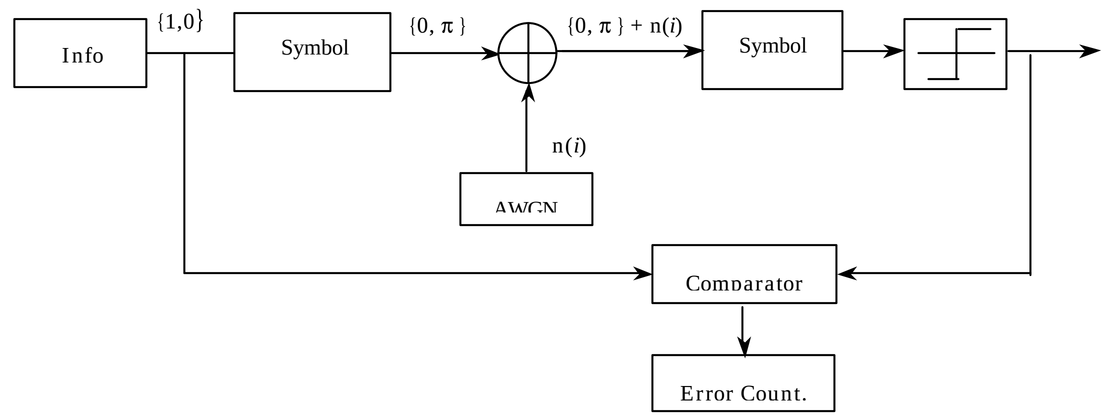

Figure (3.9) Block Diagram for BPSK Simulation

3. 3.1. Algorithm

To make a simulation program for BPSK with coherent detection we must build several functions to represent the function of block diagram at the Figure (3.9).

♦ **Info Generator**

Bit information generated randomly by PN generator function. The random number generated is set between value 0 and 1 as the binary number to represent digital signal. By using Microsoft C++ software, this function will generate a uniform random binary number.

♦ **Symbol Generator**

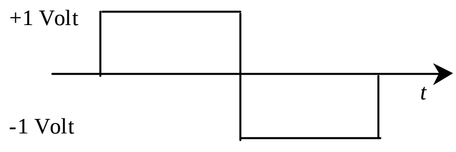

Figure (3.10) Baseband antipodal for BPSK

By using Figure (3.6) as a reference, and put a sample at t = 1. We can get the amplitude value 1. Other wise if we put a sample at t = 3, we will get the amplitude value –1. By view the waveform, we look that in the time interval t = 0 until t = 2, the waveform has an initial phase 0^o. And in the time interval t = 2 until t = 4, the waveform an initial phase 180^o or π radiant. By using this relationship we can set that +1 to represent the BPSK modulated signal by initial phase 0 and –1 to represent the BPSK modulated signal by initial phase π radiant or logic "1" and "1" from PN generator respectively. We can do it by using the function *int symbol_generate (int r)*. Finally the BPSK modulated signal can be represented as baseband antipodal signals.

- **AWGN Channel**

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.*

In the BPSK system we only have the real channel, so we didn't use the *im* part. The effect of AWGN noise will change the level of transmitted signal as like Figure (3.11).

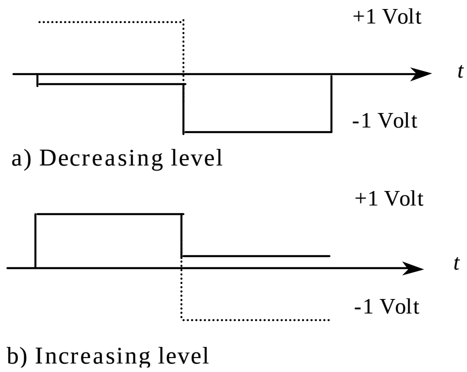

Figure (3.11) Effect AWGN at the level of antipodal signal transmission

- **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 between 0 and π radiant. Using this property the algorithm looks for one of two possible phases. When it matched we must resolve by de-rotate the phase.

- **Signal to Noise Ratio Calculation**

Signal to noise ratio is a ratio of power average of signal transmitted and noise power average. In case of BPSK the power average of signal transmitted is 1 and the average power noise is equivalent with variant value. The result of this calculation is in dB value..

- **Symbol Recover and Decision**

This part is combination of noise addition and bit recover. Using input from the *amplitude_nosie* and *symbol_generate* functions, we add the both variable; Rx = signal + noise. The result then we compare to the threshold level 0. Rx >= 0 we make a decision the bit received as "1" otherwise if Rx < 0 the bit received as "0".

- **Error Detector**

The bit received then detect bi compare it to the original bit information at the transmitter part. If bit received has a same value with the bit information, the error is not happen, other wise if the value of bit received is different with the bit information the error is happen. This function represent the part Comparator in the Figure (3.9).

- **BER Calculation**

The purpose of this part is to get ho many errors was happened during bit transmission process and compare this total error value to the total bit was transmitted. The result is known as *bit error rate* (BER), and the value is between 0 and 1. If we transmit 1000000 bits and 100 bits received by error, we get the information the *bit error rate* 100/1000000 = 10^{-4}.

- **Pb Calculation**

This part is to calculate the value of probability of error that will happen if we transmit a signal through an AWGN channel, which has some value of noise variants. By the equation (3-5) we can calculate this value. But *erfc* or *Q-Marcum* function can't found at the library of C, so we must use an formulation approximation:

$$erfc(x) = \frac{e^{-x^2}}{x\sqrt{p}}\left[1 - \frac{1}{2x^2} + \frac{1*3}{2^2x^4} - \frac{1*3*5}{2^3x^6} + ...\right]$$

In the case of BPSK, this equation can rewrite as this equation:

$$P_e = \frac{1}{2}\frac{e^{-x^2}}{x\sqrt{p}}\left[1 - \frac{1}{2x^2} + \frac{1*3}{2^2x^4} - \frac{1*3*5}{2^3x^6} + ...\right]$$

where is $x = \sqrt{E_b / N_o}$

3. 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.

Re_info_Rx = information recovered by receiver part.

pi = π radiant.

Re_Tx = symbol generate by transmitter part.

pha_Tx = phase generate by transmitter part

Re_AWGN = noise AWGN

Re_Rx = symbol received at receiver part

Re_anti_Rx = symbol recovered at receiver part in the antipodal format

pha_receiv = phase receive after transmission

phase_error = phase error at a synchronization process

Ave_Pha_err = average phase error in the all the synchronization process

SNR = Signal to Noise Ratio

BER = bit error rate

Pb_BPSK = probability of error of BPSK system

The listing program for simulation of BPSK transmission system is as follows.

```
//Simulation Program for BPSK
//by Tri Budi Santoso
//Mobile Communication Laboratory, Tokyo Institute of Technology, Japan
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define S 1000000
int nn=S-1,i;
```

```
int Re_info[S],Re_info_Rx[S];
double pi=acos(-1.0),var=0.0;
double Re_Tx[S],pha_Tx[S],Re_Rx[S],Re_anti_Rx[S],Re_AWGN[S];
double pha_receiv,deg_error,phase_error[S];
double SNR,Pha_err,Ave_Pha_err,BER, Pb_BPSK;

void info_generate(int Re_info[]);
void symbol_generate(double Re_Tx[],double pha_Tx[]);
void costas(double phase_error[],double *pha_er);
void synchronization(double pha_receiv,double *deg_er);
void AWGN_channel(double Re_AWGN[]);
void received_symbol(double Re_Rx[]);
void decission(double Re_anti_Rx[]);
void info_recover(int Re_info_Rx[]);
void error_rate(double *ber);
void signal_noise_ratio(double *snr);
void Pb_theory(double Ave_Pha_err, double *pb);

void main()
{
 info_generate(Re_info);
 symbol_generate(Re_Tx,pha_Tx);
 costas(phase_error,&Pha_err);
  AWGN_channel(double Re_AWGN[]);
  received_symbol(Re_Rx);
 decission(Re_anti_Rx);
 info_recover(Re_info_Rx);
 error_rate(&BER);
 signal_noise_ratio(&SNR);
  Pb_theory(Ave_Pha_err ,&Pb_BPSK)
}

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

void symbol_generate(double Re_Tx[],double pha_Tx[])
{
 for (i=1;i<=nn;++i)
 {
   Re_Tx[i]=((double)Re_info[i]-0.5)*(-2.0);
   pha_Tx[i]=acos(Re_Tx[i])/2/pi*360.0;
 //printf("\nRe_Tx[%d]:%3.2f  pha_Tx[%d]:%4.2f", i,Re_Tx[i],i,pha_Tx[i]);
 }
}

void costas(double phase_error[],double *pha_er)
{
   Pha_err=0.0;
```

```
 for(i=1;i<=nn;++i)
 {
   pha_receiv=pha_Tx[i];
   synchronization(pha_receiv,&deg_error);
   phase_error[i]=deg_error;
// printf("\n phase_error[i]:%f",phase_error[i]);
   Pha_err += fabs(phase_error[i]);
 }
 *pha_er=Ave_Pha_err;
}

void synchronization(double pha_revceiv,double *deg_er)
{
 int d=1,i=1,symbol_rate=16000;
 double rad_error,fc=800e+6,T,delta_fc,phase_estim=0.0;
 double VCO,K=1.0;
 T=1.0/(double)symbol_rate;
 delta_fc=1e-6 * fc *T;
 printf("\n\tpha_receiv:%f",pha_receiv);
   do
    {
        deg_error=(delta_fc + pha_receiv - phase_estim);
        if (deg_error < -45.00)
        {deg_error=deg_error + 360.0;}
        else if (deg_error > 360.0)
        {deg_error=deg_error - 360.0;}
        else if ((deg_error>0.00) && (deg_error<360.00))
        {deg_error=deg_error;}
        else
        printf("");
        rad_error=deg_error/360*2*pi;
        VCO=K*sin(2*rad_error);
        phase_estim = phase_estim + VCO;
        i++;
    }while(i<=4);

if(deg_error>-90.00 && deg_error<=90)
{deg_error=deg_error;}
else if(deg_error>90.00 && deg_error<=270.00)
{deg_error=deg_error - 180.0;}
else {printf("");}
 *deg_er=deg_error;
}
void AWGN_channel(double Re_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);
 }
```

```
}

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

void decission(double Re_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;
}
}

void info_recover(int Re_info_Rx[])
{ double re_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;
  //printf("\nRe_info_Rx[%d]:%d",i,Re_info_Rx[i]);
 }
}


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

void signal_noise_ratio(double *snr)
{
   SNR=10*log10(1/var);
   *snr= SNR;
}
```

```
void Pb_theory(double Ave_Pha_err, double *pb)
{
double gama, term_1,term_3,term_5,costas_effect;
   costas _effect=cos(Ave_Pha_err/360*2*pi);
   gama= costas _effect *(1.0/var)
  term_1= 1.0/((2.0)*pow(gama,1));
   term_3= (1.0*3.0)/((2.0)*(2.0)*pow(gama,2));
   term_5= (1.0*3.0*5.0)/((2.0)*(2.0)*(2.0)*pow(gama,3));
   Pb_BPSK=(1.0/(2.0*sqrt(pi*gama)))*exp(-gama)*(1-term_1+term_3-term_5);
*pb= Pb _BPSK;
}
```

3. 3.3. The Simulation Result

We can run the program after set the bit information = 1000000. It means that we send the 1000000 bits information every variants value. After run and change the value of variants from 0.5 until 0.1, we get the result as like in the Table1.

Table 3.1. The result of BPSK Simulation

SNR (in dB)	Pb:	BER
3.01	0.018981	0.02304100
3.47	0.015504	0.01769600
3.98	0.011716	0.01274100
4.56	0.008018	0.00851700
5.23	0.004778	0.00503500
6.02	0.002306	0.00241900
6.99	0.000778	0.00087100
8.24	0.000130	0.00014400
9.03	0.000032	0.00004500
10.00	0.000004	0.00001600

Some different value between theoretical (Pb) values with simulation result (BER) was happen.

For the variants value 0.5, which give the energy bit per noise (E_b/N_o) = 3,01 dB. The P_b shows the value; the meaning is if we send 1000000 bits information, probability of total bit will be received by error are 18981 bit. But the simulation result showed the value 238 bit were received by error or BER = 0.023041. From this data we look that the different value is 0.02304100 - 0.018981= 0,00406 or in the percent value is 21%.

For a value of E_b/N_o = 6,99 dB. The theoretical estimate Pb = 0.000778, the meaning is if we send 1000000 bits information data yang the probability of bit received by error are 778 bits. But the simulation result showed the value of BER is 0.00087100. From this data we get the different value is 0.00087100 - 0.000778 = 0.000093 or in the percent value is 11.9%.

Performance of BPSK Coherent

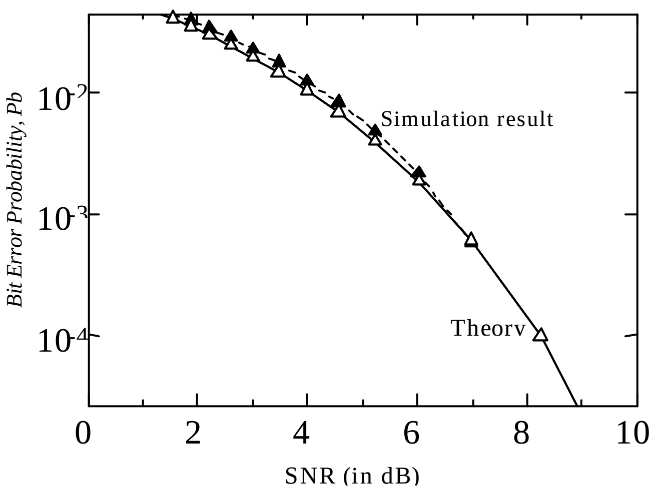

Figure (3.12) Simulation result of BPSK performance in the AWGN Channel

The trend of percent error of simulation error and the theoretical value is decrease if we increase the value of Eb/No. And from the result of simulation that indicates the small deviation from theoretical value, it is indicate that this simulation has good performance.

By using the Table 1 we can build a graph for the BPSK performance from the simulation result as Figure (3.12). From this figure we see that the simulation result is very closed to the theoretical value.