Digital Communications Lab

Laboratory report submitted for the partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

Kartikeya Acharya 21UEC072

Course Coordinator
Dr. Anirudh Agrawal/ Dr. Santosh Shah/ Dr. Soumitra Debnath

LNMIIT
The LNM Institute of Information Technology

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

September 2023

Contents

Chapter *1*

Experiment - 07

1 Aim

1. Performance analysis of BPSK/QPSK/QAM over AWGN channel.

2 Software Used

* MAT LAB

3 Theory

3.1 Binary phase-shift keying (BPSK)

BPSK (also sometimes called PRK, phase reversal keying, or 2PSK) is the simplest form of phase shift keying (PSK). It uses two phases which are separated by 180° and so can also be termed 2-PSK. It does not particularly matter exactly where the constellation points are positioned, and in this figure they are shown on the real axis, at 0° and 180°. Therefore, it handles the highest noise level or distortion before the demodulator reaches an incorrect decision. That makes it the most robust of all the PSKs. It is, however, only able to modulate at 1 bit/symbol (as seen in the figure) and so is unsuitable for high data-rate applications. Yet there is the possibility of extending this bit/symbol, given the modulators symbol encryption / decryption logic system.

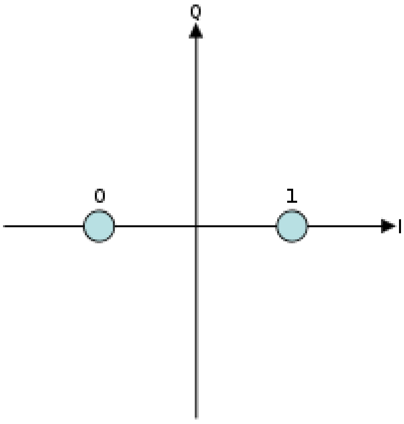


Figure 1: Constellation diagram example for BPSK

Implementation

The general form for BPSK follows the equation:

$$s_n(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f t + \pi(1 - n)), \quad n = 0, 1. \tag{1}$$

This yields two phases, 0 and π. In the specific form, binary data is often conveyed with the following signals:

$$s_0(t) = \sqrt{\frac{2E_b}{T}} \cos(2\pi f t + \pi) = -\sqrt{\frac{2E_b}{T}} \cos(2\pi f t), \quad for binary "0" (2)$$

Link: https://en.wikipedia.org/wiki/Phase-shift$_k$*eyingBinary*$_p$*hase* − *shift*$_k$*eying*$_($*BPSK*)

3.2 Quadrature phase-shift keying (QPSK)

Sometimes this is known as quadriphase PSK, 4-PSK, or 4-QAM. (Although the root concepts of QPSK and 4-QAM are different, the resulting modulated radio waves are exactly the same.) QPSK uses four points on the constellation diagram, equispaced around a circle. With four phases, QPSK can encode two bits per symbol, shown in the diagram with Gray coding to minimize the bit error rate (BER) – sometimes misperceived as twice the BER of BPSK.

The mathematical analysis shows that QPSK can be used either to double the data rate compared with a BPSK system while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK but halving the bandwidth needed. In this latter case, the BER of QPSK is exactly the same as the BER of BPSK – and believing differently is a common confusion when considering or describing QPSK. The transmitted carrier can undergo numbers of phase changes.

Given that radio communication channels are allocated by agencies such as the Federal Communi- cations Commission giving a prescribed (maximum) bandwidth, the advantage of QPSK over BPSK becomes evident: QPSK transmits twice the data rate in a given bandwidth compared to BPSK - at the same BER. The engineering penalty that is paid is that QPSK transmitters and receivers are more complicated than the ones for BPSK. However, with modern electronics technology, the penalty in costis very moderate.

Implementation The implementation of QPSK is more general than that of BPSK and also indicates the implementation of higher-order PSK. Writing the symbols in the constellation diagram in terms of the sine and cosine waves used to transmit them:

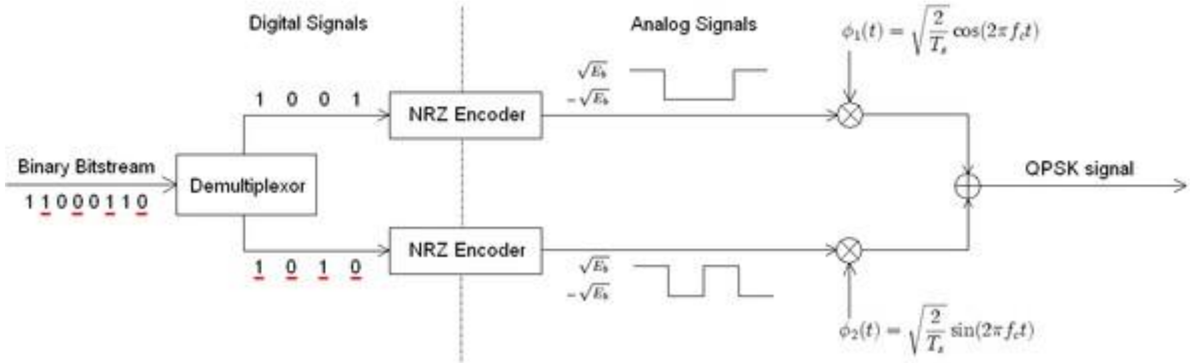


Figure 2: Transmitter structure for QPSK

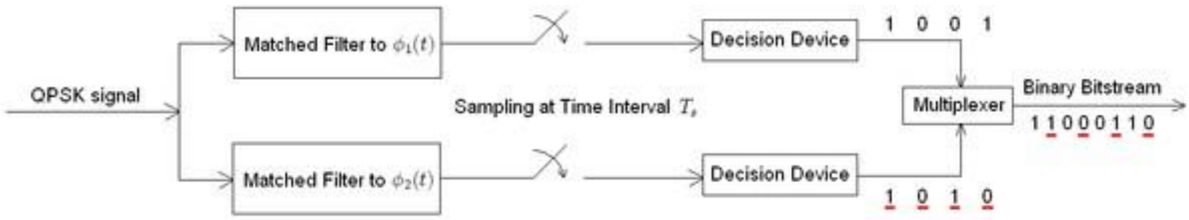
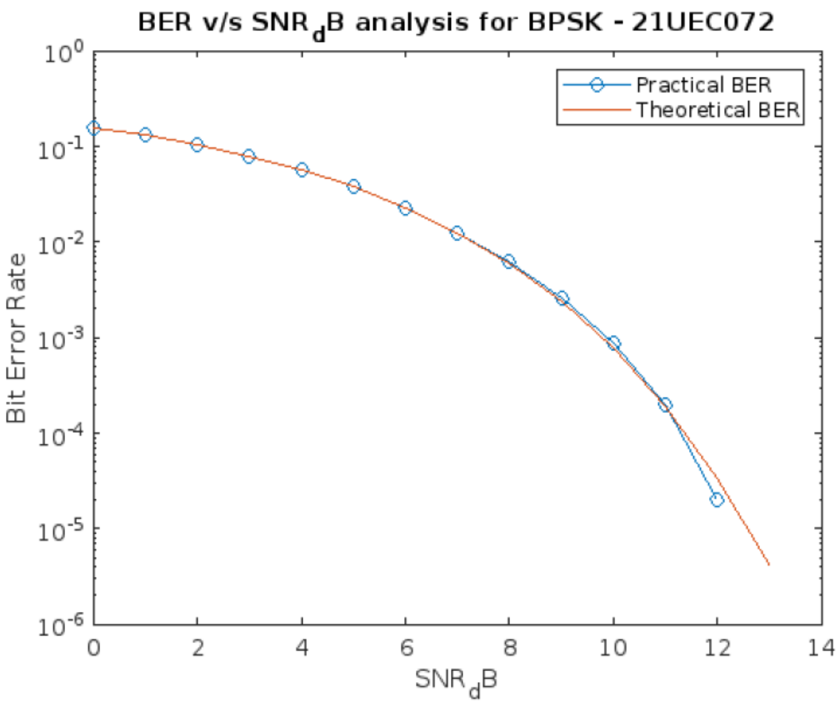


Figure 3: Receiver structure for QPSK

4 Observation

BPSK

```
clc
close all
clear all

SNRdB = 0:1:13;
SNR = 10.^(SNRdB/10);
N = 1e5;
x = randsrc(1, N, [0,1]);
n = randn(1, N);
x_bpsk = (1 - 2*x);

for k = 1:length(SNR)
    y = sqrt(SNR(k))*x_bpsk + n;
    noisy_bits = y.*x_bpsk;
    indices_corrupted = find(noisy_bits<0);
    num_corrupted_bits(k) = length(find(indices_corrupted));
end

ber = num_corrupted_bits/N;
ber_th = qfunc(sqrt(SNR));

semilogy(SNRdB, ber, 'o-', 'DisplayName', 'Practical BER');
hold on;
semilogy(SNRdB, ber_th, 'DisplayName', 'Theoretical BER');
xlabel("SNR_dB");
ylabel("Bit Error Rate");
title("BER v/s SNR_dB analysis for BPSK - 21UEC072");
legend;
```



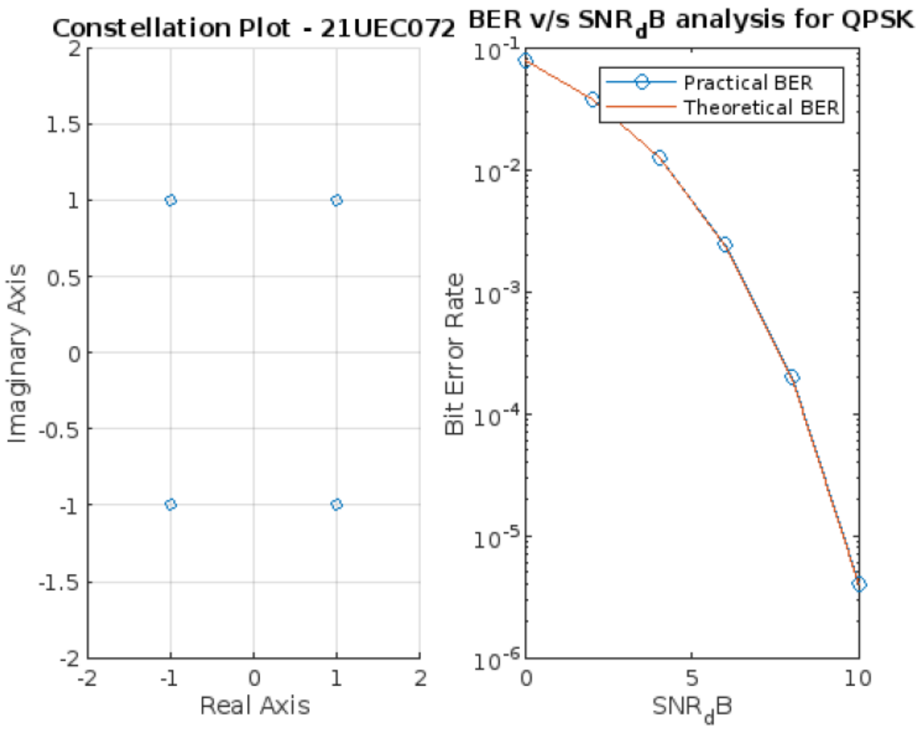
Published with MATLAB® R2023b

QPSK

```
clc
close all
clear all

SNRdB = 0:2:10;
SNR = 10.^(SNRdB/10);
N = 2e6;
n = (randi([0,3], 1, N)*pi/2) - pi/4;
x_qpsk = sqrt(2)*(cos(n) + (1j)*sin(n));

data = randi([0, 1], 1, N);
data = reshape(data, 2, N/2);

% QPSK modulation
I = 2 * data(1, :) - 1; % In-phase component
Q = 2 * data(2, :) - 1; % Quadrature component
mod_sig = I + 1i * Q;
bitErrors = zeros(1, length(SNRdB));

for snrIndex = 1:length(SNRdB)
    noise = sqrt(0.5 / (10^(SNRdB(snrIndex) / 10))) * (randn(1, N/2) + 1i *
randn(1, N/2));
    received_signal = mod_sig + noise;
    % QPSK demodulation
    demod_i = real(received_signal) > 0;
    demod_q = imag(received_signal) > 0;
    demodulatedData = [demod_i; demod_q];
    demodulatedData = demodulatedData(:)'; % Reshape into a row vector
    % Calculate bit errors
    bitErrorsI = sum(demod_i ~= data(1, :));
    bitErrorsQ = sum(demod_q ~= data(2, :));
    bitErrors(snrIndex) = bitErrorsI + bitErrorsQ;
end
BER = bitErrors / N;

subplot(1,2,1);
scatter(real(x_qpsk),imag(x_qpsk),20);
grid on;
axis([-2 2 -2 2]);
xlabel("Real Axis");
ylabel("Imaginary Axis");
title("Constellation Plot - 21UEC072");

subplot(1,2,2);
semilogy(SNRdB, BER, 'o-', 'DisplayName', 'Practical BER');
hold on;
EbN0 = 10.^(SNRdB/10);
BER_Th = 0.5 * erfc(sqrt(EbN0));
semilogy(SNRdB, BER_Th, 'DisplayName', 'Theoretical BER');
xlabel("SNR_dB");
ylabel("Bit Error Rate");
```



```
title("BER v/s SNR_dB analysis for QPSK");
legend;
```



Published with MATLAB® R2023b

Precautions

Observation should be taken properly.