

디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001



9 주차 실습과제

정보통신공학과

12191765

박승재

Introduction

QPSK 는 동시에 두 비트를 하나의 심볼로 변조하는 PSK 기반의 변조 방식이다. QPSK 변조 방식을 활용하면 binary 비트 쌍 (b_1, b_2) 는 아래와 같이 변조된다.

Index n	Bit pair	symbol	Index n	Bit pair	symbol
1	(1,1)	$s_1(t)$	3	(0,0)	$s_3(t)$
2	(0,1)	$s_2(t)$	4	(1,0)	$s_4(t)$

$$\begin{cases} s_1(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f t - \frac{\pi}{4}\right) \\ s_2(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f t - \frac{3\pi}{4}\right) \end{cases} \quad \begin{cases} s_3(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f t - \frac{5\pi}{4}\right) \\ s_4(t) = \sqrt{\frac{2E_s}{T_s}} \cos\left(2\pi f t - \frac{7\pi}{4}\right) \end{cases}$$

이번 실습에서는 QPSK 변조/복조를 통해 Symbol Error Rate 와 Bit Error Rate 을 측정해볼 것이다.

```
Eb_No_dB = 10;  
Eb_mW = 1;  
Eb_dBm = pow2db(Eb_mW);  
No_dBm = Eb_dBm - Eb_No_dB;  
No_mW = db2pow(No_dBm);
```

E_b 는 0 dBm/Hz 로 고정하고 E_b/N_0 에 맞도록 N_0 값을 정한다.

```
N = 10000; % # of symbols  
symbols = zeros(2, N);  
symbol_errors = zeros(1, N);  
bit_errors = zeros(1, N);  
  
for i = 1:N  
    bits = rand(2, 1) >= 0.5; % random binary bit: 0 or 1  
    encoded_bits = bits * 2 - 1; % random binary bit: -1 or 1  
  
    Es_mW = 2 * Eb_mW; % symbol energy  
    symbol = sqrt(Es_mW / 2) * (encoded_bits(1) + 1j * encoded_bits(2)); %  
    QPSK  
  
    noise = sqrt(No_mW / 2) * (randn() + 1j * randn()); % noise  
    y = symbol + noise;  
  
    symbols(:, i) = [real(y); imag(y)]; % generated symbol  
  
    decoded_symbol = ((real(y) > 0) * 2 - 1) + 1j * ((imag(y) > 0) * 2 - 1);
```

```

symbol_error = (symbol ~= decoded_symbol);
symbol_errors(i) = symbol_error; % error symbol number

% decision device
reconstructed_bits = [real(decoded_symbol) > 0; imag(decoded_symbol) >
0];

bit_error = sum(bits ~= reconstructed_bits);
bit_errors(i) = bit_error; % error bit number
end

```

10000 개의 심볼을 비교하여 SER 과 BER 을 측정할 것이다. QPSK 는 2 비트를 사용하기 때문에 E_s 는 E_b 를 2 배 해야 한다. 오류가 발생한 심볼과 비트 개수는 `symbol_errors`와 `bit_errors`에 저장된다.

```

ser = sum(symbol_errors) / N
ber = sum(bit_errors) / (2 * N)
ser_ = erfc(sqrt(db2pow(Eb_No_dB)))
ber_ = berawgn(Eb_No_dB, 'psk', 4, 'nodiff')

```

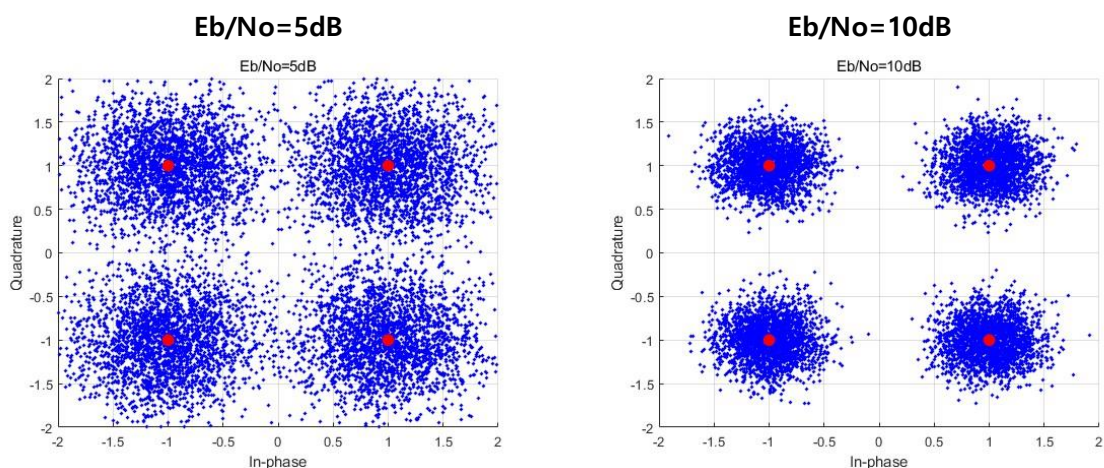
`symbol_errors`와 `bit_errors`를 이용해 SER 과 BER 을 계산한다. `ser_`과 `ber_`는 수식으로 계산한 이론값이다.

```

grid on
hold on
qpsk_symbols = [1 1 -1 -1; 1 -1 1 -1];
plot(symbols(1, :), symbols(2, :), '*b', 'MarkerSize', 2)
plot(qpsk_symbols(1, :), qpsk_symbols(2, :), 'o', 'MarkerSize', 8,
'MarkerFaceColor', 'r', 'MarkerEdgeColor', 'r')
axis([-2 2 -2 2])
title(['Eb/No=' int2str(Eb_No_dB) 'dB'])
xlabel('In-phase')
ylabel('Quadrature')

```

QPSK 로 변조된 심볼의 위치를 좌표계에 나타낸다. 빨간색 점은 오류가 발생하지 않은 심볼이고, 파란색 점은 오류가 발생한 심볼이다.



$E_b/N_0=5\text{dB}$ 와 $E_b/N_0=10\text{dB}$ 를 비교해 보았다. E_b/N_0 가 클수록 N_0 값이 작아진다. (E_b 는 고정이므로) 따라서 $E_b/N_0=5\text{dB}$ 보다 $E_b/N_0=10\text{dB}$ 가 오류가 적기 때문에 파란색 점이 빨간색 점에 몰려서 찍히게 된다. 파란색 점이 퍼져서 찍히는 것은 큰 오류가 발생하는 것을 의미한다.

Problem

1. E_b/N_0 를 0dB 부터 10dB 까지 1dB 간격으로 증가시키면서 SER 과 BER 측정
 - MATLAB 내 built-in 함수를 활용하여 QPSK 의 이론적 BER 값을 구하기
 - ✓ `berawgn(Eb_No_dB_vec,'psk',4,'nondiff')`
 - ✓ $SER=1-(1-BER)^2$ 관계임
 - X 축은 E_b/N_0 [dB], Y 축은 SER/BER 값을 log-scale 로 그리기
 - y 축은 $1e-5$ 까지만 보기
 - 실험을 통해 얻은 SER/BER 은 marker 를 사용해서 그리기
 - 이론값은 선으로 그리기
2. BPSK 와 QPSK 비교하기

Result

```
Eb_No_dBs = 0:10;
bers = zeros(1, length(Eb_No_dBs));
sers = zeros(1, length(Eb_No_dBs));

j = 1;
for Eb_No_dB = Eb_No_dBs
    Eb_mW = 1;
    Eb_dBm = pow2db(Eb_mW);
    No_dBm = Eb_dBm - Eb_No_dB;
    No_mW = db2pow(No_dBm);

    N = 10000; % # of symbols
    symbols = zeros(2, N);
    symbol_errors = zeros(1, N);
    bit_errors = zeros(1, N);

    for i = 1:N
```

```

bits = rand(2, 1) >= 0.5; % random binary bit: 0 or 1
encoded_bits = bits * 2 - 1; % random binary bit: -1 or 1

Es_mW = 2 * Eb_mW; % symbol energy
symbol = sqrt(Es_mW / 2) * (encoded_bits(1) + 1j *
encoded_bits(2)); % QPSK

noise = sqrt(No_mW / 2) * (randn() + 1j * randn()); % noise
y = symbol + noise;

symbols(:, i) = [real(y); imag(y)]; % generated symbol

decoded_symbol = ((real(y) > 0) * 2 - 1) + 1j * ((imag(y) > 0) * 2 -
1);

symbol_error = (symbol ~= decoded_symbol);
symbol_errors(i) = symbol_error; % error symbol number

reconstructed_bits = [real(decoded_symbol) > 0; imag(decoded_symbol)
> 0]; % decision device

bit_error = sum(bits ~= reconstructed_bits);
bit_errors(i) = bit_error; % error bit number
end

bers(1, j) = sum(bit_errors) / (2 * N);
sers(1, j) = sum(symbol_errors) / N;
j = j + 1;
end

bers_ = berawgn(Eb_No_dBs, 'psk', 4, 'nodiff');
sers_ = 1 - (1 - bers_) .^ 2;

figure()
subplot(1, 2, 1)
grid on
hold on
plot(Eb_No_dBs, bers, '*')
plot(Eb_No_dBs, bers_)
set(gca, 'yscale', 'log')
axis([0 10 1e-5 1])
title('QPSK')
xlabel('Eb/No [dB]')
ylabel('BER')
legend('BER(측정값)', 'BER(이론값)')

subplot(1, 2, 2)

```

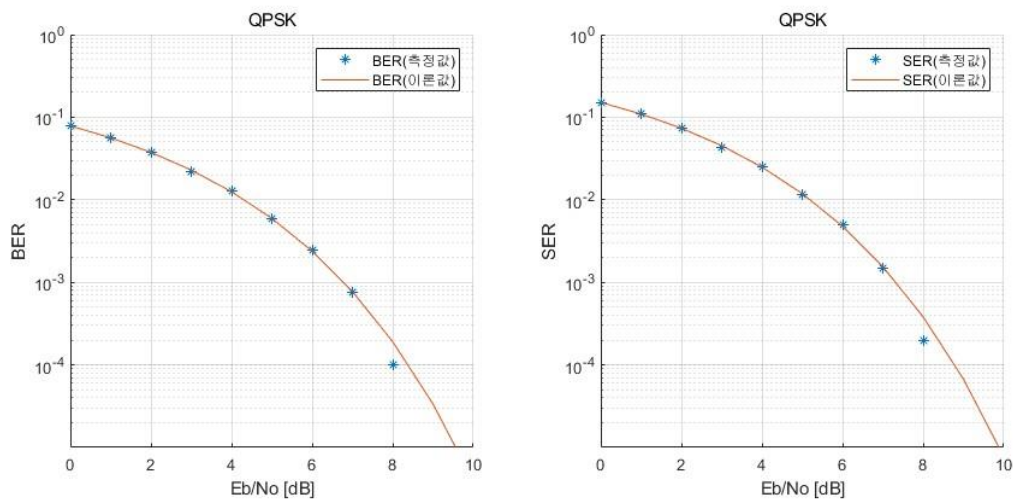
```

grid on
hold on
plot(Eb_No_dBs, sers, '*')
plot(Eb_No_dBs, sers_)
set(gca, 'yscale', 'log')
axis([0 10 1e-5 1])
title('QPSK')
xlabel('Eb/No [dB]')
ylabel('SER')
legend('SER(측정값)', 'SER(이론값)')

figure()
grid on
hold on
plot(Eb_No_dBs, 1/2 * erfc(sqrt(db2pow(Eb_No_dBs))), '-o')
plot(Eb_No_dBs, bers_, '-x')
set(gca, 'yscale', 'log')
axis([0 10 1e-5 1])
title('BPSK vs QPSK')
xlabel('Eb/No [dB]')
ylabel('BER')
legend('BPSK(이론값)', 'QPSK(이론값)')

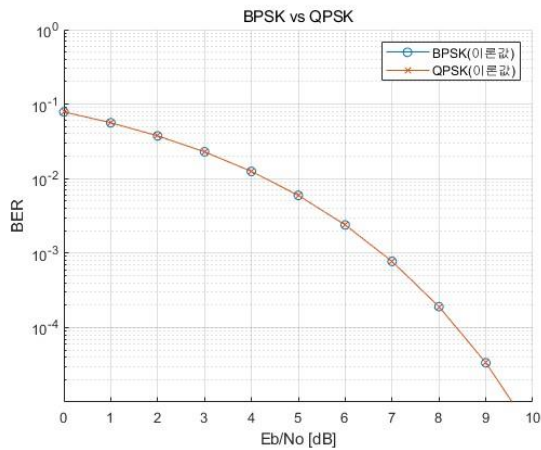
```

E_b/N_0 를 0dB 부터 10dB 까지 1dB 간격으로 증가시키면서 SER 과 BER 측정해 그래프를 그리는 코드이다. BER 과 SER 의 이론 값은 $SER=1-(1-BER)^2$ 관계인 것을 이용해 계산했다.



이론값과 유사하게 측정값 marker 가 그려지는 것을 확인할 수 있었다.

BPSK 는 위상을 2 등분하여 0 or 1 을 보내는 기법이고, QPSK 는 위상을 4 등분하여 00 or 01 or 11 or 10 을 보내는 기법이다. 위상을 더 많이 쪼개서 보내게 되면 하나의 심볼로 더 많은 비트를 전송할 수 있지만, 오류에 더 취약해지게 된다.



BPSK 와 QPSK 의 BER 을 그래프로 그리면 일치하게 나온다. 그 이유는 BPSK 의 BER 은 $P_e = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$ 으로 표현되고, QPSK 는 $P_e \approx \text{erfc} \left(\sqrt{\frac{E}{2N_0}} \right)$ 으로 표현되는데 심볼 하나가 2 비트이기 때문에 $E = 2E_b$ 라 $P_e = \frac{1}{2} \text{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$ 으로 BPSK 와 같은 BER 를 가지기 때문이다.

Conclusion

E_b 가 고정일 때, E_b/N_0 dB 가 증가하는 것은 잡음이 감소한다는 것을 의미한다. 잡음이 감소하면 오류도 감소하므로, E_b/N_0 dB 가 증가할수록 BER 은 감소하는 그래프가 그려지게 된다. $E_b/N_0=9$ dB 일 때는 모두 값이 찍히지 않았는데, 그 이유는 오류가 발생하지 않아 측정 BER 이 0 이 되었기 때문이다. 그래프의 y 축은 log-scale 인데 $\log 0$ 은 정의되지 않으므로 그래프에 표시되지 않는다.

BPSK 실습과 다르게, 이번에는 허수부 계산도 수행하여 QPSK 변조된 점들이 빨간색 점을 중심으로 구름형태로 그려졌다. BPSK 에는 허수부 계산을 하지 않아 수직선 형태로 나타났지만, QPSK 실습에서는 평면상에 원형으로 점들이 찍히게 되었다.

QPSK 는 심볼 하나에 2 비트를 전송하기 때문에 BER 계산에서 `sum(bit_errors) / (2 * N);` N 에 2 를 곱해 나눠줘야 한다. 이는 10000 개의 심볼에는 20000 개의 비트가 들어있기 때문이다.