디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001

14 주차 실습과제

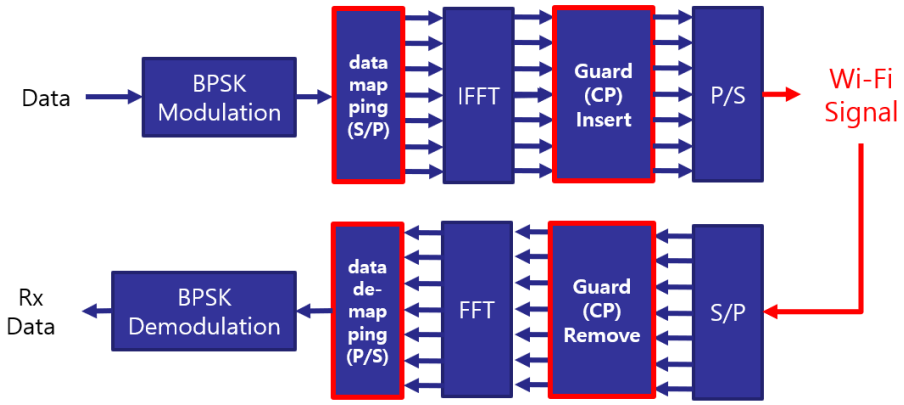
정보통신공학과

12191765

박승재

Introduction

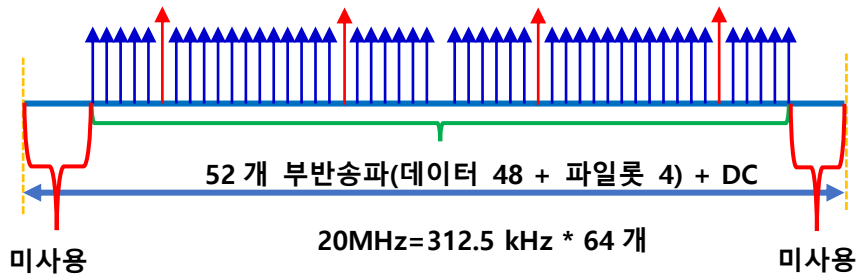
송수신기 구조



- ✓ IFFT 를 하기 전에 S/P 과정에서 data 를 주어진 subcarrier 에 매칭되도록 맵핑
- ✓ FFT 이후에는 P/S 과정에서 data 에 해당하는 심볼만 찾아서 BPSK demodulation 함
- ✓ CP 추가는 IFFT 이후의 신호의 뒷부분(전체 길이의 ¼)을 신호에 앞에 추가

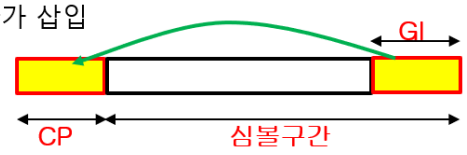
802.11a OFDM PHY Parameters	
BW	20 MHZ
OBW	16.6 MHZ
Subcarrer Spacing	312.5 Khz (20MHz/64 Pt FFT)
Information Rate	**6**/9/**12**/18/**24**/36/48/54 Mbits/s
Modulation	BPSK, QPSK, 16QAM, 64QAM
Coding Rate	1/2, 2/3, 3/4
Total Subcarriers	52 (Freq Index -26 to +26)
Data Subcarriers	48
Pilot Subcarriers*	4 (-21, -7, +7, +21) *Always BPSK
DC Subcarrier	Null (0 subcarrier)

Total subcarrier 가 52 이므로 `sample = 52;`
BPSK 를 사용하므로 `mod_order = 2;`

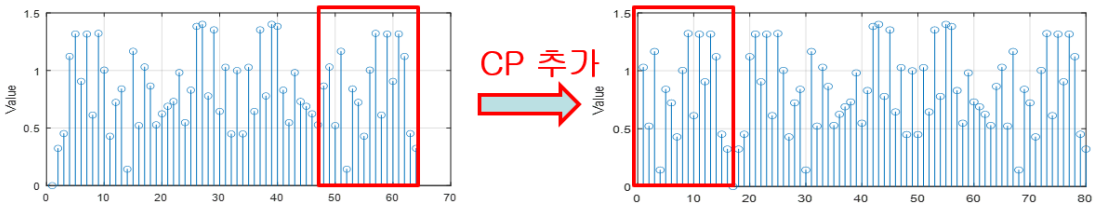


64 개의 주파수를 사용하므로 `n_fft = 64;`

- Guard Interval(GI)의 경우, CP(Cyclic Prefix)를 이용
  - 부반송파 간의 직교성 유지를 위해 OFDM 신호의 마지막 구간의 신호를 복사해서 신호의 앞에다가 삽입



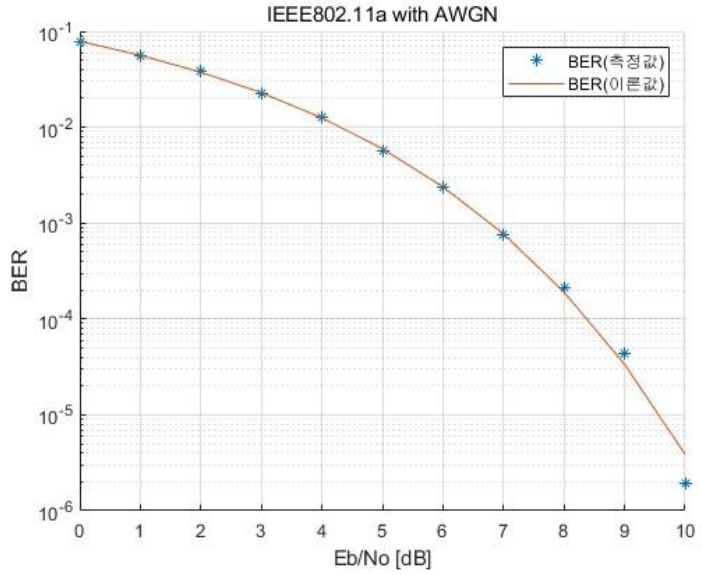

```
%% Guard Interval(CP) Insert
nSampGI = nFFTSize/4;
tx_ofdm = [outputiFFT(nFFTSize-nSampGI+1:end); outputiFFT];
```



송신에서 CP 를 넣어서 보냈으면, 수신할 때는 CP 부터 제거하고 FFT 를 해야 한다.

```
sample = 52;
mod_order = 2;

tx = randi([0 (mod_order - 1)], 1, sample);

% BPSK modulation
tx_mod = pskmod(tx, mod_order, 0);

% Data mapping
n_fft = 64;
tx_map = zeros([1 n_fft]);
subcarrier_idx = [-26:-1 1:26];
tx_map(subcarrier_idx + n_fft / 2 + 1) = tx_mod;

% S/P
tx_map = tx_map';

% IFFT
tx_ifft = ifft(tx_map, n_fft) * sqrt(n_fft);

% CP(Cyclic Prefix) insertion
n_cp = n_fft / 4;
tx_ofdm = [tx_ifft((n_fft - n_cp + 1):end); tx_ifft];

% P/S
tx_ofdm = tx_ofdm';

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

```
k = 1;

for Eb_No_dB = Eb_No_dBs
    N = 10000;

    for i = 1:N
        rx_ofdm = awgn(tx_ofdm, Eb_No_dB);

        % S/P
        rx_ofdm = rx_ofdm';

        % CP removing
        rx_signal = rx_ofdm((n_cp + 1):end);

        % FFT
        rx_fft = fft(rx_signal, n_fft) / sqrt(n_fft);

        % Data de-mapping
        rx_demap = rx_fft(subcarrier_idx + n_fft / 2 + 1);

        % P/S
        rx_demap = rx_demap';

        % BPSK demodulation
        rx = pskdemod(rx_demap, mod_order, 0);

        bers(1, k) = bers(1, k) + sum(tx ~= rx);
    end

    bers(1, k) = bers(1, k) / (N * sample);
    k = k + 1;
end

bers_ = 1/2 * erfc(sqrt(db2pow(Eb_No_dBs)));

figure()
grid on
hold on
plot(Eb_No_dBs, bers, '*')
plot(Eb_No_dBs, bers_)
set(gca, 'yscale', 'log')
axis([0 10 1e-6 1e-1])
title('IEEE802.11a with AWGN')
xlabel('Eb/No [dB]')
ylabel('BER')
legend('BER(측정값)', 'BER(이론값)')
```



측정값과 이론값은 거의 일치한다. BER 이론값은 아래와 같이 구할 수 있다. BPSK 의 BER 공식과 동일한 것을 확인할 수 있다.

```
bers_ = 1/2 * erfc(sqrt(db2pow(Eb_No_dBs)));
```

아래는 송수신 과정을 그래프로 출력하는 코드이다.

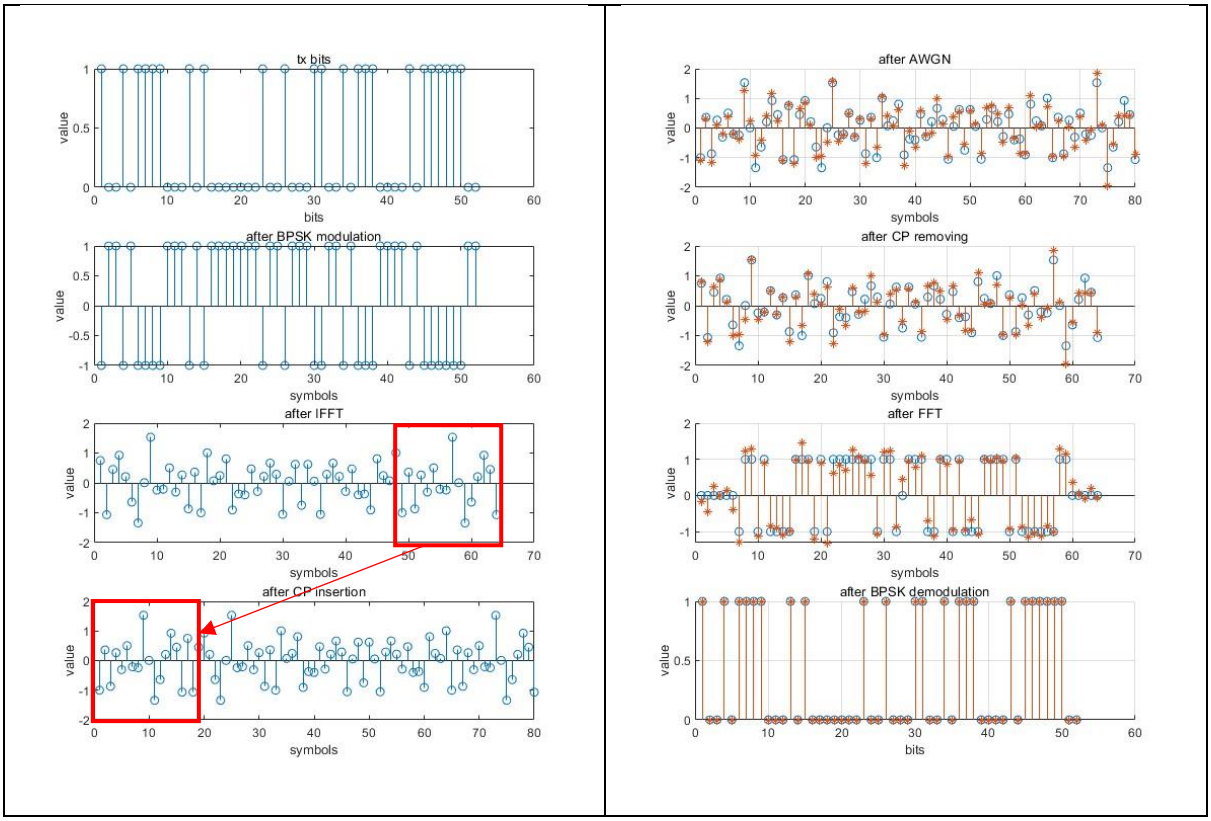
```
f = figure();
f.Position = [680 0 560 800];
subplot(4, 1, 1)
grid on
stem(tx)
title('tx bits')
xlabel('bits')
ylabel('value')

subplot(4, 1, 2)
grid on
stem(tx_mod)
title('after BPSK modulation')
xlabel('symbols')
ylabel('value')

subplot(4, 1, 3)
grid on
stem(tx_ifft)
title('after IFFT')
xlabel('symbols')
ylabel('value')
```

```
subplot(4, 1, 4)
grid on
stem(tx_ofdm)
title('after CP insertion')
xlabel('symbols')
ylabel('value')
```

tx 를 송신하는 과정을 그래프로 출력한다.

```
f = figure();
f.Position = [680 0 560 800];
subplot(4, 1, 1)
grid on
hold on
stem(tx_ofdm)
stem(rx_ofdm, '*')
title('after AWGN')
xlabel('symbols')
ylabel('value')

subplot(4, 1, 2)
grid on
hold on
stem(tx_ifft)
stem(rx_signal, '*')
title('after CP removing')
xlabel('symbols')
ylabel('value')

subplot(4, 1, 3)
grid on
hold on
stem(tx_map)
stem(rx_fft, '*')
title('after FFT')
xlabel('symbols')
ylabel('value')

subplot(4, 1, 4)
grid on
hold on
stem(tx)
stem(rx, '*')
title('after BPSK demodulation')
xlabel('bits')
ylabel('value')
```

tx 에 AWGN 이 추가된 rx 를 수신하는 과정을 그래프로 출력한다.



파란색은 송신부, 주황색은 수신부를 의미한다.

좌측 하단을 보면 Cyclic prefix 가 생기는 것을 관찰할 수 있다. 우측 최하단의 stem 그래프가 tx bit 와 rx bit 를 출력한 것으로 동일하게 나오는 것을 확인할 수 있다.

Problem

1. 실습시간에 활용한 WiFi 규격 파라미터를 이용하여, QPSK 로 변조한 경우의 이론값, 시뮬레이션 BER 그래프를 그리시오 (AWGN 만 고려).

   QPSK 는 gray coding 되어있다고 고려

2. 실습시간에 진행한 BPSK 로 변조한 경우와 BER 결과를 비교하고, 결과차이에 대한 분석을 적으시오.

   Eb/N0 = [0:1:10] dB
   Modulation : BPSK, QPSK

Result

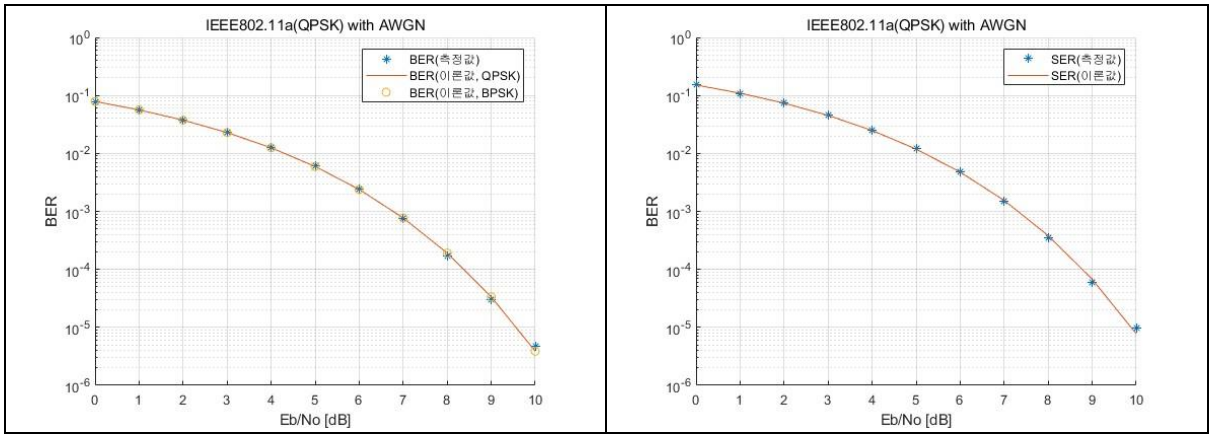
```
sample = 52;
```

```
mod_order = 4;

tx = randi([0 (mod_order - 1)], 1, sample);

% QPSK modulation
bits = log2(mod_order);
tx_mod = sqrt(bits) * pskmod(tx, mod_order);

% Data mapping
n_fft = 64;
tx_map = zeros([1 n_fft]);
subcarrier_idx = [-26:-1 1:26];
tx_map(subcarrier_idx + n_fft / 2 + 1) = tx_mod;

% S/P
tx_map = tx_map';

% IFFT
tx_ifft = ifft(tx_map, n_fft) * sqrt(n_fft);

% CP(Cyclic Prefix) insertion
n_cp = n_fft / 4;
tx_ofdm = [tx_ifft((n_fft - n_cp + 1):end); tx_ifft];

% P/S
tx_ofdm = tx_ofdm';

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

k = 1;

for Eb_No_dB = Eb_No_dBs
    No_mW = db2pow(-Eb_No_dB);
    N = 10000;

    for i = 1:N
        rx_ofdm = awgn(tx_ofdm, Eb_No_dB);

        % S/P
        rx_ofdm = rx_ofdm';

        % CP removing
        rx_signal = rx_ofdm((n_cp + 1):end);

        % FFT
```

```
        rx_fft = fft(rx_signal, n_fft) / sqrt(n_fft);

        % Data de-mapping
        rx_demap = rx_fft(subcarrier_idx + n_fft / 2 + 1);

        % P/S
        rx_demap = rx_demap';

        % QPSK demodulation
        rx = pskdemod(rx_demap, mod_order);

        bers(1, k) = bers(1, k) + sum(de2bi(tx) ~= de2bi(rx), 'all');
        sers(1, k) = sers(1, k) + sum(tx ~= rx);
    end

    bers(1, k) = bers(1, k) / (N * sample * bits);
    sers(1, k) = sers(1, k) / (N * sample);
    k = k + 1;
end

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

bers_bspk_ = berawgn(Eb_No_dBs, 'psk', 2, 'nodiff');

figure()
grid on
hold on
plot(Eb_No_dBs, bers, '*')
plot(Eb_No_dBs, bers_)
plot(Eb_No_dBs, bers_bspk_, 'o')
set(gca, 'yscale', 'log')
axis([0 10 1e-6 1])
title('IEEE802.11a(QPSK) with AWGN')
xlabel('Eb/No [dB]')
ylabel('BER')
legend('BER(측정값)', 'BER(이론값, QPSK)', 'BER(이론값, BPSK)')

figure()
grid on
hold on
plot(Eb_No_dBs, sers, '*')
plot(Eb_No_dBs, sers_)
set(gca, 'yscale', 'log')
axis([0 10 1e-6 1])
title('IEEE802.11a(QPSK) with AWGN')
xlabel('Eb/No [dB]')
```

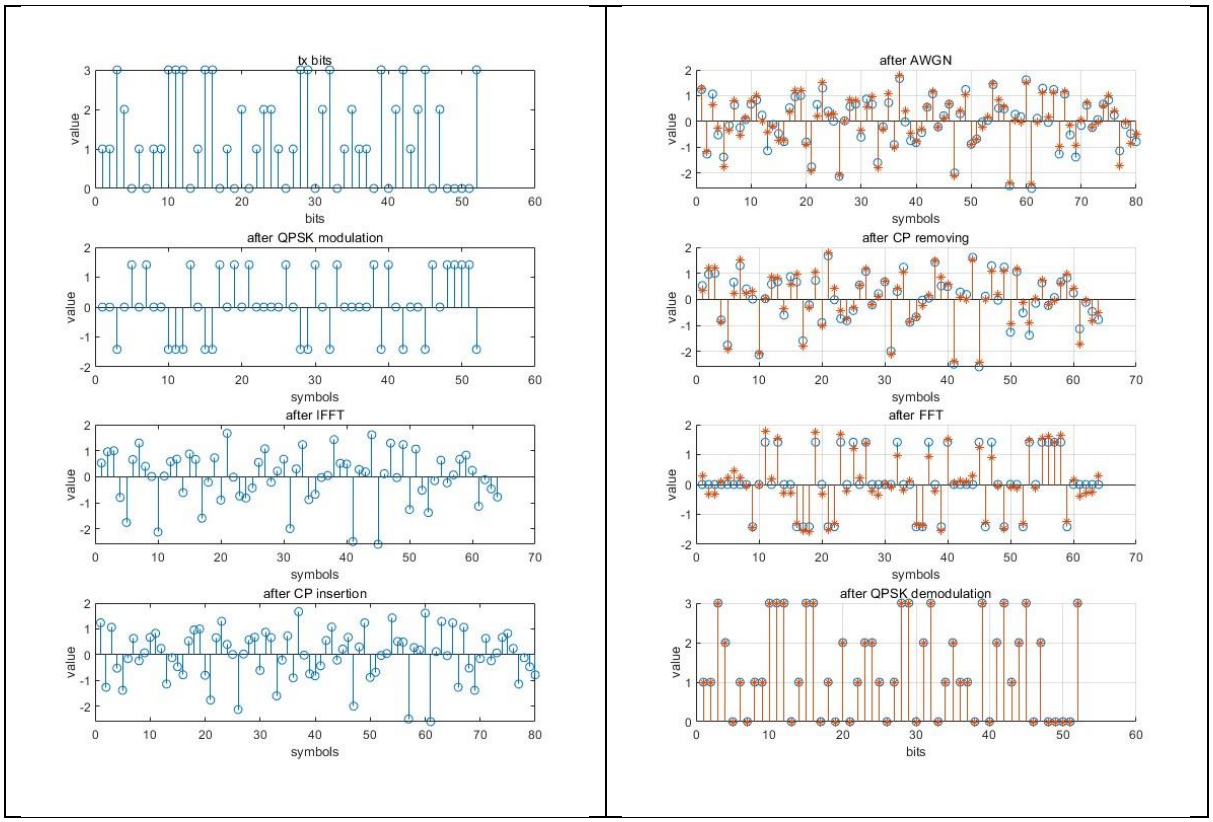
```
ylabel('BER')
legend('SER(측정값)', 'SER(이론값)')
```

BER, SER 그래프를 그려본다.



당연하게도 BER 보다 SER 이 크다. 그 이유는 QPSK symbol 에 한 비트만 오류가 발생해도 심볼은 오류가 발생하기 때문이다.

BPSK 나 QPSK 모두 BER 이론값은 동일하다. 측정값도 이론값과 거의 유사하게 나왔다.

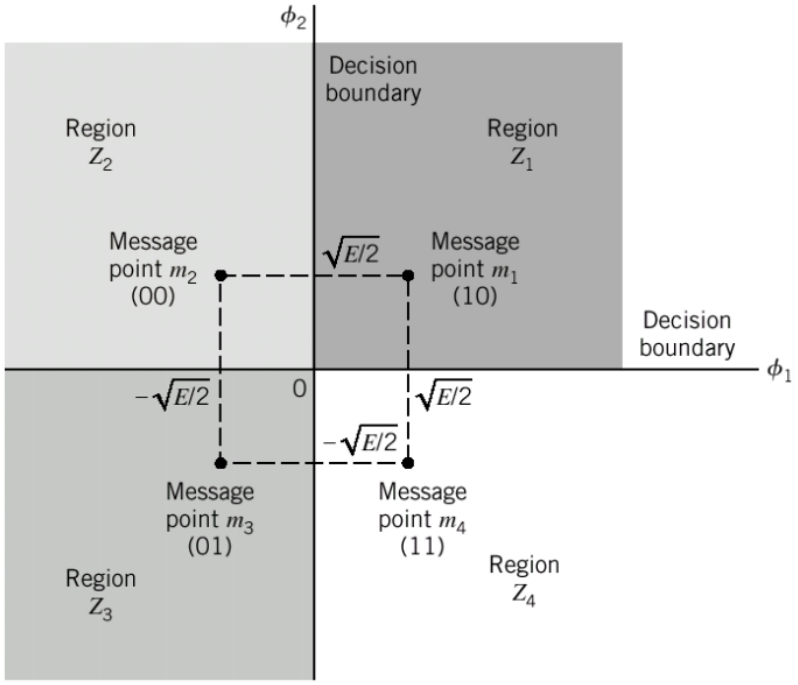


실습코드를 이용해 tx 와 rx 를 처리하는 과정도 출력해보았다. CP 가 정상적으로 생기는 것과, rx 와 tx bit 가 일치하게 나오는 것을 확인할 수 있었다.

Conclusion

실습 내용과 거의 동일한데 일단 QPSK 이므로 `mod_order = 4;`로 고친다.

주요 수정부분 위주로 설명을 하면,



QPSK 는 각각의 점이 $\left(\pm\sqrt{\frac{E}{2}}, \pm\sqrt{\frac{E}{2}}\right)$로 구성되어 있다. 여기서 $E = 2E_b$고 $E_b = 1$으로 가정했기에 실제로는 $(\pm\sqrt{2}, \pm\sqrt{2})$ 형태가 되어야 한다. 하지만, `pskmod` 는 $(\pm 1, \pm 1)$을 출력하기 때문에 $\sqrt{2}$를 곱해줘야 한다.

```
bits = log2(mod_order);
tx_mod = sqrt(bits) * pskmod(tx, mod_order);
```

BER 을 구하기 위해 오류 비트 개수를 구할 때도 `de2bi` 를 이용해 비트로 쪼갠 후 `sum` 으로 합하여 계산한다. 단순히 abs(tx – rx)를 해버리면 3(=11)이 1(=01)이 되었을 때 오류가 난 비트는 1 비트지만, abs(3 – 1) = 2 비트가 오류났다고 계산하게 된다.

```
bers(1, k) = bers(1, k) + sum(de2bi(tx) ~= de2bi(rx), 'all')
```

최종적으로 BER 을 구할 때도 `bits` 로 나눠줘야 한다. 왜냐하면 QPSK symbol 하나당 2 비트가 전송되기 때문이다.

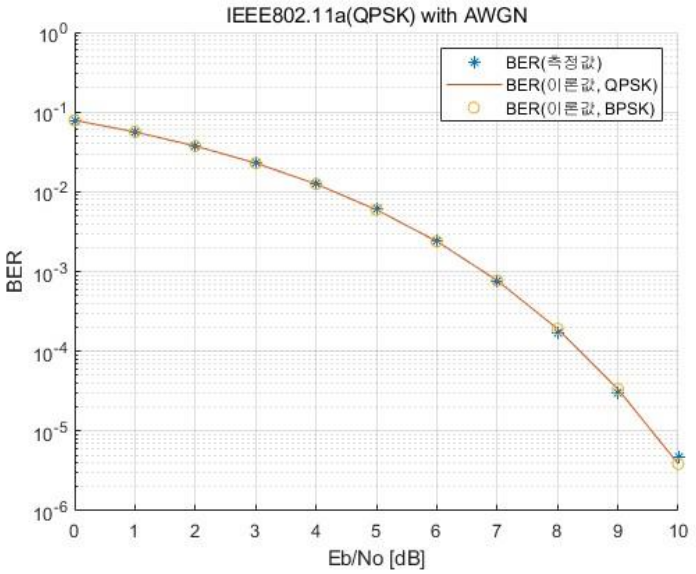
```
bers(1, k) = bers(1, k) / (N * sample * bits);
```

그렇게 해선 구한 BER 은 아래 공식과 동일하다.

$$P_e = \frac{1}{2} erfc\left(\frac{2\sqrt{\frac{E}{2}}}{2\sqrt{N_0}}\right) = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right)$$

SER 은 아래와 같이 계산한다.

$$P_e' = 1 - \left(\textit{비트 오류가 전혀 안 났을 때}\right) = 1 - (1 - P_e)^2 = 2P_e - P_e^2 \approx erfc\left(\sqrt{\frac{E_b}{N_0}}\right)$$



BPSK 를 썼을 때와 QPSK 를 썼을 때 모두 BER 이 동일한데, 그 이유는 각 비트에 오류가 발생할 확률은 독립적이기 때문이다.

비트 오류가 발생할 확률은 Log-likelihood 함수를 이용해 구한다.

$$l(m_i) = \log L(m_i) = -\frac{1}{N_0}\sum_{j=1}^{N}\left(x_j - s_{ij}\right)^2$$

오류 확률은 Constellation 에서 점과 symbol 사이 거리에 영향을 받는데, 공식은 아래와 같다.

$$P_e = \frac{1}{2}erfc\left(\frac{d_{ik}}{2\sqrt{N_0}}\right)$$

BPSK 의 경우 $d_{ik} = 2\sqrt{E_b}$고, QPSK 의 경우 $d_{ik} = 2\sqrt{\frac{E}{2}} = 2\sqrt{E_b}$이다. $(E = 2E_b)$

따라서 BPSK 와 QPSK 의 P_e는 동일하게 나온다.