

디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001



7 주차 실습과제

정보통신공학과

12191765

박승재

Introduction

BSPK에서는 아래와 같은 방식으로 신호를 변환한다.

$$\begin{cases} s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f t) = \sqrt{E_b} \phi(t), & \text{if } b = 1 \\ s_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f t + \pi) = -\sqrt{\frac{2E_b}{T_b}} \cos(2\pi f t) = -\sqrt{E_b} \phi(t), & \text{if } b = 0 \end{cases}$$

아래 코드는 Symbolic을 이용해서 $\phi(t)$ 를 구현 것이다.

```
syms Eb t Tb

f = 1 / Tb;
phi_t = sqrt(2 / Tb) * cos(2 * pi * f * t); % basis function

Eb_N0_dB = 3;
```

E_b/N_0 [dB]는 3으로 잡았다.

```
tic

N = 1000; % # of symbols
symbols = zeros(2, N);
symbol_errors = zeros(1, N);

for i = 1:N
    bit = rand() >= 0.5; % random binary bit

    if bit == 1
        s_t = sqrt(Eb) * phi_t;
    else
        s_t = -sqrt(Eb) * phi_t;
    end

    s_t_ = subs(s_t, Eb, 1); % set Eb = 1mW
    noise = sqrt(db2pow(-Eb_N0_dB) / 2) * randn() * phi_t; % noise
    x_t = s_t_ + noise;

    s = int(x_t * phi_t, t, [0 Tb]);

    % decision device
    if s > 0
        b_est = 1;
    else
        b_est = 0;
    end
end
```

```

symbols(1, i) = s;
symbol_errors(i) = (b_est ~= bit);
end

ber = sum(symbol_errors) / N
ber_awgn = 1/2 * erfc(sqrt(db2pow(Eb_N0_dB)))

toc

```

점 1000 개를 찍어 그래프를 그릴 것이다. N 은 찍을 점의 개수를 의미한다. `symbols` 와 `symbol_errors` 는 각각 점의 위치와 점의 오류 여부를 의미한다. `symbols` 는 복소수 형태로 표현되지만, 코드에서는 실수부만 계산한다. Eb 는 1mW 로 잡았다. BER 는 오류난 점의 개수를 통해 계산하고 `ber_awgn` 는 BER 의 이론값이다.

```

s1_t = sqrt(Eb) * phi_t;
s0_t = -sqrt(Eb) * phi_t;

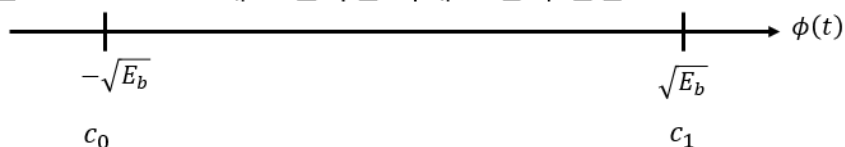
s1_t_ = subs(s1_t, Eb, 1);
c1_ = int(s1_t_ * phi_t, t, [0 Tb]);
s0_t_ = subs(s0_t, Eb, 1);
c0_ = int(s0_t_ * phi_t, t, [0 Tb]);

```

그래프의 파란 점 위치를 계산하는 코드이다. 0 과 1 일 때를 나누어 계산해 각각 -1 과 1 위치에 점을 찍는다.

$$c_n = \int_0^{T_b} s_n(t) \phi(t) dt = \begin{cases} \sqrt{E_b}, & \text{if } n = 1 \\ -\sqrt{E_b}, & \text{otherwise} \end{cases}$$

c_n 을 constellation 내 표현하면 아래 그림과 같음



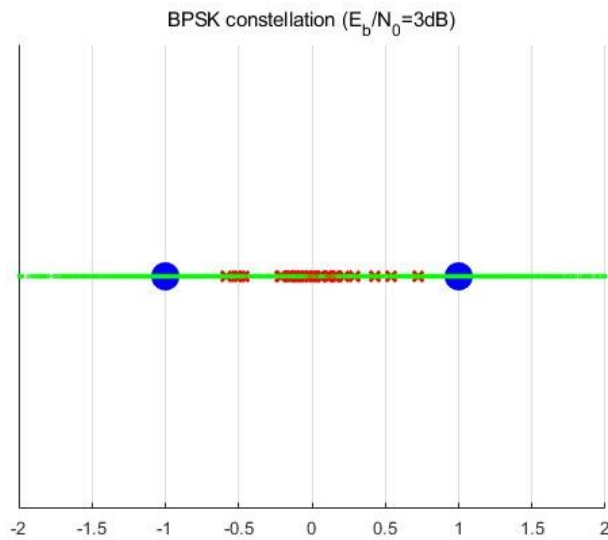
```

grid on
hold on
plot(c1_, 0, 'o', 'markersize', 15, 'markerEdgeColor', 'b',
'MarkerFaceColor', 'b')
plot(c0_, 0, 'o', 'markersize', 15, 'markerEdgeColor', 'b',
'MarkerFaceColor', 'b')
plot(symbols(1, find(symbol_errors)), symbols(2, find(symbol_errors)), 'x',
'markersize', 8, 'color', 'r', 'LineWidth', 2)
plot(symbols(1, setdiff(1:N, find(symbol_errors))), symbols(2, setdiff(1:N,
find(symbol_errors))), 'o', 'markersize', 2, 'color', 'g')
set(gca, 'ytick', -1:2:1)
axis([-2 2 -0.1 0.1])

```

```
title(['BPSK constellation (Eb/N0=', int2str(Eb_N0_dB), 'dB')])
```

계산한 값을 그래프에 그리는 코드이다.



```
>> bpsk
```

```
ber =
```

```
0.0310
```

```
ber_awgn =
```

```
0.0229
```

경과 시간은 9.432602초입니다.

빨간색은 오류 난 점이고, 초록색은 정상적으로 인식된 점이다. 파란색은 잡음이 섞이지 않은 신호를 의미한다. ber 과 ber_awgn 역시 유사하게 나오는 것을 확인할 수 있었다.

Problem

E_b/N_0 를 -2dB 부터 10dB 까지 2dB 간격으로 증가시키면서 BER 측정

시뮬레이션으로 얻은 BER 값과 이론 값을 비교

AWGN 채널에서 BER 이론 값: $\frac{1}{2} \text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$

X 축은 E_b/N_0 [dB]

Y 축은 BER 값을 log-scale 로 그리기

Result

```
syms Eb t Tb
```

```
f = 1 / Tb;
```

```
phi_t = sqrt(2 / Tb) * cos(2 * pi * f * t); % basis function
```

```
Eb_N0_dBs = -2:2:10;
```

```
bers = zeros(1, length(Eb_N0_dBs));
```

```

ber_awgns = zeros(1, length(Eb_N0_dBs));

tic

j = 1;
for Eb_N0_dB = -2:2:10
    N = 1000; % # of symbols
    symbols = zeros(2, N);
    symbol_errors = zeros(1, N);

    for i = 1:N
        bit = rand() >= 0.5; % random binary bit
        if bit == 1
            s_t = sqrt(Eb) * phi_t;
        else
            s_t = -sqrt(Eb) * phi_t;
        end

        s_t_ = subs(s_t, Eb, 1); % set Eb = 1mW
        noise = sqrt(db2pow(-Eb_N0_dB) / 2) * randn() * phi_t; % noise
        x_t = s_t_ + noise;

        s = int(x_t * phi_t, t, [0 Tb]);

        % decision device
        if s > 0
            b_est = 1;
        else
            b_est = 0;
        end

        symbols(1, i) = s;
        symbol_errors(i) = (b_est ~= bit);
    end

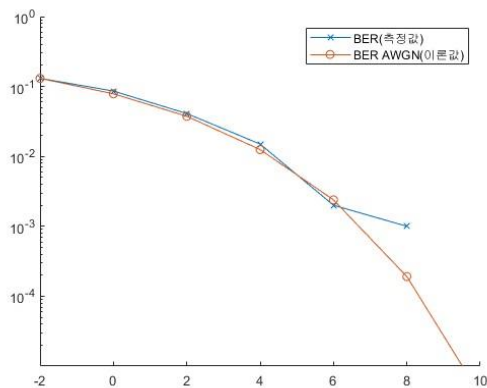
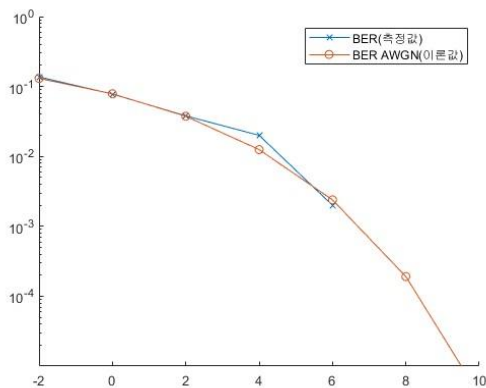
    bers(1, j) = sum(symbol_errors) / N
    ber_awgns(1, j) = 1/2 * erfc(sqrt(db2pow(Eb_N0_dB)))
    j = j + 1;
end

toc

hold on;
plot(Eb_N0_dBs, bers, '-x')
plot(Eb_N0_dBs, ber_awgns, '-o')
set(gca, 'yscale', 'log')
axis([-2 10 1e-5 1])

```

```
legend('BER(측정값)', 'BER AWGN(이론값)')
```



측정한 BER 이 중간에 끊기는 것을 확인할 수 있는데, E_b/N_0_{dB} 가 커지면서 ber 이 0 이 되어 그래프 범위에 나오지 않기 때문이다.

```
bers =
    0.1370    0.0780    0.0380    0.0200    0.0020         0         0

ber_awgns =
    0.1306    0.0786    0.0375    0.0125    0.0024    0.0002    0.0000
```

경과 시간은 105.680890초입니다.

Conclusion

E_b/N_0_{dB} 를 반복문을 이용해 -2 부터 증가시키며 BER 이 감소하는 것을 확인하였다. E_b/N_0_{dB} 가 증가하는 것은 잡음이 감소한다는 것을 의미한다. BER 은 Bit Error Rate 라는 뜻으로, 오류의 확률을 의미한다. 그러면 당연히잡음도 감소하면 오류도 감소해야 한다. 따라서 E_b/N_0_{dB} 가 증가할수록 BER 은 감소하는 그래프가 그려지게 된다.

오류가 발생할 확률은 정규분포 그래프에서 0 을 넘어간 부분의 넓이로 표현된다.

$$p_{10} = \int_0^{\infty} f_{x_1}(x_1 | 0) dx_1$$

$$= \frac{1}{\sqrt{\pi N_0}} \int_0^{\infty} \exp\left[-\frac{1}{N_0}(x_1 + \sqrt{E_b})^2\right] dx_1 \quad P_e = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

N_0 가 작아지면 0 에서부터 무한대까지의 exp 의 넓이는 감소한다. (곡선이 x 축으로 더 붙으므로) 따라서 식을 이용해 계산하더라도, E_b/N_0_{dB} 가 증가할수록 BER 은 감소하는 게 옳다.