

1. MATLAB 소스 코드 설명

1.1 랜덤 바이너리 송신 신호 생성

랜덤 바이너리 송신 신호 $a\{n\}$, $n = 1, 2, \dots, 10^7$, $a(n) \in \{-1, 1\}$ 를 생성한다.
단, $P(1) = P(-1) = 0.5$

```
5 - close all;clear;clc;
6
7 - %% 1) Random binary 송신 신호 생성
8 - n = 10^7;
9 - a_n = 2*floor(2*rand(1,n)) - 1; % P(1) = P(-1) = 0.5
10
```

Figure 1. random binary 송신 신호 생성

① line 9

$a(n)$ 을 생성하기 위해, 다음 코드를 작성하였다.

```
a_n = 2*floor(2*rand(1,n)) - 1;
```

rand 함수는 구간(0,1)에 균일하게 분포된 난수를 반환한다. floor 함수는 해당 요소보다 작거나 같은 가장 가까운 정수로 내린다.

floor 와 rand 함수를 $n=10$ 의 간단한 예시로 살펴보면 다음과 같다.

```
명령 창
>> floor(2*rand(1,10))

ans =

    0    0    0    1    0    0    1    1    0    0
```

위의 값에 2를 곱한 뒤 1을 빼면, $P(1) = P(-1) = 0.5$ 로 생성되었음을 확인할 수 있다.

```
명령 창
>> 2*floor(2*rand(1,10)) - 1

ans =

    1   -1    1   -1    1    1   -1   -1   -1    1
```

1.2 AWGN 채널

$\{a(n)\}$ 이 BPSK 로 변조되어 AWGN 채널을 통과하여 수신기에 도착한다고 가정한다.

즉, $y = x + n$, $n \sim \mathcal{N}(0, \frac{N_0}{2})$ 수신기에서 ML 방법으로 수신 신호를 디코딩 한다고 했을 때, SNR 값에 대한 BER 을 구한다.

```
11 %% 2) BPSK 변조 후 AWGN 채널 통과할 때 BER
12 % y = x + n
13 % n ~ N(0, N0/2)
14
15 - SNR_dB = 0:2:20;
16 - SNR = 10.^(SNR_dB/10);
17
18 - x = a_n;
19 - BER = zeros(1, length(SNR_dB));
20 - noise_AWGN = sqrt(0.5).*randn(1,n);
21
22 - for k=1:length(SNR)
23 -     y = (sqrt(SNR(k))*x) + noise_AWGN;
24 -     BER(k) = length(find((y.*x)<0));
25 - end
26 - BER=BER/n;
27
28 % AWGN 채널 시뮬레이션
29 - semilogy(SNR_dB,BER,'co','linewidth',2.0);
30 - hold on
31
32 % AWGN 채널 이론값
33 - AWGN_theory_ber = 0.5.*erfc(sqrt(SNR)); % Pb = Q(sqrt(2SNR))
34 - semilogy(SNR_dB,AWGN_theory_ber,'b--','linewidth',2.0);
35 - hold on
```

Figure 2. AWGN 채널 통과할 때 BER vs SNR(dB)

① line 15 - 16

$SNR(dB) = 10 \log_{10} SNR$ 이고, SNR = 0, 2, 4, ..., 20 (dB) 값에 대한 BER 을 구해야 한다.

```
SNR_dB = 0:2:20;
SNR = 10.^(SNR_dB/10);
```

② line 20

AWGN 채널의 noise 를 계산한다. $n \sim \mathcal{N}(0, \frac{N_0}{2})$ 이므로,

```
noise_AWGN = sqrt(0.5).*randn(1,n);
```

③ line 22 - 26

find 함수는 () 안에 있는 조건을 만족하는 요소들을 찾는다. 랜덤 바이너리 송신 신호는 -1 과 1 로 이루어져 있기 때문에, $y.*x < 0$ 이면 에러가 발생한 것이다. 다음 코드와 같이 for 문 내부에 find 함수와 length 함수를 이용해 $y.*x < 0$ 을 만족시키는 요소들이 몇 개 있는지 찾고, for 문이 종료된 후 n 으로 나누어 최종 BER 을 구한다.

```
for k=1:length(SNR)
    y = (sqrt(SNR(k))*x) + noise_AWGN;
    BER(k) = length(find((y.*x)<0));
end
BER=BER/n;
```

④ line 28 - 30

AWGN 채널의 BER vs SNR 시뮬레이션 결과값을 그래프에 그리는 코드이다. (청록색 원)

```
semilogy(SNR_dB,BER,'co','linewidth',2.0);
```

⑤ line 32 - 35

Q function 은 Matlab 의 erfc()를 이용한다. $Q(x) = \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2}}\right)$ 이다.

AWGN 채널의 이론값 $P_b = Q(\sqrt{2SNR})$ 을 그래프에 출력한다. (파란색 점선)

```
AWGN_theory_ber = 0.5.*erfc(sqrt(SNR)); % Pb = Q(sqrt(2SNR))
semilogy(SNR_dB,AWGN_theory_ber,'b--','linewidth',2.0);
```

1.3 Rayleigh 채널

$$y = hx + n, \quad h \sim \mathcal{CN}(0,1), \quad n \sim \mathcal{CN}(0, N_0)$$

위의 Rayleigh 페이딩 채널에 대해서 문제 2를 반복해서 수행한다.

```

37 % 3) Rayleigh 페이딩 채널
38 % y = hx + n
39 % h ~ CN(0,1)
40 % n ~ CN(0,N0)
41
42 - h = sqrt(0.5).*(randn(1,n) + 1i*randn(1,n));
43 - noise_Rayleigh = sqrt(0.5).*(randn(1,n) + 1i*randn(1,n));
44
45 - for k=1:length(SNR)
46 -     y = h.*(sqrt(SNR(k))*x) + noise_Rayleigh;
47 -     r = conj(h)./abs(h).*y; % coherent detection
48 -     BER(k)=length(find((r.*x)<0));
49 - end
50 - BER=BER/n;
51
52 % Rayleigh 채널 시뮬레이션
53 - semilogy(SNR_dB, BER, 'ms', 'linewidth', 2.0);
54 - hold on
55
56 % Rayleigh 채널 이론값
57 - Rayleigh_theory_ber = 0.5*(1-(sqrt(SNR./(1+SNR))));
58 - semilogy(SNR_dB, Rayleigh_theory_ber, 'r-', 'linewidth', 2.0);
59
60 % 최종 결과 그래프 설정
61 - title('Bit Error Rate, AWGN vs. Rayleigh')
62 - xlabel('SNR (dB)')
63 - ylabel('P_{b}')
64 - legend('AWGN (sim)', 'AWGN (theory)', 'Rayleigh (sim)', 'Rayleigh (theory)')
65 - axis ([0 20 1e-6 10^0])
66 - grid

```

Figure 3. Rayleigh 페이딩 채널 통과할 때 BER vs SNR(dB)

① line 42

Rayleigh 페이딩 채널 h 이다.

$$h = h_I + jh_Q, \quad h_I \sim \mathcal{CN}\left(0, \frac{1}{2}\right), \quad h_Q \sim \mathcal{CN}\left(0, \frac{1}{2}\right)$$

```
h = sqrt(0.5).*(randn(1,n) + 1i*randn(1,n));
```

② line 43

Rayleigh 페이딩 채널의 noise 값이다. ($N_0 = 1$)

$$n = n_I + jn_Q, \quad n_I \sim \mathcal{CN}\left(0, \frac{N_0}{2}\right), \quad n_Q \sim \mathcal{CN}\left(0, \frac{N_0}{2}\right)$$

```
noise_Rayleigh = sqrt(0.5).*(randn(1,n) + 1i*randn(1,n));
```

③ line 45 – 50

Rayleigh 페이딩 채널의 경우, 수신기에서는 채널 h 를 추정하고, 그 값으로 $r = \frac{h^*}{|h|}y$ 의 수신 신호처리(coherent detection)를 하고 난 이후 ML 검출 방법을 수행한다.

conj 함수는 켤레복소수를 구하고, abs 함수는 절댓값을 구한다.

```
for k=1:length(SNR)
    y = h.*((sqrt(SNR(k))*x)) + noise_Rayleigh;
    r = conj(h)./abs(h).*y;           % coherent detection
    BER(k)=length(find((r.*x)<0));
end
BER=BER/n;
```

④ line 52 – 54

Rayleigh 페이딩 채널의 BER vs SNR 시뮬레이션 결과값을 그래프에 그리는 코드이다.
(자홍색 사각형)

```
semilogy(SNR_dB,BER,'ms','linewidth',2.0);
```

⑤ line 56 – 58

Rayleigh 페이딩 채널의 이론값 $P_b = \frac{1}{2} \left(1 - \sqrt{\frac{SNR}{1+SNR}} \right)$ 을 그래프에 출력한다. (빨간색 실선)

```
Rayleigh_theory_ber = 0.5*(1-(sqrt(SNR./(1+SNR))));
semilogy(SNR_dB, Rayleigh_theory_ber,'r-','linewidth',2.0);
```

⑥ line 60 – 66

최종 결과 그래프의 title, label, legend, axis 등을 설정한다.

```
title('Bit Error Rate, AWGN vs. Rayleigh')
xlabel('SNR (dB)')
ylabel('P_{b}')
legend('AWGN (sim)','AWGN (theory)','Rayleigh (sim)','Rayleigh (theory)')
axis([0 20 1e-6 10^0])
grid
```

2. 시뮬레이션 결과

