Simulating Different Receivers in a Rayleigh Fading, SISO Environment

Project #1

Intelligent Communication Systems (ICS) Lab.
노용재

Winter Intern Seminar (2023-1)

모든 실험은 다음의 조건하에 진행되었다.

- Es/N0는 -2dB 20dB(2dB 간격)
- SISO; Single Input, Single Output

1 Binary Moduation (M=2)

```
% Simulation
NumberOfSignals = 10^2;
LengthBitSequence = NumberOfSignals*log2(M); % log2(M) bits per signal

NumberIteration = 10^3;

Es = 1;

EsN0_dB = -2:2:20;
EsN0 = db2pow(EsN0_dB);

EbN0 = EsN0 / log2(M);
EbN0_dB = pow2db(EbN0);

ErrorCount_ZF = zeros(1, length(EbN0_dB));
ErrorCount_MMSE = zeros(1, length(EbN0_dB));
ErrorCount_MLD = zeros(1, length(EbN0_dB));

alphabet = qammod([0:M-1], M, 'UnitAveragePower', true);

for iTotal = 1 : NumberIteration

    BitSequence = randi([0 1], 1, LengthBitSequence); % Bit Generation
    SymbolSequence = qammod(BitSequence.', M, 'InputType', 'bit', '
        UnitAveragePower', 1).';
    NoiseSequence = (randn(1, length(SymbolSequence)) + 1j * randn(1,
        length(SymbolSequence))) / sqrt(2); % Noise (n) Generation
    H = (randn(1, length(SymbolSequence)) + 1j * randn(1, length(
        SymbolSequence))) ./ sqrt(2); % Channel (h) Generation
    for indx_EbN0 = 1 : length(EbN0)
```

```
        ReceivedSymbolSequence = H .* SymbolSequence + NoiseSequence *
            sqrt(1 / EsN0(indx_EbN0)); % Received Signal (y = s + n)
            Generation

        % ZF Receiver
        w_zf = H.^(-1);
        DetectionSymbolSequence_ZF = ReceivedSymbolSequence .* w_zf; %
            Detection (Zero-Forcing: y / h)

        % MMSE Receiver
        w_mmse = (H.*conj(H)+1/EsN0(indx_EbN0)).^(-1) .* conj(H);
        z = ReceivedSymbolSequence .* w_mmse;
        arg = (ones(length(alphabet),1) * z) - (alphabet.' * H .* w_mmse);
        arg = arg .* conj(arg);
        [val,idx] = min(arg);
        DetectionSymbolSequence_MMSE = alphabet(idx);
        DetectionSymbolSequence_MMSE = z;

        % MLD Receiver
        arg = (ones(length(alphabet),1) * ReceivedSymbolSequence) - (
            alphabet.' * H);
        arg = abs(arg).^2;
        [val,idx] = min(arg);
        DetectionSymbolSequence_MLD = alphabet(idx);

        % Symbol Sequence -> Bit Sequence
        DetectionBitSequence_ZF = qamdemod(DetectionSymbolSequence_ZF.', M
            , 'OutputType', 'bit', 'UnitAveragePower', 1)';
        DetectionBitSequence_MMSE = qamdemod(DetectionSymbolSequence_MMSE
            .', M, 'OutputType', 'bit', 'UnitAveragePower', 1)';
        DetectionBitSequence_MLD = qamdemod(DetectionSymbolSequence_MLD.',
             M, 'OutputType', 'bit', 'UnitAveragePower', 1)';

        ErrorCount_ZF(1, indx_EbN0) = ErrorCount_ZF(1, indx_EbN0) + sum(
            DetectionBitSequence_ZF~=BitSequence);
        ErrorCount_MMSE(1, indx_EbN0) = ErrorCount_MMSE(1, indx_EbN0) +
            sum(DetectionBitSequence_MMSE~=BitSequence);
        ErrorCount_MLD(1, indx_EbN0) = ErrorCount_MLD(1, indx_EbN0) + sum(
            DetectionBitSequence_MLD~=BitSequence);
    end
end

BER_Simulation_ZF = ErrorCount_ZF / (LengthBitSequence * NumberIteration);
BER_Simulation_MMSE = ErrorCount_MMSE / (LengthBitSequence *
    NumberIteration);
BER_Simulation_MLD = ErrorCount_MLD / (LengthBitSequence * NumberIteration
    );

if M==2
    modtype='psk'
else
    modtype='qam'
end
BER_Theory = berfading(EbN0_dB, qam, M, 1);
```

```
70
71 % Plot
72 figure()
73 semilogy(EsN0_dB, BER_Theory, 'r--');
74 hold on
75 semilogy(EsN0_dB, BER_Simulation_ZF, 'bo');
76 semilogy(EsN0_dB, BER_Simulation_MMSE, 'bx');
77 semilogy(EsN0_dB, BER_Simulation_MLD, 'b^');
78
79
80 axis([-2 20 10^-3 0.5])
81 grid on
82 legend('Theory (Rayleigh)', 'ZF (Rayleigh)', 'MMSE (Rayleigh)', 'MLD (
       Rayleigh)');
83 xlabel('Es/No [dB]');
84 ylabel('BER');
85 title('BER for QAM (M='+string(M)+')');
```

1.1 ZF(Zero-forcing)

1.2 MMSE(Minimum Mean Square Error)

1.3 MLD(Maximum Likelihood Detection)

2 M-ary QAM

$M = 2^2 n$ $(n = 1, 2, 3, ...)$의 상황을 가정하였다. 한 가지 생각해볼 만한 사항은 Normalization Factor이다. 이 Normalization Factor를 사용하여 평균 전력이 1W가 되게끔 둘 수 있다.

QAM의 일반적인 Constellation Diagram을 살펴보면 실수 $\sqrt{M}$개, 허수 $\sqrt{M}$개의 point를 갖는 것을 알 수 있다.
하나의 지점을 하나의 alphabet이라고하자. M개의 alphabet은 다음과 같다.

$$alphabet = \pm(2n-1) \pm j \cdot (2n-1) \qquad n \in 1, 2, ..., \sqrt{M} \tag{1}$$

신호의 평균전력은 다음과 같이 일반화 가능하다.

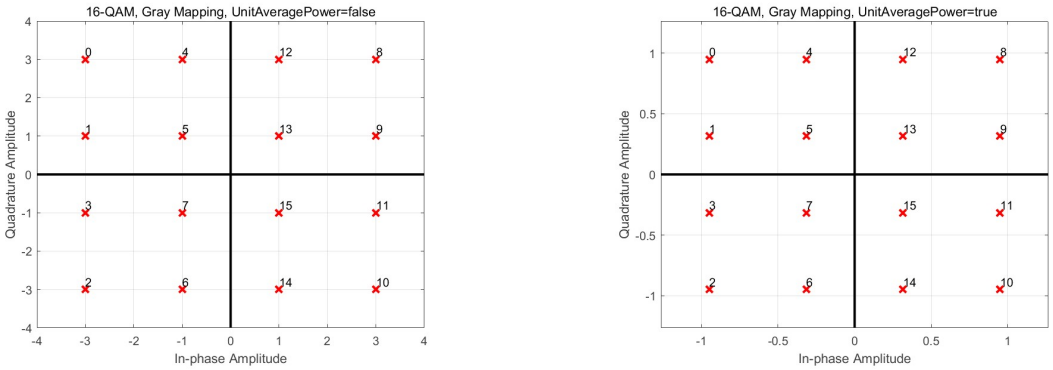
$$\begin{aligned} E_s = E[|s|^2] &= \frac{1}{M}\sum_{n=1}^{M}|s_n|^2 \\ &= \frac{1}{M}\sum_{n=1}^{\sqrt{M}}\sum_{m=1}^{\sqrt{M}}|(2m-1)^2 + (2n-1)^2| \\ &= \frac{1}{M}\cdot 4\sum_{n=1}^{\frac{\sqrt{M}}{2}}[(2n-1)^2 \cdot \sqrt{M}] \\ &= \frac{4}{M}\sum_{n=1}^{\frac{\sqrt{M}}{2}}[4n^2 - 4n + 1] \\ &= \frac{2}{3}(M-1) \end{aligned} \tag{2}$$

(2)에서의 결과를 토대로 normalization이 이뤄진 alphabet을 구할 수 있다.

$$normalized \quad alphabet = \Big[\pm\frac{2n-1}{\sqrt{\frac{2}{3}(M-1)}} \pm j \cdot \frac{2n-1}{\sqrt{\frac{2}{3}(M-1)}}\Big] \qquad n \in \{1, 2, ..., \sqrt{M}\} \tag{3}$$

해당 결과를 토대로 다시 평균 전력을 구한다면 E_s가 1W임을 확인할 수 있다.

참고자료
다음은 16-QAM의 Constellation이다.



(a) Non-normalized Constellation (b) Normalized Constellation

Figure 1: 16-QAM Constellation

2.1 ZF(Zero-forcing)

2.2 MMSE(Minimum Mean Square Error)

2.3 MLD(Maximum Likelihood Detection)

3 과제 외적 의문점 및 질문

- 왜 $M = 4$ 일때의 BER은 ZF와 MMSE의 경우 다르지만, $M = 16$일 때는 왜 모든 값이 같은가?
- SNR이 얼마나 커져야 ZF와 MMSE의 BER이 같아질까? 수학적 공식으로 나타낼 수 있는가?