

# 디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001



12 주차 실습과제

정보통신공학과

12191765

박승재

# Introduction

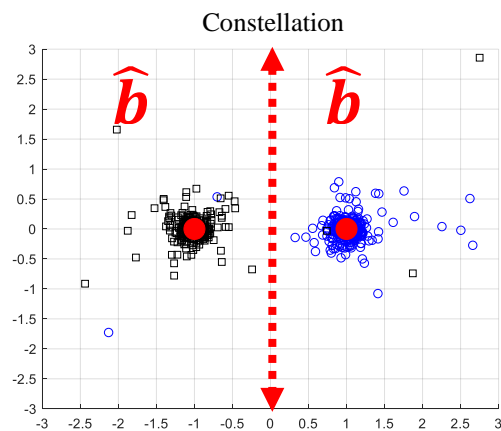
이미지 파일을 비트 스트림으로 변환시킨 후 BPSK 변조하는 실습이다.

$$s = \begin{cases} \sqrt{E_b}, & \text{if } b = 1 \\ -\sqrt{E_b}, & \text{if } b = 0 \end{cases}$$

이때, 레일리 채널을 생성하고 잡음을 더해 수신 신호를 생성한다.

$$r = \frac{h^*}{|h|^2} y = s + n'$$

비트 추정은 아래와 같이 신호의 실수부의 부호를 통해 원래의 신호를 추정한다.



$$\hat{b} = \begin{cases} 0, & \text{if } \Re\left(\frac{h^*}{|h|^2} y\right) < 0 \\ 1, & \text{otherwise} \end{cases}$$

```
Eb_No_dB = 10;  
Eb_mW = 1;  
Eb_dBm = pow2db(Eb_mW);  
No_dBm = Eb_dBm - Eb_No_dB;  
No_mW = db2pow(No_dBm);  
  
image_original = imread("original.jpg");  
  
image_height = size(image_original, 1);  
image_width = size(image_original, 2);  
image_channel = size(image_original, 3);  
image_level = double(max(max(max(image_original))));  
image_level_bin = ceil(log2(image_level));  
  
image_vec = image_original(:);  
image_bit = de2bi(image_vec);
```

```

bit_stream = image_bit(:);
N = length(bit_stream);

symbol = double(bit_stream) * 2 - 1; % symbol: -1 or 1

h = (randn(N, 1) + 1j * randn(N, 1)) / sqrt(2);
noise = sqrt(No_mw / 2) * (randn(N, 1) + 1j * randn(N, 1)); % noise

y = symbol .* h + noise;

r = conj(h) ./ (abs(h) .^ 2) .* y;
bit_stream_re = real(r) > 0;

image_bit_re = reshape(bit_stream_re, [image_height * image_width *
image_channel, image_level_bin]);
image_vec_re = bi2de(image_bit_re);
image_re = uint8(reshape(image_vec_re, [image_height, image_width,
image_channel]));

imshow(image_re);

```

$E_b/N_0$  를 수정하며 이미지를 출력하면 아래와 같이 나온다.

**$E_b/N_0 = -10\text{dB}$**



**$E_b/N_0 = 0\text{dB}$**



**$E_b/N_0 = 10\text{dB}$**



$E_b/N_0$  가 커질수록 이미지가 선명해지는 것을 확인할 수 있다.

## Problem

상이한  $E_b/N_0$  를 통해 생성된 세 가지 수신신호를 'homework.mat' 파일에 저장하였음

'homework.mat'에 저장된 변수들(struct 형태) :

- 저장된 수신신호의 파일명: 'y1', 'y2', 'y3'
- 세 가지 수신신호는 동일한 Rayleigh 페이딩 채널을 통해 수신됨: 'h'
- 이미지 관련 파라미터: 'Height', 'Width', 'RGB Channel', 'Level\_binary'

'y1', 'y2', 'y3'로부터 이미지 파일을 각각 복원해보고 세 수신 신호들을 Eb/No 가 높은 순서대로 정하십시오.

## Result

```
r1 = conj(h) ./ (abs(h) .^ 2) .* y1;
bit_stream_re1 = real(r1) > 0;
image_bit_re1 = reshape(bit_stream_re1, [Width_ * Height_ * CH_,
Level_binary]);
image_vec_re1 = bi2de(image_bit_re1);
image_re1 = uint8(reshape(image_vec_re1, [Height_, Width_, CH_]));

r2 = conj(h) ./ (abs(h) .^ 2) .* y2;
bit_stream_re2 = real(r2) > 0;
image_bit_re2 = reshape(bit_stream_re2, [Width_ * Height_ * CH_,
Level_binary]);
image_vec_re2 = bi2de(image_bit_re2);
image_re2 = uint8(reshape(image_vec_re2, [Height_, Width_, CH_]));

r3 = conj(h) ./ (abs(h) .^ 2) .* y3;
bit_stream_re3 = real(r3) > 0;
image_bit_re3 = reshape(bit_stream_re3, [Width_ * Height_ * CH_,
Level_binary]);
image_vec_re3 = bi2de(image_bit_re3);
image_re3 = uint8(reshape(image_vec_re3, [Height_, Width_, CH_]));

figure()
imshow(image_re1);
figure()
imshow(image_re2);
figure()
imshow(image_re3);
```

아래는 y 에 따라 출력된 이미지이다.

y1



y2



y3



$E_b/N_0$  는 이미지의 선명도를 결정한다. 따라서  $y_2, y_3, y_1$  순서대로  $E_b/N_0$  가 높을 것이다.

## Conclusion

$E_b/N_0$  는 잡음 대비 에너지로, 클수록 잡음의 영향이 줄어든다. 잡음은 전송하는 이미지를 왜곡시킨다. 따라서 잡음이 적을수록 수신된 이미지는 왜곡이 덜 할 것이다. 수신된 이미지에 노이즈가 많이 생겨 원본과 차이가 많이 발생한 것은, 잡음의 영향으로 이미지가 왜곡되었기 때문이다.  $y_1$  은 가장 많은 노이즈가 생겼는데, 그렇기 때문에  $E_b/N_0$  가 가장 낮다는 것을 유추할 수 있다. 실습 때  $E_b/N_0$  가  $-10\text{dB}$  일 때를 실습해봤는데, 그 때보다 더 이미지가 왜곡되는 것으로 보아,  $-10\text{dB}$  이하의  $E_b/N_0$  를 가질 것이라 생각한다.

$y_2$  는 가장 선명한 이미지를 출력했는데,  $E_b/N_0$  값이 커서 잡음의 영향을 가장 덜 받았기 때문이다.  $E_b/N_0$  가 크면 오류가 발생할 확률  $P_e$  가 줄어들어 잡음의 영향이 줄어든다.

$y_3$  는 중간 정도의 선명도를 가졌기 때문에,  $E_b/N_0$  가 중간정도로 유추할 수 있다. 따라서 수신 신호의  $E_b/N_0$  는  $y_2, y_3, y_1$  순서대로 높을 것이다.