

# 디지털통신시스템설계

Digital Communication System Capstone Design

ICE4009-001



실습 프로젝트

정보통신공학과

12191765

박승재

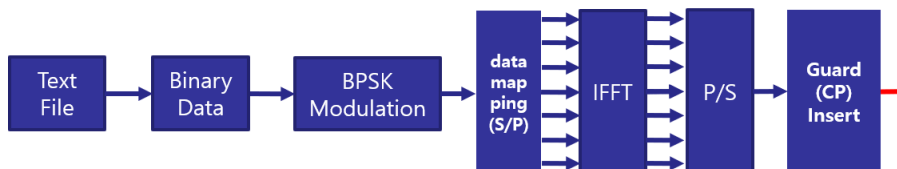
# Introduction

## 프로젝트 목표

1. PSK-변조 방식이 적용된 수신신호를 복조하여 얻어진 이미지 데이터를 출력  
임의의 PSK-변조 방식으로 변조된 이미지 파일을 복원하여 송신기에서 사용한 변조 방식 역추적
2. OFDM 수신기 생성하여 문자 데이터를 출력  
사전에 정해진 파라미터들을 활용해 만들어진 OFDM 신호를 복조할 수 있는 수신기 생성

## Problem

1. 수신한 이미지 파일을 복원하고 상기 이미지 파일에 쓰여있는 문자를 맞추기
  - ✓ 복원된 이미지
  - ✓ 예측한 변조방식 (BPSK or QPSK)
  - ✓ 채널의 SNR
2. 아래 그림과 테이블에 기반하여 설계된 OFDM 송신기 구조를 이해하고 OFDM 송신기에 의해 생성된 신호를 복원할 수 있는 OFDM 수신기 생성



OFDM 관련 파라미터	값
Subcarrier spacing, $\Delta f$ (kHz)	15
OFDM symbol ( $\mu s$ )	$1/\Delta f$
Guard Interval : OFDM symbol ratio	144 : 2048
FFT Size, $N_{FFT}$	2048
Subcarriers, $N_{subs}$	1300
Subcarrier Index	$[2:\frac{N_{subs}}{2} + 1, N_{FFT} - \frac{N_{subs}}{2} + 1:N_{FFT}]$ $\rightarrow [2:651, 1399:2048]$

- ✓ 이전 슬라이드의 OFDM 송신기 정보를 기반으로 올바른 수신기를 생성한 후 복원된 wav 파일
- ✓ 노래 제목

# Result

## Problem 1

```
r = conj(h) ./ (abs(h) .^ 2) .* y;

bit_stream_re = pskdemod(r, 2, pi); % BPSK: % 1+0j -> 1, -1+0j -> 0

image_bit_re = reshape(bit_stream_re, [Height_ * Width_ * RGB_CH,
Level_binary]);
image_vec_re = bi2de(image_bit_re);
image_re = uint8(reshape(image_vec_re, [Height_, Width_, RGB_CH]));

figure()
imshow(image_re);

N = Width_ * 25;
ber = sum(~bit_stream_re(1:N)) / N

Eb_No = (1 - 4 * ber + 4 * (ber ^ 2)) / (4 * ber - 4 * (ber ^ 2));
Eb_No_dB = pow2db(Eb_No)
```



```
>> project1

ber =

    0.0933

Eb_No_dB =

    2.9133
```

- ✓ 복원된 이미지: "참! 잘했어요"
- ✓ 예측한 변조방식: BPSK
- ✓ 채널의 SNR: 2.91dB

## Problem 2

```
% OFDM symbols
rx_symbols = reshape(y', [], N_OFDM_symbols)';

symbols = zeros(N_OFDM_symbols, nSubcarrier);
for t = 1:N_OFDM_symbols
    rx_ofdm = rx_symbols(t, :);

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

    % S/P
    rx_signal = rx_signal';

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

    % Data de-mapping
    rx_demap = rx_fft(subcarrierIndex);

    % P/S
    rx_demap = rx_demap';

    % BPSK demodulation
    rx = pskdemod(rx_demap, modOrder, pi); % 1+0j -> 1, -1+0j -> 0

    symbols(t, :) = rx;
end
symbols = symbols';

x_en = symbols(:);
x_en = x_en(1:end - 2); % drop last 2 bits

% De-quantization
Q_level = 64;

x_de = bin2dec(num2str(reshape(x_en, log2(Q_level), []))');
x_s = zeros(size(x_de));
for t = 1:length(x_de)
    x_s(t) = Sampling_values(x_de(t) + 1);
end

% Reconstruction
t_start = 0; % sampling start time
t_end = 7.4; % sampling end time
t_s = 1 / Sampling_frequency; % sampling interval
```

```

t = t_start:t_s:t_end;

x = zeros(1, length(t));
for idx = 1:length(x_s)
    x = x + sinc(t / t_s - (idx - 1)) * x_s(idx);
end
x(x < -1) = -1; % normalization: -1 < x < 1

audiowrite('project2.wav', x, Sampling_frequency);

```



UNFORGIVEN

LE SSERAFIM

이브, 프시케 그리고 푸른 수염의 아내

LE SSERAFIM (르세라핌)

앨범	UNFORGIVEN	댓글	810개 >
발매일	2023.05.01	공유	
장르	댄스		
FLAC	Flac 16/24bit		어제의 차트 순위 6위

57,748  
 > 곡 다운 >  
 > FLAC 다운 >  
 > 선물하기 >

## 가사

프린트

가사오류신고

I'm a mess mess mess mess  
 mess mess mess  
 I'm a mess mess mess mess  
 mess mess mess  
 I'm a mess in distress  
 but we're still the best dressed  
 Fearless say yes, we don't dress to impress  
 괜찮단다 뭐를 해도 거짓말인 걸 난 알아  
 괜찮겠지 뭐 해도 착한 얼굴에 네 말 잘 들을 땐  
 괜찮지 않아 그런 건 내 룰은 나만 정할래 yeah  
 볼 거야 금지된 걸  
 Never hold back 더 자유롭게  
 Boom boom boom 내 심장이 뛰네  
 Get it like boom boom boom  
 Get it like boom boom boom

✓ 노래 제목: 이브, 프시케 그리고 푸른 수염의 아내

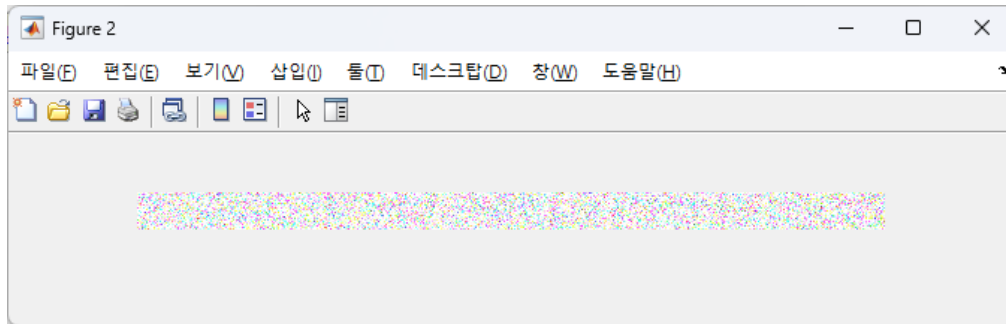
## Conclusion

### Problem 1

수신신호에 채널 보상을 곱하면 잡음이 포함된 원 신호를 복원할 수 있다. `pskdemod` 를 사용할 때 주의해야 할 점이 위상이 0(기본값)이면 Constellation 상에서  $1+0j$ 가 0,  $-1+0j$ 가 1로 매핑되기 때문에 위상을 180 도 돌려줘야 실습 때 배웠던 대로,  $1+0j$ 가 1,  $-1+0j$ 가 0으로 매핑된다.

먼저 BPSK 방식으로 변조되었을 것이라 예측해 이미지를 복원하니, "참! 잘했어요"라는 이미지가 나왔다. QPSK 방식으로 이미지 복원을 시도했지만, 해당 방식으로는 유의미한 이미지를 얻지 못했다.

채널의 SNR 은 약 2.91dB 가 나왔는데, 그 이유는 이미지 상단의 흰색 여백(RGB: 255, 255, 255)은 모든 비트가 1 로 나와야 하기 때문에 해당 부분을 이용해 BER 을 구하고 이를 토대로 SNR 을 역산했다.



```
image_re_top = image_re(1:25, :, :);
figure()
imshow(image_re_top);
```

Width\_ \* 25 을 하면 위와 같이 흰색 여백만 뽑아낼 수 있고 여기서 1 이 아닌 비트 개수를 구해 BER 을 계산했다. 그 결과 BER 은 0.09 정도가 나왔다.

BER 을 구했으므로 이제 SNR 을 구할 수 있다. 레일리 채널에서의 BER 이론값은 아래와 같다.

$$BER = \frac{1}{2} \left( 1 - \sqrt{\frac{SNR}{SNR + 1}} \right)$$

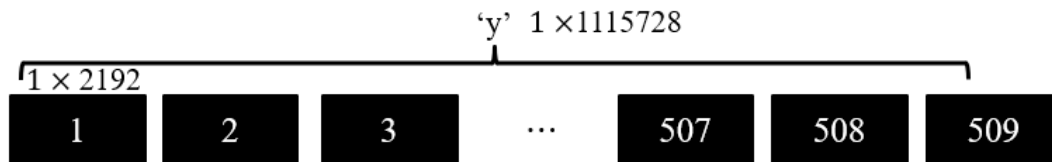
해당 식은 아래와 같이 SNR 로 정리될 수 있다.

$$\begin{aligned} 2BER &= 1 - \sqrt{\frac{SNR}{SNR + 1}} \\ 1 - 2BER &= \sqrt{\frac{SNR}{SNR + 1}} \\ (1 - 2BER)^2 &= \frac{SNR}{SNR + 1} \\ (1 - 2BER)^2(SNR + 1) &= SNR \\ (1 - 2BER)^2 SNR + (1 - 2BER)^2 &= SNR \\ (-4BER + 4BER^2)SNR + (1 - 2BER)^2 &= 0 \\ SNR &= \frac{(1 - 2BER)^2}{(4BER - 4BER^2)} = \frac{(1 - 4BER + 4BER^2)}{(4BER - 4BER^2)} \end{aligned}$$

구한 SNR 값은 1.96 이고 dB 로 변환하면 SNR 은 2.91dB 가 나온다.

## Problem 2

제시된 OFDM 수신기를 따라서 신호를 복원한다.



$y$  는 509 개의 OFDM symbol 이 포함되어 있고, 각 심볼에는 Cyclic Prefix 를 포함한 IFFT 된 신호가 들어있다.

 `rx_symbols`      509x2192 complex ...

```
rx_symbols = reshape(y', [], N_OFDM_symbols)';
```

위 코드를  $y$  를 symbol 별로 나누어 각 행에 저장한다.

 `rx_signal`      2048x1 complex do...

```
rx_signal = rx_ofdm((nSampGI + 1):end);
```

각 symbol 에서 CP 를 떼고 나면 2048 크기의 벡터가 나온다.

 `rx_demap`      1x1300 complex do...

```
rx_demap = rx_fft(subcarrierIndex);
```

De-mapping 을 하면 1300 크기의 벡터가 나온다.

BPSK 로 수신한 데이터를 복원하고 하나의 벡터로 합쳐준다. Problem 1 과 마찬가지로, `pskdemod` 를 사용할 때는 위상을 180 도 돌려야 실습 때 배웠던 대로 데이터가 복원된다.

 `x_en`      661698x1 double

```
x_en = symbols(:);  
x_en = x_en(1:end - 2); % drop last 2 bits
```

마지막에 2 비트를 버리는 이유는 수신한 데이터의 비트 수가 6 으로 나누어 떨어지지 않기 때문이다.

 `x_de`      110283x1 double

```
x_de = bin2dec(num2str(reshape(x_en, log2(Q_level), []))');
```

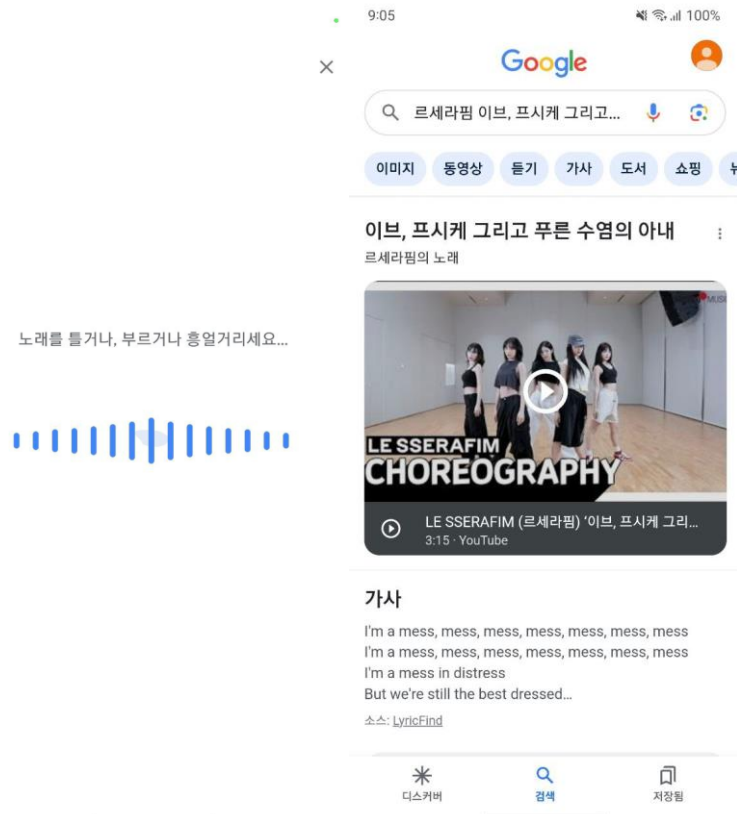
64 등분해서 양자화 했기 때문에, 위와 같이 복원할 수 있다. 첫 번째 비트가 MSB 이다. 111001 은 이진수로 57 로 변환된다.

주어진 `Sampling_values` 를 이용해 샘플링된 값을 추출하고, sinc 함수를 이용해 샘플링된 값을 interpolation 한다. sinc 함수를 사용하지 않아도 샘플 간격이 촘촘하기 때문에 노래를 맞출 수 있지만, sinc 함수를 사용하면 더 선명한 노래 파일이 나온다.

$x(x < -1) = -1$ ; % normalization:  $-1 < x < 1$

마지막에 위와 같이 -1 과 1 사이로 정규화하는 부분이 있는데, audiowrite 의 입력은 -1 과 1 사이가 되어야 하기 때문이다. 양자화와 샘플링 오차로 인해 -1 미만의 값이 생긴 것으로 추정된다.

노래 파일이 나왔지만 들어도 무슨 노래인지 몰라서 기계의 힘을 빌렸다.



구글 검색한데 노래를 들려주니 “이브, 프시케 그리고 푸른 수염의 아내”라고 찾아주었다. 해당 음악을 검색해 직접 들어보며 동일한 노래인 것을 확인했다.

프로젝트를 하면서 실험적인 BER 값을 구하고 이를 통해 채널의 SNR 값을 유추하는 방법을 배웠다. OFDM 과 양자화, 샘플링이 복합적으로 응용된 과제를 풀면서 디지털 통신에 대한 개념도 다시금 복습할 수 있었다. 특히, 과제를 풀면 변조된 그림과 음악 파일이 나오는 점이 미니게임을 하는 것 같아 재미있었다. 한 학기 동안 배운 내용을 검토할 수 있는 유의미한 프로젝트 같다.