OFDM
Computer Simulation

Student: SU PEI-JIN

2022/03/09 (Thursday)

Outline

- OFDM System Parameters
- Linear Time-invariant (LTI) Channel
- Block Diagram
- Signal Model
- Theoretical BER
- Simulation Results
- Discussion

OFDM System Parameters

- Modulation: BPSK
- Number of subcarriers (FFT size): $N = 64$ (samples)
- Length of cyclic prefix (CP): $l_{CP} = \frac{N}{4} = 16$ (samples)
- Number of data bits: $num_{bits} = 32 \times 10^6$ (bits)

- Channel: Frequency-selective & AWGN channel
- Number of discrete path delay: $num_{taps} = 8$

Linear Time-invariant (LTI) Channel

- Choose a LTI channel

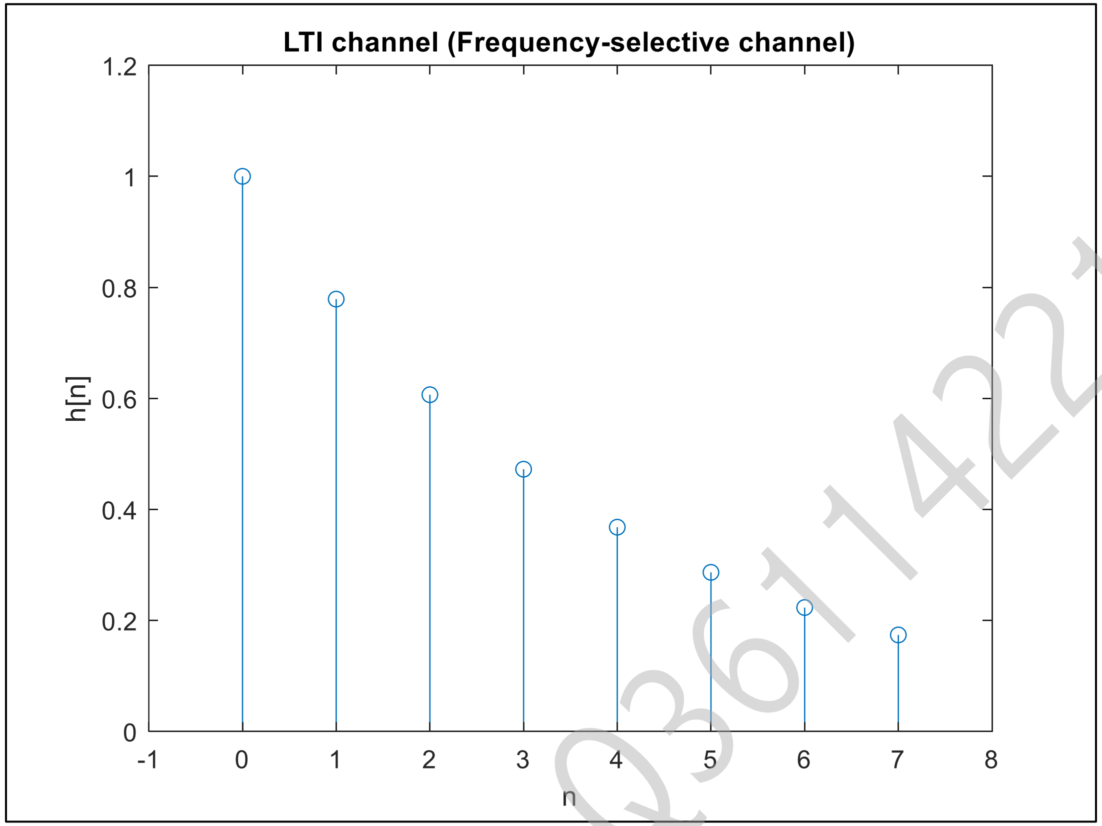
$$h[n] = e^{-0.25n}, n = 0,1,\dots,7$$

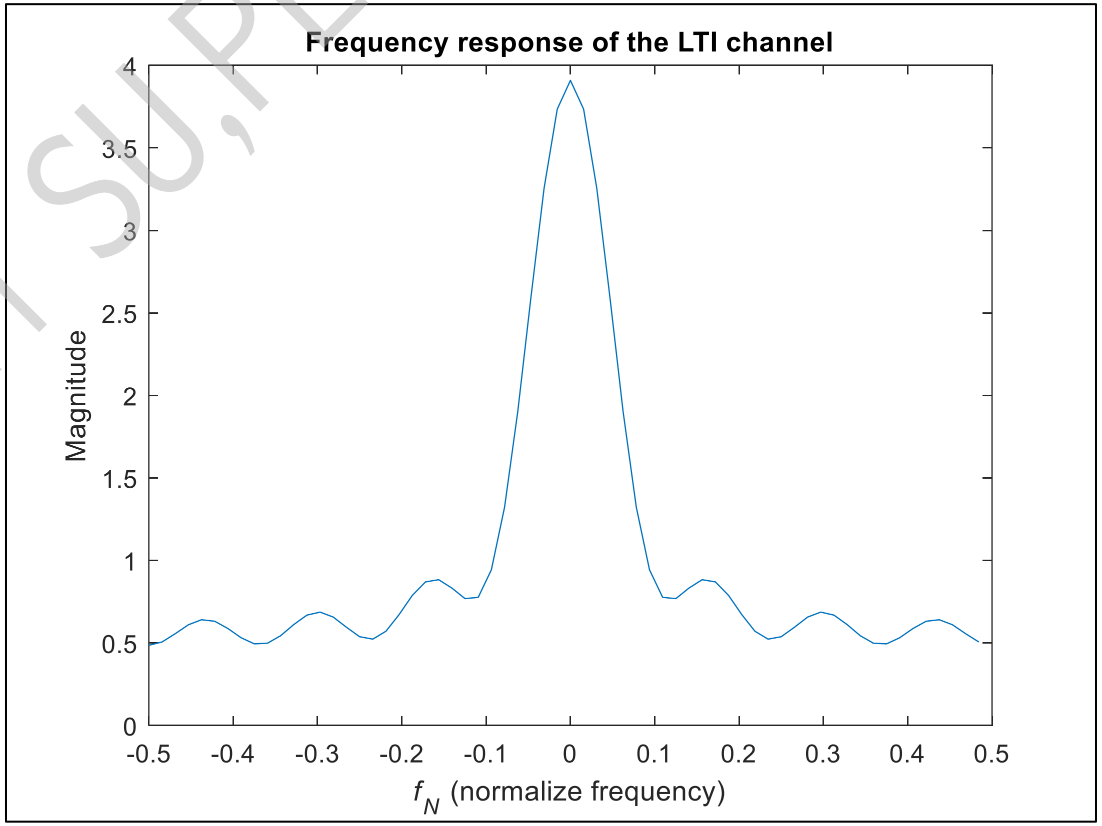
```
% Frequency-selective Channel (LTI Channel)
num_taps = 8;    % Number of discrete path delay
h = exp(-0.25*(0:num_taps-1));
```

- Number of discrete path delay: $num_{taps} = 8$

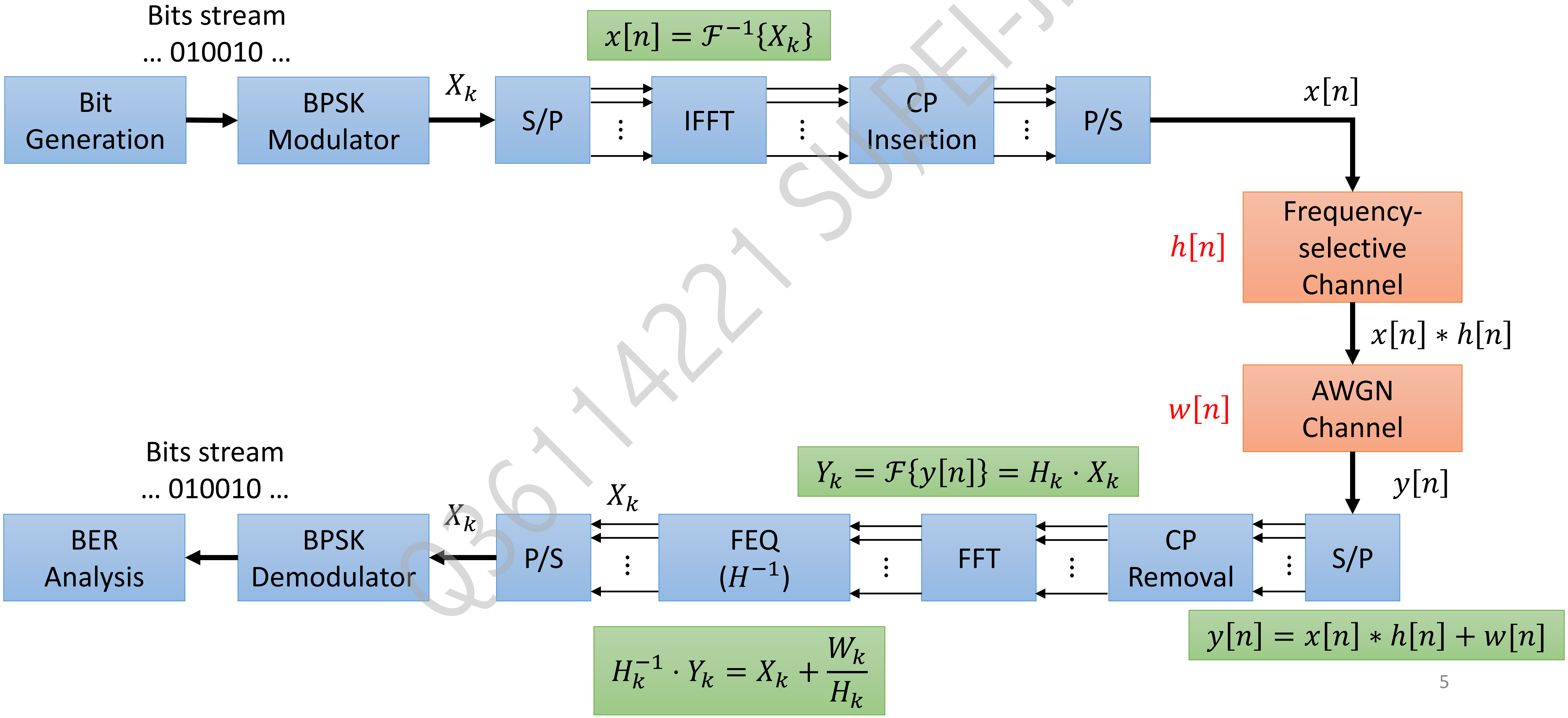
$$h[n] = [1, 0.7788, 0.6065, 0.4724, 0.3679, 0.2865, 0.2231, 0.1738]$$

Linear Time-invariant (LTI) Channel





Block Diagram



Signal Model

- Received OFDM signal

$$y[n] = h[n] * x[n] + w[n], \qquad n = 0, 1, \ldots, N-1.$$

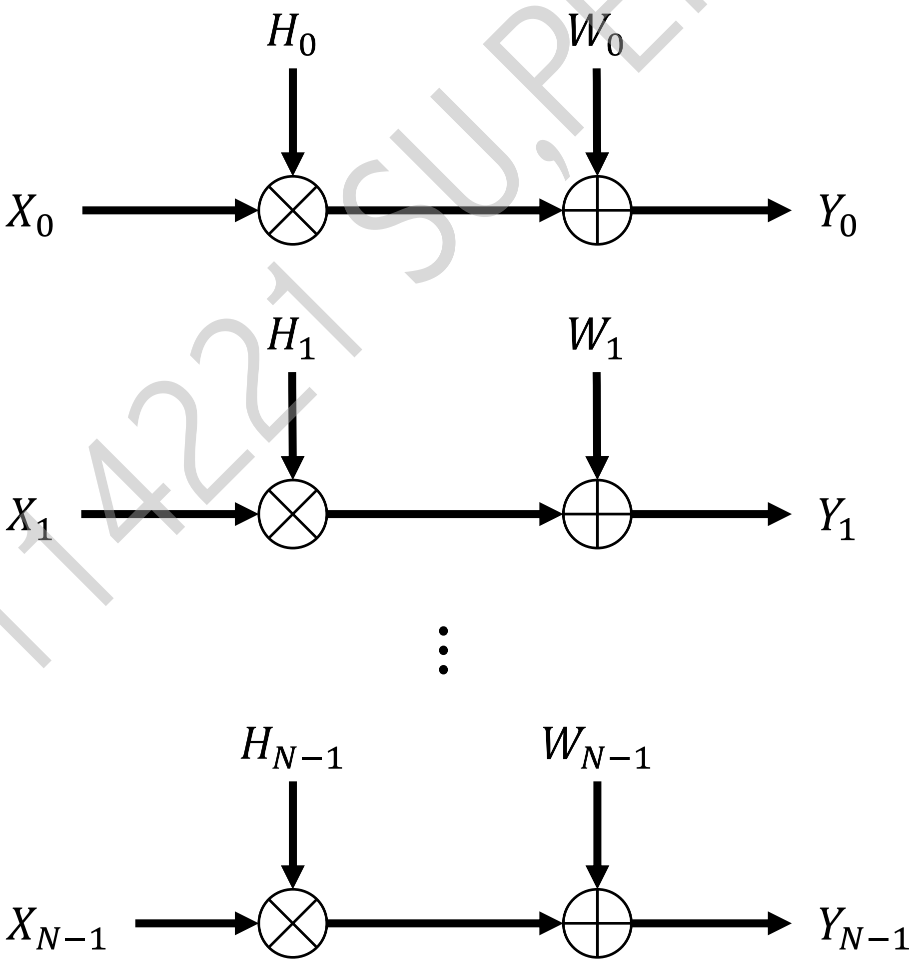
$$\xrightarrow{\text{DFT}} \quad Y_k = X_k \cdot H_k + W_k, \qquad k = 0, 1, \ldots, N-1.$$

$$\begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{N-1} \end{bmatrix} = \begin{bmatrix} H_0 & 0 & \cdots & 0 \\ 0 & H_1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & H_{N-1} \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} + \begin{bmatrix} W_0 \\ W_1 \\ \vdots \\ W_{N-1} \end{bmatrix}$$

$$\boldsymbol{Y} = \mathbf{D_H} \cdot \boldsymbol{X} + \boldsymbol{W}$$

Signal Model

- Parallel Channels



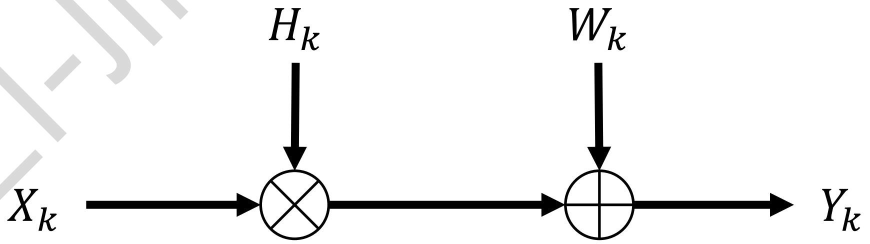
Theoretical BER

- Assume $\quad \mathrm{SNR} = \frac{2E_b}{N_0}$
- Noise: $\mathcal{N}(0, \sigma^2)$, $\sigma = \frac{N_0}{2}$
- Theoretical BER for BPSK modulation in an AWGN channel

$$P_e = Q\left(\sqrt{2E_b/N_0}\right)$$
$$= Q\left(\sqrt{\mathrm{SNR}}\right)$$

Theoretical BER

- For k-th subcarrier



$$Y_k = H_k \cdot X_k + W_k$$

$$\mathrm{SNR}_{Y_k} = \frac{E[|H_k \cdot X_k|^2]}{E[|W_k|^2]} = \frac{|H_k|^2 \cdot E[|X_k|^2]}{E[|W_k|^2]} = |H_k|^2 \cdot \mathrm{SNR}_{X_k}$$

$$\boxed{\mathrm{SNR}_{X_k} = \frac{E[|X_k|^2]}{E[|W_k|^2]} = \frac{E[|x[n]|^2]}{E[|w[n]|^2]} = \frac{2E_b}{N_0}}$$

Theoretical BER

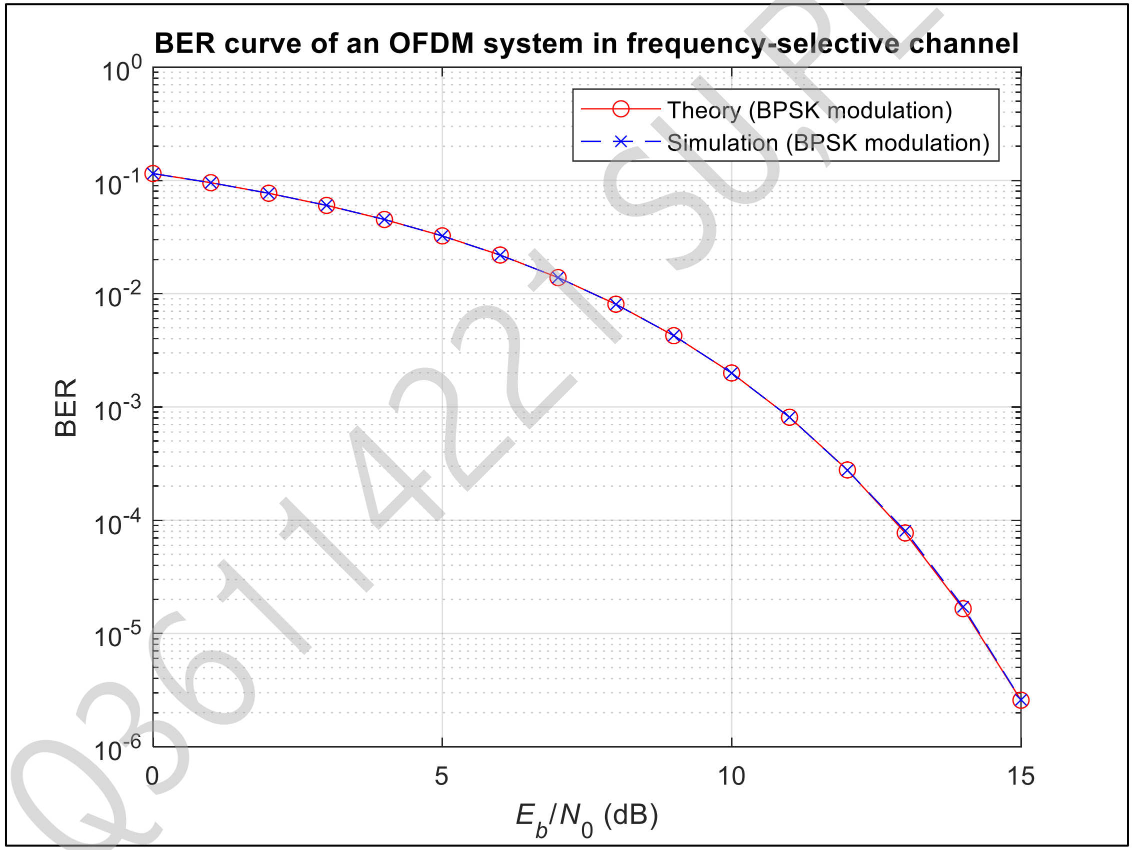
- BER for k-th subcarrier

$$\mathrm{BER}_{Y_k} = Q\left(\sqrt{\mathrm{SNR}_{Y_k}}\right) = Q\left(\sqrt{|H_k|^2 \mathrm{SNR}_{X_k}}\right) = Q\left(|H_k|\sqrt{\mathrm{SNR}_{X_k}}\right)$$

- Theoretical BER of an OFDM system with BPSK modulation for the LTI channel

$$\begin{aligned}\mathrm{BER}_{\mathrm{OFDM,BPSK}} &= E\left[\mathrm{BER}_{Y_k}\right], \qquad k = 0, 1, \ldots, N-1 \\ &= \frac{1}{N}\sum_{k=0}^{N-1} \mathrm{BER}_{Y_k} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} Q\left(|H_k|\sqrt{\mathrm{SNR}_{X_k}}\right)\end{aligned}$$

Simulation Results



Discussion

- Q: Will you do something different if the modulation is QAM instead of BPSK?
- A:
    1. 當 OFDM 系統的數位調變方式改變為 QAM 調變時，則模擬程式只需調整前端的 Signal Mapper/Demapper 部分，而 OFDM 的主架構不用更改。
    2. 而使用 QAM 進行調變的 OFDM 系統，其理論錯誤率 (i.e., SER/BER) 仿照前面 BPSK 調變的 BER 計算方式，重新進行推導。

Discussion

- Q: If you don’t add noise, will you be able to achieve 0% error rate?
- A:
  - 當傳送的 OFDM 訊號無加入雜訊時，由於接收端有使用頻率域等化器 (Frequency-domain equalizer) ，所以會將接收訊號受 frequency-selective channel 產生的通道增益進行補償。

```matlab
num_bit = 32*10^6;  % Number of data bits

Eb = 1;     % Energy per symbol
BPSK = [-sqrt(Eb), sqrt(Eb)];   % BPSK symbol mapping

SNR_in_dB = 0:1:20;     % SNR (Eb/N0) in dB
SNR = 10.^(SNR_in_dB/10);     % SNR (Eb/N0)
num_bit_error = zeros(size(SNR_in_dB));    % Number of bit error
BER_theo = zeros(size(SNR_in_dB));

% Initialize the random number generator
rng('default');

% Bit generation
tx_bit = ceil(2.*rand(1, num_bit))-1;

% BPSK Mapper
tx_BPSK_sym = BPSK(tx_bit(1:num_bit)+1); %% Transmit signal

% Serial to Parallel Converter
FFT_size = 64;  % Number of subcarriers
num_OFDM_sym = num_bit/FFT_size;
tx_sig = reshape(tx_BPSK_sym,[FFT_size,num_OFDM_sym]);

% IFFT
tx_IFFT_sig = ifft(tx_sig)*sqrt(FFT_size);

% CP Insertion
CP_size = FFT_size/4;
tx_OFDM_sym = [tx_IFFT_sig((FFT_size-CP_size+1):end,:); tx_IFFT_sig];
```

```
% Parallel to Serial Converter
tx_OFDM_sig = reshape(tx_OFDM_sym,1,(FFT_size+CP_size)*num_OFDM_sym);

% Frequency-selective Channel (LTI Channel)
num_taps = 8;   % Number of discrete path delay
h = exp(-0.25*(0:num_taps-1));

H = fft(h,FFT_size);
fre_sel_ch_sig = conv(tx_OFDM_sig,h);

for snr_Idx = 1:length(SNR_in_dB)
    % AWGN Channel
    sigma = sqrt(Eb/(2*SNR(snr_Idx)));
    noise = randn(1, length(fre_sel_ch_sig))*sigma + 1i*randn(1, length(fre_sel_ch_sig))*sigma;
    AWGN_ch_sig = fre_sel_ch_sig + noise;

    rx_sig = AWGN_ch_sig(1:(FFT_size+CP_size)*num_OFDM_sym);

    % Serial to Parallel Converter
    rx_OFDM_sig = reshape(rx_sig,[FFT_size+CP_size,num_OFDM_sym]);

    % CP Removal
    shift_FFT_window = 0;   % different starting time of FFT window
    rx_OFDM_sym = rx_OFDM_sig((CP_size + 1)-shift_FFT_window:end-shift_FFT_window,:);
    rx_OFDM_sym = circshift(rx_OFDM_sym,-shift_FFT_window,1);

    % FFT
    rx_FFT_sig = fft(rx_OFDM_sym)/sqrt(FFT_size);

    % Frequency domain equalizaion (FEQ)
    H_transpose = H.';
    rx_FEQ_sig = rx_FFT_sig(:,1:end)./H_transpose;
```

```matlab
    % Parallel to Serial Converter
    rx_BPSK_sym = reshape(rx_FEQ_sig,1,FFT_size*num_OFDM_sym);

    % BPSK Demapper
    rx_bit=zeros(1,FFT_size*num_OFDM_sym);
    rx_bit(rx_BPSK_sym>0) = 1; % decision of bit '1'

    % BER analysis
    err_pat = xor(tx_bit, rx_bit);      % compare tx and rx bits
    num_bit_error(snr_Idx) = num_bit_error(snr_Idx) + sum(err_pat); % error counting

    % Theoretical BER curve
    BER_theo(snr_Idx) = mean(qfunc(abs(H).*sqrt(2*SNR(snr_Idx))));
end

% BER performance via simulation
BER_sim = num_bit_error/num_bit;    % Simulated BER

semilogy(SNR_in_dB, BER_theo,'r-o');
hold on;
semilogy(SNR_in_dB, BER_sim,'b--x');
axis([0 15 1e-6 1]);
title('BER curve of an OFDM system in frequency-selective channel');
legend('Theory (BPSK modulation)','Simulation (BPSK modulation)');
xlabel('\itE_{b}\rm/\itN\rm_{0} (dB)');
ylabel('BER');
grid;
```