Alec Mabhiza Chirawu - 亚历克上 - M202161029

Wireless Communication - Assignment 3

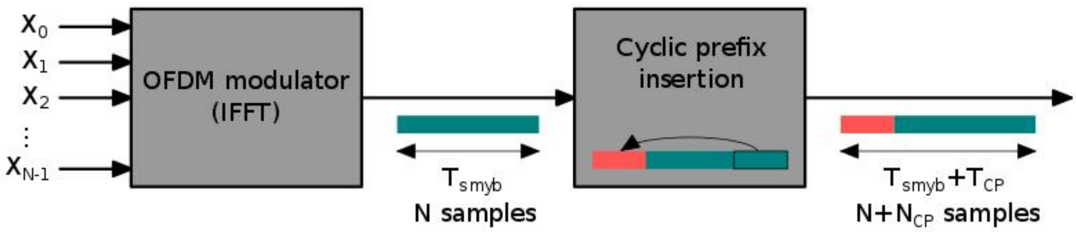
Task 1: OFDM Algorithm Simulation

Orthogonal frequency-division multiplexing(OFDM) is a type of digital transmission and a method of encoding digital data on multiple carrier frequencies. OFDM has developed into a popular scheme for wideband digital communication, used in applications such as wireless networks, power line networks, and 4G/5G mobile communications. In OFDM, multiple closely spaced orthogonal subcarrier signals with overlapping spectra are transmitted to carry data in parallel. Each signal is modulated with a conventional modulation scheme such as quadrature amplitude modulation or phase-shift keying at a low symbol rate. This maintains total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions. For example, attenuation of high frequencies in a long copper wire, narrowband interference and frequency-selective fading due to multipath without the need for complex equalization filters. Channel equalization is simplified because OFDM may be viewed as using many slowly modulated narrowband signals rather than one rapidly modulated wideband signal. The low symbol rate makes the use of a guard interval between symbols affordable, making it possible to eliminate intersymbol interference (ISI) and use echoes and time-spreading to achieve a diversity gain, therefore a signal-to-noise ratio improvement. This mechanism also facilitates the design of single frequency networks (SFNs) where several adjacent transmitters send the same signal simultaneously at the same frequency, as the signals from multiple distant transmitters may be re-combined constructively, sparing interference of a traditional single-carrier system.

In coded OFDM, forward error correction (convolutional coding) and time/frequency interleaving are applied to the signal being transmitted. This is done to overcome errors in mobile communication channels affected by multipath propagation and Doppler effects.

- 1 example of 2 x 2 OFDM is shown below :



For every message send the Additive White Gaussian Noise(AWGN) is different, this is to avoid network collision. Its clearly shown on (**Table 1: First iteration** and **Table 2: Second iteration**). The sender amplitude and the receiver amplitude is the same. This is a proof that the message transmitted and received is the same , in case its different that means there was an error during transmission. A sample code is shown below with the conditions of OFDM algorithm.

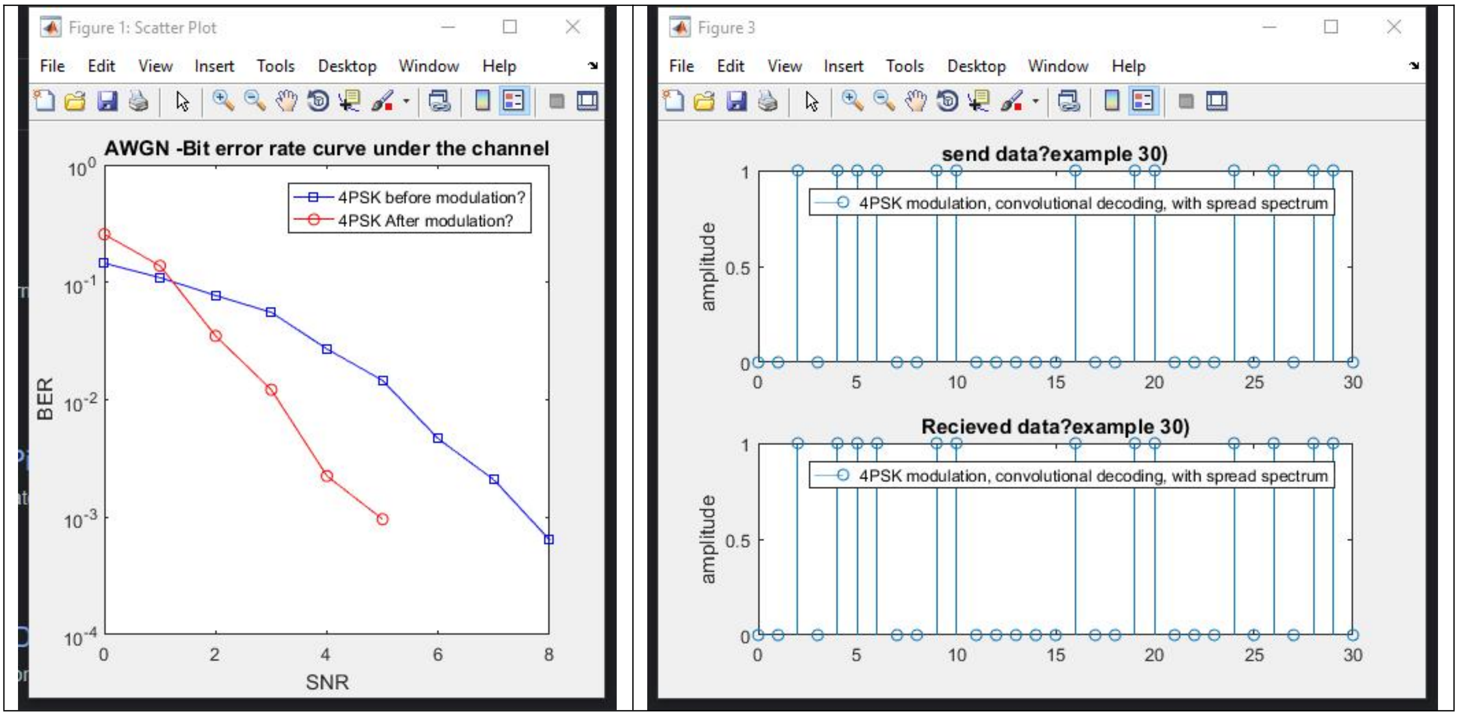


Table 1: First iteration

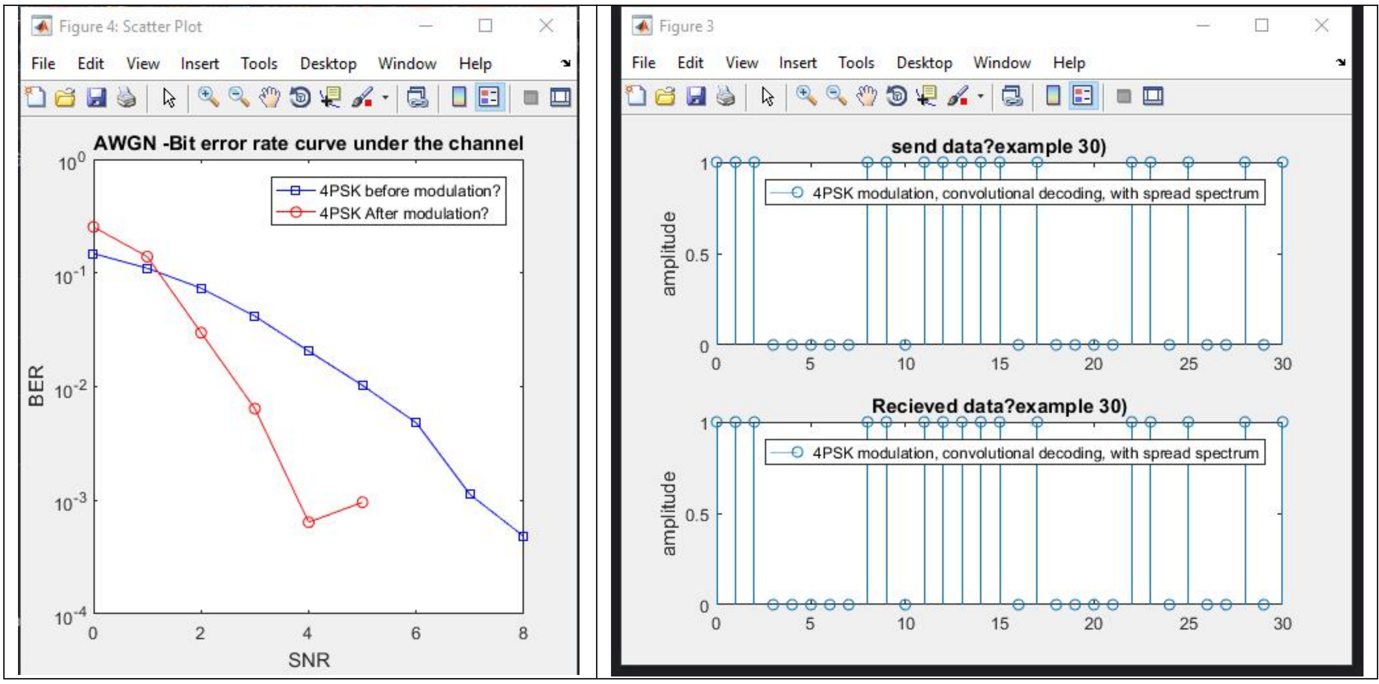


Table 2: Second iteration

```
%% Calculate the bit error rate
    [~,Ber2(jj)] = biterr(De_Bit(1:length(code_data)),code_data);%bit error rate before decoding
    [err, Ber(jj)] = biterr(rx_c_de(1:length(P_data)),P_data);%Bit error rate after decoding

end
 figure(2);
 semilogy(SNR,Ber2,'b-s');
 hold on;
 semilogy(SNR,Ber,'r-o');
 hold on;
 legend('4PSK before modulation) ','4PSK After modulation) ');
 hold on;
 xlabel('SNR');
 ylabel('BER');
 title('AWGN -Bit error rate curve under the channel');

 figure(3)
 subplot(2,1,1);
 x=0:1:30;
 stem(x,P_data(1:31));
 ylabel('amplitude');
 title('send data (example 30)');
 legend('4PSK modulation, convolutional decoding, with spread spectrum');

 subplot(2,1,2);
 x=0:1:30;
 stem(x,rx_c_de(1:31));
 ylabel('amplitude');
 title('Recieved data (example 30)');
 legend('4PSK modulation, convolutional decoding, with spread spectrum');
```

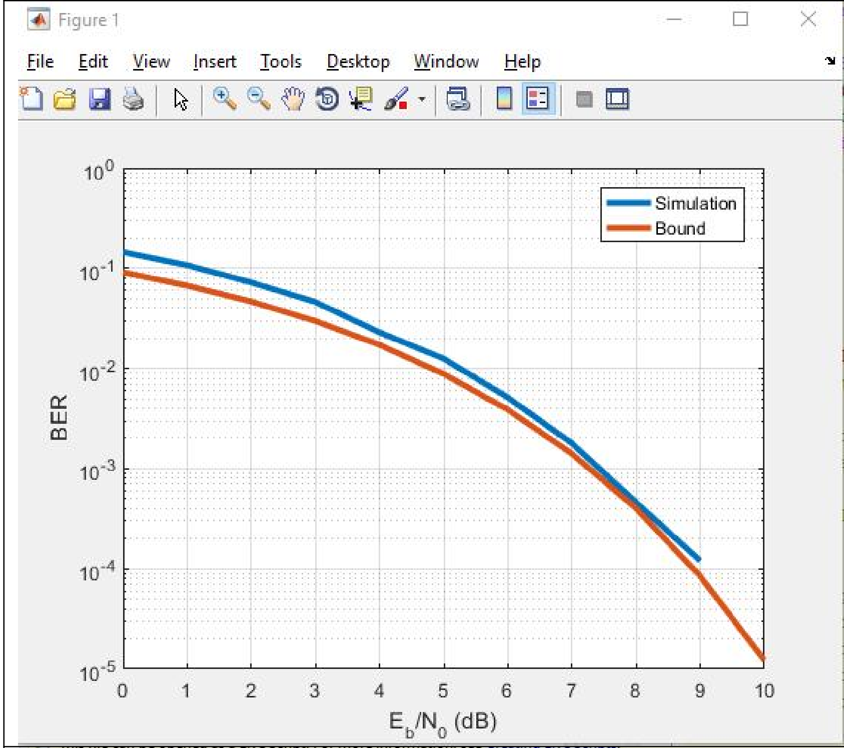
Table 3: sample peace of the algorithm

Task 2: Viterbi Algorithm Simulation

The Viterbi algorithm is a dynamic programming algorithm for obtaining the maximum a posteriori probability estimate of the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models.

The algorithm has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications, and 802.11 wireless LANs. It is now also commonly used in speech recognition, speech synthesis, diarization, computational linguistics, and bioinformatics. For example, in speech-to-text (speech recognition), the acoustic signal is treated as the observed sequence of events, and a string of text is considered to be the "hidden cause" of the acoustic signal. The Viterbi algorithm finds the most likely string of text given the acoustic signal.

The actual known is put alongside the simulated data as shown in (**Table 4: model and sample algorithm).** This simulation is for likelihood sequence detection Viterbi receiver.




```
clc
clear
close all
%% Pulse shape & Variable initialization
MAIN              = MainFunctions;
pulse             = 2;      % 1 -> lorentzian pulse
                            % 2 -> GMSK pulse BT = 0.3
                            % 3 -> LRC pulse
                            % 4 -> LREC pulse
pulse_length      = 3;      % 1  -> Full response
                            % >1 -> Partial response
os                = 2^2;    % Over sampling frequency
Ts                = 1/os;   % Sampling Time
M_ary             = 2^1;    % M_ary symbols used 2 -> Binary
modulation_index = 0.5;     % Modulation index
width             = 0.7;    % This variable is used for Lorentzian Pulse only. (Not be used for pulse > 1)
dmin              = 1.78;   % GMSK BT=0.3
% 'decision_delay' Decide after how many observation symbols
% the Viterbi receiver can make a decision on the detected bit.
decision_delay    = 50;
%------------- Modulated data ---------------------
snr            = 0:10;
BER_all        = zeros(1,length(snr));
%% frequency pulse
[g_t,q_t]      = MAIN.CREATECPMPULSE(pulse,pulse_length,width,os,0); % Function return the CPM pulse and phase.
                                                            % g_t = g(t) is the CPM pulse shape.
                                                            % q_t -> is the phase, integral of g_t.
m  = 1:M_ary;
if(pulse~=1)
    A_m    = 2*m-1-M_ary;       % Different modulation level
else
```

Table 4: model and sample algorithm