北京科技大学
University of Science and Technology Beijing

Wireless Communications

Homework 3

OFDM matlab simulation

Viterbi Algorithm

Name: Oliger Verónica Mendoza Betancourt 奥莉歌

Student ID: D202161028

School of Computer and Communication Engineering

Information and Communication Engineering

Lecturer: Du Bing.

University of Science and Technology Beijing

30 Xueyuan Road，Haidian District

Beijing 100083，P.R.CHINA

Date of Submission: 2022/05/30

1- OFDM Algoritm Simulation

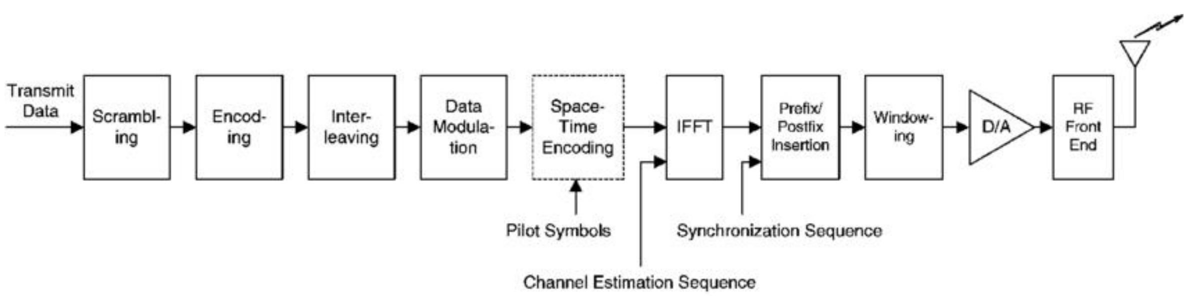
OFDM is a special multicarrier transmission scheme that can be considered as a modulation technique or a multiplexing technique.

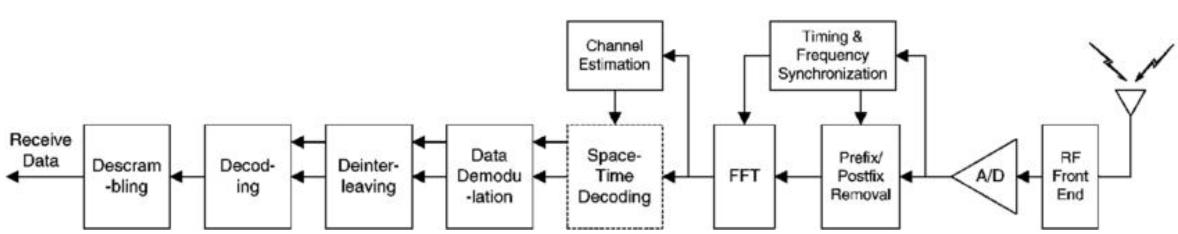
In a nutshell: OFDM is a multi-carrier transmission method. It divides the frequency band into multiple sub-channels to transmit data in parallel, splits the high-speed data stream into multiple parallel low-speed data streams, and then modulates them onto the subcarriers of each transmission channel. Because it converts the non-flat fading wireless channel into multiple flat fading orthogonal sub-channels, it can eliminate the interference between channel waveforms and achieve the purpose of combating multipath fading.

Orthogonal Frequency Division Multiplexing (OFDM) is an enhancement to Multi-Carrier Modulation (MCM). It is characterized by: The subcarriers are orthogonal to each other, so the spectrum after spread spectrum modulation can overlap each other, which not only reduces the mutual interference between the subcarriers, but also greatly improves the utilization of the spectrum.

A big reason to choose OFDM is that the system can resist frequency selective fading and narrowband interference well. In a single-carrier system, a fade or interference will cause the entire link to fail, but in a multi-carrier system, only a small portion of the subchannels will be affected by deep fading at any given time.

The typical OFDM system transceiver block diagram (parts can be added/removed according to actual needs) is as follows:





Among them, the upper half corresponds to the transmitter link and the lower half to the receiver link.

The transmitting end converts the transmitted digital signal into subcarrier amplitude and phase mapping, and performs Discrete Fourier Transformation (IDFT) to change the spectral expression of the data to the time domain. IFFT and IDFT transform have the same function, but have higher calculation efficiency, so it is suitable for all application systems. The receiving end performs the opposite operation to the sending end and uses the FFT transform to decompose, and the amplitude and phase of the subcarrier are finally converted back to a digital signal.

It is understood here that the signal transmitted in the frequency domain is due to the fact that the IFFT is from the frequency domain to the time domain. In fact, the IFFT here acts as a function of realizing the orthogonality of the subcarrier, and the DFT formula can be specifically derived.

OFDM modulation and demodulation

An OFDM symbol includes a signal composed of multiple modulated subcarriers, each of which can be modulated by psk (phase shift keying) and qam (quadrature amplitude modulation).

The OFDM transmitter maps the information bit stream into a psk or qam symbol sequence and then converts the serial symbol sequence to a parallel symbol stream. Every N symbols that undergo serial-to-parallel conversion are modulated by different subcarriers.

An OFDM symbol is a signal composed of N parallel symbols. If the transmission time (period) of a single serial symbol is Ts, the duration (period) of one OFDM symbol is Tsym = N * Ts.

The frequency of the modulation signal in the frequency domain X[k] is: fk = k / Tsym, the number of subcarriers is N, so k = 0, 1, 2 ... N-1. (Deduced by the DFT principle)

Simulation

It is appreciated that with OFDM, the similarity of results is due to the optimization of the OFDM output with a output sequence in average power very similar for different types of carriers.

It is also noted that OFDM systems usually have a high PAPR, that is, the average power of the signal is much smaller than the signal peaks that eventually appear, which is a problem when transmitting signals. One of the techniques to reduce this unwanted effect is the Clipping where the PAPR is reduced by cutting the signal through a limiter, although It must be taken into account that the clipping of the peaks of the signal produces non-linear distortion inside and outside the transmission band.

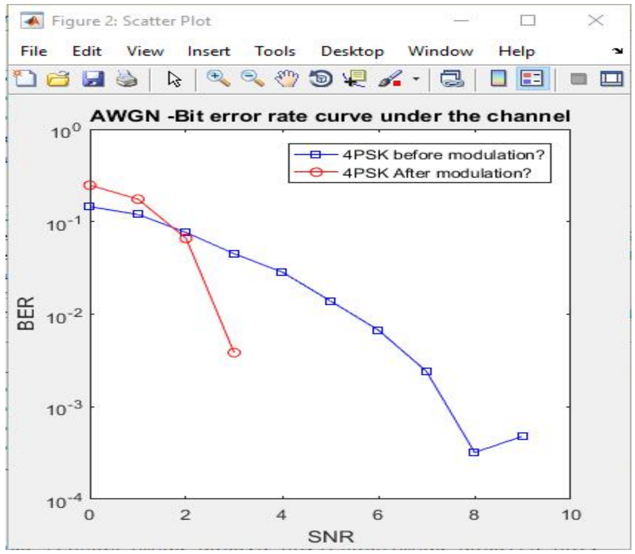
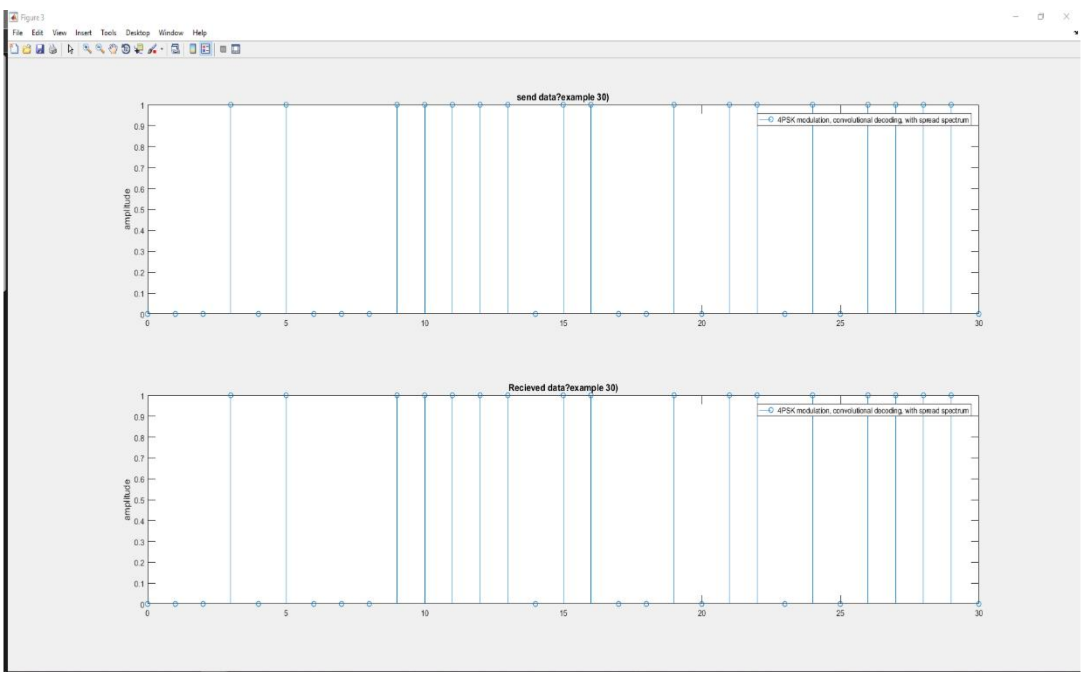


Figure 2: Scatter Plot
File Edit View Insert Tools Desktop Window Help
AWGN -Bit error rate curve under the channel
4PSK before modulation?
4PSK After modulation?
BER
SNR

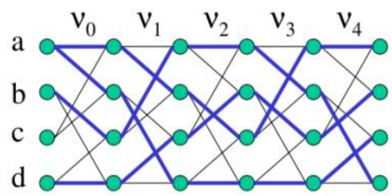

send data?example 30)
4PSK modulation, convolutional decoding, with spread spectrum
amplitude
Recieved data?example 30)
4PSK modulation, convolutional decoding, with spread spectrum
amplitude

2- Viterbi Algorithm

Optimal Maxima Decoding Algorithm Verisimilitude. s= MAX s(ln p(y/s)

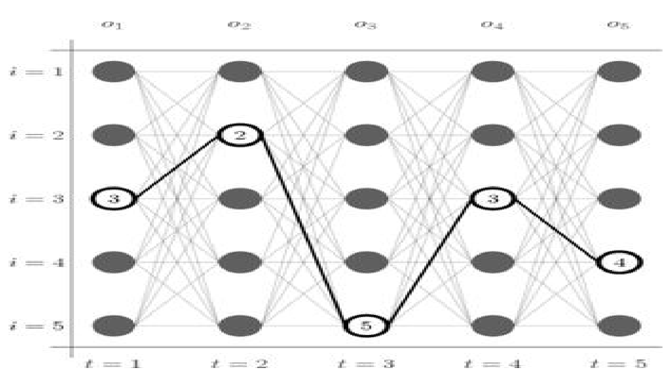
Used for convolutional or trellis codes.

Determine the most probable path according to the weighted jumps.

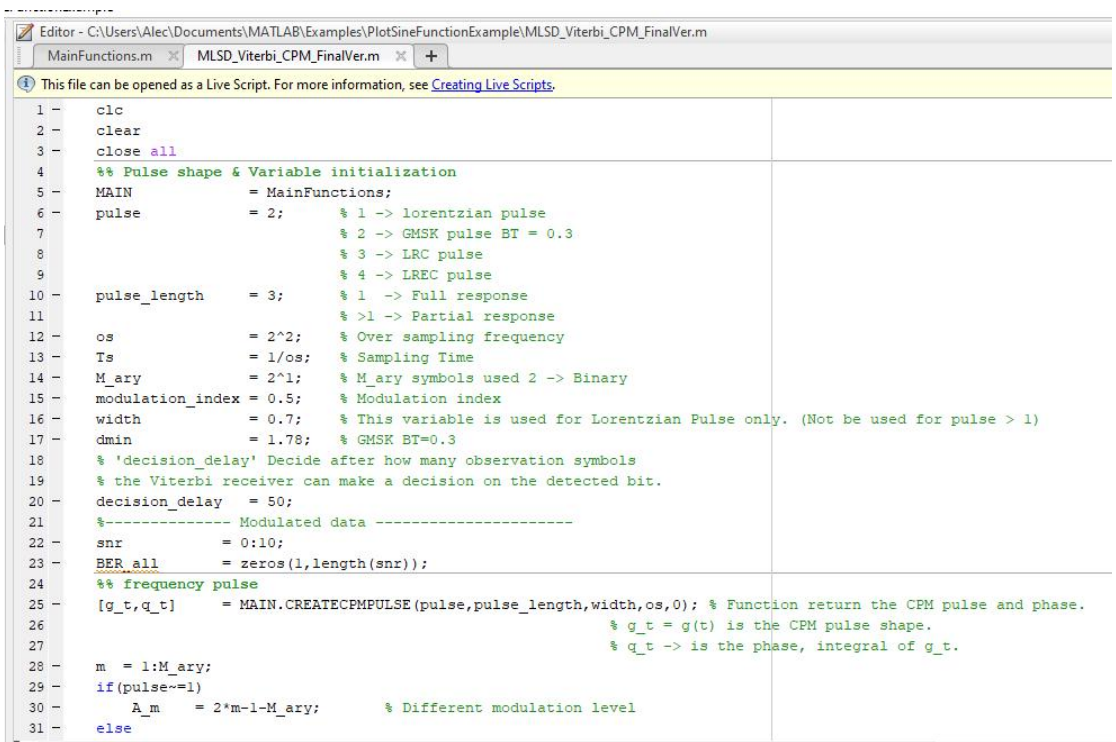


The Viterbi algorithm is an algorithm devised by Andrew Viterbi and generally used to find the best sequence of states (called the Viterbi Path) in a sequence of observed events in a Markovian process. The algorithm is used to decode convolutional codes in case high signal decoding gains are required.

Based on a process that is Markov, that is, a process in which "the probability of being in a state at a given time depends only on the state at the previous time", the algorithm chooses the path that is closest to the sequence of symbols received within the frame, or field of all possibilities. The criterion for choosing among the possibilities can be once the criterion is chosen, the same law of decoding is applicable. At each step, the algorithm eliminates the least likely paths until only one survivor remains. The algorithm is all the more efficient when the number of steps is high. Obviously, the greater the number of steps and the greater the decoding time and the greater the resource expenditure. You can estimate the complexity of computing the decoder by computing that for a code with states and observation steps T that has 2 (Me · (t − 1)) {\displaystyle \mathrm {2^{(I \ cdot (t - 1))}} } possible paths. At each step there are 2 Me {\displaystyle \mathrm {2^{i}} } paths reaching all states. Of all the paths, only one will be the one that is at the minimum distance to that step. The search for the optimal solution with an exhaustive technique quickly becomes inapplicable with the growth of i and t above fairly low values. Techniques such as the Viterbi algorithm are applicable, which reduce the complexity of the problem by applying dynamic programming techniques.



The Viterbi algorithm is very general and therefore it is possible to adapt it to the description of phenomena of different types. Some typical applications of this system are digital transmission problems in the field of telecommunications, in particular space transmissions where the channel noise conditions are such that it is difficult to receive data. It is also applied for Optical Text Recognition (OCR).



Editor - C:\Users\Alec\Documents\MATLAB\Examples\PlotSineFunctionExample\MLSD_Viterbi_CPM_FinalVer.m

MainFunctions.m | MLSD_Viterbi_CPM_FinalVer.m

This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
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
```