

# OFDM 完整仿真过程及解释(MATLAB)

## 子木

原载：知乎“子木”号，经作者授权转载。

### 目录：

- 一、说明
- 二、ofdm 总体概述
- 三、基本原理
- 四、过程中涉及的技术
- 五、OFDM 基本参数的选择
- 六、OFDM 的 MATLAB 仿真程序

## 一、说明

能找到这篇文章，说明对 ofdm 已经有一点了解，所以其原理就不再赘述，这篇代码的目的只是希望能对 ofdm 整个过程有一个理解；

- 1.看书上 ofdm 介绍挺简单的，自己来仿真才发现很多知识点都不知道；
- 2.花了很长时间才理清整个 ofdm 过程，网上的程序都是一段一段的，不能直接理解整个过程。所以想着自己来做一个完整过程的仿真，加深理解；
- 3.基带信号能完成整个过程，但是想加进频带传输这一部分，就完整了；
- 4.信道部分想用瑞利信道的，程序写出来了，但是误差和信道估计这一块还不是很明白，所以就先用的高斯信道；
- 5.不足之处欢迎指正。。。。

## 二、概述

OFDM 是一种特殊的多载波传输方案，它可以被看作是一种调制技术，也可以被当作一种复用技术。

简单来说：OFDM 是一种多载波的传输方法，它将频带划分为多个子信道并行传输数据，将高速数据流分成多个并行的低速数据流，然后调制到每个信道的子载波上进行传输。由于它将非平坦衰落无线信道转化成多个正交平坦衰落的子信道，从而可消除信道波形间的干扰，达到对抗多径衰落的目的。

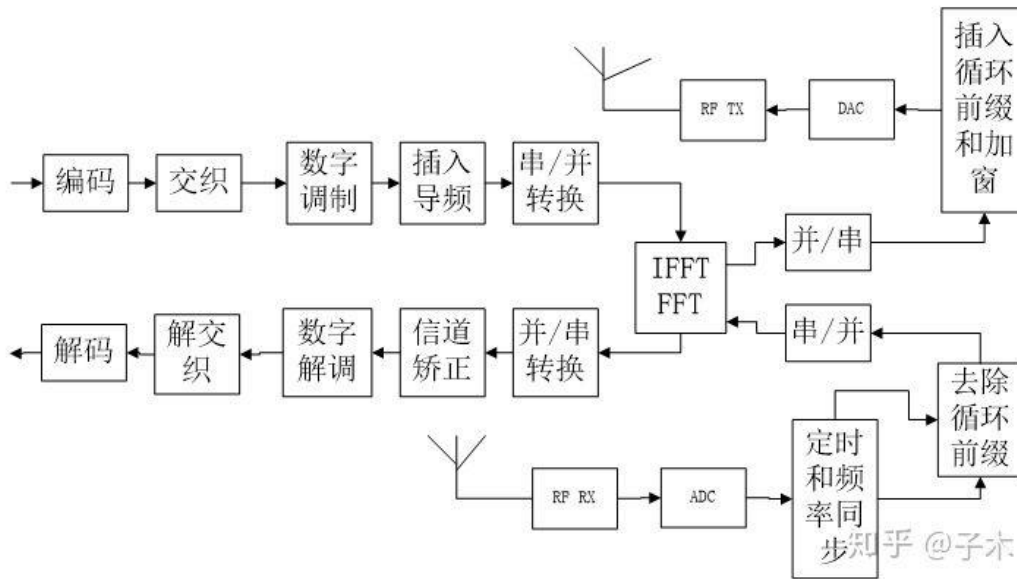
正交频分复用（OFDM）是对多载波调制（MCM）的一种改进，在。它的特点是：**各子载波相互正交，所以扩频调制后的频谱可以相互重叠，不但减少了子载波间的相互干扰，还大大提高了频谱利用率。**

选择 OFDM 的一个很大的原因是该系统能够很好的对抗频率选择性衰落和窄带干扰。在单载波系统中，一次衰落或者干扰会导致整个链路失效，但是在多载波系统中，某一时刻只会有少部分的子信道受到深衰落的影响。

## 三、基本原理

### 3.1 OFDM 系统收发机模型

OFDM 系统收发信机的框图如下(根据实际需要可添/删部分模块)：



OFDM 收发机框图

其中，上半部分对应于发射机链路，下半部分对应于接收机链路。

发送端将被传输的**数字信号**转换成子载波幅度和相位的映射，并进行离散傅里叶变换（IDFT），将数据的频谱表达式变到时域上。IFFT 和 IDFT 变换的作用相同，只是有更高的计算效率，所以适用于所有的应用系统。接收端进行与发送端相反的操作，用 FFT 变换分解，子载波的幅度和相位最终转换回数字信号。

这里理解为传输的频域信号是因为 IFFT 是从频域到时域，实际上这里 IFFT 充当的是一个实现子载波正交的作用，具体可以推导其 DFT 公式。知乎里公式编辑太麻烦了。

### 3.2 OFDM 调制与解调

一个 OFDM 符号之内包括多个经过调制的子载波的合成信号，其中每个子载波都可以收到 psk（相移键控）和 qam（正交幅度调制）的调制。

OFDM 发射机将信息比特流映射成一个 psk 或 qam 符号序列，之后将串行的符号序列转换为并行符号流。每  $N$  个经过串并转换的符号被不同的子载波调制。

OFDM 符号是  $N$  个并行符号的复合信号，若单个串行符号的传输时间（周期）是  $T_s$ ，则一个 OFDM 符号的持续时间（周期） $T_{\text{sym}}=N \cdot T_s$ 。

频域调制信号  $X[k]$  的频率为： $f_k=k/T_{\text{sym}}$ ，子载波数量为  $N$ ，则  $k=0,1,2,\dots,N-1$ 。（由 DFT 原理推导）

## 四、过程中涉及的技术

为什么要用？怎么用？

### 4.1 保护间隔

多径信道会对 OFDM 符号造成 ISI 影响，破坏了子载波间的正交性。故需要采取一些方法来消除多径信道带来的符号间干扰（ISI）影响，即插入保护间隔。

保护间隔有两种插入方法：一种是补零（zp），即在保护间隔中填充 0；另一种是插入循环前缀（cp）或循环后缀（cs）实现 OFDM 的循环扩展（为了某种连续性）。

zp 是在保护间隔内不插入任何信号，但是在这种情况下，由于多径传播的影响，会产生载波间干扰（ICI），即不同的子载波间会产生干扰。

一般采用 cp。cp 是将 OFDM 后部的采样复制到前面，长度为  $T_{cp}$ ，故每个符号的长度为  $T_{sym} = T_{sub} + T_{cp}$ ， $T_{sub}$  为数据部分子载波数。 $T_{cp}$  大于或等于多径时延，符号间的 ISI 影响将被限制在保护间隔中，因此不会影响下一个 OFDM 的 FFT 变换。

## 4.2 交织

交织的作用是将突发错误转换为随机错误，有利于前向纠错码的译码，提高了整个通信系统的可靠性。交织由两个变换过程组成。第一次变换保证了相邻的编码比特被映射到不相邻的子载波上。第二次变换保证了相邻的编码比特被分别映射到星座图的重要和非重要比特上，避免出现长时间的低比特位映射。

交织块的长度  $N_{cbps}$  对 qpsk、16qam、64qam 分别为 2、4、6  $s = N_{cbps}/2$ ， $d=16$ 。

## 4.3 信道编码

由于移动通信存在干扰和衰落，在信号传输过程中将出现差错，故对数字信号必须采用纠、检错技术，即纠、检错编码技术，以增强数据在信道中传输时抵御各种干扰的能力，提高系统的可靠性。

这里的信道编码一般采用卷积编码，Viterbi 译码。

卷积编码是现代数字通信系统中常见的一种前向纠错码,区别于常规的线性分组码,卷积编码的码字输出不仅与当前时刻的信息符号输入有关,还与之前输入的信息符号有关。

## 4.4 扩频

“扩频通信技术是一种信息传输方式,其信号所占有的频带宽度远大于所传信息必需的最小带宽;频带的扩展是通过一个独立的码序列来完成,用编码及调制的方法来实现的,与所传信息数据无关;在接收端则用同样的码进行相关同步接收、解扩及恢复所传信息数据”

**根据香农定理,带宽和信噪比可用互换,扩频扩展了带宽,则对信噪比的要求可降低。**

## 4.5 导频

导频不携带信息,导频是双方已知的数据,是用来做信道估计的。

在接收机中,虽然利用接收到的段训练序列、长训练序列可以进行信道均衡、频率偏差校正,但符号还会存在一定的剩余偏差,且偏差会随着时间的累积而累积,会造成所有子载波产生一定的相位偏移。因此,还需要不断地对参考相位进行跟踪。要能实现这个功能,需要在 52 个非 0 子载波中插入导频符号。

## 4.6 RF ( 射频 ) 调制

OFDM 调制器的输出产生了一个基带信号，将此基带信号与所需传输的频率进行混频操作，利用模拟技术或数字上变频可完成。由于数字调制技术提高了处理 I、Q 信道之间的匹配性和数字 IQ 调制器相位的准确性，将会更加精确。

## 五、OFDM 基本参数的选择

### 5.1 基本参数的选择原则

各种 OFDM 参数的选择就是需要在多项要求冲突中进行折衷考虑。通常来说，首先要确认 3 个参数：带宽、比特率、及保护间隔。

#### 5.1.1 保护间隔

按照惯例，保护间隔的时间长度应该为应用移动环境信道的时延扩展均方根值的 2~4 倍。

#### 5.1.2 OFDM 符号周期长度

确定保护间隔之后，则 OFDM 符号周期长度就确定了。为了最大限度的减少由于插入保护比特所带来的信噪比的损失，OFDM 符号周期长度远远大于保护间隔长度。但是符号周期又不能任意大，否则就需要更多的子载波，带宽不变，子载波间隔就变小，系统的实现复杂度就提高了，而且还加大了系统的峰值平均功率比，同时系统对频率偏差更加敏感。因此，**一般选择符号周期长度是保护间隔的 5 倍**，这样，由于插入保护比特所造成的信噪比损耗只有 1dB 左右。

### 5.1.3 子载波的数量

确定保护间隔和符号周期长度之后，子载波的数量可由-3dB 带宽除以子载波间隔（即去掉保护间隔之后的符号周期的倒数）得到。或者可由所要求比特速率除以每个子信道的比特速率来确定子载波的数量。每个信道中所传输的比特速率可由调制类型、编码速率、和符号速率来确定。

## 5.2 有用符号持续时间 T

T 对子载波之间间隔、译码的等待周期都有影响，为了保持数据的吞吐量，子载波数目和 FFT 的长度要有相对较大的数量，这就导致符号持续时间变长。总之，符号周期长度的选择以保证信道的稳定为前提。

## 5.3 子载波数

$$N=1/T$$

其数值与 FFT 处理过的复数点数相对应，需适应数据速率和保护间隔的要求。

## 5.4 调制模式

OFDM 系统的调制模式基于功率和频谱利用率来选择，可采用 qam、psk。



为了使所有的点有相同的平均功率，二进制序列映射后的复数要归一化。  
( BPSK\QPSK\16QAM\64QAM 分别对应乘以 1、 $1/\sqrt{2}$ 、 $1/\sqrt{10}$ 、 $1/\sqrt{42}$  )，解调的时候再变回去。

## 5.5 以具体实例说明

**要求: ( 1 ) 比特率为 25Mbit/s ( 2 ) 可容忍的时延扩展为 200ns ( 3 ) 带宽小于 18MHz。**

- 1) 由 200ns 时延扩展得保护间隔为 800ns ；
- 2) 由保护间隔 800ns 得符号周期长度  $6*800ns=4.8us$  ；
- 3) 子载波的间隔选取  $4.8-0.8=4us$  的倒数，即 250KHz ；
- 4) 由所要求的比特速率与 OFDM 符号速率的比值，每个符号需要传送的比特： $( 25Mbit/s ) / ( 1/4.8us ) =120bit$ 。
- 5) 为了完成上面 120bit/符号，有两种选择：利用 16QAM 和码率为 1/2 的编码方法，这样每个子载波携带 2bit 的有用信息，因此需要 60 个子载波；另一种是利用 QPSK 和码率为 3/4 的编码方法，每个子载波携带 1.5bit 信息。因此需要 80 个子载波，然而 80 个子载波意味着带宽： $80*250KHz=20MHz$ ，大于所给带宽要求，故取第一种，即 60 个子载波。可利用 64 点 IFFT 来实现，剩余 4 个子载波补 0。

## 六、OFDM 的 MATLAB 仿真主程序

```
clc;
clear;

%-----%
%q1:ifft 点数难道不是应该等于子载波数吗？子载波数与 ifft 点数的关系？
%a:ifft 点数等于子载波数
%q2: 对矩阵进行 fft?
%a:y 可以是一向量或矩阵，若 y 为向量，则 Y 是 y 的 FFT，并且与 y 具有相同的长度。若 y 为一矩阵，则 Y 是对矩阵的每一列向量进行 FFT。
%q3: 怎么对 ofdm 信号上变频
%-----%

%% 参数设置

N_sc=52;          %系统子载波数（不包括直流载波）、number of subcarrier
N_fft=64;         % FFT 长度
N_cp=16;          % 循环前缀长度、Cyclic prefix
N_symbo=N_fft+N_cp; % 1 个完整 OFDM 符号长度
N_c=53;           % 包含直流载波的总的子载波数、number of carriers
M=4;              %4PSK 调制
SNR=0:1:25;       %仿真信噪比
N_frm=10;         % 每种信噪比下的仿真帧数、frame
Nd=6;             % 每帧包含的 OFDM 符号数
P_f_inter=6;     %导频间隔
data_station=[]; %导频位置
L=7;              %卷积码约束长度
tblen=6*L;       %Viterbi 译码器回溯深度
stage = 3;       % m 序列的阶数
ptap1 = [1 3];   % m 序列的寄存器连接方式
regi1 = [1 1 1]; % m 序列的寄存器初始值

%% 基带数据数据产生
P_data=randi([0 1],1,N_sc*Nd*N_frm);

%% 信道编码（卷积码、或交织器）
%卷积码：前向纠错非线性码
%交织：使突发错误最大限度的分散化
trellis = poly2trellis(7,[133 171]); % (2,1,7) 卷积编码
code_data=convenc(P_data,trellis);
```

```

%% qpsk 调制
data_temp1= reshape(code_data,log2(M),[])' ;           %以每组 2
比特进行分组, M=4
data_temp2= bi2de(data_temp1);                       %二进制
转化为十进制
modu_data=pskmod(data_temp2,M,pi/M);                 % 4PSK 调制
% figure(1);
scatterplot(modu_data),grid;                          %星座图(也可以取实部
用 plot 函数)

%% 扩频
%-----%
%扩频通信信号所占有的频带宽度远大于所传信息必需的最小带宽
%根据香农定理, 扩频通信就是用宽带传输技术来换取信噪比上的好处, 这就是
扩频通信的基本思想和理论依据。
%扩频就是将一系列正交的码字与基带调制信号内积
%扩频后数字频率变成了原来的 m 倍。码片数量 = 2 (符号数) * m (扩频系数)
%-----%

code = mseq(stage,ptap1,regi1,N_sc);    % 扩频码的生成
code = code * 2 - 1;                    %将 1、0 变换为 1、-1
modu_data=reshape(modu_data,N_sc,length(modu_data)/N_sc);
spread_data = spread(modu_data,code);    % 扩频
spread_data=reshape(spread_data,[],1);

%% 插入导频
P_f=3+3*1i;                             %Pilot frequency
P_f_station=[1:P_f_inter:N_fft];%导频位置 (导频位置很重要, why?)
pilot_num=length(P_f_station);%导频数量

for img=1:N_fft                            %数据位置
    if mod(img,P_f_inter)~=1                %mod(a,b)就是求的是 a 除以 b
的余数
        data_station=[data_station,img];
    end
end
data_row=length(data_station);
data_col=ceil(length(spread_data)/data_row);

pilot_seq=ones(pilot_num,data_col)*P_f;%将导频放入矩阵
data=zeros(N_fft,data_col);%预设整个矩阵
data(P_f_station(1:end),:)=pilot_seq;%对 pilot_seq 按行取

```

```
if data_row*data_col>length(spread_data)

data2=[spread_data;zeros(data_row*data_col-length(spread_data)
,1)];%将数据矩阵补齐，补 0 是虚载频~
end;

%% 串并转换
data_seq=reshape(data2,data_row,data_col);
data(data_station(1:end),:)=data_seq;%将导频与数据合并

%% IFFT
ifft_data=ifft(data);

%% 插入保护间隔、循环前缀
Tx_cd=[ifft_data(N_fft-N_cp+1:end,:);ifft_data];%把 ifft 的末尾
N_cp 个数补充到最前面

%% 并串转换
Tx_data=reshape(Tx_cd,[],1);%由于传输需要

%% 信道（通过多经瑞利信道、或信号经过 AWGN 信道）
Ber=zeros(1,length(SNR));
Ber2=zeros(1,length(SNR));
for jj=1:length(SNR)
    rx_channel=awgn(Tx_data,SNR(jj),'measured');%添加高斯白噪声

%% 串并转换
    Rx_data1=reshape(rx_channel,N_fft+N_cp,[]);

%% 去掉保护间隔、循环前缀
    Rx_data2=Rx_data1(N_cp+1:end,:);

%% FFT
    fft_data=fft(Rx_data2);

%% 信道估计与插值（均衡）
    data3=fft_data(1:N_fft,:);
    Rx_pilot=data3(P_f_station(1:end),:); %接收到的导频
    h=Rx_pilot./pilot_seq;

H=interp1( P_f_station(1:end)',h,data_station(1:end)', 'linear'
,'extrap');%分段线性插值：插值点处函数值由连接其最邻近的两侧点的线性
函数预测。对超出已知点集的插值点用指定插值方法计算函数值
```

```
%% 信道校正
    data_afterreq=data3(data_station(1:end),:)./H;
%% 并串转换
    data_afterreq=reshape(data_afterreq,[],1);
    data_afterreq=data_afterreq(1:length(spread_data));

data_afterreq=reshape(data_afterreq,N_sc,length(data_afterreq)/N_sc);

%% 解扩
    demspread_data = despread(data_afterreq,code);           % 数据解扩
扩

%% QPSK 解调
    demodulation_data=pskdemod(demspread_data,M,pi/M);
    De_data1 = reshape(demodulation_data,[],1);
    De_data2 = de2bi(De_data1);
    De_Bit = reshape(De_data2',1,[]);

%% （解交织）
%% 信道译码（维特比译码）
    trellis = poly2trellis(7,[133 171]);
    rx_c_de = vitdec(De_Bit,trellis,tblen,'trunc','hard');   %硬
判决

%% 计算误码率
    [err,Ber2(jj)] =
biterr(De_Bit(1:length(code_data)),code_data);%译码前的误码率
    [err, Ber(jj)] = biterr(rx_c_de(1:length(P_data)),P_data);%
译码后的误码率

end
figure(2);
semilogy(SNR,Ber2,'b-s');
hold on;
semilogy(SNR,Ber,'r-o');
hold on;
legend('4PSK 调制、卷积码译码前（有扩频）','4PSK 调制、卷积码译码后
（有扩频）');
hold on;
xlabel('SNR');
ylabel('BER');
title('AWGN 信道下误比特率曲线');
```

```
figure(3)
subplot(2,1,1);
x=0:1:30;
stem(x,P_data(1:31));
ylabel('amplitude');
title('发送数据 (以前 30 个数据为例)');
legend('4PSK 调制、卷积译码、有扩频');

subplot(2,1,2);
x=0:1:30;
stem(x,rx_c_de(1:31));
ylabel('amplitude');
title('接收数据(以前 30 个数据为例)');
legend('4PSK 调制、卷积译码、有扩频');
```

## 七 原文

### 整个过程

```
1  % 基带数据产生
2  %% 信道编码 (卷积码、或交织器)
3  %% QPSK调制
4  %% 扩频
5  %% 插入导频
6  %% 串并转换
7  %% IFFT
8  %% 插入保护间隔、循环前缀
9  %% 并串转换
10 %% DA
11 %% 上变频
12 %% 信道 (通过多径瑞利信道、或信号经过AWGN信道)
13 %% 下变频
14 %% AD
15 %% 串并转换
16 %% 去掉保护间隔、循环前缀
17 %% FFT
18 %% 信道估计与插值 (均衡)
19 %% 信道校正
20 %% 并串转换
21 %% 解扩
22 %% QPSK解调
23 %% (解交织)
```

本来对每一步都有讲解注释的，但是程序编辑多了感觉不美观，就删掉了。比如扩频，其原理、作用、如何实现~

### 三、代码及说明

1. 尽量把每一句程序都注释，能达到初学者拿到程序就能懂的程度；

2. 下面这段程序是上变频之前的，包含了画图，对 ofdm 信号有一个直观的感受（与上面图片中的流程可能冲突，这里仅仅是为了画图解释，所以这也是最开始学容易绕晕的地方）

```
clear;
%% 参数设置
sub_carriers=2048;%子载波数
T = 1 / sub_carriers;
time = [0:T:1-T];% Nifft 份，每份相隔 T
Lp=4984;
P_Tx=(rand(1,Lp)>0.5);%(bits)%产生 1 个长为 Lp 的数据包:
conv_out=convolutional_en(P_Tx);% (卷积编码):
interleave_table = interleav_matrix(ones(1,2*(Lp+8)));
interleav_out = interleaving(conv_out ,interleave_table);% (交织器)
x=qpsk(interleav_out);% (4QAM 调制)
L=length(x);%信号长度
```

```
s=48;
symbol_used_len=L/s;%把输入分为 S 个符号，每个符号长为 symbol_used_len
%循环前缀的长度
cp=256;
%每一个 OFDM 符号的抽样值应补 '0' 个数 zeros_pad
zeros_pad=sub_carriers-symbol_used_len;
%每一个 OFDM 符号一侧应该补 '0' 个数 zeros_pad_side
zeros_pad_side=zeros_pad/2;
%对输入信号进行分割，分割为 s 个符号，再对每个符号进行 FFT 运算，实现 OFDM 解调，
并保证能量不变
time_domain_x_link=[];
for I=0:(s-1)
    %对输入进行分割
    x_temp=x(I*symbol_used_len+1:I*symbol_used_len+symbol_used_len);
    %对每个分割的部分进行补零操作，使其长为 sub_carriers
    x_temp_pad=[zeros(1,zeros_pad_side),x_temp,zeros(1,zeros_pad_side)];
    %对每个符号进行 IFFT 运算
    time_domain_x_temp=ifft(x_temp_pad)*sqrt(sub_carriers);
    %对每个符号添加循环前缀
    time_domain_x_cp_temp=[time_domain_x_temp(sub_carriers-cp+1:sub_carriers),time_domain_x_temp];
    %将符号连接成为串行数据流
    time_domain_x_link=[time_domain_x_link,time_domain_x_cp_temp];
end
sum_xI = real(time_domain_x_link);
sum_xQ = imag(time_domain_x_link);
figure;
num=1000;%画出前 num 个点
xaxis = zeros(length(time(1:num)));
plot(time(1:num), sum_xI(1:num), 'b:', time(1:num), sum_xQ(1:num), 'g:',
time(1:num), abs(sum_xI(1:num)+j*sum_xQ(1:num)), 'k-', time(1:num), xaxis, 'r-');
ylabel('y'),xlabel('t'),
title(['前', num2str(num),'个点经 ifft 的 QAM 符号实部之和虚部之和以及实部与虚部的绝对值波形']),
```



```
legend('实部之和','虚部之和','绝对值');
```

### 3.与上面图片流程相符的代码

代码前面的问题也是我在这个过程中遇到的，困扰了好久，可以带着问题看看。欢迎讨论。

```
clc;
```

```
clear;
```

```
%
```

```
_____ %
```

```
%q1:fft 点数难道不是应该等于子载波数吗？子载波数与 ifft 点数的关系？
```

```
%q2：对矩阵进行 fft？
```

```
%q3：怎么对 ofdm 信号上变频
```

```
%q4：基带速率是多少？怎么实现？
```

```
%q5 传输频带是多少？怎么实现？
```

```
%q6 子载波间隔是多少？怎么实现？
```

```
%q7 符号周期是多少？怎么实现？
```

```
%
```

```
_____ %
```

```
%% 参数设置
```

```
N_sc=52; %系统子载波数（不包括直流载波） number of subcarrier
```

```
N_fft=64; % FFT 长度
```

```
N_cp=16; % 循环前缀长度、Cyclic prefix
```

```
N_symbo=N_fft+N_cp; % 1 个完整 OFDM 符号长度
```

```
N_c=53; % 包含直流载波的总的子载波数、number of carriers
```

```
M=4; %4PSK 调制
```

```
SNR=0:1:25; %仿真信噪比
```

```
N_frm=10; % 每种信噪比下的仿真帧数、frame
```

```
Nd=6; % 每帧包含的 OFDM 符号数
```

```
P_f_inter=6; %导频间隔
```

```
data_station=[]; %导频位置
```

```
L=7; %卷积码约束长度
```

```
tblen=6*L; %Viterbi 译码器回溯深度
```

```
stage = 3; % m 序列的阶数
```

```
ptap1 = [1 3]; % m 序列的寄存器连接方式
```

```
regi1 = [1 1 1]; % m 序列的寄存器初始值
```

```
%% 基带数据数据产生
```

```

P_data=randi([0 1],1,N_sc*Nd*N_frm);
%% 信道编码 (卷积码、或交织器)
%卷积码：前向纠错非线性码
%交织：使突发错误最大限度的分散化
trellis = poly2trellis(7,[133 171]);      %(2,1,7)卷积编码
code_data=convenc(P_data,trellis);
%% qpsk 调制
data_temp1= reshape(code_data,log2(M),[]);      %以每组 2 比特进行分组,
M=4
data_temp2= bi2de(data_temp1);                %二进制转化为十进
制
modu_data=pskmod(data_temp2,M,pi/M);          % 4PSK 调制
% figure(1);
scatterplot(modu_data),grid;                  %星座图(也可以取实部用 plot 函数)
%% 扩频
%-----
%-----%
%扩频通信信号所占有的频带宽度远大于所传信息必需的最小带宽
%根据香农定理,扩频通信就是用宽带传输技术来换取信噪比上的好处,这就是扩频通信的
基本思想和理论依据。
%扩频就是将一系列正交的码字与基带调制信号内积
%扩频后数字频率变成了原来的 m 倍。码片数量 = 2 (符号数) * m (扩频系数)
%-----
%-----%
code = mseq(stage,ptap1,regi1,N_sc);      % 扩频码的生成
code = code * 2 - 1;                      %将 1、0 变换为 1、-1
modu_data=reshape(modu_data,N_sc,length(modu_data)/N_sc);
spread_data = spread(modu_data,code);      % 扩频
spread_data=reshape(spread_data,[],1);

%% 插入导频
P_f=3+3*1i;                               %Pilot frequency
P_f_station=[1:P_f_inter:N_fft];%导频位置 (导频位置很重要, why?)
pilot_num=length(P_f_station);%导频数量

```

```
for img=1:N_fft %数据位置
    if mod(img,P_f_inter)~=1 %mod(a,b)就是求的是 a 除以 b 的余数
        data_station=[data_station,img];
    end
end
data_row=length(data_station);
data_col=ceil(length(spread_data)/data_row);
pilot_seq=ones(pilot_num,data_col)*P_f;%将导频放入矩阵
data=zeros(N_fft,data_col);%预设整个矩阵
data(P_f_station(1:end),:)=pilot_seq;%对 pilot_seq 按行取
if data_row*data_col>length(spread_data)
    data2=[spread_data;zeros(data_row*data_col-length(spread_data),1)];%将数据
    矩阵补齐,补0是虚载频~
end;
%% 串并转换
data_seq=reshape(data2,data_row,data_col);
data(data_station(1:end),:)=data_seq;%将导频与数据合并

%% IFFT
ifft_data=ifft(data);
%% 插入保护间隔、循环前缀
Tx_cd=[ifft_data(N_fft-N_cp+1:end,:);ifft_data];%把 ifft 的末尾 N_cp 个数补充到最前面
%% 并串转换
Tx_data=reshape(Tx_cd,[],1);%由于传输需要
%% 信道 ( 通过多经瑞利信道、或信号经过 AWGN 信道 )
Ber=zeros(1,length(SNR));
Ber2=zeros(1,length(SNR));
for jj=1:length(SNR)
    rx_channel=awgn(Tx_data,SNR(jj),'measured');%添加高斯白噪声
    %% 串并转换
    Rx_data1=reshape(rx_channel,N_fft+N_cp,[]);
    %% 去掉保护间隔、循环前缀
    Rx_data2=Rx_data1(N_cp+1:end,:);
%% FFT
```

```
fft_data=fft(Rx_data2);
%% 信道估计与插值 ( 均衡 )
data3=fft_data(1:N_fft,:);
Rx_pilot=data3(P_f_station(1:end),:); %接收到的导频
h=Rx_pilot./pilot_seq;
H=interp1( P_f_station(1:end)',h,data_station(1:end)', 'linear', 'extrap');%分段线性
插值 :插值点处函数值由连接其最邻近的两侧点的线性函数预测。对超出已知点集的插值点
用指定插值方法计算函数值
%% 信道校正
data_afterreq=data3(data_station(1:end),:)./H;
%% 并串转换
data_afterreq=reshape(data_afterreq,[],1);
data_afterreq=data_afterreq(1:length(spread_data));
data_afterreq=reshape(data_afterreq,N_sc,length(data_afterreq)/N_sc);
%% 解扩
demsread_data = despread(data_afterreq,code); % 数据解扩
%% QPSK 解调
demodulation_data=pskdemod(demsread_data,M,pi/M);
De_data1 = reshape(demodulation_data,[],1);
De_data2 = de2bi(De_data1);
De_Bit = reshape(De_data2',1,[]);
%% ( 解交织 )
%% 信道译码 ( 维特比译码 )
trellis = poly2trellis(7,[133 171]);
rx_c_de = vitdec(De_Bit,trellis,tblen,'trunc','hard'); %硬判决
%% 计算误码率
[err,Ber2(jj)] = biterr(De_Bit(1:length(code_data)),code_data);%译码前的误码率
[err, Ber(jj)] = biterr(rx_c_de(1:length(P_data)),P_data);%译码后的误码率
end
figure(2);
semilogy(SNR,Ber2,'b-s');
hold on;
semilogy(SNR,Ber,'r-o');
hold on;
```

```
legend('4PSK 调制、卷积码译码前 ( 有扩频 ) ','4PSK 调制、卷积码译码后 ( 有扩频 ) ');  
hold on;  
xlabel('SNR');  
ylabel('BER');  
title('AWGN 信道下误比特率曲线');  
figure(3)  
subplot(2,1,1);  
x=0:1:30;  
stem(x,P_data(1:31));  
ylabel('amplitude');  
title('发送数据 ( 以前 30 个数据为例)');  
legend('4PSK 调制、卷积译码、有扩频');  
subplot(2,1,2);  
x=0:1:30;  
stem(x,rx_c_de(1:31));  
ylabel('amplitude');  
title('接收数据(以前 30 个数据为例)');  
legend('4PSK 调制、卷积译码、有扩频');
```

4.上面就是整个过程（还缺上变频和下变频，也就是上面是基带传输，还缺频带传输），写好了就来补上。

5.本来打算解释原理的，但是想着网上资料很多，就不献丑了。

本文原载：知乎“子木”号：<https://zhuanglan.zhihu.com/p/57967971>，需要原程序者，可与作者联系。



临菲信息技术港



临菲信息技术港微信公众号