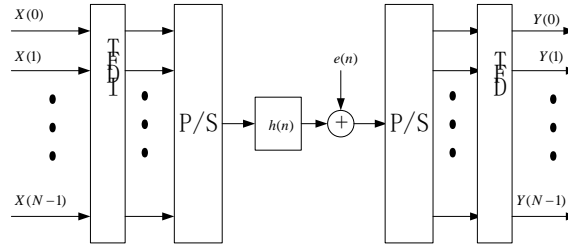


第一次课程设计——Project-OFDM 系统信道估计

黄琅飞 201921220305

一、原理分析：



$$x(n) = IDFT(X(k)) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}$$

如 X 的 Ts 为 2us, N=64 为子信道个数, 则 X 的总带宽为 500kHz, OFDM 将 500kHz 划分为 64 个子信道。

为防止 ISI, 在 x 中加入循环前缀 cp, 再经过并串转换后发射信号。

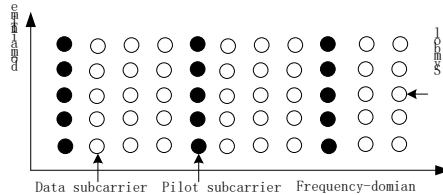
在 OFDM 系统信道估计中, 信道

$$h(n) = \sum_{m=0}^{L-1} \alpha_m e^{-j2\pi f_c \tau_m} \delta(n-m)$$

e 为测量噪声, 服从零均值复高斯分布。

$$\mathbf{Y} = \mathbf{X}\mathbf{H} + \mathbf{n} = \mathbf{X}\mathbf{F}\mathbf{h} + \mathbf{n}$$

为了解决信号分析的问题, 在真实数据前面加入导频信号 Xp, 用于接收端进行信道估计。并且为减小信道不稳定而带来的偏差, 每相隔几个数据就需要加入一个新的导频信号。



$$\mathbf{Y}_p = \mathbf{X}_p \mathbf{H}_p + \mathbf{n}_p = \mathbf{X}_p \mathbf{F}_p \mathbf{h} + \mathbf{n}_p$$

$$\hat{\mathbf{h}} = \left((\mathbf{X}_p \mathbf{F}_p)^H \mathbf{X}_p \mathbf{F}_p \right)^{-1} (\mathbf{X}_p \mathbf{F}_p)^H \mathbf{Y}_p$$

$$\hat{\mathbf{H}} = \mathbf{F} \hat{\mathbf{h}}$$

$$\hat{\mathbf{X}} = \hat{\mathbf{Y}} / \hat{\mathbf{H}}$$

二、仿真步骤：

1、N=64 个子信道, CP 长度为 4 个码片, 码片宽度为 Ts=2us, 即 Tcp=8us, 传输速率约为 1.9Mb/s, 载频为 1Ghz, 1 个帧包括 1 个导频+4 个数据, 假设 1 帧内信道不变。

2、产生 16QAM 信号 (randsrc, qammod), 并将基带信号调制到 64 个子信道去 (ifft)。

3、并串转换, 并加入 CP, 使之变为 68 个码片, 每帧共 340 个码片。

4、仿真信道冲击响应：在 $[0, T_{cp}]$ 范围内均匀产生3个时延 τ ；求取每个时延向对应的功率 P ，由功率再得到幅度 $\alpha_m(\alpha_m^2=P)$ ，

$$P = 4e^{-\tau/\tau_{rms}}, \tau_{rms} = 2\mu s$$

然后计算每个时延对应的相位，由相位和幅度得到3条路径的冲击响应 h

$$h(n) = \sum_{m=0}^{L-1} \alpha_m e^{-j2\pi f_c \tau_m} \delta(n-m)$$

5、得到基带回波信号 y ，并加入测量误差

$$y(n) = \sum_{m=0}^{L-1} \alpha_m e^{-j2\pi(f_c+B_s)\tau_m} x(n)$$

6、去cp后再串并转换，然后通过fft得到最后的接收信号 Y

7、在 X 和 Y 中取出导频信号，估计信道频域估计 H

8、在数据帧中用 Y/H 校正信道影响

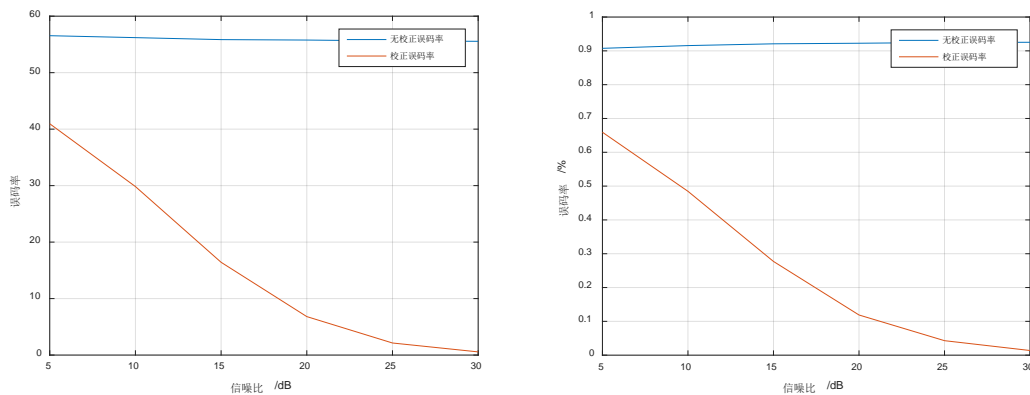
9、解码 qamdemod

10、统计误码率 symerr

三、仿真结果及分析：

在不进行信道估计并以此校正接收信号的前提下，直接将接收信号 y 作为原信号处理，具有极大的不准确，会产生严重的错误。

实际仿真结果如下：（误码率取平均）



图示与分析基本一致。信号一共64个码片，可以明显看出，不经过校正的信号与原信号存在很大的偏差，而且随着信噪比的增大，误码率也从90%逐步上升。

经过信道估计并校正的信号，误码率会随着信噪比的提升而大幅度降低，在 $SNR=30dB$ 时，误码率已降低到1%。所以实际应用中的接收信号都会先经过校正再进行使用，以减少错误。

四、Matlab 仿真程序：

```
clear all
close all
Ts=2e-6;
Rb=1.9e6;
fc=1e9;
M=16;
```

```

N=64;
x0=[];
Num=1000;
for n=1:Num
for i=0:4
    x_0=randsrc(1,N,[0 1]);
    eval(['x0',num2str(i),'=', 'x_0']);
    X=qammod(x_0,M);    %生成 16QAM 信号
    if i==0
        X0=X;
    end
    x=ifft(X,64,2);
for j=1:4                %加入 CP
    xc(1,j)=x(1,60+j);
    xc(:,5:68)=x;
end
    Xa(:,i*68+1:68*(i+1))=xc;
end
for i=1:3                %信道估计
    Tm(1,i)=4*Ts*rand;
    p(1,i)=4*exp(-Tm(1,i)/Ts);
    Am(1,i)=sqrt(p(1,i));
end
y_0=Am(1,1)*exp(-1i*2*pi*(fc+1/Ts)*Tm(1,1))*Xa+Am(1,2)*exp(-
1i*2*pi*(fc+1/Ts)*Tm(1,2))*Xa+Am(1,3)*exp(-1i*2*pi*(fc+1/Ts)*Tm(1,3))*Xa;
snr=[5:5:30];
for i=1:length(snr)
yn=awgn(y_0,snr(1,i));    %加入噪声那个测量误差
y=[];
Y=[];
for j=0:4                %去 CP+串并转换
    A=yn(:,j*68+5:(j+1)*68);
    eval(['y',num2str(j),'=', 'A']);
end
Y0=fft(y0); Y1=fft(y1); Y2=fft(y2); Y3=fft(y3); Y4=fft(y4);                %FFT 转换
y01=qamdemod(Y1,M);y02=qamdemod(Y2,M);y03=qamdemod(Y3,M);y04=qamdemod(Y4,M);    %直接解码
error0=(symerr(x01,y01)+symerr(x02,y02)+symerr(x03,y03)+symerr(x04,y04))/4;    %无校正误码率
Error0(n,i)=error0;
mean_Error0=mean(Error0,1);
H=Y0/X0;
X1=Y1/H; X2=Y2/H; X3=Y3/H; X4=Y4/H;                %校正
x1=qamdemod(X1,M);x2=qamdemod(X2,M);x3=qamdemod(X3,M);x4=qamdemod(X4,M);                %解码
error1=(symerr(x01,x1)+symerr(x02,x2)+symerr(x03,x3)+symerr(x04,x4))/4;    %校正误码率
Error1(n,i)=error1;

```

```
mean_Error1=mean(Error1,1);  
end  
end  
figure;  
plot(snr,mean_Error0/64);hold on;  
plot(snr,mean_Error1/64);xlabel('信噪比/dB');ylabel('误码率/%');grid on;legend('无校正误码率','校正误码率');
```