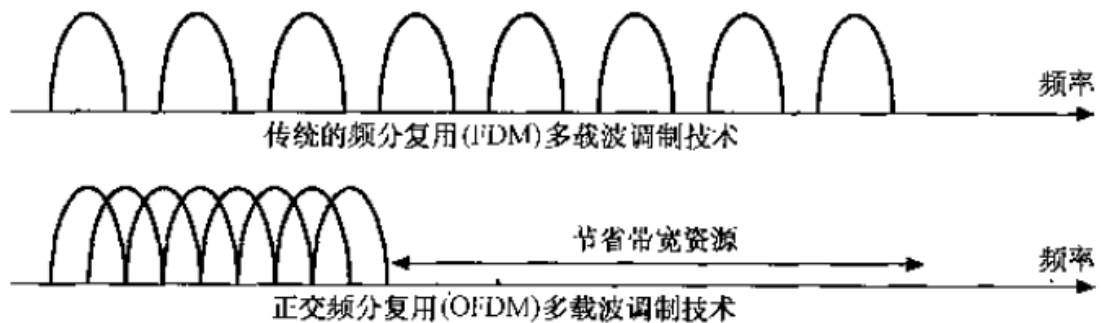


OFDM报告

OFDM概述

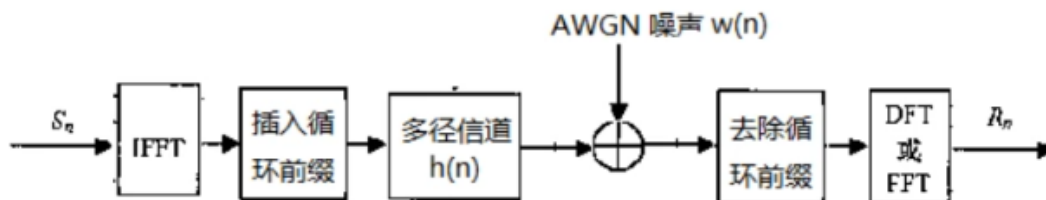


OFDM是“正交频分复用多载波调制技术”的英文缩写。由上图可知，FDM信号的子载波在频域上是没有重叠部分的，而OFDM信号的子载波在频域上具有重叠部分。OFDM在频域上由于存在重叠，因而在频域上并没有体现出正交性，而通过在频域和时域两个维度的联合使得子载波之间存在正交性。

由简化版框图以及要求，OFDM的实现分为：

1. 串并转换
2. ifft处理
3. 插入循环前缀
4. 多径信道以及AWGN噪声引入
5. 去除循环前缀
6. fft处理
7. 均衡
8. 并串转换
9. 判决

另外由于我对普通做法结果不太满意，引入了信道均衡，使得结果相对较好。



仿真参数：

1. 子载波数 $N=256$;
2. CP长度 $N_G = (1/16) N$;
3. 信道 $h(n)$ 为一个总功率为1的复高斯变量;
4. 发送信号为 BPSK调制，平均功率为1;
5. 信噪比为 0、5、10、15、20dB

代码解释

为了方便更改信道的数目，定义函数OFDM。输入参数分别为系统内block数目N，输入的0，1码流input，信道个数，以及信噪比。由于这里对5个不同信噪比进行测试，所以定义一个循环进行。期中输出uwfftoutput为经过fft的输出，用此结果来绘制星座图，uwoutput为判决后的结果，uwser为误码率。

```
for i=1:1:5
    [uwfftoutput(i,:),uwoutput(i,:),uwser(1,i)]=OFDM(N,input,1,(i-1)*5);
end
```

串并转换

在函数OFDM中的第一步。这里的N取1000。concurrent为串并转换后的并行信号。

```
%输入信号的串行/并行变换
count=0;
concurrent=zeros(N,256);
for i=1:1:N
    for j=1:1:256
        concurrent(i,j)=input(1,count+j);
    end
    count=count+256;
end
```

ifft与保护间隔

对并行信号进行ifft之后插入保护间隔。

```

OFDMinput=zeros(N,256);
for i=1:1:N
    OFDMinput(i,:)=sqrt(256)*ifft(concurrent(i,:));
end
%对并行输入信号插入保护间隔
OFDMprotect=zeros(N,272);
for i=1:1:N
    for j=1:1:16
        OFDMprotect(i,j)=OFDMinput(i,240+j);
    end
    for j=1:1:256
        OFDMprotect(i,j+16)=OFDMinput(i,j);
    end
end
end

```

信道与噪声

对 $n_channel$ 个信道赋予复高斯随机变量并归一化，将此参数再与插入保护间隔后的信号卷积得到 y_0 。

```

对OFDM的输出信号进行多径传播：其中 $ak^2$ 满足均匀分布
h=zeros(N,n_channel);
for i=1:1:N
    h(i,:)=0.5.^0.5.*complex(randn(1,n_channel),randn(1,n_channel));
end

```

再使用awgn函数引入白噪声。

```

y0=awgn(y0,SN,'measured');%将噪声加入到多径传播后产生的y0中

```

去除循环前缀并fft

去除循环前缀并fft。

```

FOFDM=zeros(N,256);
for i=1:1:N
    for j=1:1:256
        FOFDM(i,j)=y0(i,16+j);
    end
end
%对FOFDM进行DFT或FFT
FFTout=zeros(N,256);
for i=1:1:N
    FFTout(i,:)=1/sqrt(256)*fft(FOFDM(i,:));
end

```

均衡

因为不加均衡的结果不好，所以我在fft之后又加入了信道均衡。这一步主要完成的是信道均衡。之前在时域上，“多径传播”模拟真实信号传输的过程时，我们用信道抽头系数矩阵 h 与OFDM系统输出的信号进行卷积操作。假设我们通过某种方式得到了抽头系数矩阵 h 通过FFT得到的 H ，则为了消除时域上卷积引入的误差，我们在频域上将FFTout的信号点除 H （按行处理），这样做可以有效避免多径传播的误差。

```

%均衡
H=zeros(N,256);
for i=1:1:N
    H(i,:)=fft(h(i,:),256);
end
for i=1:1:N
    FFTout(i,:)=FFTout(i,:)./H(i,:);
end

```

并串转换及判决

```

FFToutput=zeros(1,(256*N));
count=0;
for i=1:1:N
    for j=1:1:256
        FFToutput(1,count+j)=FFTout(i,j);
    end
    count=count+256;
end

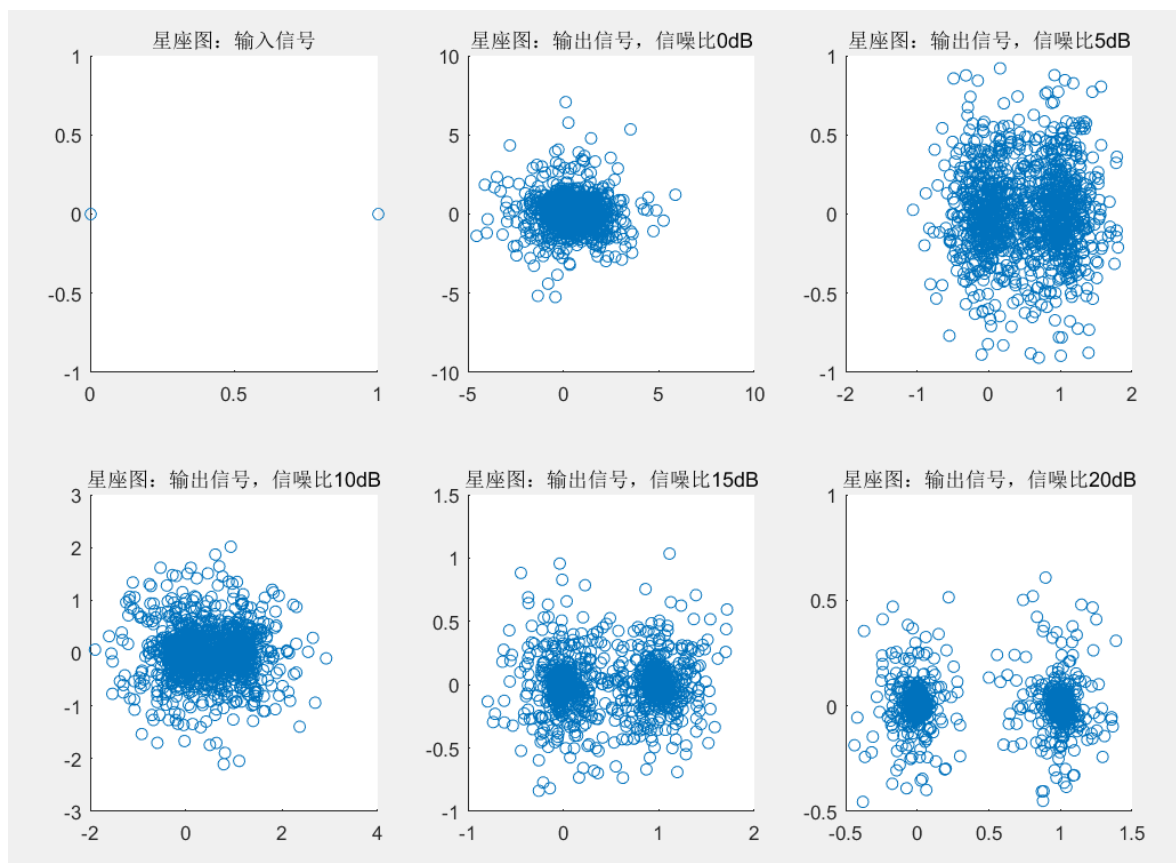
%判决
output=zeros(1,(256*N));
for i=1:(256*N)
    if real(FFToutput(1,i))>0
        output(1,i)=1;
    else
        output(1,i)=0;
    end
end
end

```

结果分析

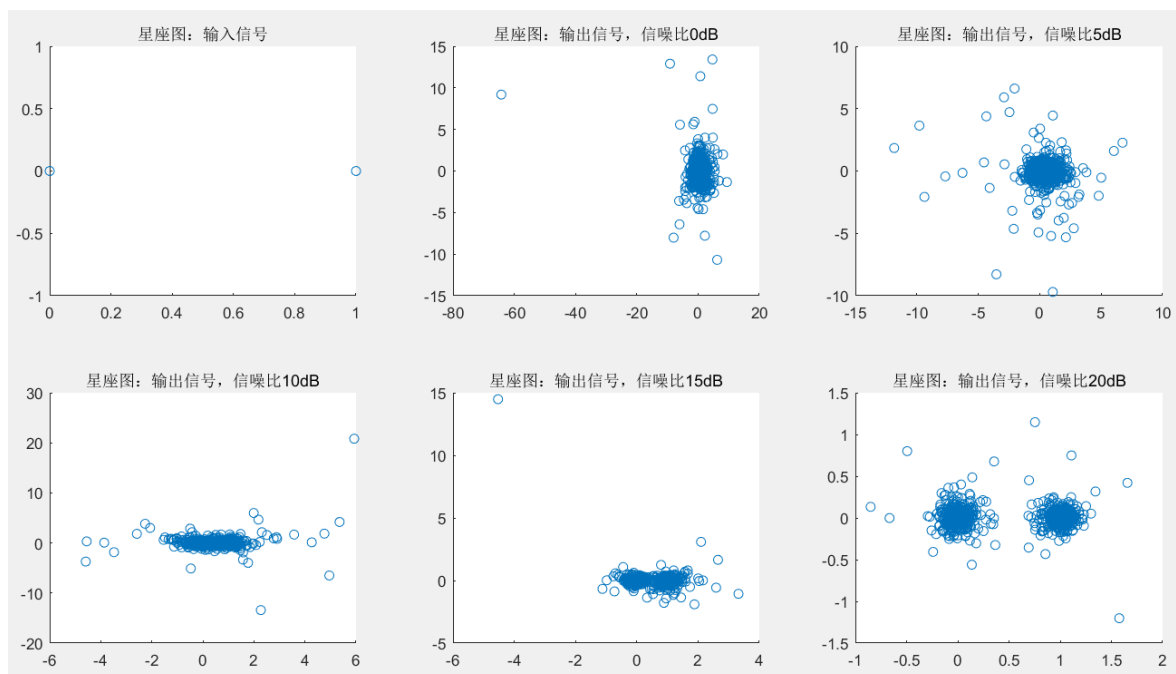
由老师的任务为抽头数为1和10的对比。

抽头数1



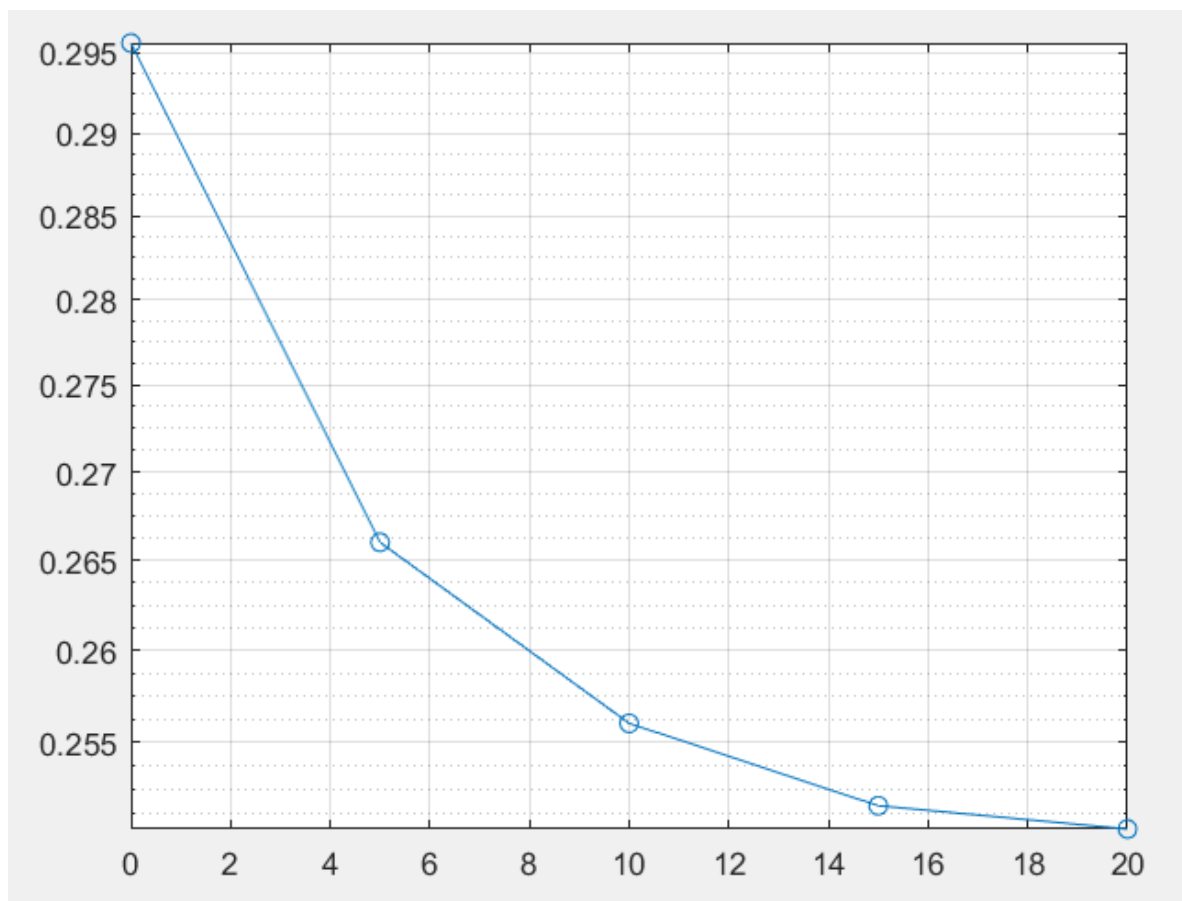
从星座图中可以看到，输入信号只有0和1两个值，之后随着信噪比的增大混在一起的点群渐渐分开，分到了0和1两侧。

抽头数为10

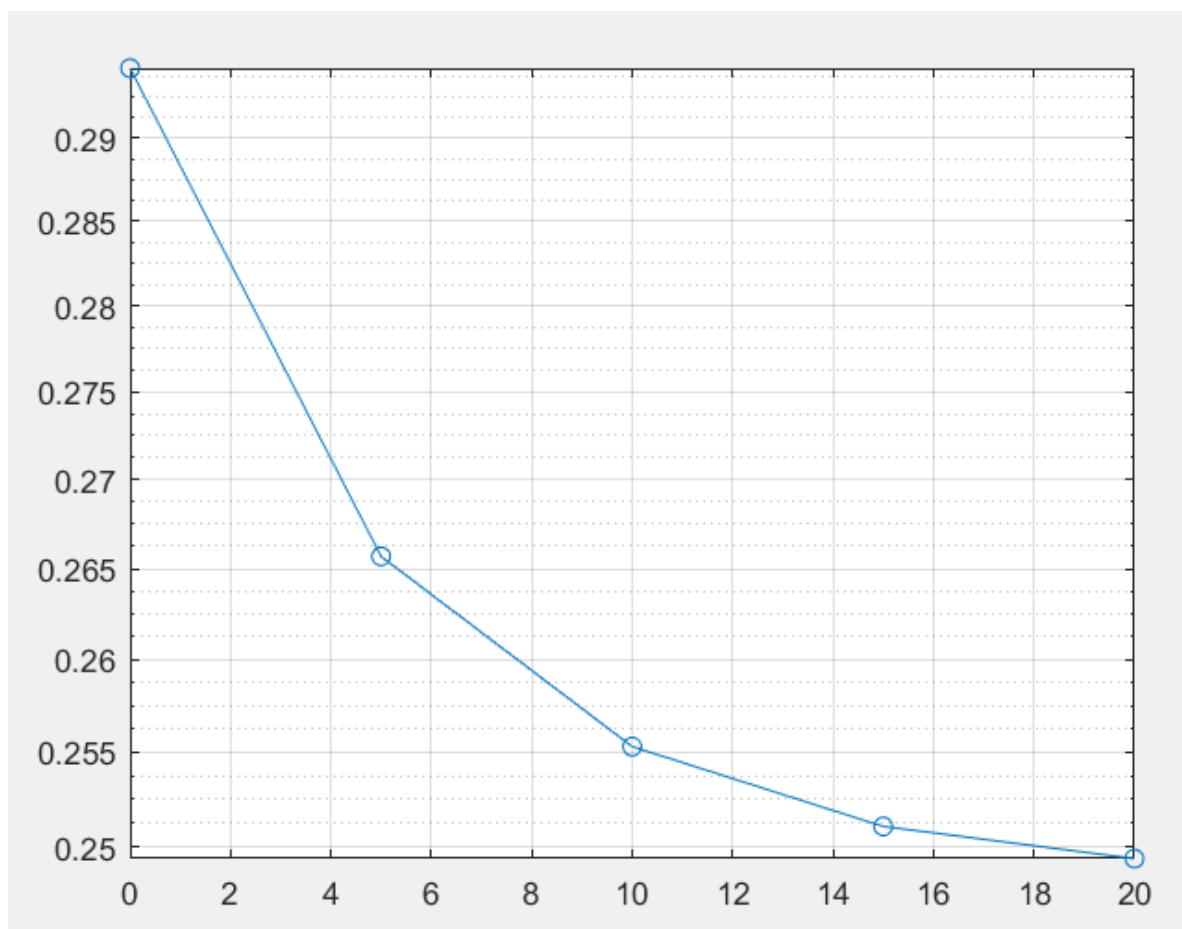


误码率对比

下图为抽头数为一的误码率。



下图为抽头数为10的误码率。



讨论

从上述图中可以得到几条结论。

1. 理论上讲，信噪比越高误码率应该越小，因为信噪比越高由噪声引入的误差影响越小。这里的仿真结果证实了这一点。通过误码率曲线可知，随着信噪比的升高，误码率曲线平滑、单调下降，符合理论分析。
2. 由于进行了信道均衡，所以总体上来看，信道的误码率比未经过信道均衡时小了许多。未经信道均衡时，不论信噪比为多大，误码率都大概在50%左右；而经过信道均衡后，误码率变到了0.2左右。
3. 改变信道抽头数目，误码率没有发现太大的变化，可以看到这种调整信道抽头的方式不能很明显地改变系统的功能。