

# 数字通信实验四 调制解调

PB16061024 陈进泽

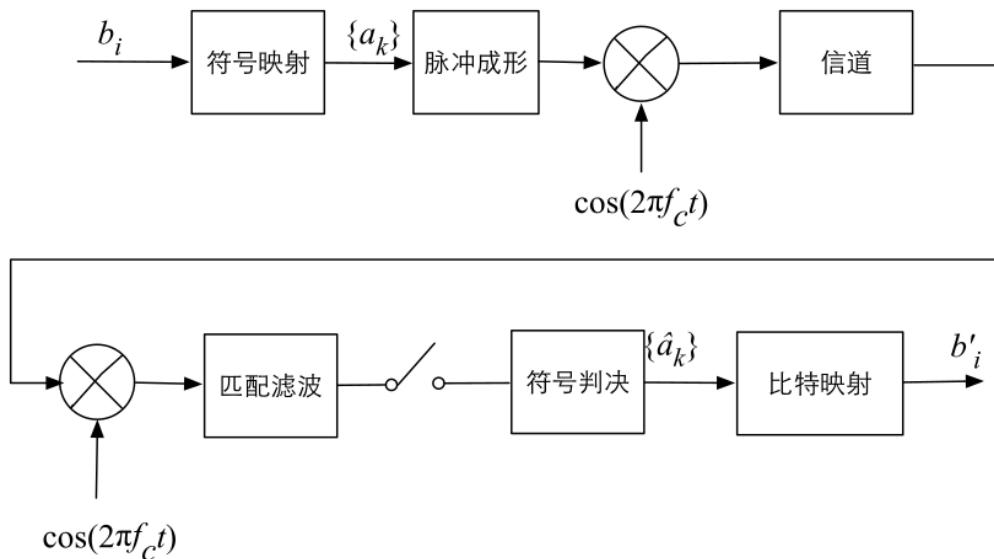
2018 年 12 月 7 日

# 1 实验目的

- 掌握数字频带传输系统调制解调的仿真过程
- 掌握数字频带传输系统误码率仿真分析方法

# 2 实验原理

以 BPSK 为例，数字频带传输过程如下：



假定：信道为加性高斯白噪声信道，其均值为 0、方差为  $\sigma^2$ ，采用矩形成形；发射端 BPSK 调制信号为：

$$s(t) = \begin{cases} A \cos 2\pi f_c t & b_k = "1" \\ -A \cos 2\pi f_c t & b_k = "0" \end{cases} \quad kT \leq t < (k+1)T$$

经信道传输，接受端输入信号为：

$$y_d(t) = s(t) + n(t)$$

经相干解调，匹配滤波，定时恢复后输出：

$$x_k = \begin{cases} A + n_k & b_k = "1" \\ -A + n_k & b_k = "0" \end{cases}$$

当 1, 0 独立等概出现时，BPSK 系统的最佳判决门限电平  $U_d^* = 0$ 。故判决规则为在取样时刻的判决值大于 0，判 1，小于 0，判 0

### 3 实验内容

#### 3.1 分别编写 BPSK 与 QPSK 调制解调系统的 Matlab 仿真程序

##### 3.1.1 BPSK 源代码

```
1 function ber = BPSK( snr )
2     fc=2;
3     %snr = 5
4     fs=16;
5     N=10000;
6     Ts=1;
7     % B=1/Ts;
8     ts=1/fs ;
9     t=0:ts :N*Ts-ts ;
10    N_sample=Ts*fs ;
11
12    P = 0.5*0.5;
13    std_v=sqrt(P/10^(snr/10));
14
15    A=2*randi([0 ,1] ,1 ,N)-1;
16
17    span = 6;
18    h = rcosdesign(0.5 ,span ,N_sample);
19    S = upsample(A,N_sample);
20    S = filter(h,1,S);
21
22    X=S.*cos(2*pi*fc*t);
23
24    X = X + randn(1,length(X))*std_v;
25
```

```

26 r = X.*cos(2*pi*fc*t);
27 baseRec = filter(h,1,r);
28 Br = 2*(baseRec(length(h):N_sample:end)>0)-1;
29 errbit = sum(Br~=A(1:N-span));
30 ber = errbit/(N-span);
31 end

```

为了方便绘制误码率曲线时调用采用了函数形式。以下解释代码中内容:

### 3.1.2 BPSK 代码解释

代码 15 行生成 N 个随机信源符号

代码 17 到 20 行产生相应的基带信号，根升余弦滤波器滚降系数为 0.5

代码 22 行进行信源的调制

代码 24 行加噪声

代码 26 到 28 行进行信号的解调，匹配滤波及采样判决

其中因为发送滤波器的影响，需根据 SNR 对噪声进行调整。这里采用了等效为码元功率的变化的方式等效了其影响，具体体现在代码 12 行

### 3.1.3 QPSK 源代码

```

1 function ber = QPSK(snr)
2     fc=2;
3     %snr = 5;
4     fs=16;
5     N=10000;
6     Ts=1;
7     % B=1/Ts;
8     ts=1/fs;
9     t=0:ts:N*Ts-ts;

```

```

10 N_sample=Ts*fs ;
11
12 P = 1*0.5^2;
13 std_v=sqrt(P/10^(snr/10));
14
15 A=2*randi([0,1],2,N)-1;
16
17 span = 6;
18 h = rcosdesign(0.5,span,N_sample);
19 S(1,:) = upsample(A(1,:),N_sample);
20 S(2,:) = upsample(A(2,:),N_sample);
21 S(1,:) = filter(h,1,S(1,:));
22 S(2,:) = filter(h,1,S(2,:));
23 X = S.*cos(2*pi*fc*t-pi/2*(0:1)');
24
25 X = X + randn(2,length(X))*std_v;
26
27 r = X.*cos(2*pi*fc*t-pi/2*(0:1)');
28 baseRec(1,:) = filter(h,1,r(1,:));
29 baseRec(2,:) = filter(h,1,r(2,:));
30 Br = 2*(baseRec(:,length(h):N_sample:end)>0)-1;
31 det = Br ~= A(:,1:N-span);
32 errbit = sum(sum(abs(det))~=zeros(1,N-span));
33 ber = errbit/(N-span);
34 end
35 %{
36 X=S.*cos(2*pi*fc*t);
37
38 X = X + randn(1,length(X))*std_v;
39
40 r = X.*cos(2*pi*fc*t);

```

```

41 baseRec = filter(h,1,r);
42 Br = 2*(baseRec(1:length(h):N_sample:end)>0)-1;
43 errbit = sum([Br,zeros(1,span)]~=A);
44 ber = errbit/N;
45 \%}

```

同 BPSK 一致采用了函数形式表达

### 3.1.4 QPSK 代码解释

代码将发送信号分两路进行发送，并分两路解调

代码 31 行判决接收到的码元各位是否与发送码元一致

代码 32 行判决若所有位的码元均相等，则接受成功

## 3.2 绘制 BPSK 与 QPSK 调制下的误码率与新噪比曲线图，并与理论曲线进行对比

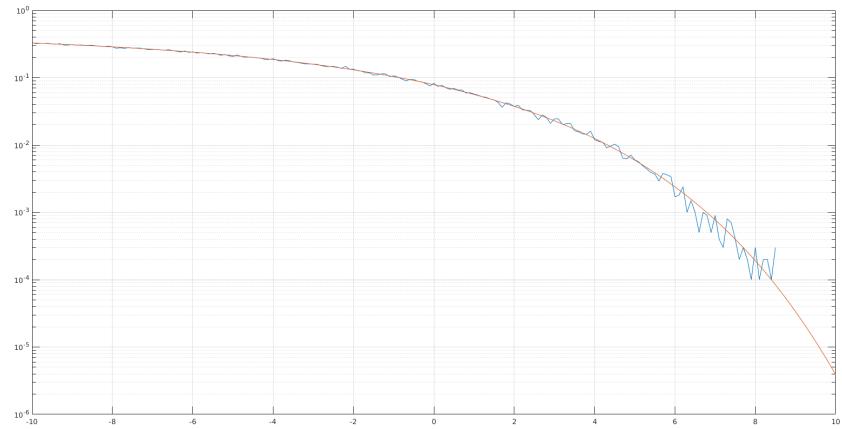
### 3.2.1 BPSK

```

1 snr = -10:0.1:10;
2 res = zeros(1,length(snr));
3 for i = 1:length(snr)
4     res(i) = BPSK(snr(i));
5 end
6 ref = 1/2*erfc(sqrt(10.^^(snr/10)));
7 semilogy(snr,res,snr,ref);

```

### 3.2.2 实验效果

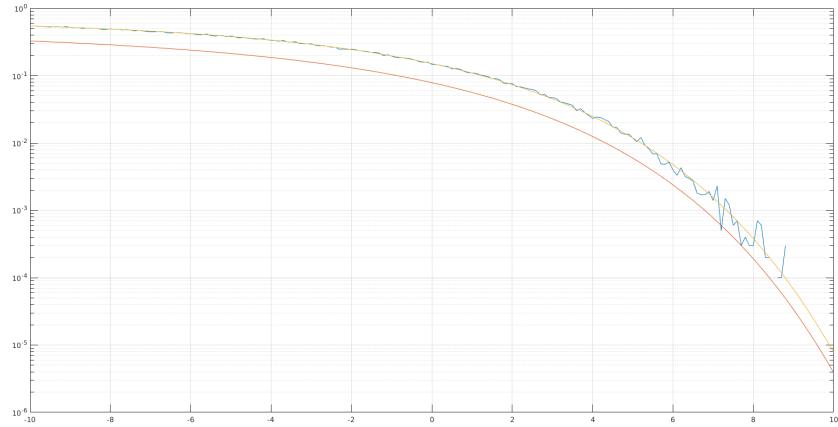


由图可见，仿真曲线与实际曲线十分相似，说明理论正确，解调方式也无误

### 3.2.3 QPSK

```
1 snr = -10:0.1:10;
2 res = zeros(1,length(snr));
3 for i = 1:length(snr)
4     res(i) = QPSK(snr(i));
5 end
6 ref = 1/2*erfc(sqrt(10.^^(snr/10)));
7 semilogy(snr,res,snr,ref,snr,1-(1-ref).^2);
```

### 3.2.4 实验效果



如图，由 QPSK 函数计算的是误码率，上方曲线代表理论误码率曲线，为  $1 - (1 - \frac{1}{2}erfc(\sqrt{snr}))^2$ 。下方曲线为理论误比特率曲线，与 BPSK 情况下的误码率曲线一致，且在对数坐标下几乎只差个平移，说明两者为倍数关系，与理论值  $P_e \approx erfc(\sqrt{snr})^2 = 2P_b$  一致，说明调制与解调方法没错，且与理论一致

## 3.3 几点说明

1. 由于实际随机生成序列长度有限，当误码率足够小时可以想到错误的点的个数也足够小，故计算获得的误码率偏差较大。这点在仿真图上也有体现，当 SNR 足够大时误码率足够小，故理论曲线与仿真曲线偏差较大，而在小 SNR 处两者吻合的很好
2. 由于 filter 进行的是卷积操作，原信号会有所偏移。且 `rcosdesign` 函数生成的函数长度为  $|h| = span \times N_{sample} + 1$ ，为使采样判决不出现偏差需将判决点右移  $|h|$  个单位
3. 关于发送滤波器对 SNR 的影响，若其在 BPSK 下的影响为  $\alpha$ ，则由于发送信号为累加的关系，在 QPSK 下它的影响为  $\alpha^2$ 。具体数值  $\alpha$  为发送滤波器的函数。由于并无太多时间分析 matlab 函数 `rcosdesign` 的具体原理，其数值  $\alpha = 0.5$  为多次尝试后所得结果，并经验证其与

span 的具体数值无关。一个可能有用的信息为  $\sum_{i=1}^{|h|} h^2(i) = 1$