

浙江大学

4ASK数字通带传输系统

课程名称：	通信原理
姓 名：	周灿松
学 院：	信息与工程学院
系：	
专 业：	信息工程
学 号：	3190105055
指导老师：	王玮

2022 年 6 月 23 日

基于 4ASK 的数字通带传输系统

周灿松；3190105055；信息与工程学院；信息工程

一、 系统说明

构建基于 4ASK 的数字通带传输系统，考虑加性高斯白噪声信道条件下，用 MATLAB 仿真采用相干解调时的数字通带传输系统。

系统的大致流程如图 1所示：



图 1: 系统工作流程

图中上半部分属于发射机工作，下半部分为接收机工作。加性高斯白噪声信道采用在调制后信号上加入一个高斯白噪声的方法引入。

二、 仿真思路

1. 输入图片与脉冲成型

在 MATLAB 中，我们采用 `imread` 函数读取彩色图像，获取到了 RGB 三个通道的灰度图。因为灰度值的范围为 0-255，而 4ASK 信道处理的是四电平信号，所以在进行图像传输前需要将图像的灰度值处理为 4 进制数据，并且将其转换为双极性，然后再使用方波进行脉冲成型。

相关代码编写如下：

Listing 1: pic2plush

```
1 function [x , n] = pic2pluse(pic , fs , fb)
   % 此函数用于将图片转换为脉冲成型序列
3
   m = fs/fb;
5 %按RGB三通道读取图片
   r_origin = reshape(pic(:,:,1),1,[]);
7   g_origin = reshape(pic(:,:,2),1,[]);
   b_origin = reshape(pic(:,:,3),1,[]);
9 % 获取单个通道的长度
   n = size(r_origin , 2);
```

```

11     r_progress = zeros(1,4*n*m);
13     g_progress = zeros(1,4*n*m);
15     b_progress = zeros(1,4*n*m);
17     for i=1:1:n
18         r = dec2base(r_origin(1,i),4,4);
19         g = dec2base(g_origin(1,i),4,4);
20         b = dec2base(b_origin(1,i),4,4);
21         for j = 1:1:4
22             r_temp = 2*str2double(r(j))-3;
23             g_temp = 2*str2double(g(j))-3;
24             b_temp = 2*str2double(b(j))-3;
25             for k = 1:1:m
26                 r_progress(4*m*(i-1)+m*(j-1)+k) = r_temp;
27                 g_progress(4*m*(i-1)+m*(j-1)+k) = g_temp;
28                 b_progress(4*m*(i-1)+m*(j-1)+k) = b_temp;
29             end
30         end
31     end
32     x = [r_progress , g_progress , b_progress];
33 end

```

实现思路：

- (1) 首先提取出图像的 RGB 三通道值，并将其转换为一个列向量。
- (2) 调用 dec2base 函数将灰度值转换为四进制，转换为整数并且将值映射到 $\{-3,-1,1,3\}$ 上。
- (3) 通过采样率与符号速率，计算出一个符号中需要有多少个点，将其脉冲成型为方波。
- (4) 最后将三通道数据进行拼接，得到我们需要传送的基带脉冲。

接收端解调出 4 进制数据后，只需要按照同样规则恢复图像即可。

2. 4ASK 调制与解调

上述步骤得到待发送序列后，我们只需要将其与载波信号相乘即得到了发送信号，在数学上表示为：

$$Tx = x(t) \cdot \cos 2\pi f_c t$$

加性高斯白噪声信道我们通过 awgn 函数进行实现。

接收端接收到信号后，采用相关解调进行解调，然后在通过低通滤波器后采样，该部分实现如下：

Listing 2: 调制与解调

```

function sample = trans(x,SNR,fs,fc)
2     T = length(x);
3     dt = 1/fs;
4     t = 0:dt:(T-1)*dt;
5     y = x.*cos(2*pi*fc*t);

```

```
6     y = awgn(y,SNR,'measured');

8     y = y.*cos(2*pi*fc*t);
    [b,a] = butter(2,2*fc/fs);
10    y = filtfilt(b,a,y)*2;
    sample = y(2:32:end);
12 end
```

3. 判决

采用最佳距离判决准则对采样后的序列进行判决，即采用 $\{-2,0,2\}$ 进行判决。

Listing 3: 判决

```
1 function judge_series = judge(sample)
    judge_series = zeros(1 , length(sample));
3 for i = 1:length(sample)
    if sample(i) > 2
5         judge_series(i) = 3;
    elseif sample(i) > 0
7         judge_series(i) = 1;
    elseif sample(i) > -2
9         judge_series(i) = -1;
    else
11        judge_series(i) = -3;
    end
13 end
end
```

4. 图像还原

采用与图像输入时同样的方法对接收端解调得到的数据进行恢复为 0-255 之间的值，并且将其重新恢复到三个通道进行显示。

Listing 4: 图像恢复

```
1 function pic_recovered = recover(judge_series , n, w, h, c)
    judge_series = (judge_series+3)/2;
3    pic_recover = zeros(1 , n*3);
    for i = 1:4:length(judge_series)
5        pic_recover((i-1)/4+1) = judge_series(i)*64+judge_series(i+1)*16+judge_series
            (i+2)*4+judge_series(i+3);
    end
7
    pic_recovered = uint8(reshape(pic_recover , [w,h,c]));
9 end
```

三、 实现结果

1. 传输图像结果



图 2: 原图



图 3: 信噪比为 0dB 时结果

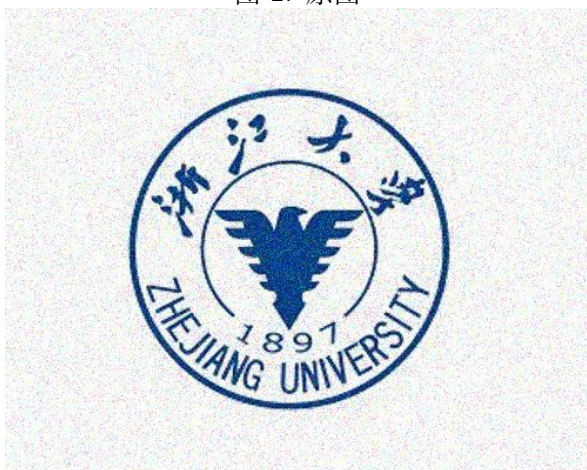


图 4: 信噪比为 5dB 时结果



图 5: 信噪比为 10dB 时结果



图 6: 信噪比为 15dB 时结果



图 7: 信噪比为 20dB 时结果

对比发送的图像可以看出，当信噪比大于 15dB 时，接收到的图像已经与原图基本没有区别，说明了该系统在信噪比达到一定程度时可以保证传输失真足够小，仿真结果准确。

2. 误码率分析

对比不同信噪比情况下的传输结果与理论计算得到的误码率，整理结果如下所示：

SNR	0	1	2	3	4	5	6
理论误码率	0.395317	0.358452	0.319431	0.278745	0.237122	0.195543	0.155235
实际误码率	0.24926	0.221094	0.192049	0.162774	0.133567	0.106082	0.080442
SNR	7	8	9	10	11	12	13
理论误码率	0.117605	0.084103	0.056001	0.034125	0.018623	0.008855	0.003545
实际误码率	0.058027	0.039293	0.025281	0.015178	0.008852	0.004814	0.00258
SNR	14	15	16	17	18	19	20
理论误码率	0.001144	0.000282	4.94E-05	5.67E-06	3.80E-07	1.30E-08	1.90E-10
实际误码率	0.001385	0.000779	0.000389	0.000177	7.60E-05	1.87E-05	3.40E-06

绘制出曲线图如下：

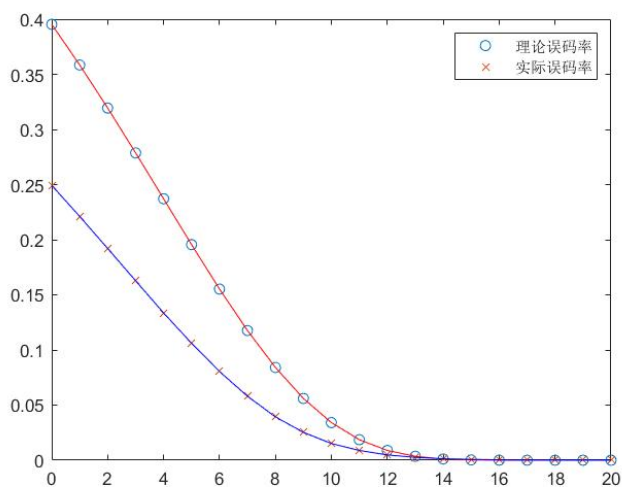


图 8: 4ASK 误码率分析

由上述结果可以看出，实际误码率比理论上的误码率有着一定的差距，分析后我认为造成这一结果的主要原因是因为我们在计算理论误差时假设了各个发送波形为等概的，但是对于我们传输的图像来说，大部分传输的数据为 255，这也就导致了出现的概率并不是等概的，这也就导致了最终的实际误码率比理论误码率小。