

通信系统仿真实验报告

马江岩

2021 年 12 月 30 日

1 介绍

本次实验利用 MATLAB 工具, 实现了通信系统的仿真实验. 具体的实验流程如图 1 所示.

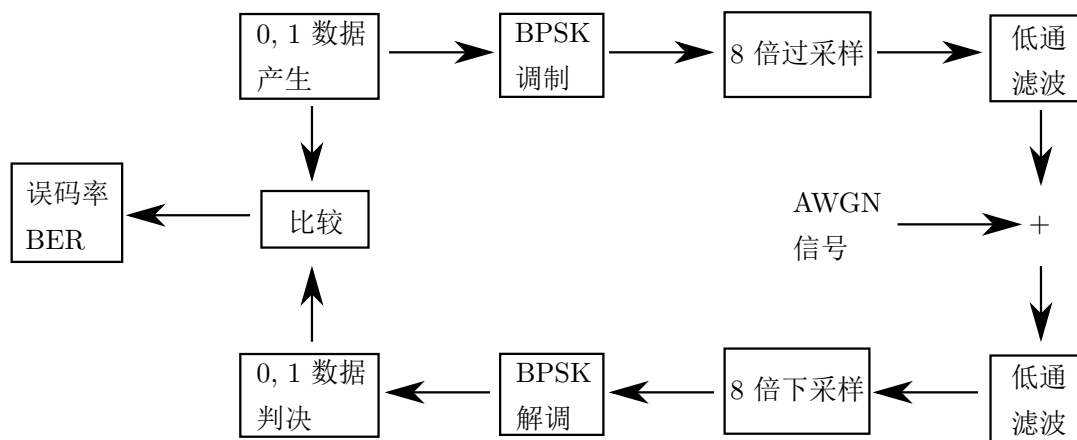


图 1: MATLAB 通信系统仿真实验流程.

发端通过信源直接生成随机 0, 1 信号 (概率相等), 通过 BPSK 方式进行调制, 上采样后进行基带成型滤波 (采用根升余弦滤波器), 模拟高斯信道的环境, 人为加入加性高斯白噪声.

收端先进行匹配滤波 (采用与发端相同的滤波器) 提高系统的信噪比, 然后进行下采样, 译码判决, 与发端的序列进行比较统计误码率.

在调制和解调阶段, 除了使用 BPSK 调制, 我还使用了 64-QAM 方式进行了调制, 并对两种方式进行了对比; 在低通滤波阶段, 采用的是平方根升余弦滤波器.

我们不妨以一个随机生成的 12 位的 0, 1 信号为例, 其时域波形如图 2 所示, 频谱如图 3 所示.

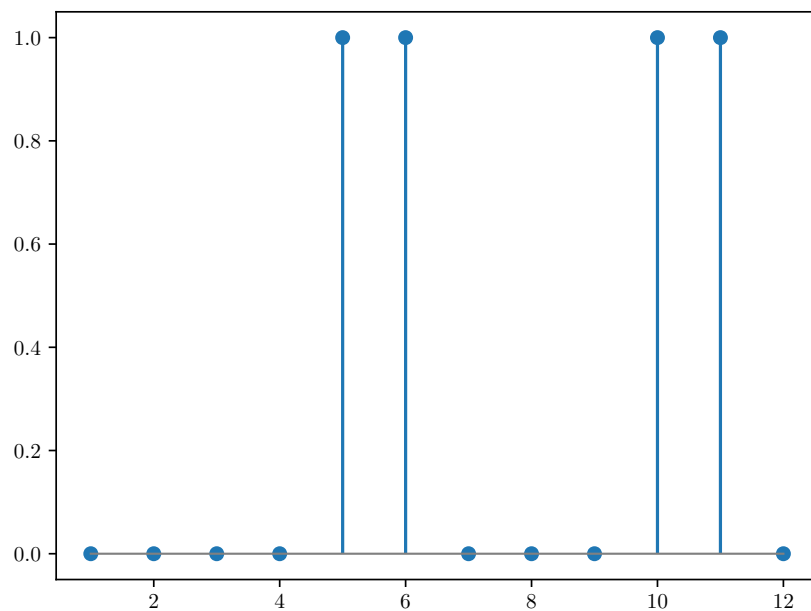


图 2: 12 位随机信号的时域波形.

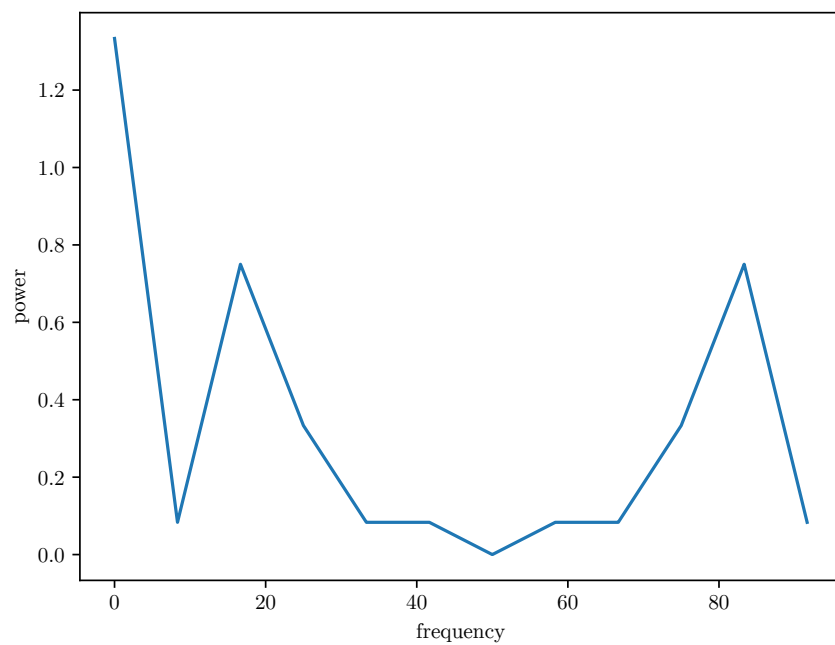


图 3: 12 位随机信号的频谱.

2 信号的调制

对由 0, 1 组成的信号, 一种比较简单的调制方式是 BPSK 调制 (binary phase-shift keying), 它将比特 0 调制成 1, 将比特 1 调制成 -1, 其星座图如图 4 所示.

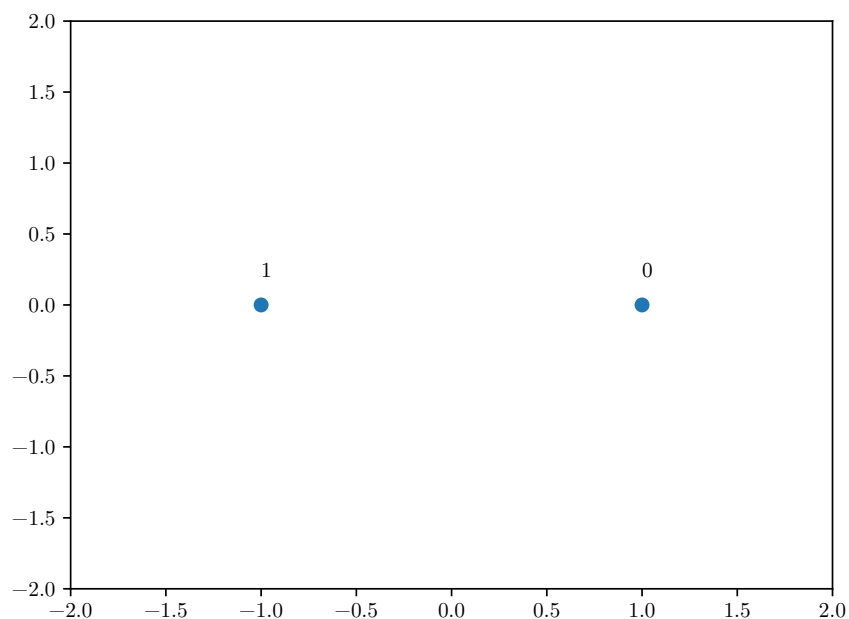


图 4: BPSK 调制星座图.

我使用的另一种调制方式是 M -QAM (quadrature amplitude modulation), 它将每 $\log_2 M$ 个比特编为一组, 给每个组赋予一个具有一定相位和幅角的复数. 若信号的长度不是 M 的整数倍, 我们将信号末尾补零, 凑成 M 的整数倍. 64-QAM 的星座图如图 5 所示.

为了比较这两种调制方式, 我在不同的信噪比下 (其他条件相同) 分别使用 BPSK 和 64-QAM 对信号进行调制, 其误码率如图 6 所示.

可以看出, 在实验条件下, 使用 64-QAM 调制的误码率更低. 并且 64-QAM 调制的效率更高, 因为它将每 6 个比特调制成一个复数, 而 BPSK 将每个比特调制成一个实数.

下文的讨论中, 我们均默认采用了 64-QAM 调制.

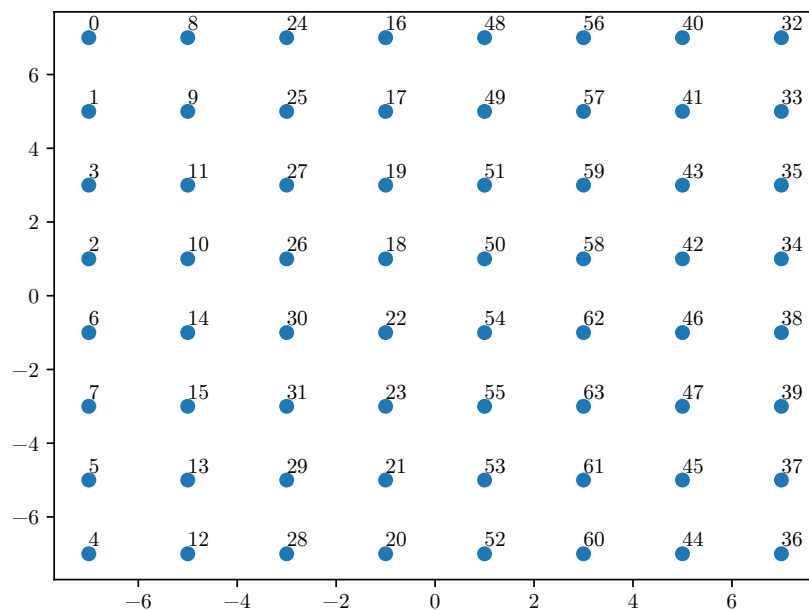


图 5: 64-QAM 调制星座图.

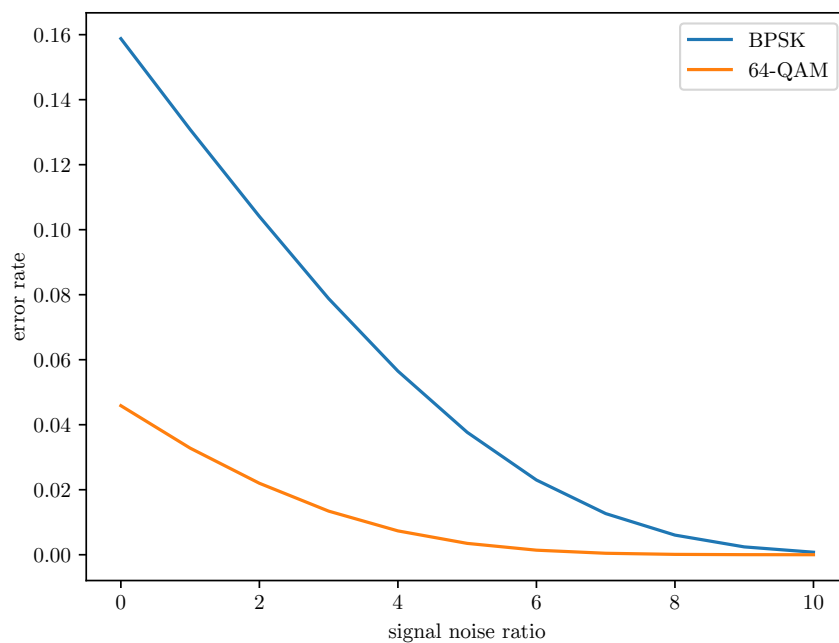


图 6: 在不同的信噪比下, 使用 BPSK 和 64-QAM 调制的误码率.

3 低通滤波

信号在经过 64-QAM 调制后, 进行 8 倍上采样, 再进行低通滤波作为发送信号. 我们在发送端和接收端采用相同的低通滤波器, 即平方根升余弦滤波器, 其时域的表达式为

$$h(t) = -\frac{4\beta}{\sqrt{T}} \frac{\cos \frac{(1+\beta)\pi t}{T} + \frac{T}{4\beta t} \sin \frac{(1-\beta)\pi t}{T}}{\pi \left(\left(\frac{4\beta t}{T} \right)^2 - 1 \right)}.$$

在具体的 MATLAB 实现中, 为了避免出现除以零的情况, 一方面, 当 $t = 0$ 时, 运用洛必达法则并代入 $t = 0$, 得

$$h(0) = -\frac{\pi(\beta - 1) - 4\beta}{\pi\sqrt{T}};$$

另一方面, 若 $h(t)$ 分母为零, 即

$$\left| \left| \frac{4\beta t}{T} \right| - 1 \right| < \sqrt{\text{eps}}$$

时, 运用洛必达法则并代入 $t = \frac{1}{4\beta}$, 得

$$h(t) = \frac{1}{2\pi\sqrt{T}} \left[\pi(\beta + 1) \sin \frac{\pi(\beta + 1)}{4\beta} - 4\beta \sin \frac{\pi(\beta - 1)}{4\beta} + \pi(\beta - 1) \cos \frac{\pi(\beta - 1)}{4\beta} \right].$$

经过 64-QAM 调制、8 倍上采样、低通滤波后的信号波形如图 7 所示, 其频谱如图 8 所示.

4 加入加性高斯白噪声

信号经低通滤波后, 我们以一定的信噪比 r 加入加性高斯白噪声 (additive white gaussian noise). 噪声的功率为

$$p = \frac{1}{10^{\frac{r}{10}}},$$

加入噪声后的信号 x' 与原信号 x 的关系为

$$x' = x + \sqrt{\frac{p}{2}} (\text{randn}_1 + i \cdot \text{randn}_2),$$

其中 $\text{randn}_1, \text{randn}_2$ 是由标准正态分布产生的随机数, 即

$$\text{randn}_1, \text{randn}_2 \sim N(0, 1).$$

改变不同的信噪比, 采用蒙特卡洛算法进行模拟, 所得到的误码率与信噪比的关系曲线如图 9 所示. 可见, 误码率与信噪比呈反比.

下文的讨论中我们默认信噪比为 10.

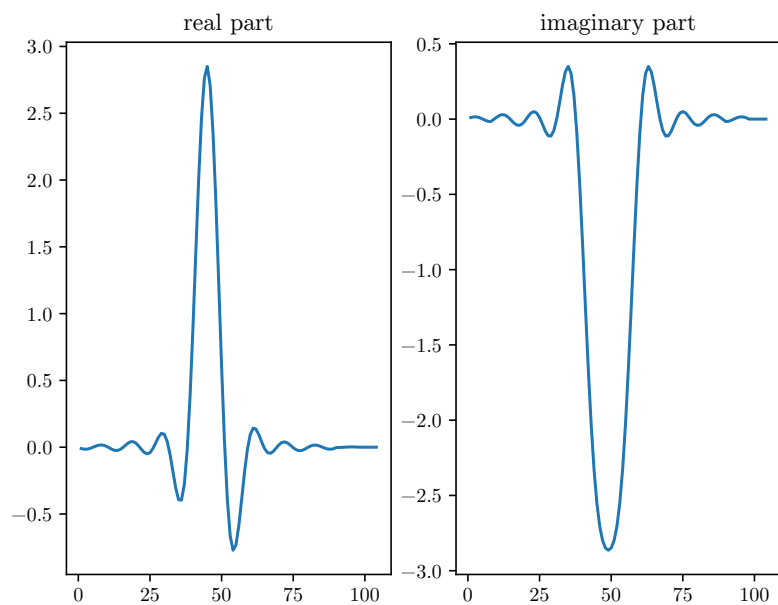


图 7: 发送端经低通滤波后的信号波形.

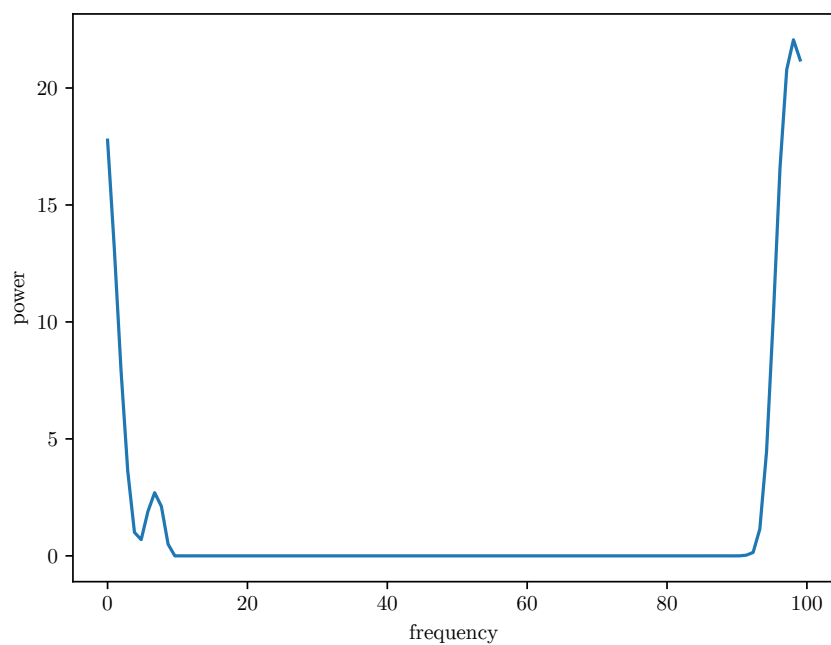


图 8: 发送端经低通滤波后的信号频谱.

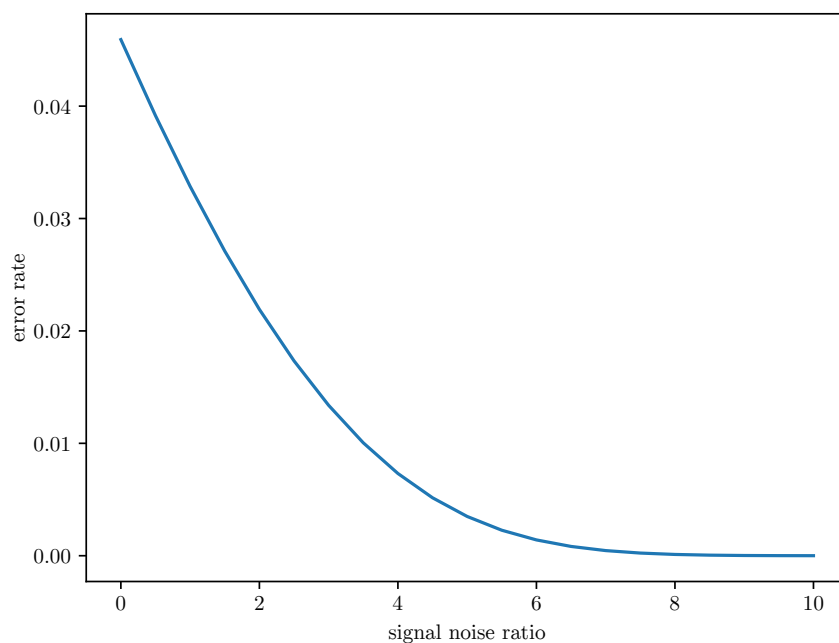


图 9: 误码率与信噪比的关系曲线.

5 匹配滤波

接收端使用与发送端相同的平方根升余弦滤波器, 经滤波后的接收信号的波形如图 10 所示, 频谱如图 11 所示.

与图 7 和图 8 对比可以看出, 恢复后的信号波形和频谱与发送端的非常相似.

6 信号的解调

我们对接收端低通滤波后的信号进行 8 倍下采样, 截去信号两端的部分使得信号与发送端调制后的信号长度相同, 然后依据图 5 中的星座图对信号进行解调. 我们的解调方式是对每个信号, 将其译码为星座图中与其欧氏距离最近的复数, 然后依据星座图将其解调为对应的 6 位比特序列. 解调后的信号的波形如图 12 所示, 频谱如图 13 所示.

与图 2 和图 3 对比可以看出, 恢复出的信号与发送端产生的随机信号完全相同, 误码率为 0%.

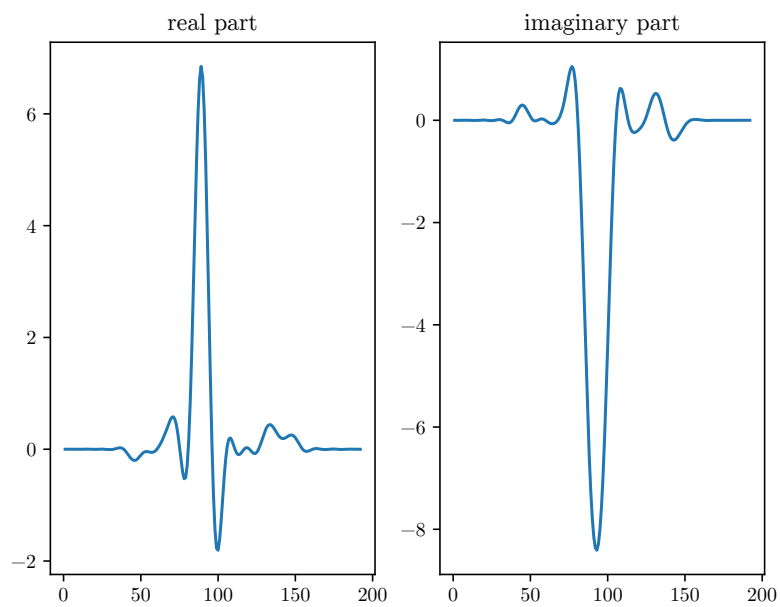


图 10: 接收端经低通滤波后的信号波形.

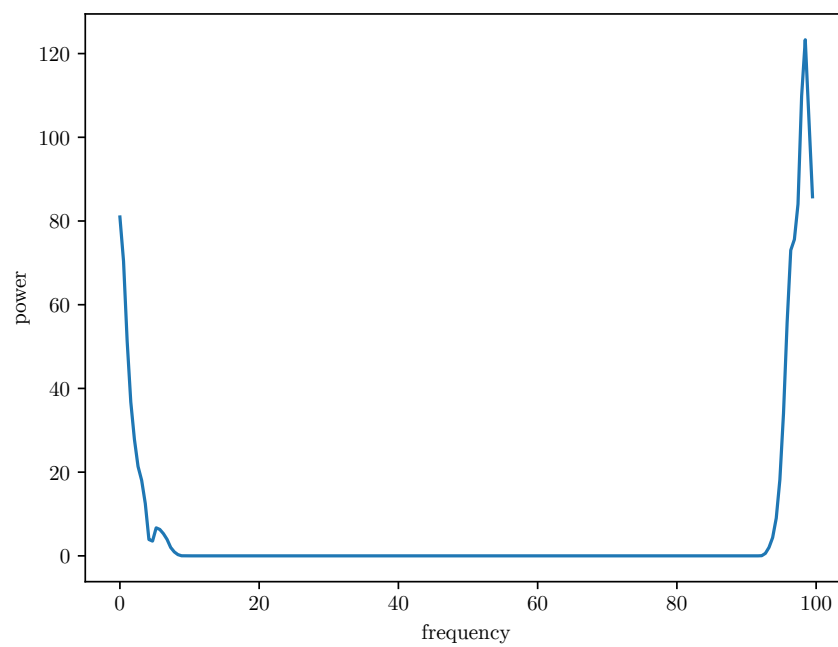


图 11: 接收端经低通滤波后的信号频谱.

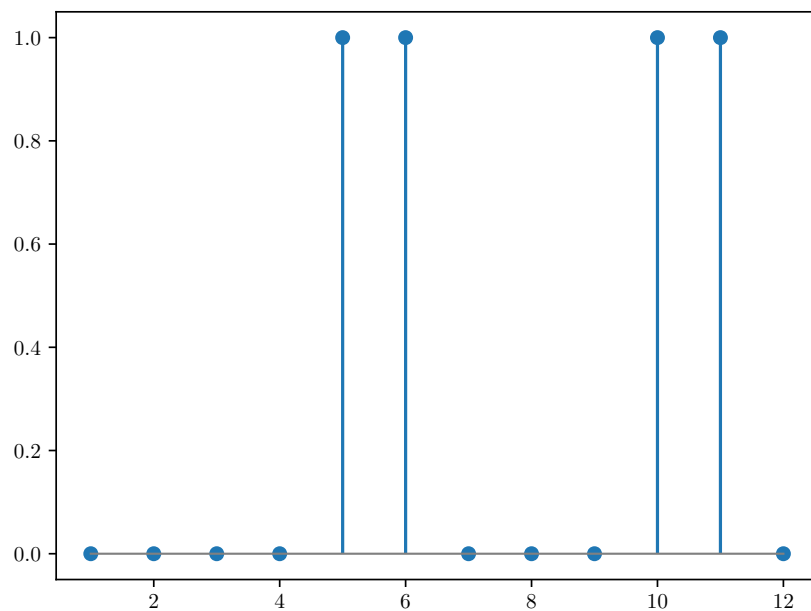


图 12: 接收端恢复出的信号波形.

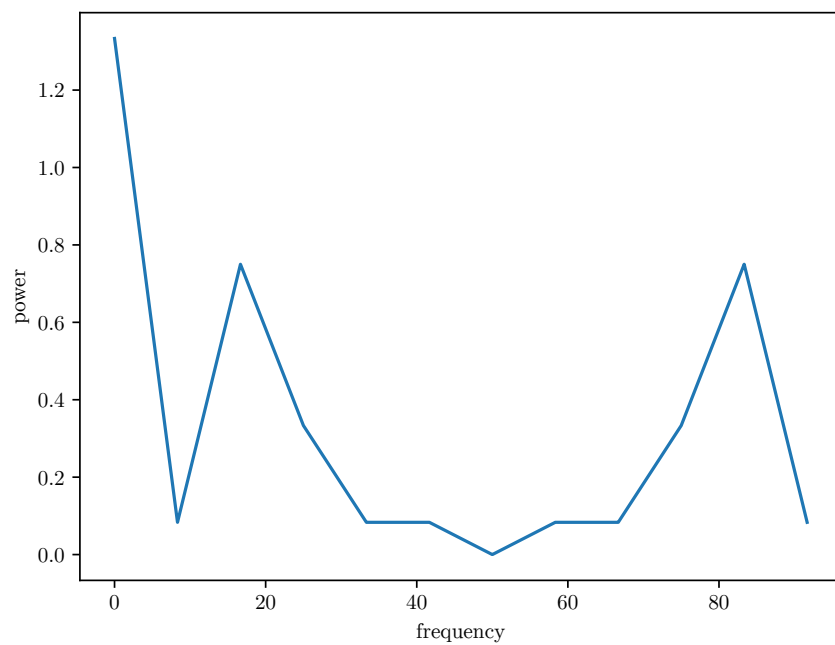


图 13: 接收端恢复出的信号频谱.

7 思考题

1. 发端为什么要经过过采样, 过采样倍数和什么有关?

在信号处理中, 过采样是指以明显高于奈奎斯特速率的采样频率对信号进行采样. 从理论上讲, 如果以奈奎斯特速率或更高的速率进行采样, 则可以完美地重建带宽受限的信号. 因此, 过采样可以更真实地重建原始信号.

对信号进行过采样, 所需的过采样率可以由下面公式确定:

$$f_{os} = 4^W \times f_s,$$

其中 f_{os} 为过采样频率, f_s 为产品所需实际采样频率, W 为额外所需增加的分辨率位数. 由上式可以看出, 利用过采样可以提升采样分辨率; 每提高 W 位分辨率, 需要提高采样率 4^W 倍.

2. 为什么要使用根升余弦滤波?

根升余弦滤波是一种低通滤波, 可以消除信号中的高频噪声, 去除信号短期的振荡, 使信号变得更平滑. 至于使用根升余弦滤波器, 是因为在发送端和接收端分别用一个平方根升余弦滤波器, 既能实现升余弦滤波器的作用, 也能满足匹配滤波器的实现, 从而既能满足奈奎斯特采样定理, 又可以提升接收端信噪比, 更便于准确接收信号.

3. 收端经过匹配滤波后信噪比有什么变化, 为什么?

收端经过匹配滤波后信噪比会增大, 原因是平方根升余弦滤波器是低通滤波器, 消除了高频噪声, 提升了信噪比. 可以证明, 使得收端信噪比最大的匹配滤波器的频率响应是输入信号频率响应的共轭, 而我们发送端使用的滤波器为平方根升余弦滤波器, 因此在接收端也应当使用同样的滤波器, 使得信噪比最大.

A 程序结构

`QAM_without_System_Functions.m` 是本项大作业的主程序, 其结构如下.

首先我们定义了信号的大小 `signal_size = 10000000`, 上采样率 `upsample_rate = 8`, 信噪比 `signal_noise_ratio = 10`, 调制方式为 M-ary QAM, 其中 $M = 64$, 调制时每 $k = 6$ 个比特为一个单位.

然后我们产生了大小为 `signal_size` 的随机 0, 1 信号. 若其长度不是 k 的倍数, 我们在其后补零使其长度为 k 的倍数. `qam_list` 为 64-QAM 的星座图中相应的复数, `power_series` 为 2 的幂构成的数组, 用于在十进制和二进制间进行转换.

之后我们对信号进行 64-QAM 调制, 并对其上采样, 得到 `upsampled_signal`.

随后我们利用平方根升余弦滤波器的相关公式, 生成平方根升余弦滤波器的参数, 用其对信号进行滤波, 并加入高斯噪声得到 `noisy_signal`.

接着我们用与发送端相同的滤波器对信号进行滤波, 再下采样、解调, 得到复原信号. 最后我们将 `received_signal` 与 `signal` 进行比较, 得到误码率 `error_rate`.

B 文件说明

所有 .m 文件需在 MATLAB R2016b 及以上版本的 MATLAB 中运行, 否则诸如 `randi`、`rcosdesign` 等函数可能会报错.

各个附件的内容如下:

- `BPSK_Using_System_Functions.m` 使用 BPSK 调制, 调用了系统函数.
- `QAM_Using_System_Functions.m` 使用 64-QAM 调制, 调用了系统函数.
- `QAM_without_System_Functions.m` 使用 64-QAM 调制, 没有调用相关系统函数. 这也是这项大作业的主要程序.
- `Monte_Carlo.m` 运用蒙特卡洛算法测量了误码率与信噪比的关系曲线.
- `BPSK 与 64-QAM 的比较.pdf` 比较了 BPSK 与 64-QAM 两种调制方式在不同信噪比下的误码率.
- 误码率与信噪比的关系图.pdf 为误码率与信噪比的关系曲线.
- 通信系统仿真实验报告.pdf 为本项大作业的实验报告.