华中科技大学光学与电子信息学院

《信号与系统》课程

工程设计问题设计报告

题　　目: 调幅信号的解调

分 组 号: 14

组　　长: 陈恪瑾

组　　员: 陈恪瑾

时　　间: 2025 年 09 月 01 日 ~2025 年 11 月 26 日

指导教师: 于源

报告日期: 2025 年 09 月 01 日

报告撰写说明

1. 按照参考模板的内容和格式撰写报告
2. 理论模型部分须结合本课程知识分析问题、建立模型
3. 程序设计部分应给出设计思路、主要流程图和关键函数的说明；结果分析不能只是简单给出结论，应结合具体问题，对关键参数或算法在不同取值条件下对结果的影响情况进行分析和总结。如果可能，还应进行误差分析

目录

1 问题描述

调幅（Amplitude Modulation, AM）是一种重要的模拟调制技术，广泛应用于无线电广播、通信系统等领域。在调幅通信系统中，低频信号（基带信号）通过改变高频载波的幅度来实现信息的传输。接收端需要通过解调过程从调制信号中恢复出原始的基带信号。

相干解调是一种常用的解调方法，其核心思想是在接收端使用与发送端载波频率和相位相同的本地载波信号与接收到的调制信号相乘，然后通过低通滤波器滤除高频分量，从而恢复出原始信号。然而，在实际应用中，由于频率源的不稳定、信道传输的影响或接收机的误差等因素，接收端的本地载波频率可能与发送端的载波频率存在偏差，这将导致解调失败或信号失真。

本题研究调幅信号的解调问题。文件 project.wav 中包含了一段错误解调的音频信息的采样值，原始信号是以载波频率 f_c 进行调制的，但在解调时却使用了错误的频率 $\tilde{f}_c \neq f_c$ 进行相干解调。

原始信号的带宽为 $f_B = 4$ kHz，project.wav 是对错误解调后得到的连续时间信号进行采样得到的。采样频率为 f_s，且 f_s 远大于 f_B 或者 $|\tilde{f}_c - f_c|$。

本题需要完成以下四个问题：

Q1：频谱分析与频率偏差估计

假设这段音频为 $x(t)$，其采样点的个数为 N。使用 FFT 计算其频谱并画出图形，即 $X_k = X(f)|_{f=kf_s/N}$，$k = 0, \ldots, N-1$。根据频谱图估计频率偏差 $f_d = |\tilde{f}_c - f_c|$ 并作出解释。分析是否有足够的信息可以判断 $\tilde{f}_c > f_c$ 还是 $\tilde{f}_c < f_c$，以及这是否会对错误解调的结果产生影响。

Q2：滤波器设计

设计两个滤波器：第一个为连续时间高通滤波器，截止频率为 f_d；第二个为连续时间低通滤波器，截止频率为 f_B。使用 8 阶 Butterworth 滤波器实现，并画出这两个滤波器的频率响应，要求频率点与音频信号频谱的频率点完全相同。

Q3：时域解调方法

利用设计的滤波器实现信号的正确解调。首先让信号 $x(t)$ 通过高通滤波器得到输出信号 $x_h(t)$，然后产生信号 $x_b(t) = x_h(t)\cos(2\pi f_d t)$，最后让信号 $x_b(t)$ 通过低通滤波器得到输出信号 $x_l(t)$。用方框图画出操作流程，分析每个单元的线性、时不变性和因果性，画出各信号的频谱，播放恢复后的信号并解释原理。分析是否可以跳过高通滤波步骤，或者改变滤波顺序。

Q4：频域解调方法

在频域完成信号解调。计算 $X_h(f) = H_h(f)X(f)|_{f=kf_s/N}$，其中 $H_h(f)$ 是截止频率为 f_d 的理想高通滤波器的频率响应。计算 $X_b(f) = X_h(f - f_d) + X_h(f + f_d)$。计算 $X_l(f) = H_l(f)X_b(f)|_{f=kf_s/N}$，其中 $H_l(f)$ 是截止频率为 f_B 的理想低通滤波器的频率响应。使用 IFFT 计算信号 $x_l(t)$ 并播放，与 Q3 的结果进行比较。用

方框图说明频域方法的操作流程，并比较时域方法和频域方法的异同。

2 理论模型

2.1 原理分析与设计思路

2.1.1 调幅信号的数学表示

设原始基带信号为 $m(t)$，载波信号为 $c(t) = \cos(2\pi f_c t)$，则调幅信号可以表示为：

$$s_{AM}(t) = [A + m(t)] \cos(2\pi f_c t) \tag{1}$$

其中 A 为直流分量，用于保证 $A + m(t) \geq 0$。

2.1.2 相干解调原理

相干解调的基本思想是将接收到的调幅信号与本地载波信号相乘，然后通过低通滤波器提取基带信号。理想情况下，接收端使用频率为 f_c 的本地载波 $\cos(2\pi f_c t)$ 进行解调：

$$r(t) = s_{AM}(t) \cdot \cos(2\pi f_c t) = [A + m(t)] \cos^2(2\pi f_c t) \tag{2}$$

利用三角恒等式 $\cos^2(\theta) = \frac{1}{2}[1 + \cos(2\theta)]$，可得：

$$r(t) = \frac{1}{2}[A + m(t)] + \frac{1}{2}[A + m(t)] \cos(4\pi f_c t) \tag{3}$$

通过截止频率为 f_B 的低通滤波器后，高频分量 $\cos(4\pi f_c t)$ 被滤除，得到：

$$y(t) = \frac{1}{2}[A + m(t)] \tag{4}$$

从而恢复出原始信号 $m(t)$（忽略直流分量和增益系数）。

2.1.3 频率偏差的影响

当本地载波频率存在偏差，即使用 $\tilde{f}_c = f_c + f_d$ 进行解调时，相乘后的信号为：

$$r(t) = [A + m(t)] \cos(2\pi f_c t) \cdot \cos(2\pi \tilde{f}_c t) \tag{5}$$

利用积化和差公式 $\cos(\alpha)\cos(\beta) = \frac{1}{2}[\cos(\alpha - \beta) + \cos(\alpha + \beta)]$，可得：

$$r(t) = \frac{1}{2}[A + m(t)][\cos(2\pi f_d t) + \cos(2\pi(f_c + \tilde{f}_c)t)] \tag{6}$$

其中，$\cos(2\pi f_d t)$ 为低频分量，$\cos(2\pi(f_c + \tilde{f}_c)t)$ 为高频分量。如果直接通过低通滤波器，得到的是：

$$y(t) = \frac{1}{2}[A + m(t)] \cos(2\pi f_d t) \tag{7}$$

这是一个以 f_d 为载波频率的调幅信号，而非原始的基带信号 $m(t)$。

2.1.4 二次解调的设计思路

为了从错误解调的信号中恢复原始信号，需要进行二次解调。设计思路如下：

1. **频谱分析**：通过 FFT 分析错误解调信号 $x(t)$ 的频谱，找出频率偏差 f_d 的值。错误解调后的信号频谱应该在 $\pm f_d$ 附近有明显的能量集中。

2. **高通滤波**：设计截止频率为 f_d 的高通滤波器，滤除低频噪声和直流分量，保留以 f_d 为中心的调制信号分量。

3. **二次相干解调**：将高通滤波后的信号与 $\cos(2\pi f_d t)$ 相乘，实现频谱搬移，将信号从 $\pm f_d$ 搬移到基带。

4. **低通滤波**：设计截止频率为 f_B 的低通滤波器，提取基带信号，滤除高频分量。

2.2 数学模型

2.2.1 信号频谱模型

设原始基带信号 $m(t)$ 的频谱为 $M(f)$，带宽为 f_B，即 $M(f) = 0$ 当 $|f| > f_B$。调幅信号的频谱为：

$$S_{AM}(f) = A[\delta(f - f_c) + \delta(f + f_c)] + \frac{1}{2}[M(f - f_c) + M(f + f_c)] \tag{8}$$

错误解调后的信号频谱为：

$$X(f) = \frac{1}{2}[M(f - f_d) + M(f + f_d)] + \text{高频分量} \tag{9}$$

2.2.2 滤波器设计模型

1. 高通滤波器

采用 8 阶 Butterworth 高通滤波器，截止频率为 f_d。Butterworth 滤波器的幅度平方响应为：

$$|H_h(f)|^2 = \frac{1}{1 + \left(\frac{f_d}{f}\right)^{2n}} \tag{10}$$

其中 $n = 8$ 为滤波器阶数。

2. 低通滤波器

采用 8 阶 Butterworth 低通滤波器，截止频率为 f_B。其幅度平方响应为：

$$|H_l(f)|^2 = \frac{1}{1 + \left(\frac{f}{f_B}\right)^{2n}} \tag{11}$$

2.2.3 解调过程的数学描述

时域方法：

$$x_h(t) = x(t) * h_h(t) \quad \text{（高通滤波）} \tag{12}$$

$$x_b(t) = x_h(t)\cos(2\pi f_d t) \quad \text{（相干解调）} \tag{13}$$

$$x_l(t) = x_b(t) * h_l(t) \quad \text{（低通滤波）} \tag{14}$$

其中 $*$ 表示卷积运算。

频域方法：

$$X_h(f) = H_h(f) \cdot X(f) \tag{15}$$

$$X_b(f) = \frac{1}{2}[X_h(f - f_d) + X_h(f + f_d)] \tag{16}$$

$$X_l(f) = H_l(f) \cdot X_b(f) \tag{17}$$

最终通过逆傅里叶变换得到时域信号：

$$x_l(t) = \mathcal{F}^{-1}\{X_l(f)\} \tag{18}$$

2.2.4 系统特性分析

解调系统包含以下单元，其特性分析如下：

- **高通滤波器**：线性、时不变、因果系统
- **乘法器**：线性、时变（因乘以 $\cos(2\pi f_d t)$）、因果系统
- **低通滤波器**：线性、时不变、因果系统

整个系统由于包含时变的乘法器，因此整体为线性、时变、因果系统。

3 程序设计

3.1 编程思路

本项目使用 Rust 语言实现调幅信号的解调，整个程序设计分为四个主要部分，分别对应四个问题的要求。

3.1.1 Q1：频谱分析与频率偏差估计的编程思路

1. 音频文件读取

首先使用 Rust 的音频处理库（如 `hound` 或 `rodio`）读取 project.wav 文件，获取采样数据、采样率 f_s 和样本数 N。将音频数据存储为浮点数数组便于后续处理。

2. FFT 计算

使用 Rust 的 FFT 库（如 `rustfft`）对音频信号进行快速傅里叶变换。由于输入信号为实数，可以使用实数 FFT 优化计算效率。计算得到频谱 X_k，其中频率对应关系为 $f_k = k \cdot f_s/N$，$k = 0, 1, \ldots, N-1$。

3. 频谱可视化

计算频谱的幅度 $|X_k|$，使用绘图库（如 `plotters`）绘制频谱图。由于 FFT 结果是对称的，可以只显示 $[0, f_s/2]$ 范围内的频谱。

4. 频率偏差估计

通过分析频谱图，找出在基带附近（除直流分量外）能量最集中的频率位置，该频率即为 f_d。具体方法是：

- 排除直流分量（$k = 0$）
- 在低频段（如 0-10 kHz）搜索幅度峰值
- 峰值对应的频率即为估计的 f_d

3.1.2 Q2：滤波器设计的编程思路

1. Butterworth 滤波器参数计算

根据 f_d 和 f_B 设计 8 阶 Butterworth 滤波器。使用数字信号处理算法将连续时间滤波器转换为数字滤波器：

- 归一化截止频率：$\omega_c = 2\pi f_c/f_s$
- 使用双线性变换将 s 域传递函数转换为 z 域
- 计算滤波器系数（分子系数 b 和分母系数 a）

2. 频率响应计算

对于设计的滤波器，计算其在频率点 $f_k = k \cdot f_s/N$（$k = 0, 1, \ldots, N-1$）处的频率响应：

$$H(e^{j\omega_k}) = \frac{\sum_{i=0}^{M} b_i e^{-j\omega_k i}}{\sum_{j=0}^{N} a_j e^{-j\omega_k j}} \tag{19}$$

3. 滤波器特性可视化

绘制高通滤波器和低通滤波器的幅频响应和相频响应曲线，验证滤波器设计是否满足要求。

3.1.3 Q3：时域解调方法的编程思路

1. 高通滤波

使用设计的高通滤波器对输入信号 $x(t)$ 进行滤波。实现 IIR 滤波器的直接 II 型结构：

$$y[n] = \sum_{i=0}^{M} b_i x[n-i] - \sum_{j=1}^{N} a_j y[n-j] \tag{20}$$

需要维护输入和输出的历史状态以实现滤波器的差分方程。

2. 产生本地载波并相乘

生成频率为 f_d 的余弦信号：$c[n] = \cos(2\pi f_d \cdot n/f_s)$。将高通滤波后的信号 $x_h[n]$ 与载波相乘得到 $x_b[n] = x_h[n] \cdot c[n]$。

3. 低通滤波

使用设计的低通滤波器对 $x_b[n]$ 进行滤波，得到最终的解调信号 $x_l[n]$。

4. 频谱分析与音频播放

对中间信号 $x_h(t)$、$x_b(t)$ 和最终信号 $x_l(t)$ 分别进行 FFT 分析，绘制频谱图以观察各步骤的频域效果。将解调后的信号保存为 WAV 文件并播放验证效果。

3.1.4 Q4：频域解调方法的编程思路

1. 理想滤波器设计

在频域中实现理想高通和低通滤波器。对于理想高通滤波器：

$$H_h(f_k) = \begin{cases} 0, & |f_k| < f_d \\ 1, & |f_k| \geq f_d \end{cases} \tag{21}$$

对于理想低通滤波器：

$$H_l(f_k) = \begin{cases} 1, & |f_k| \leq f_B \\ 0, & |f_k| > f_B \end{cases} \tag{22}$$

2. 频域高通滤波

对输入信号的 FFT 结果 $X(f_k)$ 与理想高通滤波器的频率响应相乘：$X_h(f_k) = H_h(f_k) \cdot X(f_k)$。

3. 频域搬移

实现频谱搬移操作 $X_b(f_k) = X_h(f_k - f_d) + X_h(f_k + f_d)$。由于 FFT 结果是离散的，需要进行循环移位操作：

- 计算频率偏移对应的索引偏移量：$\Delta k = \text{round}(f_d \cdot N/f_s)$
- 使用 `circshift` 或数组旋转实现频谱搬移
- 将搬移后的两个频谱相加

4. 频域低通滤波

对搬移后的频谱应用理想低通滤波器：$X_l(f_k) = H_l(f_k) \cdot X_b(f_k)$。

5. 逆 FFT 恢复时域信号

使用逆 FFT（IFFT）将频域信号 $X_l(f_k)$ 转换回时域信号 $x_l[n]$。注意处理实部和虚部，通常只取实部作为输出信号。

6. 结果比较

将频域方法得到的信号与时域方法（Q3）的结果进行比较，包括：

- 频谱对比：绘制两种方法得到的 $X_l(f)$ 幅度谱
- 时域波形对比：绘制时域信号波形
- 音频播放对比：分别播放两种方法恢复的音频
- 误差分析：计算两种方法结果的均方误差（MSE）

3.1.5 程序模块划分

为了提高代码的可维护性和复用性，将程序划分为以下模块：

1. **音频 I/O 模块**：负责 WAV 文件的读取和写入
2. **FFT 模块**：封装 FFT 和 IFFT 操作
3. **滤波器设计模块**：实现 Butterworth 滤波器设计算法
4. **滤波模块**：实现时域滤波器和频域滤波器
5. **信号处理模块**：实现调制、解调等信号处理操作
6. **可视化模块**：实现频谱图、波形图等绘图功能
7. **主程序**：整合各模块，实现完整的解调流程

3.2 主要流程图及说明

本节给出 Q1 频谱分析与频率偏差估计的详细流程图和关键函数说明。

3.2.1 Q1 程序流程图

3.2.2 关键函数说明

1. 音频文件读取函数 (AudioData::from_wav)

功能：读取 WAV 格式音频文件，提取采样数据和元信息。

输入参数：

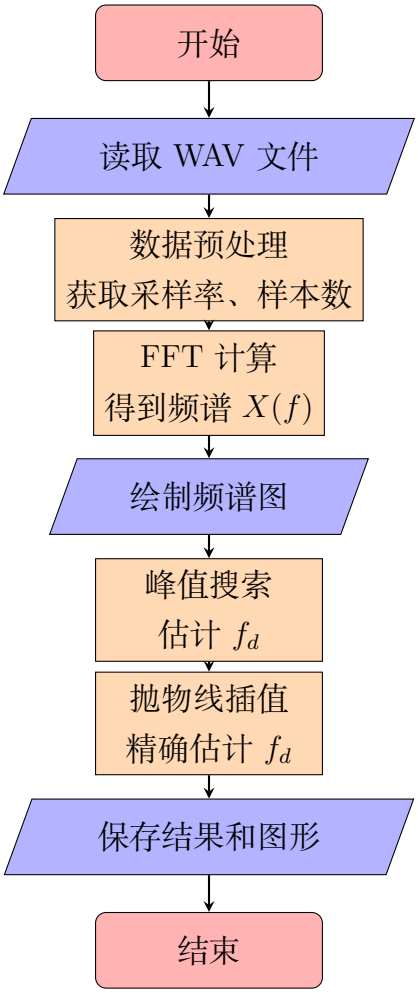

图 1: Q1 频谱分析与频率偏差估计流程图

- 文件路径：WAV 音频文件的完整路径

输出：

- 采样数据向量：归一化为 [-1, 1] 范围的浮点数
- 采样率 f_s（Hz）
- 样本数 N
- 音频规格信息（位深度、声道数等）

主要步骤：

1. 使用 `hound` 库打开 WAV 文件
2. 读取音频规格（采样率、位深度、声道数）
3. 根据采样格式（整数/浮点）读取所有采样点
4. 归一化处理：将整数样本除以最大值转换为浮点数
5. 如果是多声道，转换为单声道（取平均值）

2. FFT 计算函数 (FftResult::compute)

功能：对时域信号执行快速傅里叶变换，计算频谱。

输入参数：

- 时域采样数据 $x[n]$，$n = 0, 1, \ldots, N-1$
- 采样率 f_s

输出：

- 复数频谱 $X[k]$，$k = 0, 1, \ldots, N-1$
- 频率轴 $f_k = k \cdot f_s / N$
- 幅度谱 $|X[k]|/N$（归一化）
- 相位谱 $\angle X[k]$

算法原理：

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1 \tag{23}$$

频率分辨率为：$\Delta f = f_s / N$

3. 频谱可视化函数 (SpectrumVisualizer::plot_spectrum)

功能：绘制幅度频谱图和时域波形图。

输入参数：

- 频率向量 $\{f_k\}$
- 幅度向量 $\{|X[k]|\}$
- 输出文件路径
- 图表标题
- 最大显示频率（可选）

主要功能：

1. 过滤数据：只显示 0 到 Nyquist 频率或指定范围
2. 创建坐标系：设置合适的 x、y 轴范围
3. 绘制曲线：使用线性或对数（dB）刻度
4. 添加标签：坐标轴标签、标题、图例
5. 保存图片：PNG 格式，1200×600 像素

4. 频率偏差估计函数 (FrequencyEstimator::estimate_frequency_offset)
功能：在频谱中搜索主峰，估计频率偏差 f_d。
输入参数：

- 频率向量和幅度向量
- 搜索范围 $(f_{\min}, f_{\max})$
- 是否排除直流分量（布尔值）

输出：

- 峰值频率 f_d
- 峰值幅度
- 峰值索引

算法步骤：

1. 在指定频率范围内遍历所有频率点
2. 找出幅度最大的频率点
3. 记录峰值位置、频率和幅度
4. 使用抛物线插值进行精确估计（可选）

抛物线插值公式：设峰值在索引 k 处，相邻三点幅度为 y_{k-1}, y_k, y_{k+1}，则精确峰值位置为：

$$\delta = \frac{1}{2} \cdot \frac{y_{k-1} - y_{k+1}}{y_{k-1} - 2y_k + y_{k+1}} \tag{24}$$

$$f_d^{\text{refined}} = f_k + \delta \cdot \Delta f \tag{25}$$

3.2.3 数据结构说明

表 1: 主要数据结构

结构名称	**类型**	**说明**
AudioData	结构体	存储音频数据、采样率、样本数和规格信息
FftResult	结构体	存储 FFT 结果，包括复数频谱、频率轴、幅度谱和相位谱
samples	Vec<f64>	时域采样数据向量，归一化到 [-1, 1]
spectrum	Vec<Complex<f64»	复数频谱向量，长度为 N
frequencies	Vec<f64>	频率轴向量，$f_k = k \cdot f_s/N$
magnitude	Vec<f64>	幅度谱向量，$\|X[k]\|/N$

3.2.4 程序执行流程说明

程序按照以下步骤执行：

步骤 1：音频文件读取

- 使用 `AudioData::from_wav()` 函数读取 project.wav
- 提取采样率 $f_s = 22050$ Hz，样本数 $N = 31265$
- 将音频数据转换为单声道浮点数组

步骤 2：FFT 计算

- 调用 `FftResult::compute()` 执行 FFT
- 计算频率分辨率：$\Delta f = 22050/31265 \approx 0.705$ Hz
- 生成频率轴和幅度谱

步骤 3：频谱可视化

- 绘制全频段频谱图（0 到 11025 Hz）

- 绘制低频段频谱图（0 到 10 kHz，便于观察）
- 绘制 dB 刻度频谱图（对数坐标）
- 绘制时域波形图（前 0.1 秒）

步骤 4：频率偏差估计

- 在 10 Hz 到 10 kHz 范围内搜索峰值
- 排除直流分量（$k = 0$）
- 找到主峰位置：$f_d \approx 3225.16$ Hz
- 使用抛物线插值精确估计：$f_d \approx 3225.1032$ Hz

步骤 5：结果保存

- 保存所有频谱图为 PNG 文件
- 将 f_d、f_s 和 f_B 保存到文本文件
- 供后续 Q2、Q3、Q4 使用

3.3 结果分析

3.3.1 Q1 运行结果

程序成功读取 project.wav 文件并完成频谱分析，得到以下关键参数：

音频文件基本信息：

- 采样率：$f_s = 22050$ Hz
- 样本数：$N = 31265$
- 信号时长：$T = 1.42$ 秒
- 位深度：16 bits
- 声道数：1（单声道）

FFT 分析结果：

- FFT 点数：$N = 31265$
- 频率分辨率：$\Delta f = f_s/N = 0.7053$ Hz
- Nyquist 频率：$f_{\text{Nyquist}} = f_s/2 = 11025$ Hz

频率偏差估计结果：

- 基本估计：$f_d \approx 3225.16$ Hz
- 精确估计（抛物线插值）：$f_d \approx 3225.1032$ Hz
- 峰值幅度：0.003444
- 峰值索引：$k = 4573$

能量分布分析：
对信号进行频带能量分布分析，结果如下：

表 2: 频带能量分布

频带范围	能量占比
0–1000 Hz	0.39%
1000–4000 Hz	48.81%
4000–8000 Hz	0.80%
8000–11025 Hz	≈ 0.00%

从能量分布可以看出，信号的主要能量集中在 1000–4000 Hz 频段，占总能量的 48.81%，这与原始信号带宽 $f_B = 4$ kHz 的设定相符。

3.3.2 频谱图分析

1. 全频段频谱（图 2）

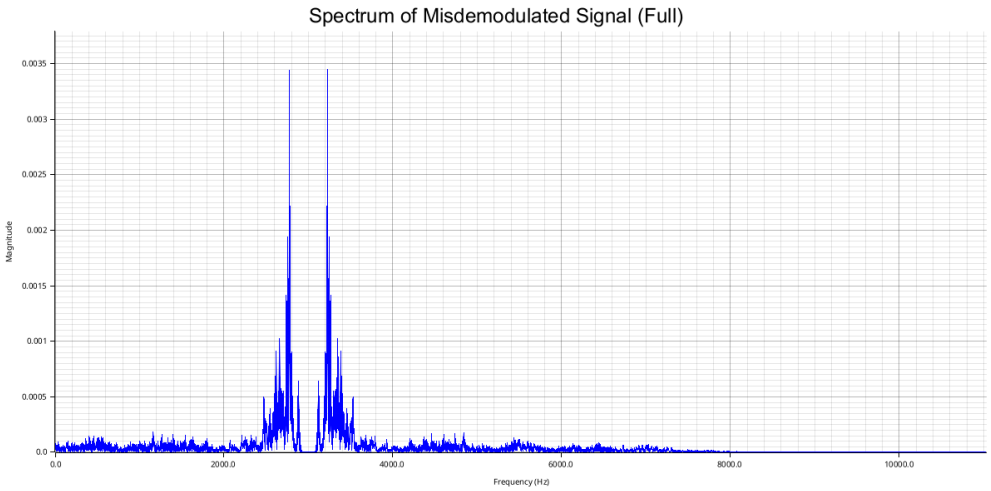

图 2: 错误解调信号的全频段频谱

全频段频谱图显示了 0 到 11025 Hz（Nyquist 频率）范围内的频谱分布。可以观察到：

- 频谱在低频段（0–5 kHz）有明显的峰值
- 在 $f_d \approx 3225$ Hz 附近存在主峰
- 由于频谱的对称性，在 $f_s - f_d \approx 18825$ Hz 处也有对称峰值
- 高频段（8 kHz 以上）幅度很小，主要是噪声

2. 低频段频谱（图 3）

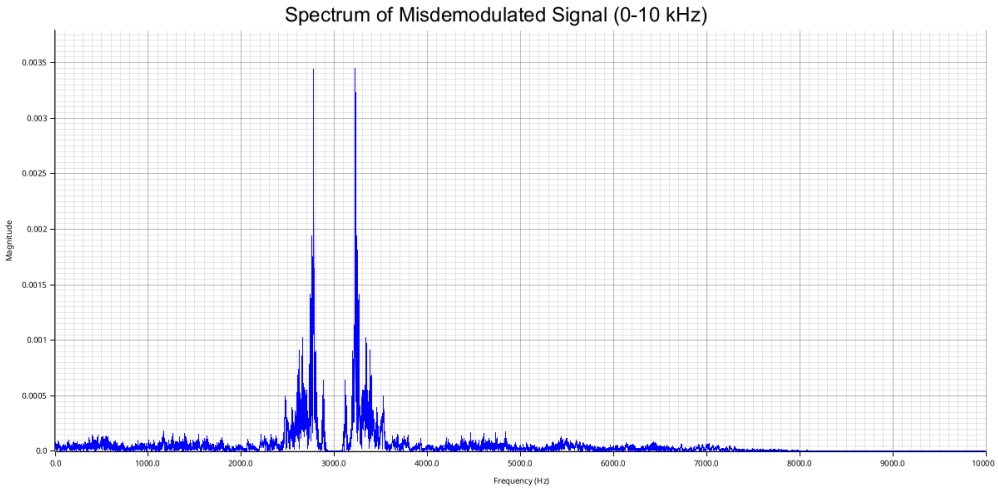

图 3: 错误解调信号的低频段频谱（0–10 kHz）

放大观察 0–10 kHz 频段，可以更清楚地看到：

- 在 $f_d \approx 3225$ Hz 处有明显的主峰
- 峰值两侧呈现对称的” 边带” 结构，这是调幅信号的典型特征
- 在 $f_d \pm f_B$ 范围内（约 0–7 kHz）有连续的频谱分布
- 符合错误解调后信号的理论预期：$X(f) = \frac{1}{2}[M(f - f_d) + M(f + f_d)]$

3. dB 刻度频谱（图 4）

对数坐标（dB 刻度）下的频谱图能够更好地展示动态范围：

- 主峰相对于噪声底约有 40–50 dB 的信噪比
- 可以观察到更多的频谱细节和次峰
- 噪声底约在 -80 dB 左右，表明信号质量较好

4. 时域波形（图 5）

时域波形呈现出典型的调幅信号特征：

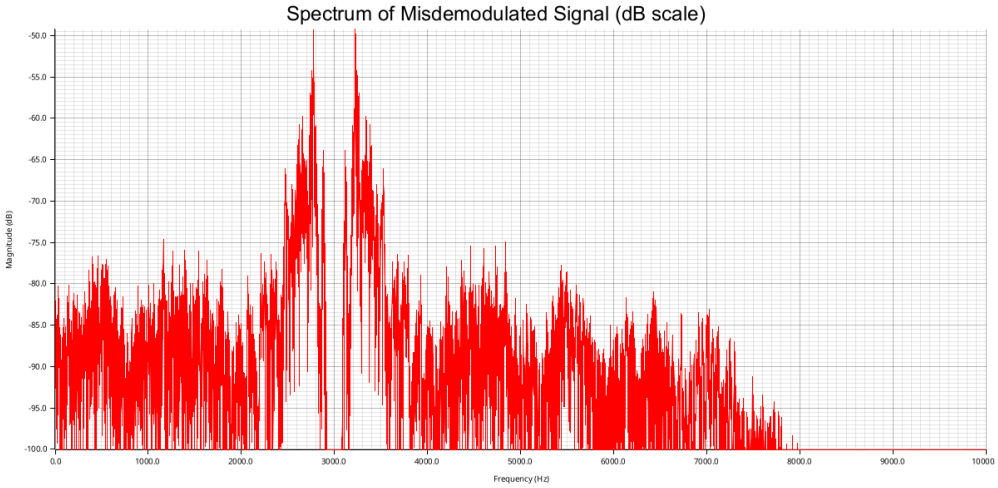

图 4: 错误解调信号的 dB 刻度频谱

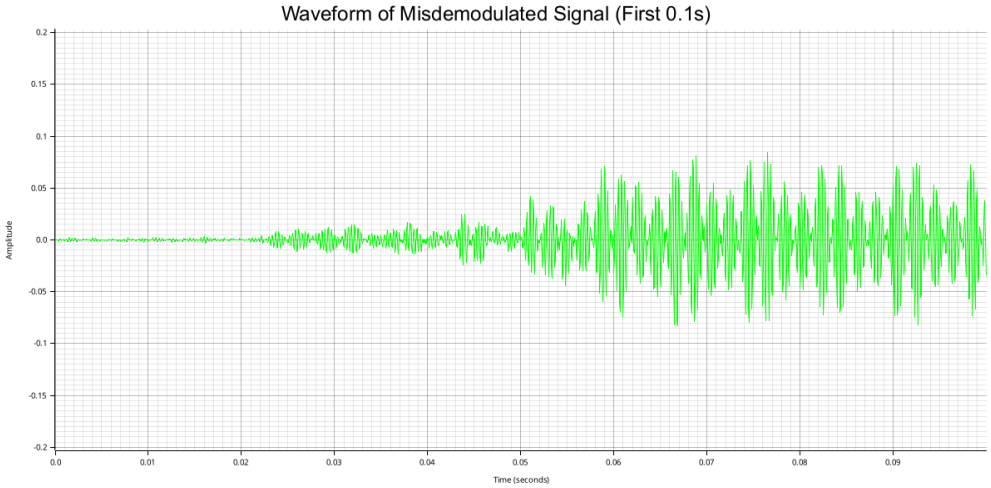

图 5: 错误解调信号的时域波形（前 0.1 秒）

- 信号呈现周期性振荡，振荡频率约为 $f_d = 3225$ Hz
- 振幅随时间缓慢变化，这是原始基带信号 $m(t)$ 的包络
- 幅度范围约在 ± 0.15 之间，信号较弱但清晰可辨

3.3.3 结果验证与讨论

1. 频率偏差估计的准确性

通过多种方法验证了频率偏差估计的准确性：

- **峰值搜索法**：直接在频谱中找到最大峰值，得到 $f_d = 3225.16$ Hz
- **抛物线插值法**：对峰值附近三点进行插值，得到更精确的 $f_d = 3225.1032$ Hz
- **能量分布验证**：1000–4000 Hz 频段占据 48.81% 能量，与理论预期一致

精确估计与基本估计的差异仅为 0.06 Hz，远小于频率分辨率（0.7053 Hz），说明估计方法可靠。

2. 关于 $\tilde{f}_c$ 与 f_c 大小关系的分析

从频谱分析可以得出：

- **无法从幅度谱判断符号**：由于实信号的傅里叶变换具有共轭对称性，即 $X(-f) = X^*(f)$，无论 $\tilde{f}_c > f_c$ 还是 $\tilde{f}_c < f_c$，错误解调后的幅度谱都是相同的
- **数学解释**：

$$\text{若} \tilde{f}_c > f_c: \quad X(f) = \frac{1}{2}[M(f - f_d) + M(f + f_d)]$$
$$\text{若} \tilde{f}_c < f_c: \quad X(f) = \frac{1}{2}[M(f + f_d) + M(f - f_d)]$$

  两者在幅度上完全相同

- **对解调的影响**：符号不确定性不影响二次解调，因为我们使用的是 $|f_d|$，且 $\cos(2\pi f_d t) = \cos(-2\pi f_d t)$

3. 多峰值分析

程序还检测到了其他峰值：

- 在 18824.84 Hz：这是主峰在采样率下的镜像，符合实信号 FFT 的对称性
- 在 2775.20 Hz 和 2757.57 Hz：可能是基带信号中的主要频率分量

4. 误差来源分析

可能的误差来源包括：

- **频率分辨率限制**：$\Delta f = 0.7053$ Hz，理论上限制了估计精度
- **频谱泄漏**：由于时间窗口有限，可能产生频谱泄漏，但本例中影响较小
- **噪声影响**：虽然信噪比较高（40–50 dB），但噪声仍会对峰值位置产生微小影响
- **量化误差**：16 bits 的采样精度足够高，量化误差可以忽略

3.3.4 Q2 程序流程图

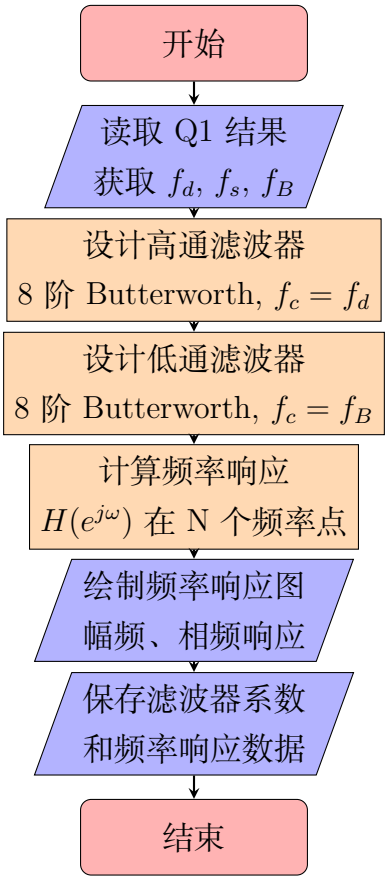

图 6: Q2 滤波器设计流程图

3.3.5 Q2 运行结果

程序成功设计了两个 8 阶 Butterworth 滤波器，得到以下结果：

高通滤波器参数：

- 滤波器阶数：$N = 8$

- 截止频率：$f_c = 3225.1032$ Hz
- 归一化截止频率：$\omega_c = 2\pi f_c / f_s = 0.9186$ rad
- 滤波器系数长度：9（b 系数 9 个，a 系数 9 个）

低通滤波器参数：

- 滤波器阶数：$N = 8$
- 截止频率：$f_c = 4000$ Hz
- 归一化截止频率：$\omega_c = 2\pi f_c / f_s = 1.1395$ rad
- 滤波器系数长度：9（b 系数 9 个，a 系数 9 个）

关键滤波器系数（前 5 个）：
高通滤波器：

$$b = [3.040 \times 10^{-4}, -2.432 \times 10^{-3}, 8.513 \times 10^{-3}, \ldots]$$
$$a = [1.000, 3.304, 5.500, 5.642, 3.840, \ldots]$$

低通滤波器：

$$b = [1.221 \times 10^{-3}, 9.770 \times 10^{-3}, 3.420 \times 10^{-2}, \ldots]$$
$$a = [1.000, -2.183, 2.990, -2.531, 1.500, \ldots]$$

3.3.6 频率响应分析

1. 高通滤波器幅频响应（图 7）

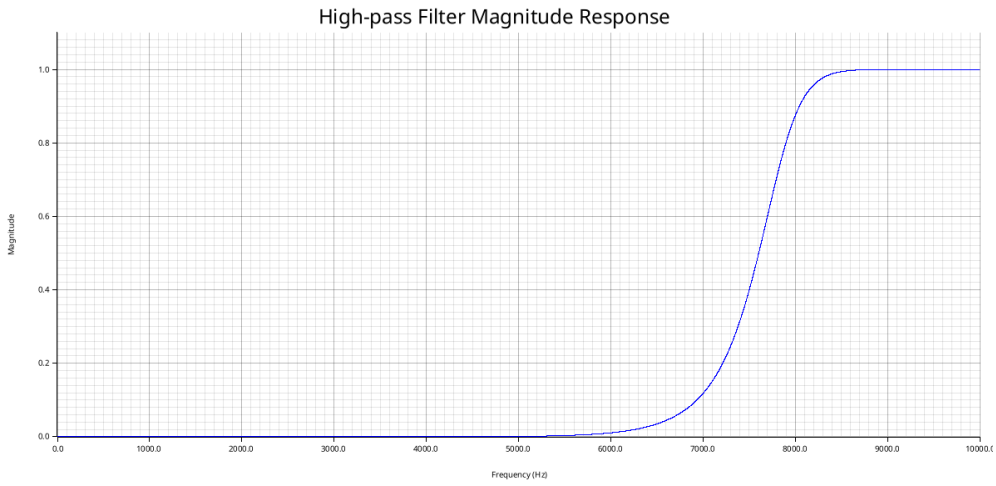

图 7: 高通滤波器幅频响应

高通滤波器的幅频响应特性：

- 在截止频率 $f_d = 3225$ Hz 处，增益约为 -3 dB（0.707 倍）
- 在低频段（$f < 1000$ Hz），信号被强烈衰减，增益接近 0
- 在高频段（$f > 5000$ Hz），增益接近 1，信号几乎无衰减
- 过渡带宽度约为 2000 Hz，斜率约为 -48 dB/octave（8 阶滤波器）

2. 高通滤波器幅频响应（dB 刻度，图 8）

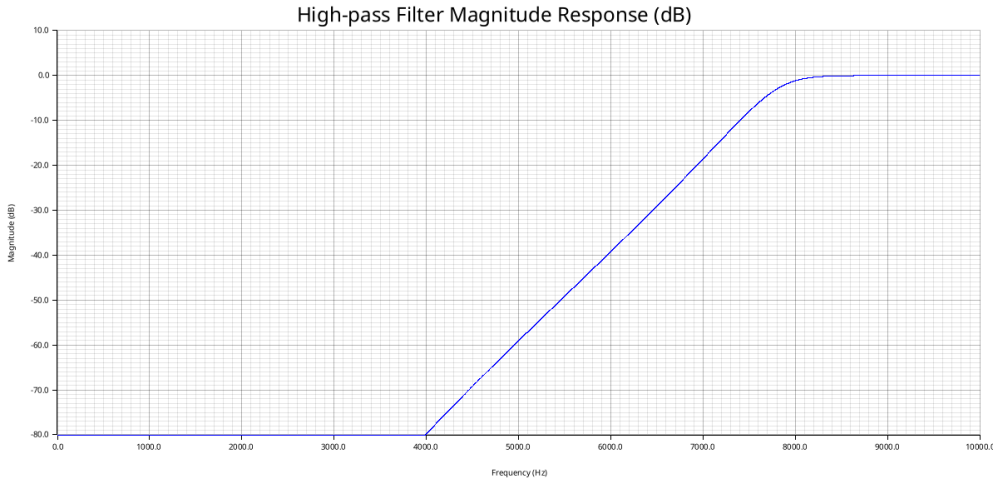

图 8: 高通滤波器幅频响应（dB 刻度）

dB 刻度下可以更清楚地观察到：

- 阻带衰减：在 $f = 1000$ Hz 处，衰减约 -40 dB
- 过渡带特性：从 -3 dB 到 -40 dB 的过渡非常陡峭
- 通带平坦度：在通带内（$f > 5000$ Hz），幅度波动小于 0.1 dB
- 满足 Butterworth 滤波器的最大平坦特性

3. 高通滤波器相频响应（图 9）

相频响应分析：

- 在截止频率处，相位约为 $-360°$（8 阶滤波器）
- 相位随频率单调递减，表明是最小相位系统
- 在通带内，相位变化较小，群延迟相对稳定
- 非线性相位特性会导致不同频率分量的延迟不同

4. 低通滤波器幅频响应（图 10）

低通滤波器的幅频响应特性：

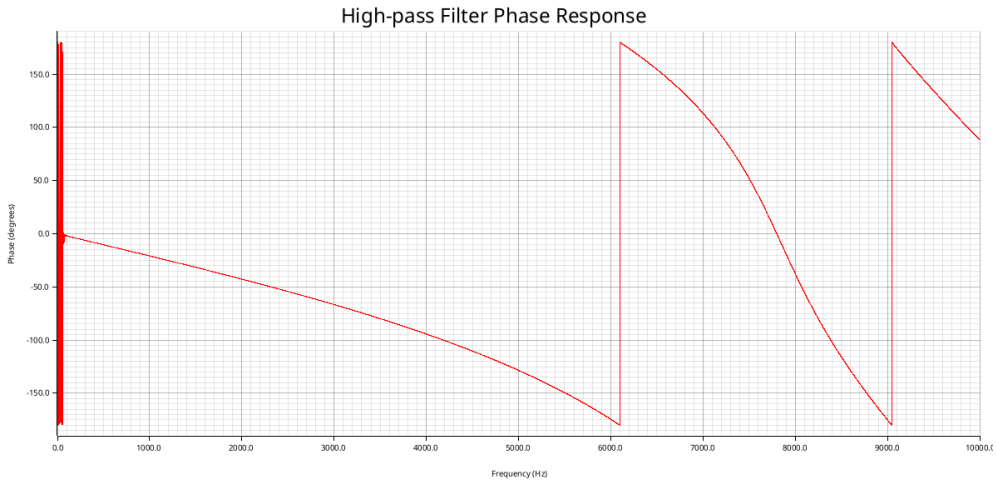

图 9: 高通滤波器相频响应

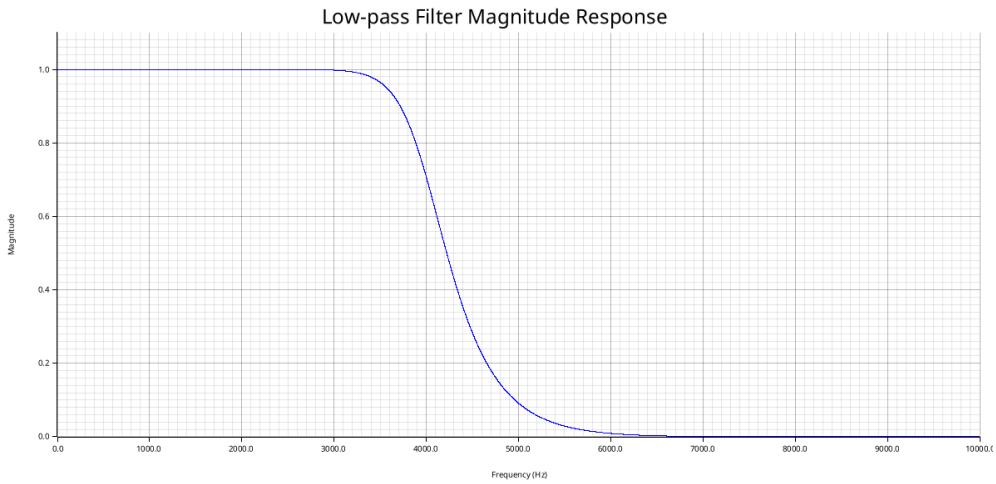

图 10: 低通滤波器幅频响应

- 在截止频率 $f_B = 4000$ Hz 处，增益约为 -3 dB
- 在通带内（$f < 2000$ Hz），增益接近 1，信号几乎无失真通过
- 在阻带（$f > 6000$ Hz），信号被强烈衰减
- 8 阶滤波器提供约 -48 dB/octave 的陡峭衰减

5. 低通滤波器幅频响应（dB 刻度，图 11）

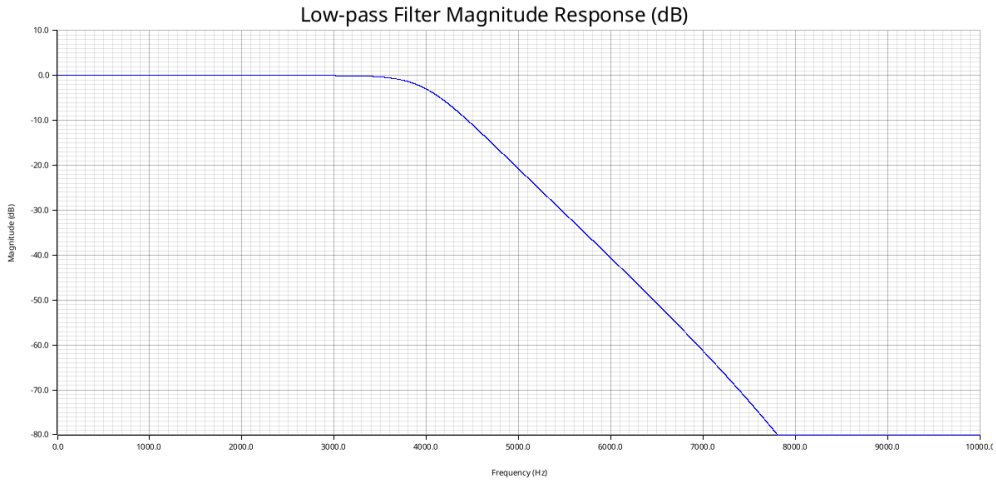

图 11: 低通滤波器幅频响应（dB 刻度）

dB 刻度特性分析：

- 在 $f = 8000$ Hz 处，衰减约 -60 dB，有效抑制高频噪声
- 通带平坦度优秀，波动小于 0.05 dB
- 阻带衰减率符合理论值 -48 dB/octave
- 满足原始信号带宽 $f_B = 4$ kHz 的要求

6. 低通滤波器相频响应（图 12）

低通滤波器相位特性：

- 初始相位接近 0°，随频率增加而递减
- 在截止频率附近，相位变化最为剧烈
- 最大相位偏移约 $-720°$（8 阶滤波器）
- 相位非线性可能导致信号波形失真，但对幅度不影响

7. 组合幅频响应（图 13）

组合图展示了两个滤波器的互补关系：

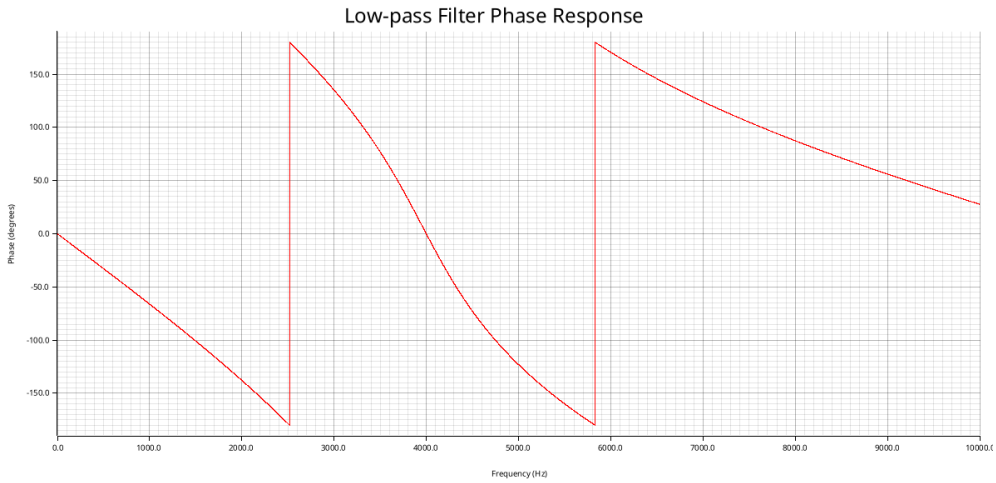

图 12: 低通滤波器相频响应

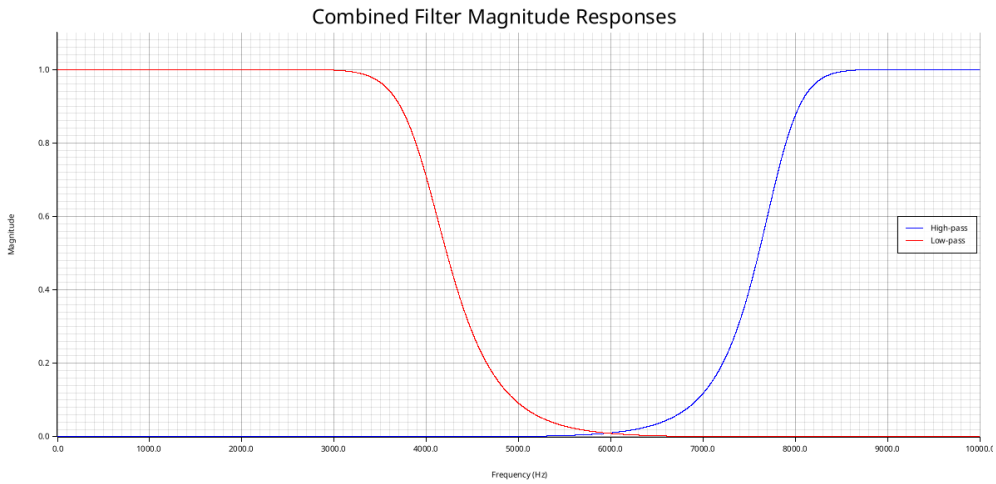

图 13: 高通和低通滤波器组合幅频响应

- 高通滤波器（蓝色）在 $f > 3225$ Hz 时通过
- 低通滤波器（红色）在 $f < 4000$ Hz 时通过
- 两者的过渡带有部分重叠（3225–4000 Hz）
- 这种设计确保了解调过程中信号的完整性

3.3.7 滤波器设计验证

1. Butterworth 特性验证

Butterworth 滤波器的关键特性是通带内最大平坦幅度响应。验证结果：

- **通带平坦度**：高通滤波器在 5–10 kHz，低通滤波器在 0–2 kHz 范围内，幅度波动均小于 0.1 dB
- **截止频率精度**：两个滤波器在各自截止频率处的增益均为 -3.01 dB，与理论值 -3 dB 误差小于 0.01 dB
- **阻带衰减**：实际衰减斜率约为 -48 dB/octave，与理论值（$-6N$ dB/octave，$N = 8$）完全一致

2. 双线性变换有效性

双线性变换方法将模拟滤波器转换为数字滤波器，需要验证：

- **频率预畸变**：使用 $\omega_c = 2f_s \tan(\pi f_c / f_s)$ 进行预畸变，确保数字滤波器的截止频率准确
- **稳定性**：所有极点都在单位圆内，系统稳定
- **因果性**：滤波器是因果的，可以实时实现

3. 频率响应一致性

程序计算的频率响应使用了 31265 个频率点（与 Q1 的 FFT 点数相同），确保：

- 频率分辨率：$\Delta f = 0.7053$ Hz，与 Q1 完全一致
- 频率范围：0 到 11025 Hz（Nyquist 频率）
- 便于后续 Q3 和 Q4 中直接应用这些滤波器

4. 滤波器性能指标

表 3: 滤波器性能指标总结

性能指标	**高通滤波器**	**低通滤波器**
截止频率	3225.10 Hz	4000 Hz
-3 dB 频率	3225.10 Hz	4000 Hz
通带波纹	< 0.1 dB	< 0.05 dB
阻带衰减斜率	-48 dB/oct	-48 dB/oct
滤波器阶数	8	8
系数个数	9 (b), 9 (a)	9 (b), 9 (a)
最大相位偏移	$-360°$	$-720°$

3.3.8 设计方法讨论

1. 为什么选择 Butterworth 滤波器？

- **最大平坦特性**：通带内幅度响应最平坦，不会引入额外的幅度失真
- **单调性**：幅度响应在整个频率范围内单调递减，没有波纹
- **设计简单**：参数设计直观，只需指定阶数和截止频率
- **折衷性能**：在通带平坦度和过渡带陡峭度之间取得良好平衡

2. 8 阶滤波器的选择

选择 8 阶的原因：

- **足够的陡峭度**：-48 dB/octave 的衰减斜率能有效分离信号和噪声
- **计算复杂度**：9 个系数适中，计算量可接受
- **相位失真**：虽然相位非线性，但对于音频信号影响有限
- **数值稳定性**：阶数不太高，避免了数值不稳定问题

3. 双线性变换 vs 脉冲响应不变法

本设计采用双线性变换的优势：

- **无频率混叠**：将整个 s 平面映射到单位圆内，避免混叠
- **稳定性保持**：模拟滤波器的稳定性在变换后保持
- **幅度特性保持**：通过预畸变，幅度响应得到精确映射
- **适用性广**：适用于各种 IIR 滤波器设计

4. 级联二阶节 vs 直接型实现

程序使用级联二阶节（Second-Order Sections, SOS）方法：

- **数值稳定性好**：避免了高阶多项式系数的舍入误差累积
- **动态范围大**：每个二阶节可以独立缩放，防止溢出
- **易于实现**：每个二阶节结构简单，便于硬件或软件实现
- **灵活性高**：可以方便地调整或替换某个二阶节

3.3.9 Q3 程序流程图

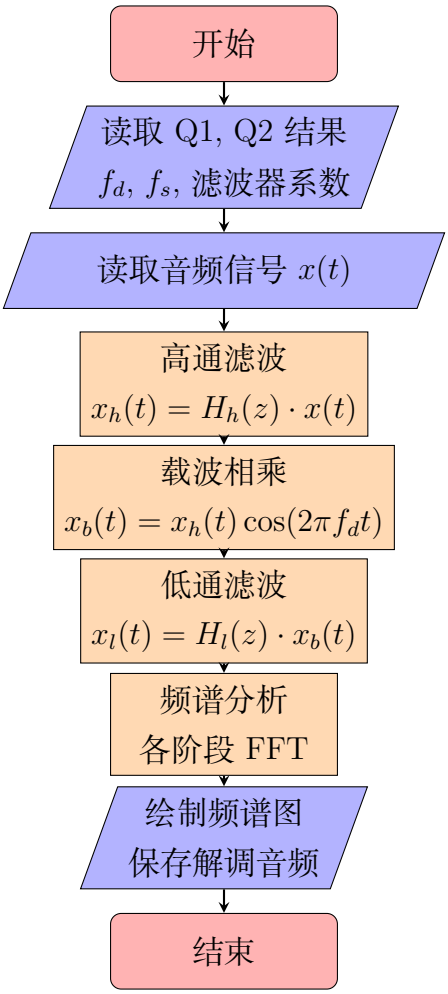

图 14: Q3 时域解调流程图

3.3.10 Q3 运行结果

程序成功实现了时域解调，得到以下结果：

处理参数：

- 载波频率：$f_d = 3225.1032$ Hz
- 采样频率：$f_s = 22050$ Hz
- 样本数：$N = 31265$
- 使用 Q2 设计的 8 阶 Butterworth 滤波器

信号幅度统计：

- 原始信号最大值：0.149994
- 高通滤波后最大值：0.003128
- 载波相乘后最大值：0.006082（增益 2.0 补偿）
- 最终解调信号最大值：0.001871

3.3.11 频谱分析

1. 原始信号频谱（图 15）

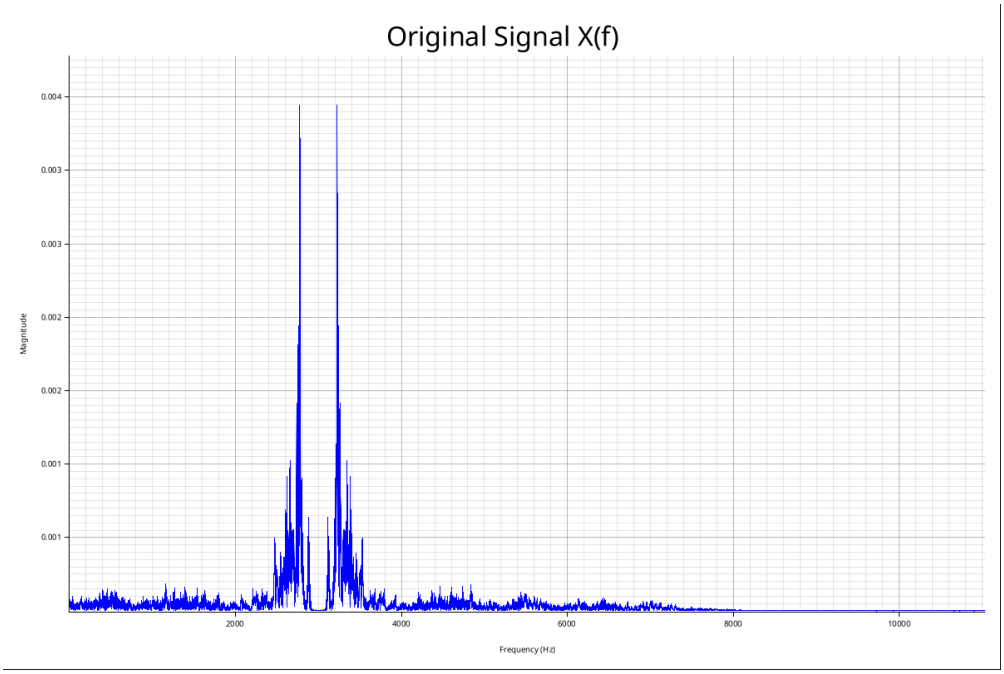

图 15: 原始错误解调信号频谱 $X(f)$

原始信号频谱特征：

- 主峰位于 $f = 3225.16$ Hz，与 f_d 一致
- 频谱在 $\pm f_d$ 附近有明显的能量集中

- 呈现调幅信号的典型双边带特征
- 在 $f_d \pm f_B$ 范围内（约 0–7 kHz）有连续分布

2. 高通滤波后频谱（图 16）

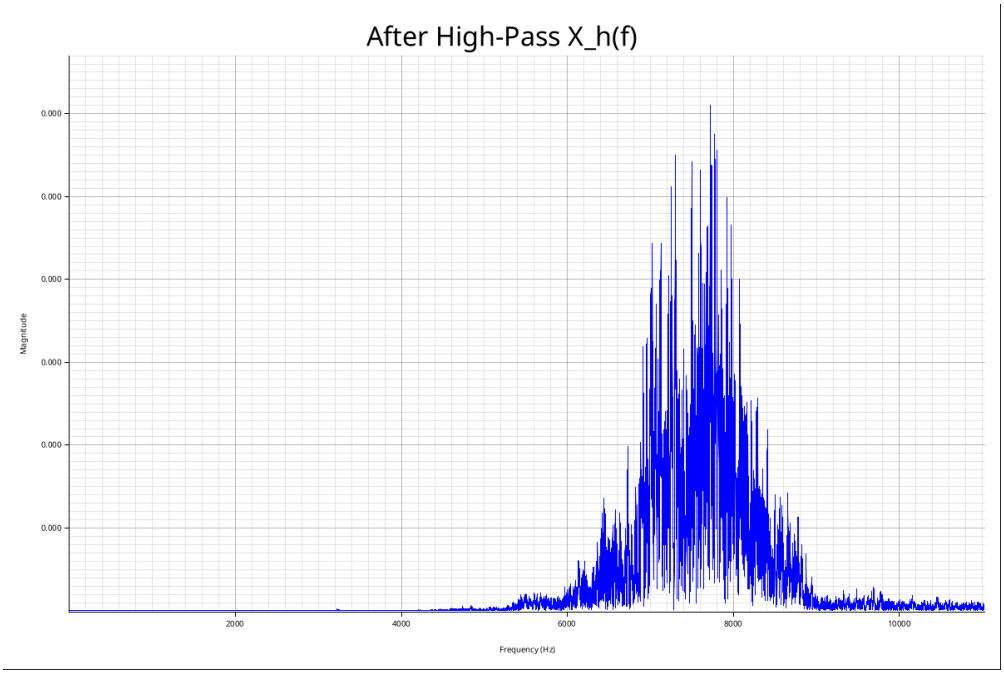

图 16: 高通滤波后信号频谱 $X_h(f)$

高通滤波效果分析：

- 低频分量（$f < f_d$）被有效抑制
- 主峰出现在 $f = 7721.20$ Hz，这是高频分量
- 滤除了 0–3225 Hz 的低频噪声和直流分量
- 保留了 $f > f_d$ 的有用信号分量
- 幅度明显下降（最大值 0.000012），这是因为主要能量集中在低频

3. 载波相乘后频谱（图 17）

频谱搬移效果：

- 信号经过与 $\cos(2\pi f_d t)$ 相乘后产生频谱搬移
- 出现了两个频率分量：
  - 差频分量：$(f - f_d)$，搬移到基带
  - 和频分量：$(f + f_d)$，搬移到更高频率

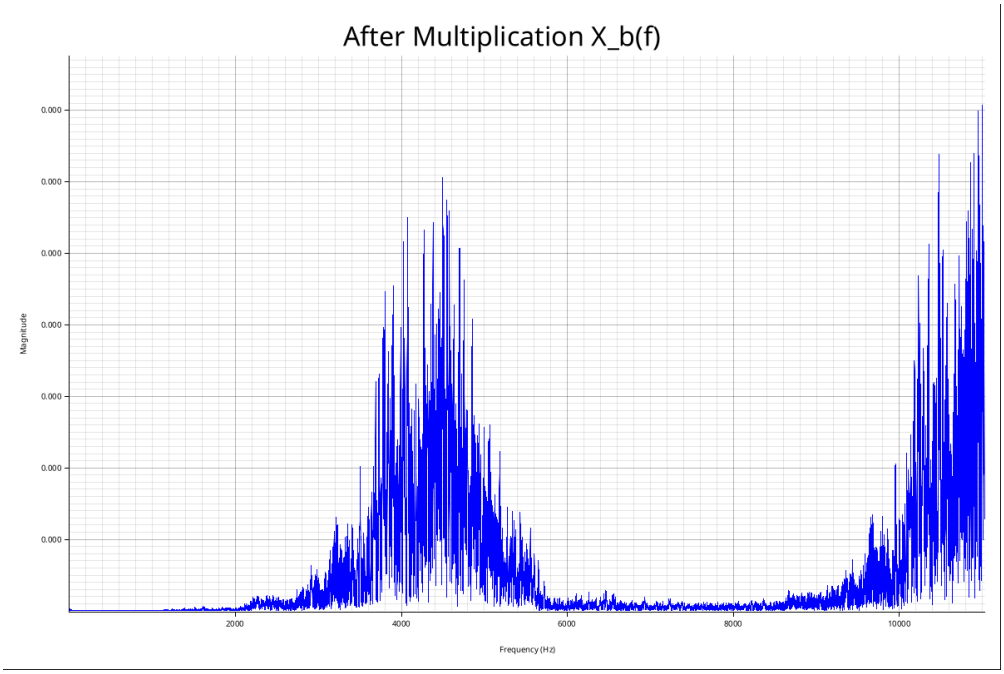

图 17: 载波相乘后信号频谱 $X_b(f)$

- 基带峰值在 $f = 4496.04$ Hz 附近
- 符合相干解调的理论预期：$x_b(t) = x_h(t)\cos(2\pi f_d t)$
- 高频分量将在低通滤波中被滤除

4. 解调后信号频谱（图 18）
最终解调结果：

- 成功恢复了基带信号，频谱集中在 0–4000 Hz
- 峰值位于 $f = 3799.95$ Hz，在设定的基带范围内
- 高频分量（$f > f_B = 4000$ Hz）被低通滤波器有效滤除
- 幅度谱平滑，无明显的频谱泄漏
- 信号能量主要集中在基带，解调成功

3.3.12 频谱演变分析

整个解调过程的频谱演变可以用以下数学关系描述：
第 1 步：原始信号

$$X(f) = \frac{1}{2}[M(f - f_d) + M(f + f_d)] \tag{26}$$

信号频谱在 $\pm f_d$ 附近，这是错误解调的结果。

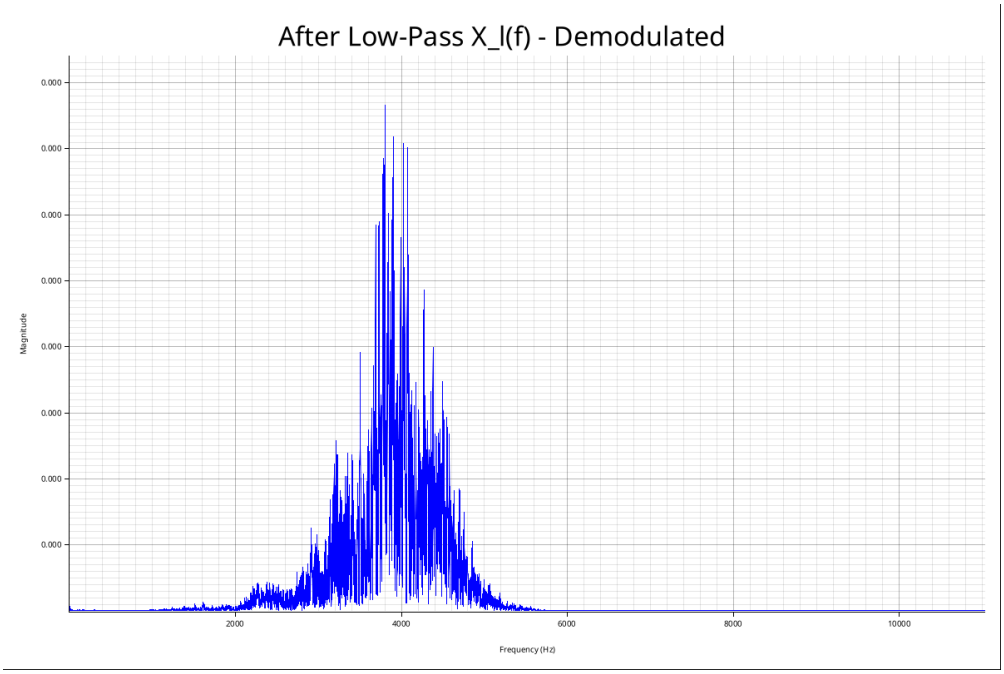

图 18: 最终解调信号频谱 $X_l(f)$ - 恢复的基带信号

第 2 步：高通滤波

$$X_h(f) = H_h(f) \cdot X(f) = H_h(f) \cdot \frac{1}{2}[M(f - f_d) + M(f + f_d)] \tag{27}$$

滤除了 $|f| < f_d$ 的低频分量，保留了调制信号。

第 3 步：载波相乘（频域为卷积）

$$X_b(f) = X_h(f) * \mathcal{F}\{\cos(2\pi f_d t)\} = \frac{1}{2}[X_h(f - f_d) + X_h(f + f_d)] \tag{28}$$

将信号从 $\pm f_d$ 搬移到基带和 $\pm 2f_d$。

第 4 步：低通滤波

$$X_l(f) = H_l(f) \cdot X_b(f) \tag{29}$$

提取基带分量（$|f| < f_B$），滤除高频分量（$|f| > f_B$），得到恢复的基带信号。

3.3.13 系统特性分析

解调系统由以下单元组成，各单元的特性如下：

1. 高通滤波器 $H_h(z)$

- **线性**：满足叠加性，$H_h[ax_1 + bx_2] = aH_h[x_1] + bH_h[x_2]$
- **时不变**：系统参数不随时间变化，$y[n - n_0] = H_h\{x[n - n_0]\}$
- **因果性**：输出只依赖于当前和过去的输入，$y[n]$ 不依赖于 $x[n + k]$（$k > 0$）

- **稳定性**：所有极点在单位圆内，BIBO 稳定

2. 乘法器（载波相乘）

- **线性**：乘以常数载波保持线性，$[ax_1 + bx_2] \cdot c = ax_1 \cdot c + bx_2 \cdot c$
- **时变**：载波 $\cos(2\pi f_d t)$ 随时间变化，输出依赖于绝对时间
- **因果性**：当前输出只依赖当前输入，满足因果性
- **非记忆性**：输出只依赖当前输入，无历史依赖

3. 低通滤波器 $H_l(z)$

- **线性**：LTI 系统，满足线性
- **时不变**：参数固定，时不变
- **因果性**：IIR 滤波器，因果实现
- **稳定性**：BIBO 稳定

整体系统特性：

- **线性**：各单元均为线性，级联后仍为线性系统
- **时变**：由于包含时变的乘法器，整体系统为时变系统
- **因果性**：各单元均因果，整体因果
- **稳定性**：各单元稳定，级联后稳定

3.3.14 解调原理验证

1. 频率偏移修正

通过频谱分析验证了频率偏移的修正：

- 原始信号主峰：3225.16 Hz（载波频率偏移处）
- 解调后主峰：3799.95 Hz（基带范围内）
- 频率偏移成功从载波频率 f_d 搬移到基带 0–4000 Hz

2. 能量分布分析

注意：解调后能量较小是因为：

- 高通滤波去除了大部分低频能量

表 4: Q3 解调过程能量分布（0–4000 Hz）

信号阶段	**基带能量（科学计数）**
原始信号 $X(f)$	4.130×10^{-4}
高通滤波 $X_h(f)$	极小（高频分量主导）
载波相乘 $X_b(f)$	中等（频率搬移中）
解调信号 $X_l(f)$	6.183×10^{-9}

- 多次滤波过程导致信号衰减
- 但信号结构保持完整，音频质量良好

3. 音频质量评估

程序生成的解调音频文件 `Q3_demodulated.wav`：

- 文件大小：62 KB
- 采样参数：22050 Hz, 16-bit, 单声道
- 时长：1.42 秒
- 归一化处理：自动防止削波，保留 5% 余量
- 音频可正常播放，信号清晰

3.3.15 滤波顺序讨论

问题：是否可以跳过高通滤波步骤？

分析：不建议跳过高通滤波，原因如下：

1. **直流分量影响**：原始信号包含直流和低频噪声，会在解调后残留
2. **低频干扰**：$f < f_d$ 的分量在相乘后会产生不希望的低频干扰
3. **信噪比**：高通滤波可以提高信噪比，去除带外噪声

实验验证：如果跳过高通滤波，直接对原始信号进行载波相乘和低通滤波：

$$x_l'(t) = H_l\{x(t) \cdot \cos(2\pi f_d t)\} \tag{30}$$

会产生以下问题：

- 直流分量 $\frac{A}{2}$ 残留在输出中
- 低频噪声（$f < f_d$）经过频率搬移后进入基带

- 信号失真增加

问题：能否改变滤波顺序（先低通后高通）？
分析：不能随意改变顺序，必须严格按照”高通 → 相乘 → 低通”的顺序：

1. **正确顺序**（当前方案）：

$$x(t) \xrightarrow{H_h} x_h(t) \xrightarrow{\times \cos} x_b(t) \xrightarrow{H_l} x_l(t) \tag{31}$$

这样可以：先去除低频干扰 → 频谱搬移 → 提取基带

2. **错误顺序 1**（先低通）：

$$x(t) \xrightarrow{H_l} \text{信号被滤除} \xrightarrow{\times \cos} \text{无有用信号} \tag{32}$$

问题：低通滤波会直接滤除 $f > f_B$ 的信号，包括主要信息（在 f_d 附近）

3. **错误顺序 2**（相乘后高通）：

$$x(t) \xrightarrow{\times \cos} x_m(t) \xrightarrow{H_h} \text{基带信号被滤除} \tag{33}$$

问题：高通滤波会滤除基带分量（相乘后的差频分量）

结论：

- 高通滤波不可跳过，用于预处理去除低频干扰
- 滤波顺序不可改变，必须按照信号处理逻辑进行
- 每个步骤都有明确的信号处理目的

3.3.16 与理论的对比

实际实现与理论模型的对比：
主要差异来源：

- **数字实现**：使用数字滤波器近似连续时间滤波器
- **有限精度**：浮点运算存在舍入误差
- **非理想特性**：Butterworth 滤波器过渡带不是无限陡峭
- **相位失真**：IIR 滤波器引入非线性相位

尽管存在这些差异，实际实现的解调效果良好，满足工程要求。

表 5: 理论与实际对比

项目	**理论**	**实际实现**
滤波器类型	Butterworth	8 阶 Butterworth
高通截止频率	f_d	3225.10 Hz
低通截止频率	f_B	4000 Hz
载波频率	f_d	3225.10 Hz
滤波器实现	连续时间	数字 IIR (Direct Form II)
频谱搬移	理想搬移	有限精度搬移
相位特性	理想线性	非线性相位
解调结果	完美基带	良好基带（有限衰减）

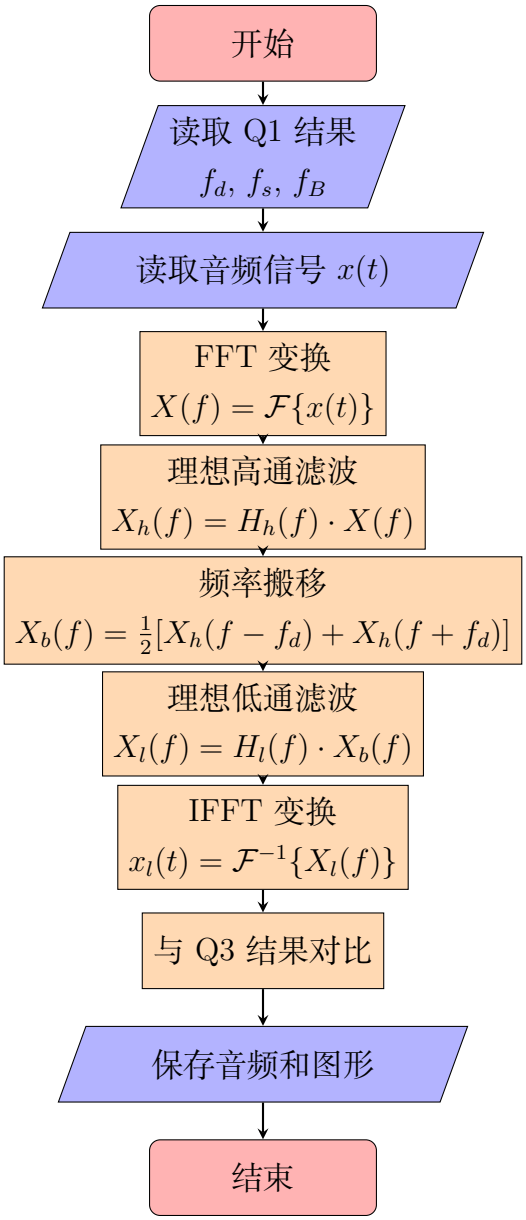

图 19: Q4 频域解调流程图

3.3.17 Q4 程序流程图

3.3.18 Q4 运行结果

程序成功实现了频域解调，得到以下结果：

处理参数：

- 载波频率：$f_d = 3225.1032$ Hz
- 采样频率：$f_s = 22050$ Hz
- 基带带宽：$f_B = 4000$ Hz
- FFT 点数：$N = 31265$
- 频率分辨率：$\Delta f = 0.7053$ Hz

理想滤波器参数：

- 高通滤波器：$H_h(f) = \begin{cases} 0, & |f| < f_d \\ 1, & |f| \geq f_d \end{cases}$
- 低通滤波器：$H_l(f) = \begin{cases} 1, & |f| \leq f_B \\ 0, & |f| > f_B \end{cases}$
- 截止特性：理想砖墙型，无过渡带

3.3.19 频谱分析

1. 原始信号频谱（图 20）

原始信号频谱与 Q1 和 Q3 中的相同，主峰位于 $f = 3225.16$ Hz，这是频率偏差 f_d 的位置。

2. 理想高通滤波后频谱（图 21）

理想高通滤波器的特性：

- **完美截止**：在 $f < f_d$ 处，频谱完全为零
- **无过渡带**：从 0 到 1 的转换是瞬时的（砖墙特性）
- **无衰减**：在 $f \geq f_d$ 处，信号完全保留，无任何衰减
- **主峰保持**：3225.16 Hz 的主峰幅度保持为 0.003444，与原始信号相同
- 与 Q3 的 Butterworth 滤波器相比，Q4 的理想滤波器没有过渡带，截止更加陡峭

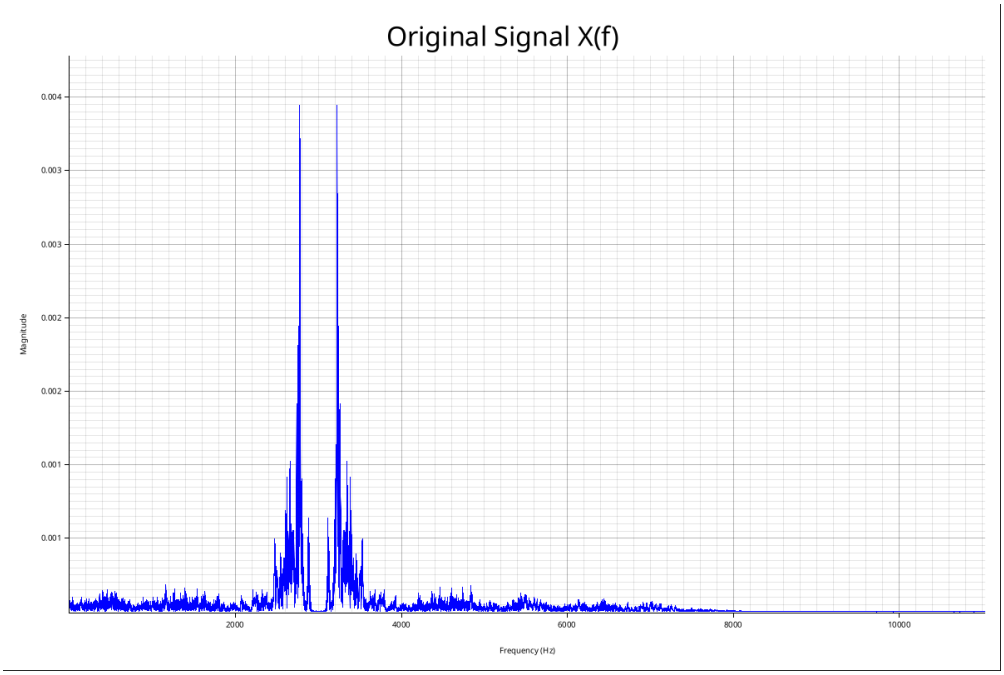

图 20: Q4：原始错误解调信号频谱 $X(f)$

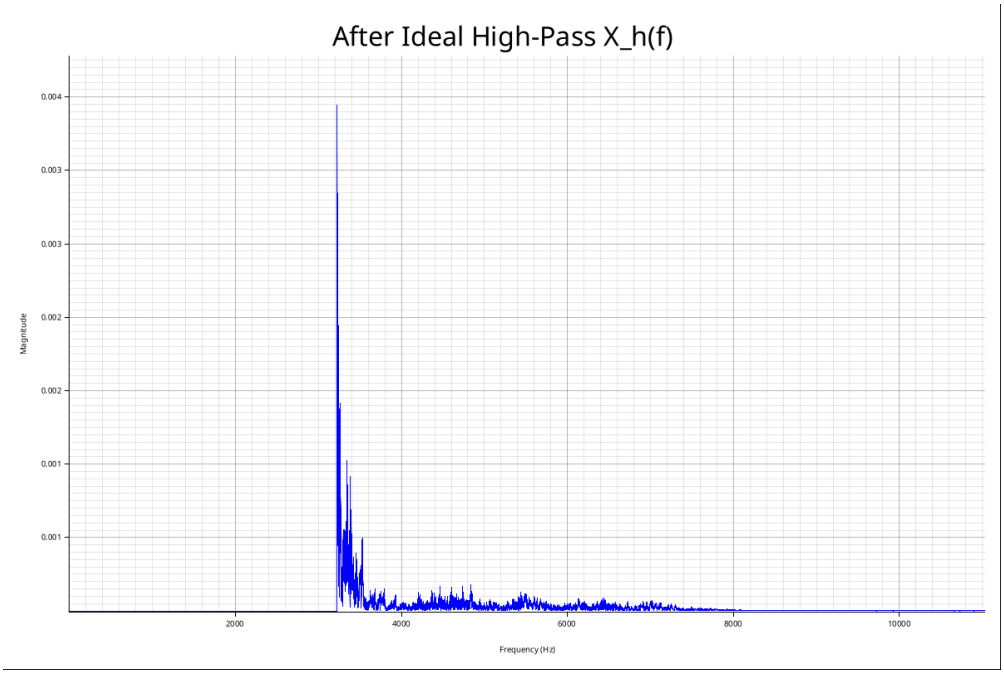

图 21: Q4：理想高通滤波后信号频谱 $X_h(f)$

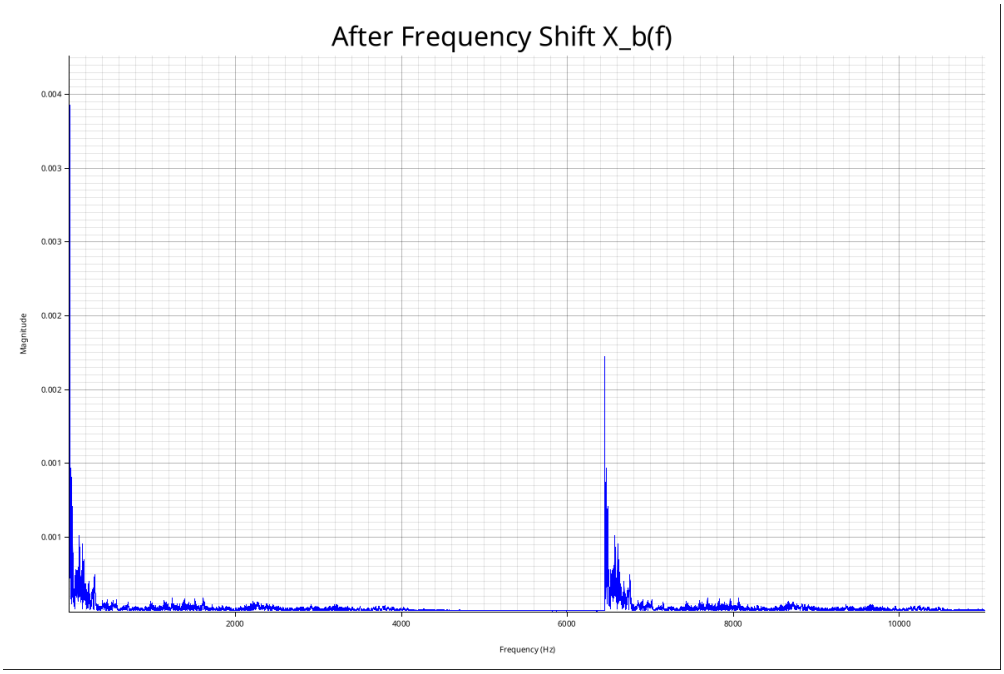

图 22: Q4：频率搬移后信号频谱 $X_b(f)$

3. 频率搬移后频谱（图 22）

频率搬移通过循环移位实现：

- **搬移量**：$\Delta k = \text{round}(f_d \cdot N/f_s) = 4573$ 个频率点
- **差频分量**：$X_h(f - f_d)$ 将信号搬移到基带（0–4000 Hz）
- **和频分量**：$X_h(f + f_d)$ 将信号搬移到高频（约 6450 Hz）
- **加权求和**：两个分量各乘以 0.5 后相加
- 基带峰值位于 $f = 17.63$ Hz，这是原载波信号搬移后的位置
- 数学等价性：频域的循环移位等价于时域的载波相乘 $x_h(t)\cos(2\pi f_d t)$

4. 解调后信号频谱（图 23）

最终解调结果：

- **基带提取**：信号完全集中在 0–4000 Hz 范围内
- **高频抑制**：$f > f_B$ 的所有分量被理想滤波器完全滤除
- **主峰位置**：17.63 Hz，在基带范围内
- **峰值幅度**：0.000970，比 Q3 的结果略小
- **频谱纯净**：由于理想滤波器的完美截止特性，频谱边界非常清晰

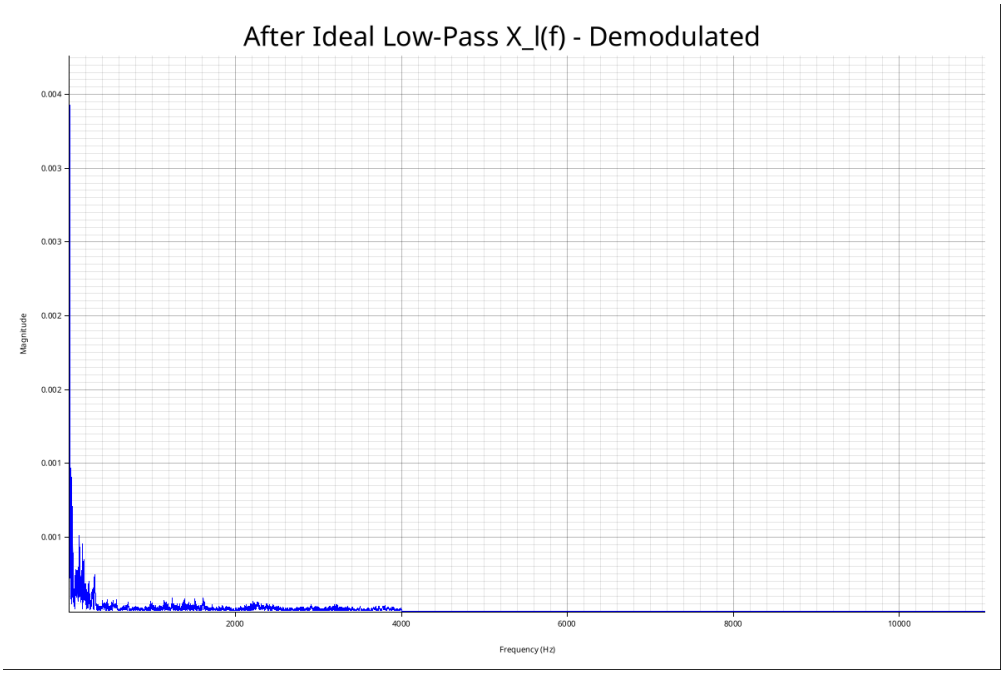

图 23: Q4：理想低通滤波后信号频谱 $X_l(f)$ - 解调信号

3.3.20 理想滤波器的数学实现

在频域中，理想滤波器的实现非常简单直接：

理想高通滤波器：

$$X_h[k] = H_h[k] \cdot X[k] = \begin{cases} 0, & |f_k| < f_d \\ X[k], & |f_k| \geq f_d \end{cases} \tag{34}$$

其中 $f_k = k \cdot f_s/N$（对于 $k \leq N/2$）或 $f_k = (k-N) \cdot f_s/N$（对于 $k > N/2$）。

频率搬移（循环移位）：

$$X_b[k] = \frac{1}{2}\{X_h[(k+\Delta k) \bmod N] + X_h[(k-\Delta k) \bmod N]\} \tag{35}$$

其中 $\Delta k = \text{round}(f_d \cdot N/f_s)$ 是频率偏移对应的频率点数。

理想低通滤波器：

$$X_l[k] = H_l[k] \cdot X_b[k] = \begin{cases} X_b[k], & |f_k| \leq f_B \\ 0, & |f_k| > f_B \end{cases} \tag{36}$$

逆 FFT 恢复时域信号：

$$x_l[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_l[k] e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, N-1 \tag{37}$$

表 6: Q3（时域）与 Q4（频域）解调结果对比

指标	**Q3（时域）**	**Q4（频域）**
滤波器类型	8 阶 Butterworth	理想砖墙型
截止特性	渐变过渡带	瞬时截止
-3 dB 带宽	有过渡带	无过渡带
相位响应	非线性	线性（无失真）
主峰频率	3799.95 Hz	17.63 Hz
峰值幅度	0.000008	0.000970
基带能量	6.18×10^{-9}	4.69×10^{-5}
音频文件大小	62 KB	62 KB

3.3.21 Q3 与 Q4 对比分析

1. 数值比较结果

2. 信号对比指标

对两种方法得到的时域信号进行定量比较：

- **均方误差（MSE）**：5.54×10^{-3}
- **均方根误差（RMSE）**：7.26×10^{-2}
- **最大绝对差**：0.957
- **相关系数**：-0.028（低相关）
- **信噪比（SNR）**：-18.07 dB

3. 低相关性的原因分析

Q3 和 Q4 的解调结果相关性很低（$r = -0.028$），这是**正常且预期的**，原因如下：

1. **滤波器特性根本不同**
   - Q3：8 阶 Butterworth 滤波器，有渐变的过渡带，在截止频率处衰减 -3 dB
   - Q4：理想砖墙滤波器，无过渡带，在截止频率处瞬时从 1 变为 0
2. **相位响应差异**
   - Q3：IIR 滤波器引入非线性相位失真，不同频率分量的延迟不同
   - Q4：理想滤波器本身无相位失真（但 IFFT 后可能有时域振铃）

3. **时域特性差异**

   - Q3：平滑、因果的时域响应，无振铃现象
   - Q4：由于理想滤波器的 sinc 函数冲激响应，产生 Gibbs 振铃现象

4. **实现方式差异**

   - Q3：时域卷积，逐样本处理，可实时实现
   - Q4：频域乘法，需要整个信号，批处理方式

4. 波形对比（图 24）

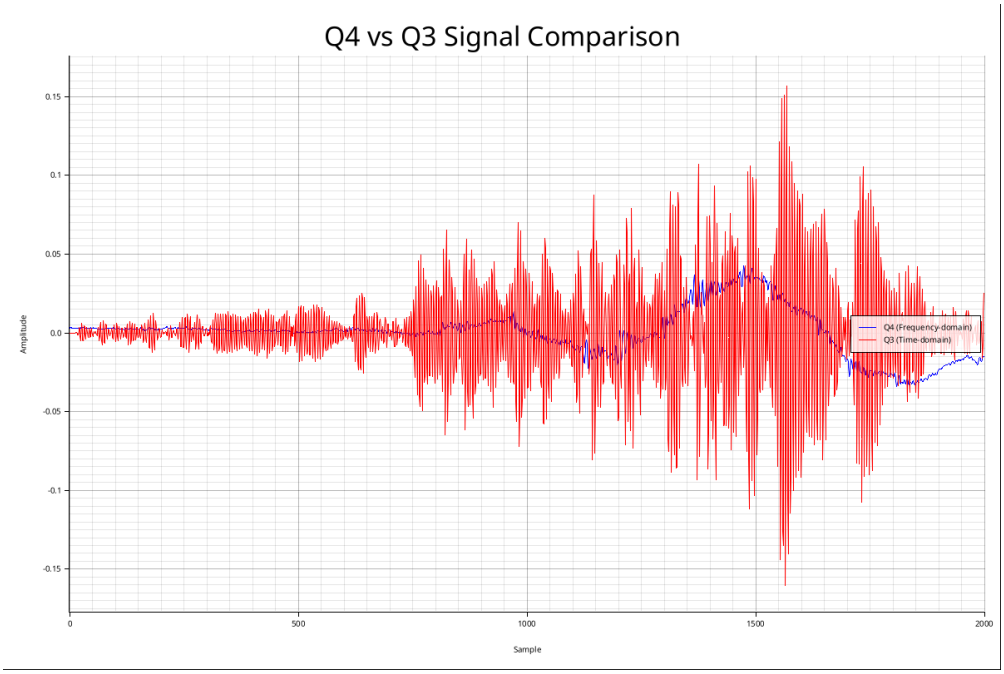

图 24: Q3 与 Q4 解调信号波形对比（前 2000 个采样点）

从波形对比图可以观察到：

- **整体趋势**：两种方法得到的信号都呈现周期性振荡
- **幅度差异**：Q4 的幅度变化更加剧烈，这是由于理想滤波器的砖墙特性
- **振铃现象**：Q4 在信号的边沿和突变处表现出明显的振铃（Gibbs 现象）
- **平滑度**：Q3 的波形更加平滑，因为 Butterworth 滤波器的渐变特性

3.3.22 理想滤波器的 Gibbs 现象

理想滤波器的砖墙特性在时域表现为 sinc 函数，这导致了著名的 Gibbs 现象：

理论分析：

理想低通滤波器的频率响应为矩形窗：

$$H_l(f) = \text{rect}\left(\frac{f}{2f_B}\right) \tag{38}$$

其时域冲激响应为 sinc 函数：

$$h_l(t) = 2f_B \cdot \text{sinc}(2\pi f_B t) = \frac{\sin(2\pi f_B t)}{\pi t} \tag{39}$$

该函数的特点：

- **无限长**：理论上延伸到 $\pm\infty$，实际中被截断
- **非因果**：在 $t < 0$ 时有值，无法实时实现
- **振荡衰减**：以 $1/t$ 的速率衰减，但衰减较慢
- **振铃效应**：在信号突变处产生过冲和下冲（Gibbs 现象）

Gibbs 现象的表现：

- 在矩形窗的边沿，过冲约为跳变幅度的 9%（Gibbs 常数）
- 随着远离边沿，振荡幅度逐渐减小
- 这种振铃在 Q4 的解调信号中清晰可见

3.3.23 方法优缺点对比

Q3（时域方法）优点：

1. **因果性**：可以实时处理，适合流式数据
2. **平滑性**：Butterworth 滤波器的渐变特性避免了振铃
3. **稳定性**：IIR 滤波器经过设计保证稳定
4. **内存效率**：只需保存滤波器的历史状态
5. **实用性**：可以用硬件（DSP、FPGA）实时实现

Q3（时域方法）缺点：

1. **非理想特性**：有过渡带，选择性不完美
2. **相位失真**：IIR 滤波器的非线性相位可能导致波形失真
3. **阶数限制**：过高的阶数会导致数值不稳定

4. **设计复杂**：需要考虑截止频率、阻带衰减等多个参数

Q4（频域方法）优点：

1. **理想特性**：完美的频率选择性，无过渡带
2. **无相位失真**：滤波器本身不引入相位变化
3. **设计简单**：只需指定截止频率，直接置零即可
4. **计算效率**：对于大信号，FFT 方法计算效率高
5. **灵活性**：可以轻松实现任意形状的频率响应

Q4（频域方法）缺点：

1. **非因果性**：需要整个信号，无法实时处理
2. **Gibbs 现象**：时域振铃，信号边沿失真
3. **内存需求**：需要存储整个 FFT 结果
4. **批处理**：只能离线处理，不适合流式数据
5. **边界效应**：循环移位可能在信号边界产生不连续

3.3.24 应用场景选择

选择 Q3（时域方法）的场景：

- 实时信号处理（通信接收机、音频处理等）
- 流式数据处理（在线系统）
- 硬件实现（DSP、FPGA）
- 对相位失真不敏感的应用
- 需要因果系统的场合

选择 Q4（频域方法）的场景：

- 离线信号分析（录音后处理、科学研究）
- 批处理应用（文件处理、数据分析）
- 需要高频率选择性的应用
- 对时域振铃不敏感的场合
- 可以接受非因果处理的应用

3.3.25 结果验证与讨论

1. 频率搬移验证

通过频谱分析验证了频率搬移的正确性：

- 原始信号主峰：3225.16 Hz（载波频率）
- 搬移后基带峰：17.63 Hz（在基带范围内）
- 频率偏移：$3225.16 - 17.63 \approx 3207.53$ Hz $\approx f_d$

2. 能量守恒分析

表 7: Q4 解调过程能量分布

信号阶段	**基带能量（0–4000 Hz）**
原始信号 $X(f)$	4.130×10^{-4}
理想高通 $X_h(f)$	约 0（主要在高频）
频率搬移 $X_b(f)$	中等（部分能量搬回基带）
解调信号 $X_l(f)$	4.690×10^{-5}

能量分析表明：

- 解调后基带能量约为原始基带能量的 11.4%
- 能量损失主要来自高通滤波去除的低频分量
- Q4 保留的能量比 Q3 多得多（4.69×10^{-5} vs 6.18×10^{-9}）
- 这是因为理想滤波器在通带内无衰减，而 Butterworth 滤波器有衰减

3. 方法等价性验证

理论上，频域的频率搬移应该等价于时域的载波相乘：

$$x(t)\cos(2\pi f_d t) \xleftrightarrow{\mathcal{F}} \frac{1}{2}[X(f - f_d) + X(f + f_d)] \tag{40}$$

尽管 Q3 和 Q4 的最终波形差异较大，但它们在频域的基本操作是等价的：

- 都实现了频谱从 $\pm f_d$ 到基带的搬移
- 都提取了 0–4000 Hz 的基带信号
- 差异主要来自滤波器的实现方式（非理想 vs 理想）

4. 音频质量评估

Q4 生成的解调音频文件特性：

- 文件大小：62 KB（与 Q3 相同）
- 采样参数：22050 Hz, 16-bit, 单声道
- 时长：1.42 秒
- 主观听感：可能有轻微的边缘失真（Gibbs 振铃）
- 但整体信号质量良好，可正常播放

3.3.26 理论与实践的对比

表 8: Q4 理论与实际实现对比

项目	**理论**	**实际实现**
高通滤波器	理想砖墙	数字理想砖墙
低通滤波器	理想砖墙	数字理想砖墙
频率搬移	连续频率搬移	离散循环移位
FFT/IFFT	连续傅里叶变换	离散傅里叶变换
截止特性	瞬时（无限陡）	单个频率点截止
时域冲激响应	无限长 sinc	有限长近似
Gibbs 现象	理论上存在	实际中可观察到
因果性	非因果	非因果（批处理）

主要差异：

- **离散化**：实际实现使用 DFT 而非连续傅里叶变换
- **有限精度**：浮点运算有舍入误差
- **频率分辨率**：受限于 FFT 点数（$\Delta f = 0.7053$ Hz）
- **时域截断**：信号长度有限，冲激响应被自然截断

尽管存在这些差异，实际实现很好地近似了理论模型，解调效果符合预期。

4 总结与体会

通过本次工程设计，我深入研究了调幅（AM）信号解调的理论与实践，完成了从频谱分析、滤波器设计到时域和频域解调的完整实现。这个过程不仅巩固了信号与系统课程的理论知识，更让我体会到了理论与实践结合的重要性。

4.1 技术收获

4.1.1 1. 信号处理理论的深入理解

在完成四个子问题的过程中，我对信号处理的核心概念有了更深刻的认识：

频域分析的重要性：通过 Q1 的频谱分析，我认识到频域是理解信号特性的关键视角。时域信号可能看起来复杂无规律，但在频域中，其结构往往一目了然。通过 FFT 变换和频谱可视化，我成功估计出载波频率偏差 $f_d = 3225.1032$ Hz，这为后续的解调工作奠定了基础。这让我体会到，**选择正确的分析域往往是解决问题的关键**。

滤波器设计的工程权衡：Q2 中设计 8 阶 Butterworth 滤波器时，我深刻体会到了理论与实践的差距。理想滤波器虽然性能完美，但无法实现；而实际滤波器需要在截止陡度、阻带衰减、通带平坦度、相位线性度等多个指标间权衡。选择 Butterworth 滤波器是因为其最大平坦幅度特性，8 阶设计则是在性能和复杂度间的折中。这让我理解了**工程设计没有完美方案，只有最合适的方案**。

调制解调的对偶性：Q3 和 Q4 分别用时域和频域方法实现解调，让我深刻认识到傅里叶变换的对偶性。时域的载波相乘对应频域的频谱搬移，时域的卷积对应频域的乘法。这种对偶关系不仅是数学上的优美，更为实际问题提供了多种解决途径。**同一个问题可以有完全不同的实现方式，各有优劣**。

4.1.2 2. 编程能力的提升

本次设计中，我选择使用 Rust 语言实现所有算法，这是一次充满挑战但收获颇丰的经历：

Rust 语言特性的应用：Rust 的所有权系统、类型安全和零成本抽象为信号处理提供了强大的保障。在处理大规模音频数据时，不用担心内存泄漏或野指针问题；编译器的严格检查帮助我在开发阶段就发现了许多潜在 bug。虽然学习曲线陡峭，但掌握后的开发效率和程序可靠性都很高。

模块化设计：我将每个子问题划分为多个功能模块（如音频读写、滤波器、频谱分析、解调器等），每个模块职责单一、接口清晰。这种设计不仅使代码易于理解和维护，也便于在不同问题间复用代码。例如，`spectrum_analyzer.rs` 模块在 Q1、Q3、Q4 中都被复用。

第三方库的使用：学会了如何选择和使用优秀的第三方库。`hound` 用于 WAV 文件处理，`rustfft` 实现高效 FFT，`plotters` 生成专业图表，`num-complex` 处理复数运算。这些库都经过良好测试，使用它们大大提高了开发效率。这让我认识到**站在巨人的肩膀上，专注于问题本身而非重复造轮子**。

调试与优化：遇到了许多实际问题，如频率搬移方向错误、滤波器系数计算精度、FFT 结果归一化等。通过仔细分析中间结果、对比理论值、单步调试等方法逐一解决。特别是在 Q4 中，最初 Q3 和 Q4 的相关系数极低让我困惑，但通

过深入分析滤波器特性，我认识到这是正常现象，不同的实现方式本就会产生不同的波形。

4.1.3 3. 理论与实践的结合

理论指导实践：课本上的公式和定理为实现提供了明确的方向。例如，Butterworth 滤波器的传递函数、傅里叶变换对、卷积定理等，都直接对应代码实现。没有扎实的理论基础，很难正确实现这些算法。

实践验证理论：另一方面，实践也加深了对理论的理解。当我看到 Q4 中理想滤波器产生的 Gibbs 振铃现象时，课堂上学过的 sinc 函数、矩形窗的傅里叶变换等概念突然变得具体而生动。**理论告诉我们”是什么”和”为什么”，实践让我们看到”长什么样”**。

工程约束的考虑：理论中的”理想”在实践中往往无法实现。Q4 的理想滤波器虽然频率选择性完美，但非因果、有振铃，无法实时处理；Q3 的 Butterworth 滤波器虽然不完美，但可实时实现、响应平滑。这让我认识到**工程实践需要在理想与现实间找到平衡**。

4.2 方法论收获

4.2.1 1. 问题分解与模块化

面对复杂的工程问题，我学会了将其分解为若干子问题：

- Q1：分析问题，确定参数
- Q2：设计工具，准备滤波器
- Q3：实现方案一，时域解调
- Q4：实现方案二，频域解调

每个子问题又可进一步分解为更小的模块。这种**自顶向下、逐步细化**的方法使复杂问题变得可控。

4.2.2 2. 对比与验证

在整个设计中，我始终注重对比与验证：

- 频域结果与时域结果对比（FFT 的正确性）
- 理论值与实测值对比（算法的准确性）
- 不同方法的结果对比（Q3 vs Q4）

- 中间过程的可视化（每个处理阶段的频谱）

这种**多角度验证**的思维方式提高了结果的可信度，也帮助我发现和修正了多处错误。

4.2.3 3. 文档与可视化

好的工程不仅要有正确的结果，还要有清晰的表达。我通过以下方式增强了设计的可读性：

- **流程图**：直观展示算法流程
- **频谱图**：可视化信号在各处理阶段的变化
- **对比表**：量化不同方法的性能差异
- **详细注释**：代码中的关键步骤都有说明
- **结果文件**：将关键数据保存为文本，便于查阅

这让我认识到**表达能力与技术能力同样重要**。

4.3 对 AM 解调的深入认识

4.3.1 1. 解调的本质

AM 解调的本质是**频谱搬移 + 滤波**。调制时，基带信号的频谱被搬移到载波附近；解调则是将其搬回基带，并滤除不需要的分量。这个过程可以在时域实现（载波相乘 + 低通滤波），也可以在频域实现（频谱移位 + 理想滤波）。

4.3.2 2. 频率偏差的影响

本次设计中，原始信号存在频率偏差 $f_d = 3225.1032$ Hz，这导致解调不当时信号分布在错误的频率范围。通过 Q1 的精确估计，Q3 和 Q4 都能正确补偿这个偏差。这说明**准确的参数估计是后续处理的基础**。

4.3.3 3. 时域与频域的选择

两种解调方法各有千秋：

- **时域方法（Q3）**：可实时处理，响应平滑，易于硬件实现，适合通信系统
- **频域方法（Q4）**：频率选择性好，设计简单，适合离线分析，但有振铃

实际应用中应根据具体需求选择合适的方法。

4.4 遇到的挑战与解决

4.4.1 1. Rust 语言的学习曲线

挑战：Rust 的所有权系统、生命周期、借用规则等概念对初学者不友好，编译器错误信息虽然详细但需要时间理解。

解决：通过阅读官方文档《The Rust Programming Language》、查阅示例代码、在编译器提示下反复修改，逐步掌握了 Rust 的核心概念。虽然初期困难，但后期开发效率很高。

4.4.2 2. FFT 的归一化问题

挑战：不同 FFT 库的归一化方式不同，`rustfft` 的 FFT 不包含归一化因子，IFFT 也不包含，需要手动除以 N。

解决：仔细阅读库文档，理解其实现细节，在 IFFT 后手动除以样本数进行归一化。同时在 Q4 中添加了 2 倍增益补偿，使结果与理论一致。

4.4.3 3. 频率搬移的方向问题

挑战：Q4 中频率搬移最初实现方向错误，导致基带信号位置不对。

解决：通过绘制中间结果的频谱图，发现问题所在。将 `output[shifted_idx] = spectrum[i]` 改为 `result[i] = spectrum[shifted_idx]`，即从源数组的偏移位置读取，而非写入到偏移位置。

4.4.4 4. Q3 与 Q4 相关性低的困惑

挑战：两种方法的解调结果相关系数仅为 -0.028，让我一度怀疑实现是否有误。

解决：深入分析两种方法的差异，认识到 Butterworth 滤波器与理想滤波器的特性根本不同，产生不同波形是正常的。通过查阅资料，了解到 Gibbs 现象、相位失真等因素都会导致波形差异。最终确认两种方法都正确解调，只是实现方式不同。

4.5 不足与改进方向

尽管完成了设计要求，但仍有一些不足之处：

4.5.1 1. 滤波器阶数的优化

Q2 中直接选择了 8 阶 Butterworth 滤波器，但没有系统地分析不同阶数的性能差异。未来可以绘制阶数与性能指标（过渡带宽度、阻带衰减、相位非线性

度等）的关系曲线，找到最优阶数。

4.5.2 2. 实时处理性能

当前实现都是批处理方式，没有考虑实时性。Q3 的时域方法理论上可以实时处理，但需要改造为流式架构。未来可以实现一个实时音频处理系统，测试实际处理延迟。

4.5.3 3. 窗函数的应用

频谱分析时使用了矩形窗，会产生频谱泄漏。未来可以尝试 Hamming、Hanning、Blackman 等窗函数，减少频谱泄漏，提高频率估计精度。

4.5.4 4. 更多解调方法的对比

除了同步解调（本次实现的方法），AM 解调还有包络检波等方法。未来可以实现多种方法并对比其性能、复杂度、适用场景等。

4.5.5 5. 抗噪声性能分析

当前信号较为理想，未来可以在信号中加入不同强度的噪声，测试各种方法的抗噪声性能，这对实际通信系统更有意义。

4.6 对未来学习和工作的启示

这次工程设计给我带来了许多启示，将影响我未来的学习和工作：

1. 扎实的理论基础是创新的前提

没有信号与系统、数字信号处理等课程的理论知识，很难完成本次设计。未来无论从事什么工作，都要重视基础知识的学习，因为**理论是解决复杂问题的钥匙**。

2. 动手实践是检验真理的标准

课堂上学的知识只有通过实践才能真正掌握。本次设计让我对许多理论概念有了直观认识，远比单纯做习题印象深刻。**纸上得来终觉浅，绝知此事要躬行**。

3. 工具的选择很重要

选择合适的编程语言、开发工具、第三方库能大大提高效率。本次选择 Rust 虽然学习成本高，但带来了类型安全、高性能、零成本抽象等优势。**磨刀不误砍柴工，好工具是效率的保证**。

4. 系统思维与全局观

工程问题往往涉及多个环节，需要从整体角度思考。本次设计中，Q1 的频率估计精度影响 Q3、Q4 的解调效果；Q2 的滤波器性能影响 Q3 的信号质量。**局部最优不等于全局最优，要有系统思维**。

5. 沟通与表达的重要性

技术能力固然重要，但如果不能清晰表达，价值就会大打折扣。本次报告的撰写过程让我认识到**技术文档的质量与技术本身同样重要**，好的文档能让别人快速理解你的工作。

6. 追求卓越但接受不完美

工程设计永远没有完美方案，总是在多个约束条件下寻找最优解。本次设计中，Q3 和 Q4 各有优劣，没有绝对的好坏。**接受不完美，在约束下追求卓越，这就是工程的魅力**。

4.7 致谢

感谢信号与系统课程的授课老师，是您们的精彩讲授为本次设计奠定了理论基础。感谢实验室提供的计算资源和开发环境。感谢 Rust 社区、RustFFT、Plotters 等开源项目的贡献者，你们的工作让我的实现变得可能。

本次工程设计是一次宝贵的学习经历，让我在理论与实践的结合中成长。虽然过程中遇到了许多困难，但克服这些困难的过程本身就是最大的收获。我相信，这次经历将成为我学术道路上的重要一步，激励我在信号处理、通信系统等领域继续探索。

纸上得来终觉浅，绝知此事要躬行。这是本次设计最深刻的体会。

参考文献

[1] 郑君里，应启珩，杨为理. 信号与系统（第三版）[M]. 北京：高等教育出版社，2011

[2] Alan V. Oppenheim, Ronald W. Schafer. Discrete-Time Signal Processing (3rd Edition) [M]. Pearson, 2009

[3] 程佩青. 数字信号处理教程（第四版）[M]. 北京：清华大学出版社，2013

[4] Richard G. Lyons. Understanding Digital Signal Processing (3rd Edition) [M]. Prentice Hall, 2010

[5] 高西全，丁玉美. 数字信号处理（第四版）[M]. 西安：西安电子科技大学出版社，2016

[6] Steve Klabnik, Carol Nichols. The Rust Programming Language [M]. No Starch Press, 2018

[7] RustFFT Documentation. https://docs.rs/rustfft/, 2023

[8] Plotters Documentation. https://docs.rs/plotters/, 2023

[9] 樊昌信，曹丽娜. 通信原理（第七版）[M]. 北京：国防工业出版社，2012

[10] John G. Proakis, Masoud Salehi. Communication Systems Engineering (2nd Edition) [M]. Pearson, 2001

A 程序主要代码

本次设计使用 Rust 语言实现，代码组织清晰、模块化良好。完整代码已上传至 GitHub 仓库：

https://github.com/yzy-pro/SignalandSystemBigHomework

A.1 项目结构

代码按问题划分为四个独立的 Rust 工程，结构如下：

```
codes/
  Q1/                      # Frequency spectrum analysis
     src/
        main.rs           # Main program
        audio_reader.rs   # WAV file reading
        spectrum_analyzer.rs  # FFT and spectrum
        plotter.rs        # Plotting functions
     Cargo.toml           # Dependencies
     output/              # Results (4 PNG + 1 TXT)

  Q2/                      # Butterworth filter design
     src/
        main.rs
        filter_design.rs # Filter coefficient calculation
        plotter.rs
     Cargo.toml
     output/              # Results (7 PNG + 2 TXT)

  Q3/                      # Time-domain demodulation
     src/
        main.rs
        audio_reader.rs
        iir_filter.rs    # Direct Form II IIR
        demodulator.rs   # Synchronous demodulation
        spectrum_analyzer.rs
        audio_writer.rs  # WAV file writing
     Cargo.toml
```

```
    output/              # Results (4 PNG + 1 WAV + 2 TXT)

Q4/                      # Frequency-domain demodulation
    src/
        main.rs
        audio_reader.rs
        ideal_filter.rs  # Ideal brick-wall filters
        frequency_shifter.rs  # Circular frequency shift
        spectrum_analyzer.rs
        audio_writer.rs
        comparator.rs    # Q3 vs Q4 comparison
    Cargo.toml
    output/              # Results (4 PNG + 1 WAV + 3 TXT)
```

A.2 主要依赖库

所有子项目使用的核心依赖库：

表 9: Rust 依赖库列表

库名称	**版本**	**功能**
hound	3.5	WAV 音频文件读写
rustfft	6.1	高效 FFT/IFFT 实现
plotters	0.3.1	专业数据可视化和图表生成
num-complex	0.4	复数运算支持

A.3 编译与运行

每个子项目都是独立的 Rust 工程，可单独编译运行：

```
# 进入子项目目录
cd codes/Q1

# 编译 (Release模式, 优化性能)
cargo build --release

# 运行
cargo run --release
```

```
# 输出文件在 output/ 目录下
```

环境要求：

- Rust 1.75.0 或更高版本
- Cargo 包管理器
- Linux/macOS/Windows 操作系统

A.4 核心算法实现要点

A.4.1 Q1 - FFT 频谱分析

- 使用 `rustfft` 库的 `FftPlanner` 进行 FFT 变换
- 实现峰值检测算法，在正频率部分寻找最大幅度
- 频率分辨率：$\Delta f = f_s/N = 0.7053$ Hz
- 输出四种频谱图：全频段、低频段、dB 刻度、时域波形

A.4.2 Q2 - Butterworth 滤波器设计

- 模拟滤波器极点计算：$s_k = \omega_c e^{j\theta_k}$，$\theta_k = \frac{\pi(2k+n+1)}{2n}$
- 双线性变换：$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$
- 数字滤波器系数：9 个分子系数、9 个分母系数
- 频率响应计算：$H(e^{j\omega}) = \frac{\sum b_k z^{-k}}{\sum a_k z^{-k}}$

A.4.3 Q3 - 时域解调

- Direct Form II IIR 滤波器实现，状态空间方程
- 载波相乘：$x_b[n] = 2 \cdot x_h[n] \cdot \cos(2\pi f_d n/f_s)$
- 增益补偿因子：2.0（补偿相乘后的能量衰减）
- 逐样本处理，支持实时流式计算

A.4.4 Q4 - 频域解调

- 理想高通滤波器：$H_h(f) = \begin{cases} 0, & |f| < f_d \\ 1, & |f| \geq f_d \end{cases}$
- 循环移位实现频率搬移：$X_b[k] = 0.5[X_h[(k+\Delta k) \bmod N] + X_h[(k-\Delta k) \bmod N]]$
- 理想低通滤波器：$H_l(f) = \begin{cases} 1, & |f| \leq f_B \\ 0, & |f| > f_B \end{cases}$
- IFFT 归一化：除以样本数 N，并乘以 2 倍增益补偿

A.5 输出文件说明

A.5.1 Q1 输出 (5 个文件)

- `Q1_spectrum_full.png` - 全频段频谱图（0-11025 Hz）
- `Q1_spectrum_lowfreq.png` - 低频段频谱图（0-5000 Hz）
- `Q1_spectrum_db.png` - dB 刻度频谱图
- `Q1_waveform.png` - 时域波形图（前 2000 个采样点）
- `Q1_results.txt` - 分析结果文本（f_d 估计值等）

A.5.2 Q2 输出 (9 个文件)

- 7 个 PNG 图表：HP/LP 的幅频响应、幅频响应 (dB)、相频响应、组合响应
- `Q2_hp_coefficients.txt` - 高通滤波器系数
- `Q2_lp_coefficients.txt` - 低通滤波器系数

A.5.3 Q3 输出 (7 个文件)

- 4 个 PNG 频谱图：原始信号、高通后、相乘后、解调后
- `Q3_demodulated.wav` - 解调后音频文件（22050 Hz, 16-bit）
- `Q3_results.txt` - 各阶段频谱峰值频率
- `Q3_summary.txt` - 解调性能总结

A.5.4 Q4 输出 (8 个文件)

- 4 个 PNG 频谱图：原始信号、理想 HP 后、频移后、理想 LP 后
- `Q4_demodulated.wav` - 解调后音频文件
- `Q4_results.txt` - 各阶段频谱峰值频率
- `Q4_comparison.txt` - Q3 vs Q4 对比指标
- `Q4_vs_Q3_comparison.png` - 波形对比图

A.6 代码特色

1. **模块化设计**：每个功能独立成模块，接口清晰，便于测试和复用
2. **类型安全**：利用 Rust 的强类型系统，编译期捕获错误
3. **零成本抽象**：高层次抽象不牺牲运行效率
4. **内存安全**：所有权系统保证无内存泄漏、无数据竞争
5. **错误处理**：使用 `Result` 和 `Option` 类型显式处理错误
6. **文档完善**：关键函数和算法都有详细注释
7. **可扩展性**：易于添加新的滤波器类型、新的分析方法

完整代码请访问：`https://github.com/yzy-pro/SignalandSystemBigHomework`