

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

报告撰写说明	1
1 问题描述	3
2 理论模型	4
2.1 原理分析与设计思路	4
2.1.1 调幅信号的数学表示	4
2.1.2 相干解调原理	4
2.1.3 频率偏差的影响	4
2.1.4 二次解调的设计思路	5
2.2 数学模型	5
2.2.1 信号频谱模型	5
2.2.2 滤波器设计模型	5
2.2.3 解调过程的数学描述	6
2.2.4 系统特性分析	6
3 程序设计	6
3.1 编程思路	6
3.1.1 Q1: 频谱分析与频率偏差估计的编程思路	7
3.1.2 Q2: 滤波器设计的编程思路	7
3.1.3 Q3: 时域解调方法的编程思路	8
3.1.4 Q4: 频域解调方法的编程思路	8
3.1.5 程序模块划分	9
3.2 主要流程图及说明	9
3.3 结果分析	10
3.3.1 运行结果	10
3.3.2 结果验证	10
4 总结与体会	10
参考文献	11
A 程序主要代码	12
附录程序主要代码	12

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 $X_k = X(f)|_{f=kf_s/N}$, $k = 0, \dots, N-1$ 。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：

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

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：

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

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

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

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：

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

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：

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

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

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

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：

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

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：

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

错误解调后的信号频谱为：

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

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：

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

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：

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

2.2.3 解调过程的数学描述

时域方法：

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

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

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

其中 $*$ 表示卷积运算。

频域方法：

$$X_h(f) = H_h(f) \cdot X(f) \quad (15)$$

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

$$X_l(f) = H_l(f) \cdot X_b(f) \quad (17)$$

最终通过逆傅里叶变换得到时域信号：

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

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 $f_k = k \cdot f_s / N$ ， $k = 0, 1, \dots, N - 1$ 。

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 $f_k = k \cdot f_s / N$ （ $k = 0, 1, \dots, N - 1$ ）处的频率响应：

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

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：

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

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：

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

对于理想低通滤波器：

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

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

图 1: xxxx 流程图

(流程图要求用适当工具绘制后插入)

由状态图进行状态编码。扭循环计数器状态数： $N = 2 \times 6 = 12$ 个。如表1所示为扭循环计数器的状态数。

表 1: 扭循环计数器的状态数

状态	编码	说明
...

3.3 结果分析

3.3.1 运行结果

在 matlab 上运行后得到...

3.3.2 结果验证

我们用画图软件打开图片...

4 总结与体会

参考文献

- [1] 康华光. 数字电子技术基础（第五版）[M] . 北京：高等教育出版社，2006
- [2] 付家才. 电子实验与实践 [M] . 北京：高等教育出版社，2004

A 程序主要代码

```
1 % 在这里添加主要代码
2 % 例如:
3 % load('data.mat');
4 % result = processData(data);
5 % figure;
6 % plot(result);
```