苏州大学电子信息学院
设计性实验报告

实验名称　基于滤波技术的信号增强

实验者姓名：王旭

合作者姓名：无

专业：通信工程

班级：22 通信 2 班

学号：2215404064

指导老师：朱哲辰

实验日期：12.11

一. 实验目的

1. 通过实验巩固 FIR 数字滤波器的认识和理解。
2. 熟悉掌握 FIR 数字低通滤波器的窗函数设计方法。
3. 了解 FIR 数字滤波器的具体应用。
4. 通过编写 MATLAB 代码来实现滤波器的设计和信号处理，提高使用 MATLAB 进行信号处理的能力。

二. 实验背景

在无线通信系统中，滤波器是一种关键的电子元件，它的作用是通过选择性地传递或抑制特定频率范围内的信号，从而对信号进行频率调整或去除干扰。从而实现频谱整形、抑制干扰、选择性放大、消除多径效应以及降低噪声等作用。

现有一台具有宽带接收能力的软件定义无线电接收机。其接收信号为实验一中的信号一，现要求利用 FIR 数字滤波器对该信号进行不同方式的滤波处理。请利用书本所学知识，更具实验内容通过 MATLAB 编写滤波器函数。计算获得滤波输出结果，并对结果进行分析。

三. 实验准备

1. 汉明窗（Hamming window）是一种在数字信号处理中常用的窗函数，它用于减少频谱泄漏并改善频谱估计的质量。汉明窗的公式可以表示为：

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right), 0 \le n \le N-1$$

其中，N是窗的长度，n是窗内的样本索引。汉明窗的主要特点是在窗的两端有一个平滑的衰减，这有助于减少频谱泄漏，尤其是在快速傅里叶变换（FFT）

2. 评价 FIR 滤波器的性能指标通常涉及以下几个关键方面：

通带截止频率：指在通带内，数字滤波器的频率响应的增益降至 -3dB 的频率点。这个参数决定了滤波器在通带内的频率响应特性。

阻带截止频率：指在阻带内，数字滤波器的频率响应的增益低于 -3dB 的频率点。这个参数决定了滤波器在阻带内的频率响应特性。

通带最大衰减：指数字滤波器在通带内最大可容忍的衰减量。通常以 dB 为单位，用来评估数字滤波器的通带性能。

阻带最小衰减：指数字滤波器在阻带内最小需要达到的衰减量。通常以 dB 为单位，用来评估数字滤波器的阻带性能。

过渡带宽度：指滤波器从通带到阻带的转换区域，越窄意味着滤波器在频率选择上越精确。

相位响应：需要检查滤波器的相位响应是否符合实际要求。

3. 在实验一中，我们得到信号 x1.mat 四个主要频率成分分别为 1700Hz，1850Hz，3550Hz，3750Hz，本次实验二的相关处理都基于这几个频率展开。

图 1 实验一中四个主要频率成分

四. 实验内容

1. 滤波器一

请利用哈明窗的方法设计一个阶数 N=33，频率为 2000Hz 的 FIR 低通滤波器。结合实验一中信号一的频谱特点，分析滤波结果及滤波器性能。

在 MATLAB 中，我们可以使用 fir1 函数结合哈明窗（Hamming window）设计一个 FIR 低通滤波器。

```
1. % 使用 hamming 窗设计 FIR 滤波器
2. b = fir1(N, Fc_normalized, hamming(N+1));


N：阶数  Fc_normalized：归一化截止频率  hamming(N+1)：滤波器类型
```

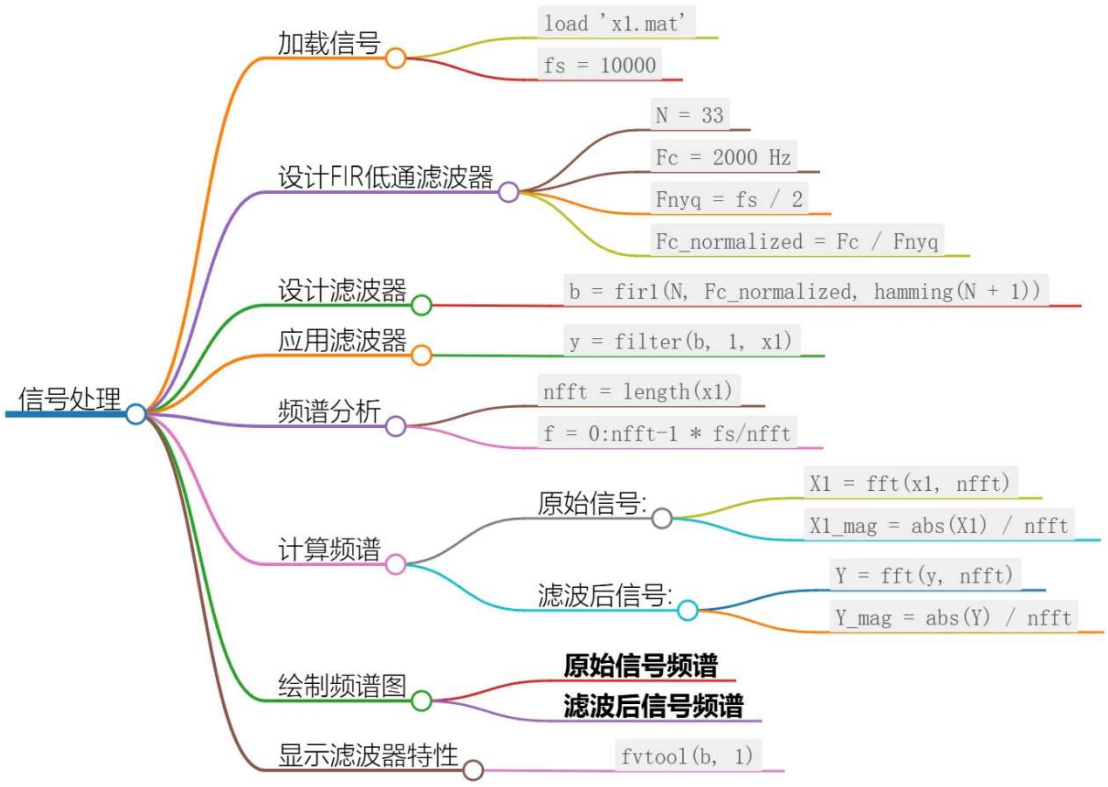

图 2 实验一思维导图

实验结果如下：

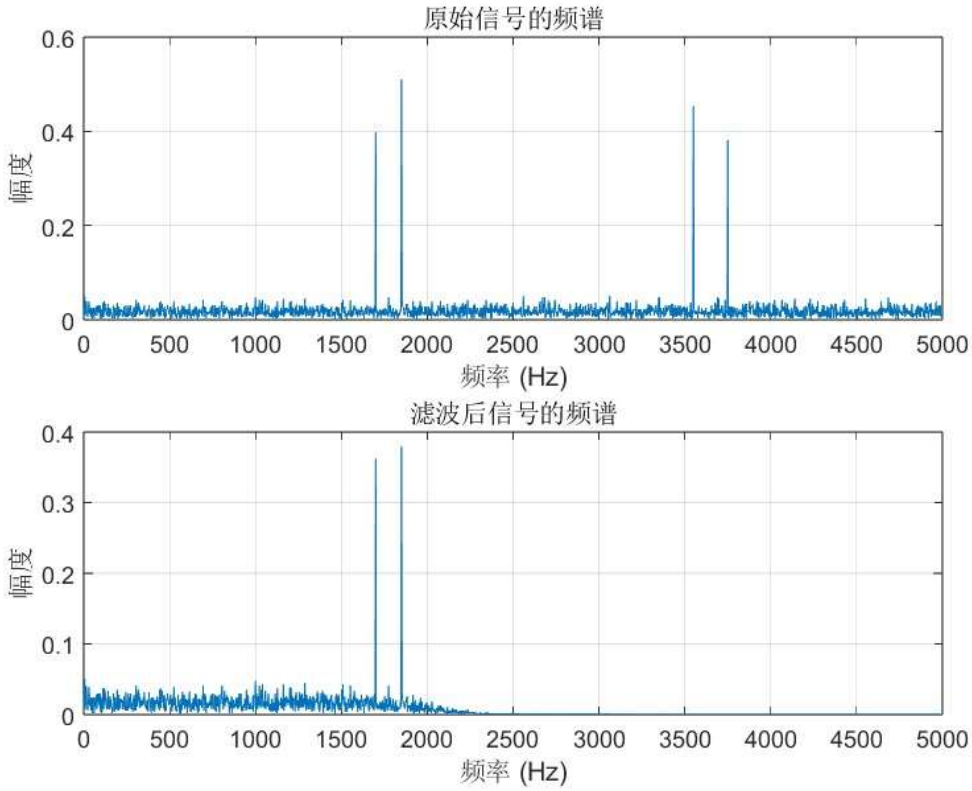

图 3 实验一结果频谱分析

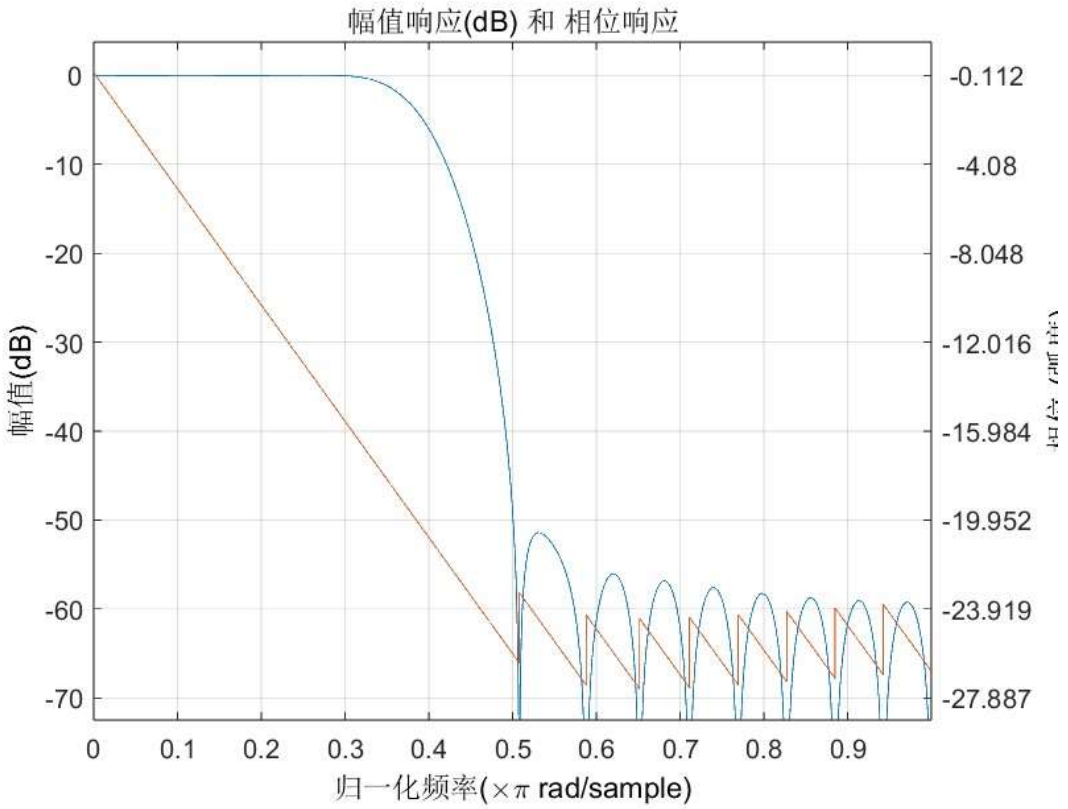

图 4 实验一结果幅值响应和相位响应

通过比较两个频谱图，我们可以发现滤波后的信号在 2000Hz 以下的频率成分被保留，而 2000Hz 以上的频率成分被有效抑制。

通带特性：在 0 到 2000Hz 的频率范围内，滤波后的信号幅度与原始信号相近，且没有明显的衰减。

阻带特性：在 2000Hz 以上的频率范围内，滤波后的信号幅度显著减小，形成明显的衰减。

相位特性：相位变化相对平稳，说明滤波器的相位失真较小，适合对波形保真度要求较高的应用。一旦到达截止频率，此平滑度保持不变，表示在通带和阻带之间没有剧烈的相位变动。

过渡带较窄，意味着频率从通带到阻带的变化较为平滑，降低了对信号的失真。

总评：该滤波器在设计上表现出良好的通带特性和相对平稳的相位响应，适合于需要保持信号完整性的应用。对于高频噪声的衰减能力也十分有效。

2. 滤波器二

请选择合适的窗函数方法及相关参数，设计滤波器使得实验一信号一中两个频率较低的成分得以保留而两个频率较高的成分被过滤。要求使用尽可能低的阶数 N，同时保留成分的衰减小于 3 dB，过滤成分衰减大于 50 dB。通过滤波结果分析所设计滤波器的性能。

设置的截止频率会影响滤波器的参数，与我们要寻找的阶数有关系。我们通过循环，不断改变截止频率，计算滤波器的最小阶数，从而获得最小阶数以及所对应的截止频率。

设置循环不断更改截止归一化频率：

```
for Fc = linspace(Fc_start, Fc_end, 20)
    Fc_normalized = Fc / Fnyq;
```

需要满足的条件：保留成分的衰减小于 3 dB，过滤成分衰减大于 50 dB

```
% 将增益转换为 dB 并检查是否满足衰减要求
if (20*log10(gain_f1) >= -3 && 20*log10(gain_f2) >= -3 && 20*log10(gain_f3) <= -50 && 20*log10(gain_f4) <= -50)
% 找到满足要求的滤波器
if N < min_N
    min_N = N;
    optimal_N = N;
    optimal_Fc = Fc;
end
```

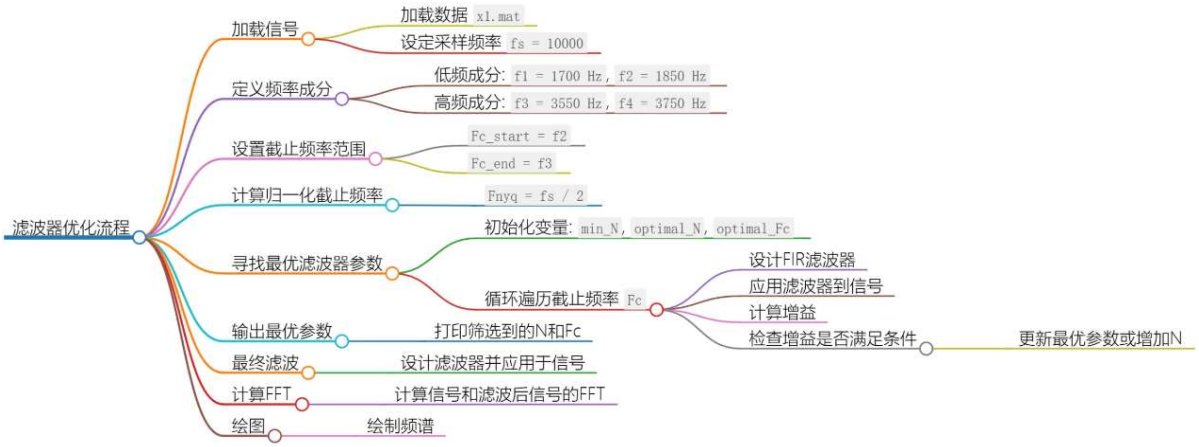

图 5 实验二思维导图

实验结果如下：

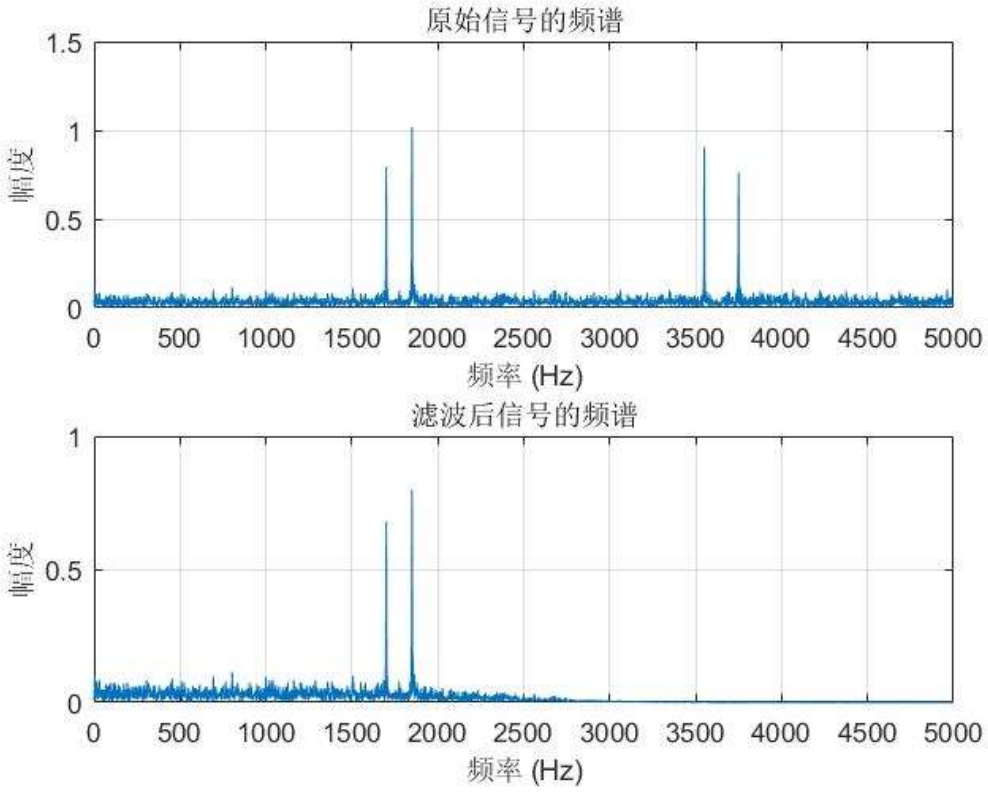

图 6 实验二结果频谱分析

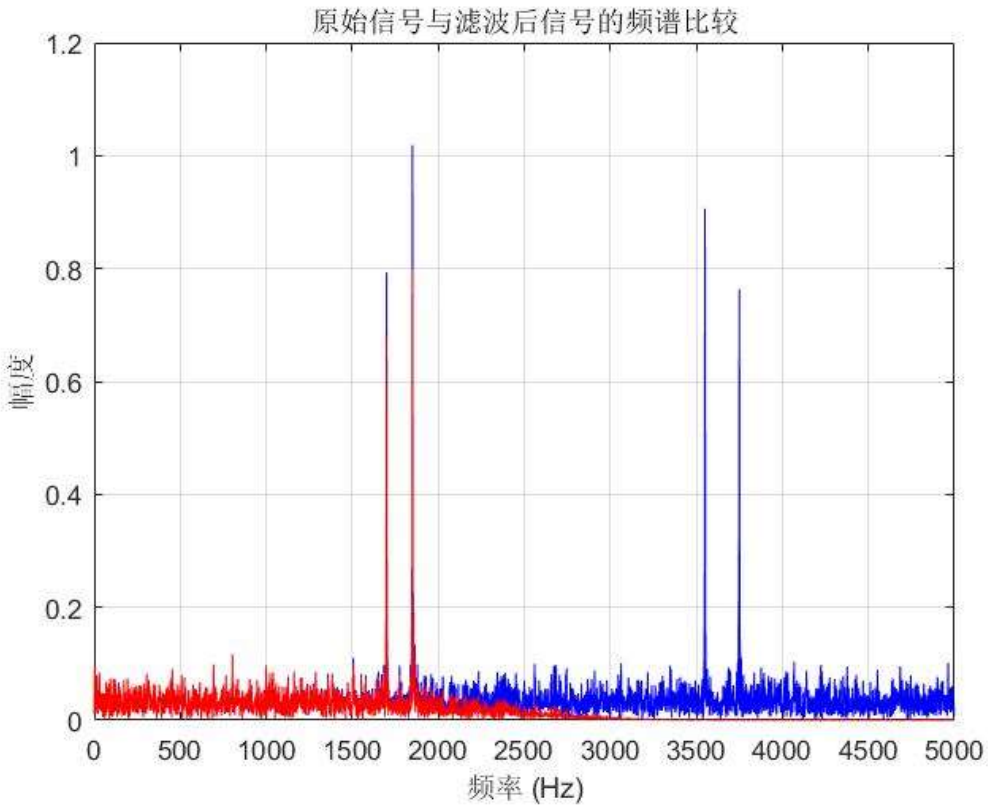

图 7 实验二结果比较

```
>> NO2
最小满足要求的滤波器阶数N为：13，对应的截止频率Fc为：2297.37 Hz
```

图 8 实验二输出结果

得到结果：最小满足要求的滤波器阶数 N 为：13，对应的截止频率 Fc 为：2297.37Hz。

从结果可以看出，该滤波器在抑制特定频率信号（高通）方面表现良好，对于目标滤除 1700Hz 和 1850Hz 而言，该滤波器成功实现用较小的阶数 N 对这两个频率的过滤实现较大的衰减。

3. 滤波器三：其他条件与滤波器二相同，但保留两个高频成分并过滤两个低频成分。

同步骤二，更改需要截止的频率为两个高频成分，同时在设计 hamming 窗时调用“high”高通滤波（无说明默认低通）

```
1.  b = fir1(optimal_N, Fc_normalized, 'high', hamming(optimal_N + 1));
```

实现结果实现如下：

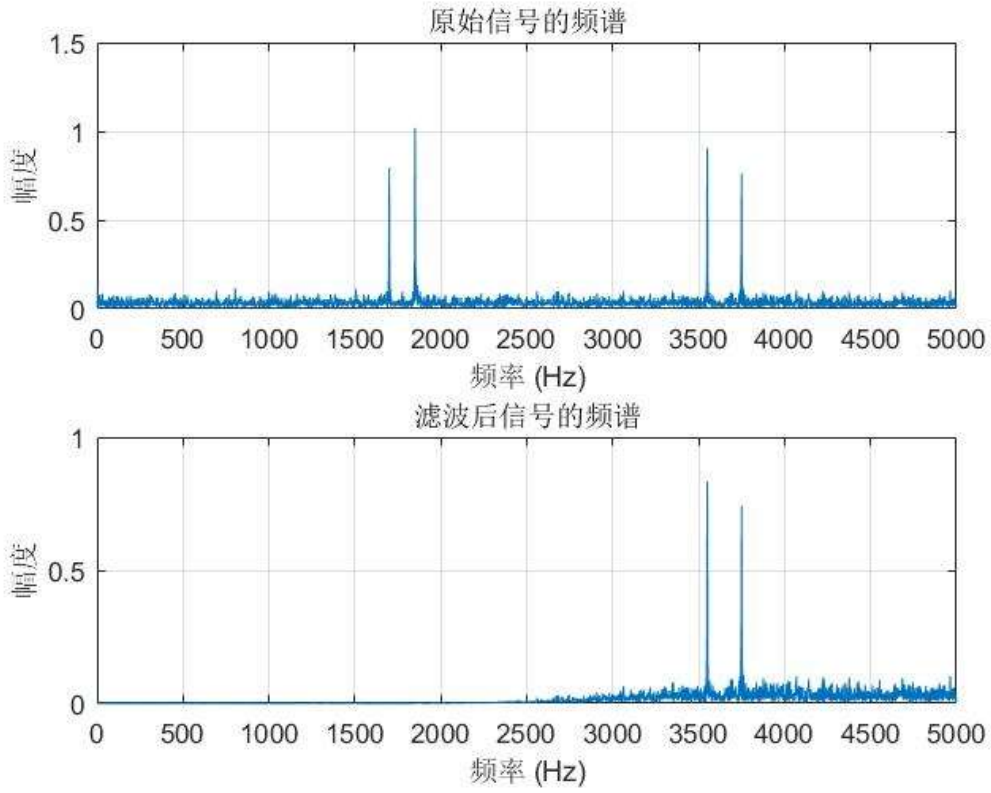

图 9 实验三结果频谱分析

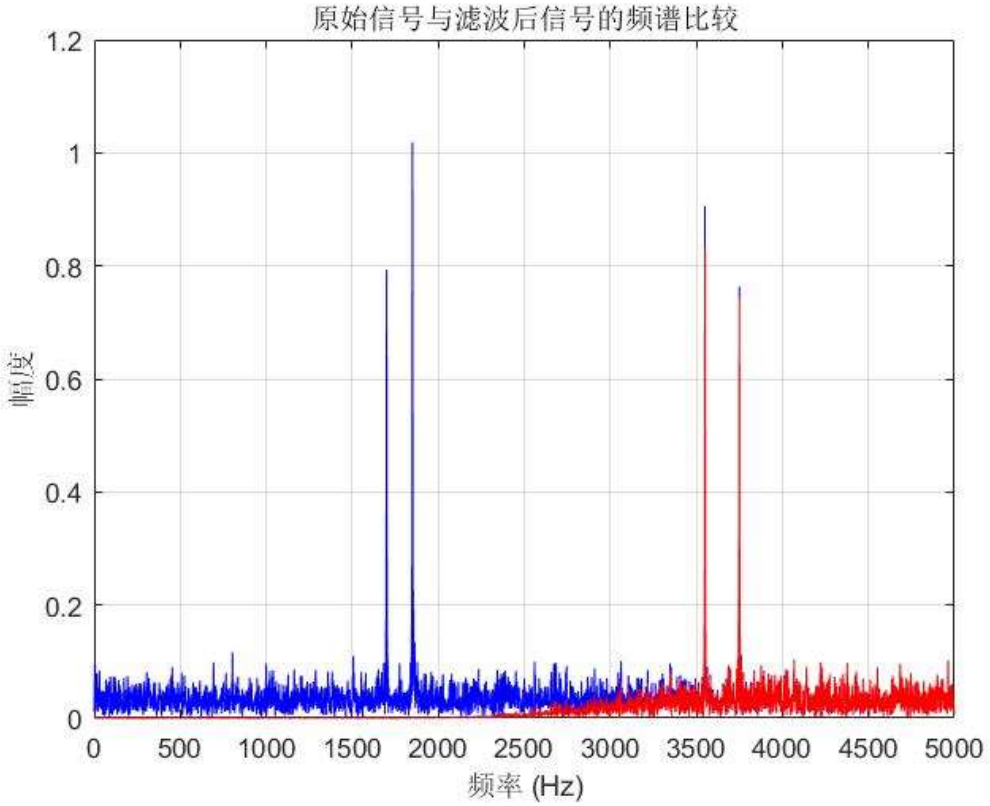

图 10 实验三结果比较

```
>> N03
最小满足要求的滤波器阶数 N 为：16，对应的截止频率 Fc 为：2923.68 Hz
>>
```

图 11 实验三输出结果

得到结果：最小满足要求的滤波器阶数 N 为：16，对应的截止频率 Fc 为：2923.68 Hz。

从结果可以看出，该滤波器在抑制特定频率信号（高通）方面表现良好，对于目标滤除 1700Hz 和 1850Hz 而言，该滤波器成功实现用较小的阶数 N 对这两个频率的过滤实现较大的衰减。

4. 滤波器四：设计滤波器，并实现 OCT 眼底图像的降噪处理。

在此步中，决定使用两种滤波器分别实现对图像的降噪处理：

1. 高斯滤波

我们通常图像上说的高斯滤波，指的是高斯模糊(Gaussian Blur)，高斯模糊也叫高斯平滑，是图像处理中常用的一种技术，主要用来降低图像的噪声和减少图像的细节。是一种高斯低通滤波，其过滤掉图像高频成分（图像细节部分），保留图像低频成分（图像平滑区域），所以对图像进行‘高斯模糊’后，图像会变得模糊（中心像素的像素值为由周围像素的像素值的和的平均值。这种“平滑化”处理在数值上表现为数值的平滑，在图形上则表现为图像细节的模糊化，中心点失去细节）。

高斯模糊是一种图像模糊滤波器，它用正态分布计算图像中每个像素的变换。N 维空间正态分布方程为

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2}^N} e^{-r^2/(2\sigma^2)}$$

在二维空间定义为

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/(2\sigma^2)}$$

其中r是模糊半径，σ是正态分布的标准偏差。在二维空间中，这个公式生成的曲面的等高线是从中心开始呈正态分布的同心圆。分布不为零的像素组成的卷积矩阵与原始图像做变换。每个像素的值都是周围相邻像素值的加权平均。原始像素的值有最大的高斯分布值，所以有最大的权重，相邻像素随着距离原始像素越来越远，其权重也越来越小。这样进行模糊处理比其它的均衡模糊滤波器更高地保留了边缘效果，参见尺度空间实现。

2. 中值滤波

中值滤波处理图像的原理是基于排序统计理论的一种非线性信号处理技术。它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值。这种方法特别适用于去除椒盐噪声等孤立噪声点，同时能够保持图像的边缘特性，不会使图像产生显著的模糊。

中值滤波特别适用于去除椒盐噪声等孤立噪声点，因为噪声点的灰度值通常与周围像素点的灰度值差异较大，通过中值滤波可以有效地将其去除。与高斯滤波等线性滤波方法相比，中值滤波能够更好地保持图像的边缘特性，因为中值滤

波是基于排序统计的，对边缘像素点的处理更加鲁棒。而且中值滤波的实现方法相对简单，计算量较小，因此在实际应用中具有较高的效率。

在 matlab 中，均可以使用函数来便捷地实现这两种滤波算法：

```
% 1. 高斯滤波器
sigma = 1; % 设置标准差
hsize = 5; % 设置滤波器大小
gaussianFilter = fspecial('gaussian', hsize, sigma);
gaussianFiltered = imfilter(double(image), gaussianFilter, 'replicate');
```

```
% 2. 中值滤波器
medianFiltered = medfilt2(image, [5 5]); % 5x5 的邻域
```

下面是两个滤波处理 OCT 眼底图片的结果：

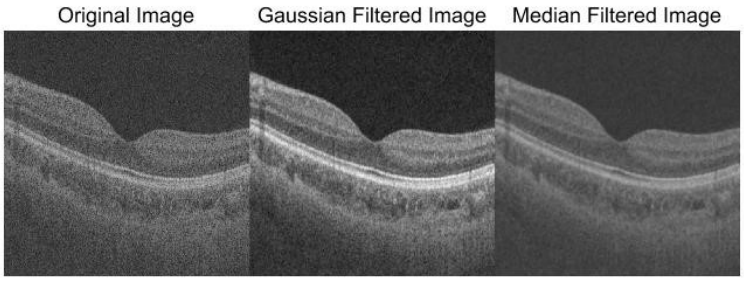

图 12 实验四滤波结果

处理后的两图均实现了对原图像的降噪处理，使图片变得更加平滑。

可以看到，对图像进行‘高斯模糊’后，图像会变得模糊；中值滤波同时能够保持图像的边缘特性。

五. 结论与总结

本次实验旨在设计和评估四种不同类型的 FIR 滤波器，以处理和分析实验一中信号一的频谱特性，并对 OCT 眼底图像进行降噪处理。实验一成功设计了一

个阶数为 33，截止频率为 2000Hz 的低通滤波器。通过哈明窗法，该滤波器在通带内保持了较小的波动，并在阻带内实现了快速衰减。根据信号一的频谱特点，实验设计了一个低通滤波器，保留了两个低频成分，同时抑制了两个高频成分。该滤波器在保留成分的衰减小于 3dB，过滤成分衰减大于 50dB。实验三与滤波器二设计条件相同，但保留了两个高频成分，过滤了两个低频成分。该滤波器满足了实验要求，展示了良好的频率选择性。实验四针对 OCT 眼底图像，设计了适合的滤波器进行降噪处理，有效降低了图像噪声，同时保留了重要的图像特征。

所有设计的滤波器均达到了预期的性能指标，展示了良好的频率响应和相位特性。滤波器二和三的设计和测试过程验证了窗函数选择和参数调整对滤波器性能的重要影响。滤波器四的图像处理结果证明了滤波器在实际应用中的有效性，特别是在医学图像处理领域。

通过本次实验，我对 FIR 滤波器的设计和应用有了更深入的理解。实验过程中，我学习到了如何根据信号特性选择合适的窗函数和参数，以及如何评估滤波器的性能。在设计带通和带阻滤波器时，我遇到了一些挑战，尤其是在满足特定衰减要求的同时保持滤波器阶数尽可能低。通过不断尝试和优化，我最终找到了满足要求的设计方案。

我也意识到在滤波器设计和图像处理方面还有许多可以进一步探索和优化的空间。我期待在未来的研究中，能够应用本次实验的经验和知识，探索更高级的滤波技术和算法。

通过本次实验，我不仅提升了自己的实践技能，也加深了对信号处理领域的兴趣和热情。这将是我在学术和职业生涯中宝贵的财富。

六. 参考文献

[1]俞一彪,孙兵.数字信号处理.东南大学出版社.

[2]姜恩华,周建芳,窦德召,等.数字信号处理课程实验教学平台建设[J].湖北师范大学学报(自然科学版),2024,44(04):39-44.

[3]王逸飞,高磊,叶曦,等.一种通用数字 fir 滤波器设计方法[J].长江信息通信,2024,37(09):111-115.DOI:10.20153/j.issn.2096-9759.2024.09.034.

[5]李振峰,高丽杰,张太行.基于滤波算法的曝光图像多特征细节识别仿真[J].计算机仿真,2023,40(12):242-246+395.

附录：

实验一代码 NO1.mat

```
load('x1.mat');
fs = 10000;

% 设计 FIR 低通滤波器
N = 33;
Fc = 2000;
Fnyq = fs / 2;
Fc_normalized = Fc / Fnyq; % 归一化截止频率

% 使用 hamming 窗设计 FIR 滤波器
b = fir1(N, Fc_normalized, hamming(N+1));

% 应用滤波器
y = filter(b, 1, x1);

% 频谱分析
nfft = length(x1);
f = (0:nfft-1)*(fs/nfft);

% 计算原始信号的频谱
X1 = fft(x1, nfft);
X1_mag = abs(X1/nfft);

% 计算滤波后信号的频谱
Y = fft(y, nfft);
Y_mag = abs(Y/nfft);

% 绘制频谱图
figure;
subplot(2,1,1);
plot(f, X1_mag);
title('原始信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
xlim([0 fs/2]);
grid on;

subplot(2,1,2);
plot(f, Y_mag);
title('滤波后信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
```

```
xlim([0 fs/2]);
grid on;

% 幅频响应和相频响应分析
fvtool(b, 1);
```

实验二代码 NO2.mat

```
load('x1.mat');
fs = 10000;

% 频率成分
f1 = 1700;
f2 = 1850;
f3 = 3550;
f4 = 3750;

% 设置截止频率的范围
Fc_start = f2;
Fc_end = f3;

% 计算归一化截止频率
Fnyq = fs / 2;

% 尝试不同的截止频率来找到最小的 N
min_N = Inf; % 初始设为无穷大
optimal_N = 0; % 最优阶数
optimal_Fc = Fc_start; % 最优截止频率

for Fc = linspace(Fc_start, Fc_end, 20)
    Fc_normalized = Fc / Fnyq;

    N = 1; % 初始滤波器阶数
    while true
        % 使用 hamming 窗设计 FIR 滤波器
        b = fir1(N, Fc_normalized, hamming(N+1));

        % 应用滤波器到信号上
        y = filter(b, 1, x1);

        % 计算滤波器的频率响应
        [h, f_resp] = freqz(b, 1, 1024, fs);

        % 检查在 f1, f2, f3, f4 处的增益是否满足要求
        gain_f1 = abs(interp1(f_resp, abs(h), f1, 'linear', 'extrap'));
```

```
        gain_f2 = abs(interp1(f_resp, abs(h), f2, 'linear', 'extrap'));
        gain_f3 = abs(interp1(f_resp, abs(h), f3, 'linear', 'extrap'));
        gain_f4 = abs(interp1(f_resp, abs(h), f4, 'linear', 'extrap'));

        % 将增益转换为 dB 并检查是否满足衰减要求
        if (20*log10(gain_f1) >= -3 && 20*log10(gain_f2) >= -3 && 20*log10(gain_f3) <= -50 && 20*log10(gain_f4) <= -50)
            % 找到满足要求的滤波器
            if N < min_N
                min_N = N;
                optimal_N = N;
                optimal_Fc = Fc;
            end
            break;
        else
            N = N + 2;
            if N > 100
                break;
            end
        end
    end
end

fprintf('最小满足要求的滤波器阶数 N 为：%d，对应的截止频率 Fc 为：%.2f Hz\n', optimal_N, optimal_Fc);

% 使用找到的最优参数对信号进行最终滤波
Fc_normalized = optimal_Fc / Fnyq;
b = fir1(optimal_N, Fc_normalized, hamming(optimal_N+1));
y_optimal = filter(b, 1, x1);

% 计算 FFT
nfft = length(x1) * 2;
X = fft(x1, nfft) / length(x1);
Y_optimal = fft(y_optimal, nfft) / length(y_optimal);
f = fs * (0:(nfft/2)) / nfft;

% 绘制原始信号和滤波后信号的频谱
figure;
subplot(2,1,1);
plot(f, 2*abs(X(1:nfft/2+1)));
title('原始信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
```

```
grid on;

subplot(2,1,2);
plot(f, 2*abs(Y_optimal(1:nfft/2+1)));
title('滤波后信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
grid on;

% 进行比较
figure;
plot(f, 2*abs(X(1:nfft/2+1)), 'b', 'DisplayName', '原始信号');
hold on;
plot(f, 2*abs(Y_optimal(1:nfft/2+1)), 'r', 'DisplayName', '滤波后信号');
title('原始信号与滤波后信号的频谱比较');
xlabel('频率 (Hz)');
ylabel('幅度');
grid on;
legend;
hold off;
```

实验三代码 NO3.mat

```
load('x1.mat');
fs = 10000;

% 频率成分
f1 = 1700;
f2 = 1850;
f3 = 3550;
f4 = 3750;

% 设置截止频率的范围
Fc_start = f2;
Fc_end = f3;

% 计算归一化截止频率
Fnyq = fs / 2;

% 尝试不同的截止频率来找到最小的 N
min_N = Inf; % 初始设为无穷大
optimal_N = 0; % 最优阶数
optimal_Fc = Fc_start; % 最优截止频率

for Fc = linspace(Fc_start, Fc_end, 20) %
```

```
    Fc_normalized = Fc / Fnyq;

    N = 2; % 初始滤波器阶数
    while true
        % 使用 Hamming 窗设计 FIR 高通滤波器
        b = fir1(N, Fc_normalized, 'high', hamming(N + 1));

        % 应用滤波器到信号上
        y = filter(b, 1, x1);

        % 计算滤波器的频率响应（用于验证）
        [h, f_resp] = freqz(b, 1, 1024, fs);

        % 检查在 f1, f2, f3, f4 处的增益是否满足要求
        gain_f1 = abs(interp1(f_resp, abs(h), f1, 'linear', 'extrap'));
        gain_f2 = abs(interp1(f_resp, abs(h), f2, 'linear', 'extrap'));
        gain_f3 = abs(interp1(f_resp, abs(h), f3, 'linear', 'extrap'));
        gain_f4 = abs(interp1(f_resp, abs(h), f4, 'linear', 'extrap'));

        % 将增益转换为 dB 并检查是否满足衰减要求
        if (20 * log10(gain_f1) <= -50 && 20 * log10(gain_f2) <= -50 && ...
            20 * log10(gain_f3) > -3 && 20 * log10(gain_f4) > -3)
            % 找到满足要求的滤波器
            if N < min_N
                min_N = N;
                optimal_N = N;
                optimal_Fc = Fc; % 记录当前的截止频率
            end
            break;
        else
            N = N + 2;
            if N > 100
                break;
            end
        end
    end
end

fprintf('最小满足要求的滤波器阶数 N 为：%d，对应的截止频率 Fc 为：%.2f Hz\n', optimal_N,
optimal_Fc);

% 使用找到的最优参数对信号进行最终滤波
Fc_normalized = optimal_Fc / Fnyq;
b = fir1(optimal_N, Fc_normalized, 'high', hamming(optimal_N + 1));
```

```
y_optimal = filter(b, 1, x1);

% 计算 FFT
nfft = length(x1) * 2;
X = fft(x1, nfft) / length(x1);
Y_optimal = fft(y_optimal, nfft) / length(y_optimal);
f = fs * (0:(nfft/2)) / nfft;

% 绘制原始信号和滤波后信号的频谱
figure;
subplot(2,1,1);
plot(f, 2 * abs(X(1:nfft/2 + 1)));
title('原始信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
grid on;

subplot(2,1,2);
plot(f, 2 * abs(Y_optimal(1:nfft/2 + 1)));
title('滤波后信号的频谱');
xlabel('频率 (Hz)');
ylabel('幅度');
grid on;

% 进行比较
figure;
plot(f, 2 * abs(X(1:nfft/2 + 1)), 'b', 'DisplayName', '原始信号');
hold on;
plot(f, 2 * abs(Y_optimal(1:nfft/2 + 1)), 'r', 'DisplayName', '滤波后信号');
title('原始信号与滤波后信号的频谱比较');
xlabel('频率 (Hz)');
ylabel('幅度');
grid on;
legend;
hold off;
```

实验四代码 NO4.mat

```
% 读取图像
image = imread('image.jpg');
if size(image, 3) == 3
    image = rgb2gray(image); % 转换为灰度图像
end

% 1. 高斯滤波器
```

```
sigma = 1; % 设置标准差
hsize = 5; % 设置滤波器大小
gaussianFilter = fspecial('gaussian', hsize, sigma);
gaussianFiltered = imfilter(double(image), gaussianFilter, 'replicate');

% 2. 中值滤波器
medianFiltered = medfilt2(image, [5 5]); % 5x5 的邻域

% 结果展示
figure;
subplot(1, 3, 1);
imshow(image);
title('Original Image');

subplot(1, 3, 2);
imshow(gaussianFiltered, []);
title('Gaussian Filtered Image');

subplot(1, 3, 3);
imshow(medianFiltered);
title('Median Filtered Image');
```