IIR数字滤波器的设计与实现实验报告

班级：2022211116 姓名：方小倩 学号：2022210582

实验步骤

1. 生成发送信号

在Matlab上生成并发送一个模拟信号s（t），使得S(t)= 1\*sin(2\*pi\*50\*t)+ 2\*sin(2\*pi\*100\*t)+ 0.5\*sin(2\*pi\*150\*t)，并用subplot画出图像。

Matlab源代码如下：

%% 生成模拟信号

fs = 1000; % 采样率

t = 0:1/fs:1; % 时间向量

f1 = 50; % 信号频率1

f2 = 100; % 信号频率2

f3 = 150; % 信号频率3

x = 1\*sin(2\*pi\*f1\*t) + ... % 发送信号

2\*sin(2\*pi\*f2\*t) + ...

0.5\*sin(2\*pi\*f3\*t);

subplot(3,1,1); % 创建第一个子图

plot(t, x);

xlabel('时间/s');

ylabel('功率/dB');

xlim([0,0.5]);

title('发送信号');

1. 添加随机噪声

利用randn函数生成信噪比为7的随机噪声，并叠加到发送信号上，得到接收信号，并同样用subplot画出图像。

Matlab源代码如下：

%% 添加随机噪声

SNR\_dB = 7; % 信噪比 (dB)

SNR = 10^(SNR\_dB/10); % 信噪比(线性值)

P\_signal = rms(x)^2; % 信号功率

P\_noise = P\_signal / SNR; % 噪声功率

noise = sqrt(P\_noise) \* randn(size(x)); % 生成随机噪声

y = x + noise; % 叠加噪声

subplot(3,1,2); % 创建第二个子图

plot(t, y);

xlabel('时间/s');

ylabel('功率/dB');

xlim([0,0.5]);

title('接收信号');

1. 生成Butterworth滤波器

调用matlab中的滤波器设计函数designfilt，设计一个阶数为20，上3-dB频率为120Hz，下-3dB频率为80Hz且采样率为1000Hz的IIR带通滤波器，并查看滤波器的幅值响应图。再将叠加随机噪声的信号通过该滤波器，利用subplot画出滤波后的信号

Matlab源代码如下：

%% 设计 IIR 滤波器

% 带通 IIR 滤波器，20 阶，下3-dB 频率为80 HZ，上3-dB 频率为120 HZ

% 滤波器类型为 Butterworth，采样率为 1000 HZ

bpFilt = designfilt('bandpassiir','FilterOrder',20,...

'HalfPowerFrequency1',80,'HalfPowerFrequency2',120,...

'DesignMethod','butter', ...

'SampleRate' ,fs);

fvtool(bpFilt) % 查看滤波器幅值响应

filt\_rx = filter(bpFilt,y);% 进行滤波

subplot(3,1,3); % 创建第三个子图，显示滤波后的信号

plot(t, filt\_rx);

xlabel('时间/s');

ylabel('功率/dB');

xlim([0,0.5]);

title('滤波后的信号');

1. 对比实验，改变滤波器参数

绘制发送信号、接收信号，以及滤波后的信号的时域、频域波形，观察。之后改变Butterworth参数，先将阶数从20调整为10，观察滤波后信号时频域波形的变化；再令阶数变为20，改上下-3dB分别为140Hz和60Hz，观察滤波后信号时频域波形变化。（Butterworth参数变化需在上一段代码中改变）

Matlab源代码如下：

%% 绘制频域波形图

figure(2);

N = length(x);%信号长度

f= (0:N-1)\*(fs/N);% 频率向量

subplot(3,1,1)

X=fft(x)/N;% FFT 变换并归一化

plot(f,abs(X))

xlabel("频率/Hz");

ylabel('功率/dB');

title('发送信号');

subplot(3,1,2)

RX = fft(y)/N;% FFT 变换并归一化

plot(f,abs(RX))

xlabel('频率/Hz');

ylabel('功率/dB');

title('接收信号');

subplot(3,1,3);

FILT\_RX = fft(filt\_rx)/N;% FFT 变换并归一化

plot(f,abs(FILT\_RX))

xlabel('频率/Hz');

ylabe1("功率/dB");

title('滤波信号');

实验数据分析

1. 生成模拟信号

在生成模拟信号时，使用了三个不同频率的正弦波成分，分别是50hz、100hz和150hz。

这样做可以使实验更接近真实情况，因为真实信号往往是由多个频率成分叠加而成的。

1. 生成随机噪声

在引入随机噪声可以模拟信号传输过程中的干扰，例如来自电磁干扰、电路噪声等。且设置信噪比为7dB，以确保噪声对信号的影响在一个合理的范围内，同时仍然能够清晰地观察到滤波效果。

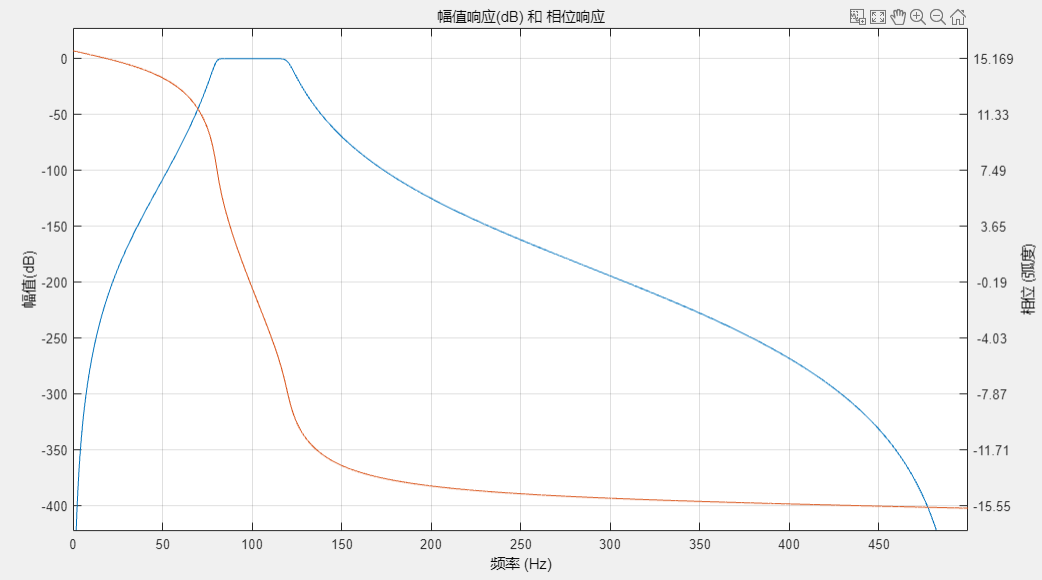
1. 设计滤波器

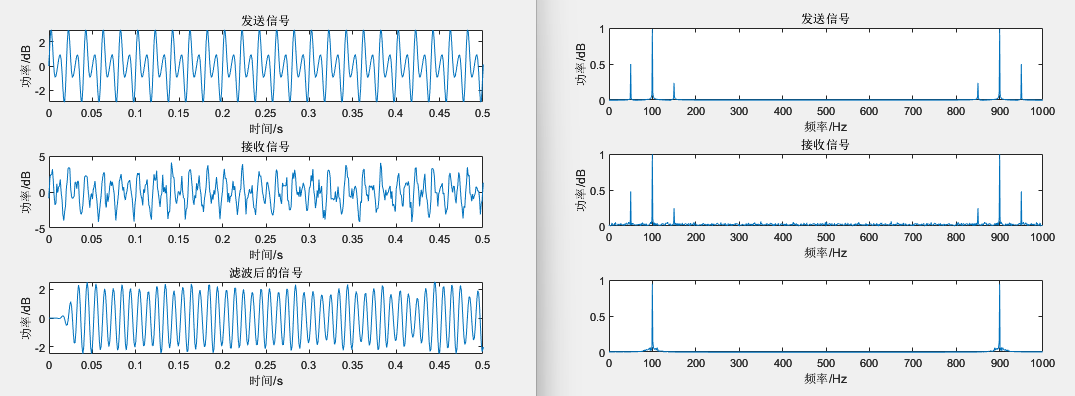
设计了一个80—120hz的20阶带通滤波器，以保留该频率范围内的信号成分，并抑制其他频率的干扰信号。体现在该信号上就是滤除了50hz和150hz的正弦信号，留下的100hz的正弦信号。

实验结果分析

1. 阶数为20，通带为80—120Hz

20阶，通带为80—120Hz的实验结果如下：

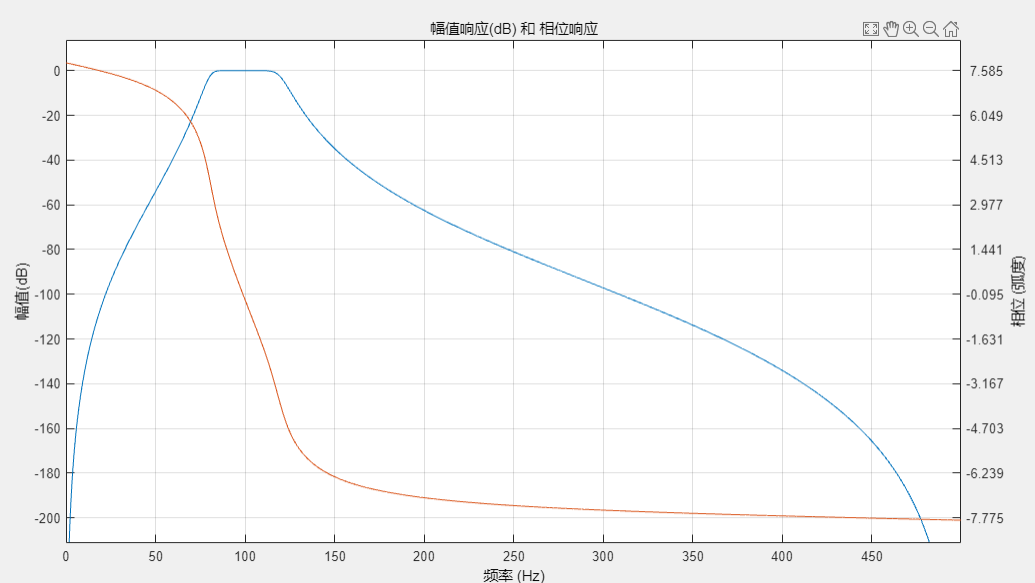


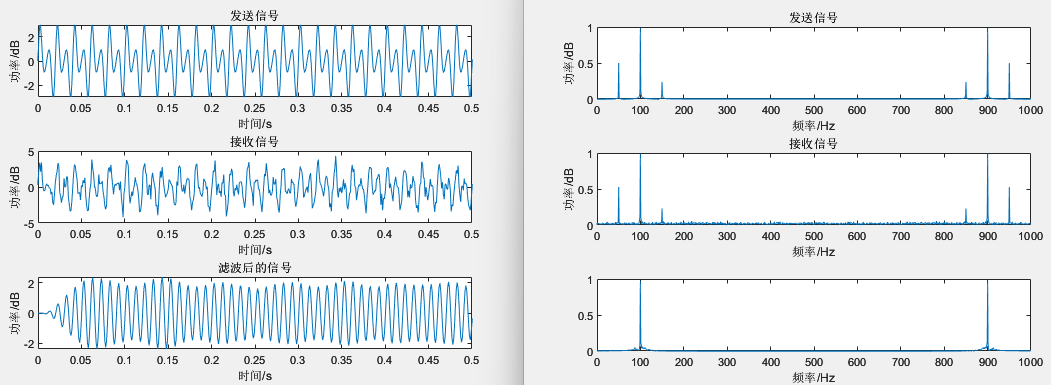


由结果可知，所得滤波器通带在80—120hz，幅值为0dB，通带边缘锐利。该滤波器使得输出的滤波后信号50hz和150hz的频率分量都被滤除，只剩下100hz的频率分量，且该分量与原信号幅值一致。由图还可以看出，信号中的随机噪声大部分也被滤除了，只剩下通带内的小部分噪声。

1. 阶数为10，通带为80—120Hz

10阶，通带为80—120Hz的实验结果如下：





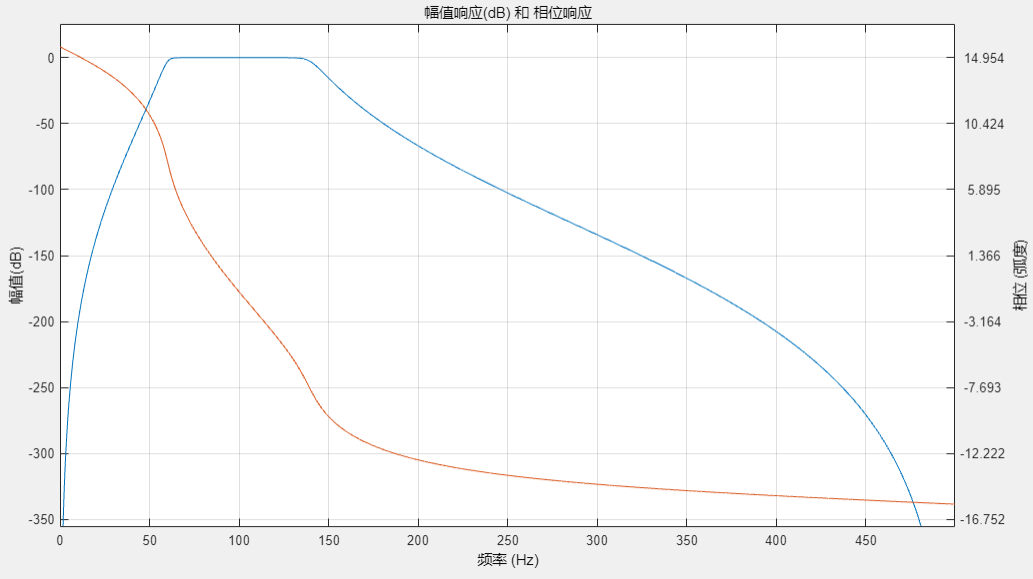
由结果可知，该10阶滤波器的实验结果大体上与20阶的一致，不同地地方如下：

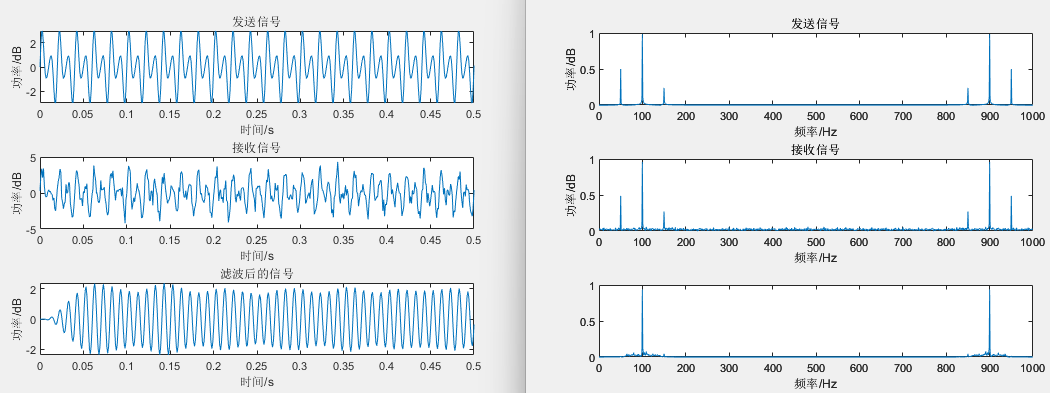
10阶滤波器的自由度更少，故频率响应比起20阶的更加平滑，观察滤波器幅值响应可发现该滤波器通带边缘较平滑。且10阶滤波器的相位范围比20阶的相位范围来的小，10阶约为7到-7弧度，20阶约为14到-14的弧度。

观察时域的滤波后信号可以看出，10阶的滤波器还引入了一些相位延迟。

1. 阶数为20，通带为60—140Hz

20阶，通带为60—140Hz的实验结果如下：





该滤波器的变化，最显著的就是通带变宽，导致相位的变化在更宽的频率范围内完成了。

可以观察到滤波后的信号有更多频段的噪声被保留了下来。

该滤波后信号的时域图中，也显示出了一些相位的失真。