**实验四 FIR数字滤波器的设计与实现**

一，实验原理

FIR滤波器的设计方法以直接逼近所需离散设计系统的频率响应为基础。**FIR滤波器可以很容易地获得线性相移特性，不存在不稳定地问题。**

FIR滤波器的设计，通常有窗函数设计法，频率抽样设计法和最佳逼近设计法。**窗函数设计法比较简单，它的频率特性是理想滤波器频谱与窗的频谱的卷积，因而，其频率特性取决于窗的类型和长度。**频率抽样设计法比较直观，但由于频域的采样会造成时域的混叠，从而滤波器性能不可能很高，为提高滤波器的性能，可以在过渡带加上0到1之间的过渡点。

二.实验代码，结果以及结果分析

1，用conv函数求两个已知序列的卷积和，并绘制三个序列的波形。

代码如下：

**clear;**

**%4.1**

**M=20;**

**N=10;**

**x=zeros(1,21);**

**y=zeros(1,11);**

**for i=1:(M+1)**

**x(i)=0.8^(i-1);**

**end**

**for j=1:(N-1)**

**y(j)=1;**

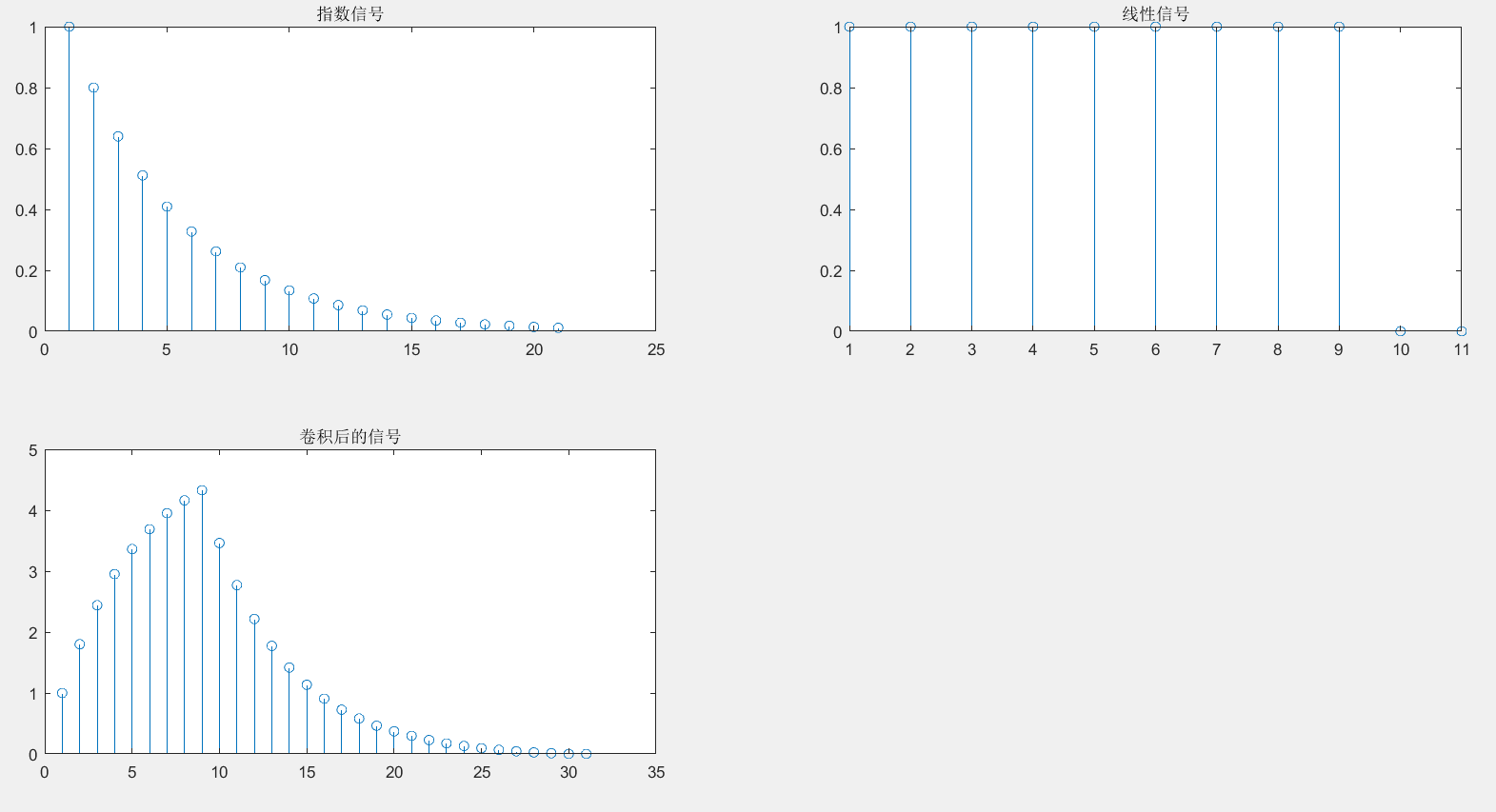
**end**

**z=conv(x,y);**

**subplot(2,2,3),stem(z),title('卷积后的信号');**

**subplot(2,2,1),stem(x),title('指数信号');**

**subplot(2,2,2),stem(y),title('线性信号');**

运行结果：  


2，利用wavread函数（或者audioread函数）将加噪声的音频文件读入Maatlab的工作空间，利用load函数从h50.mat文件中读取50阶离散时间带通滤波器的脉冲相应h1[n]，用conv函数求两个序列的卷积和，并利用wavwrite函数将结果写入规定的音频文件，，绘制两个音频信号的波形。

代码如下：

**% %4.2**

**[y,fs]=audioread('D:\文档\DSP\实验四的数据和系数\第一场雪加噪声.wav');%加入音频**

**load('D:\文档\DSP\实验四的数据和系数\h50.mat');%加入滤波器**

**hn=h50;**

**stem(hn);**

**y1=y';**

**y2=y1(1,1:end);**

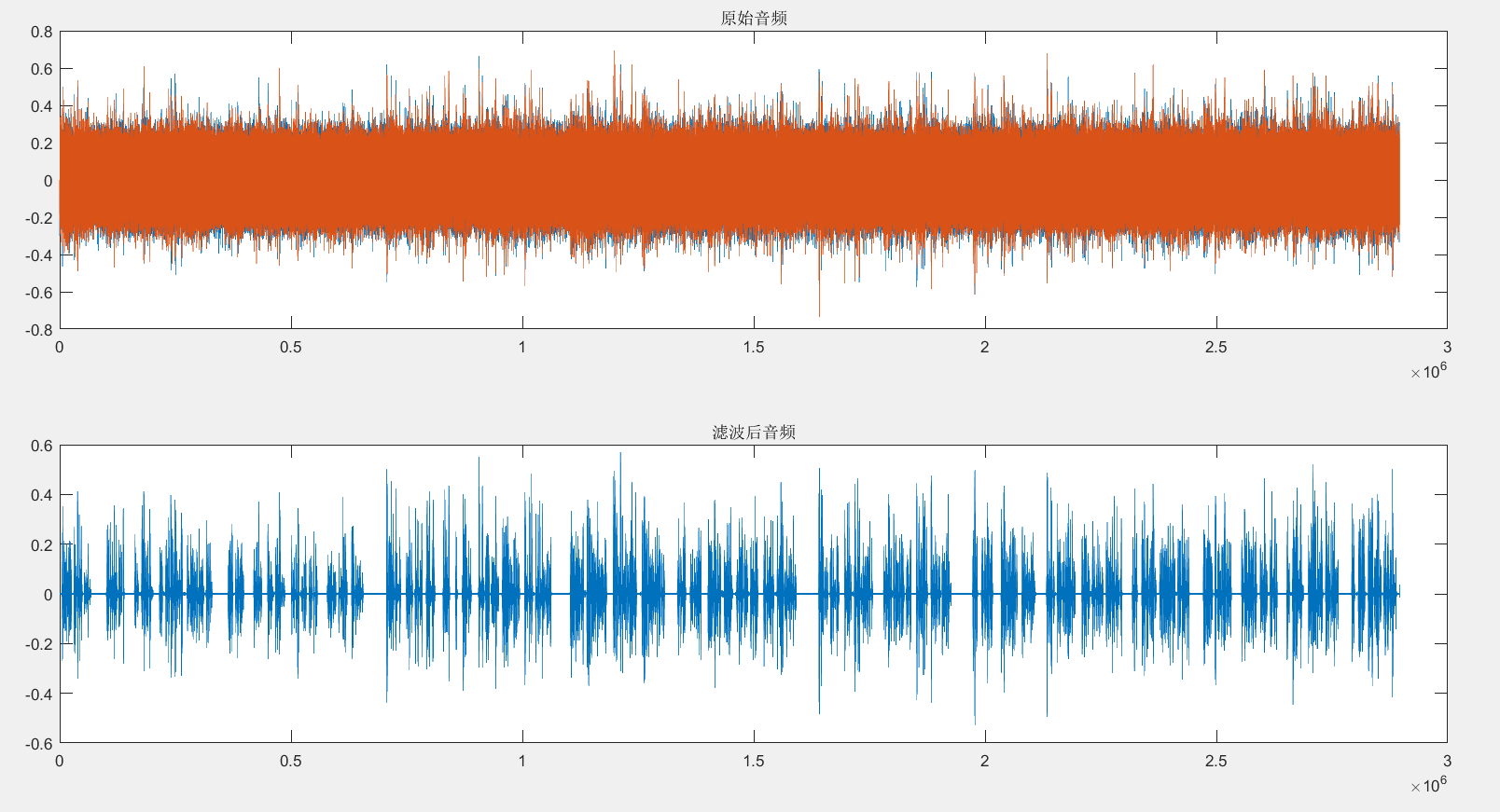
**z=conv(y2,hn);**

**subplot(2,1,2),plot(z),title('滤波后音频');**

**subplot(2,1,1),plot(y),title('原始音频');%绘制音频信号波形**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪 50.wav',z,fs);**

运行结果：



**结果分析：经过处理后的音频文件中的电流声明显减小。**

3，利用wavread函数（或者audioread函数）将加噪声的音频文件读入Maatlab的工作空间，利用load函数从h50.mat文件中读取471阶离散时间带通滤波器的脉冲相应h2[n]，用conv函数求两个序列的卷积和，并利用wavwrite函数将结果写入规定的音频文件，，绘制两个音频信号的波形。

代码如下：

**%4.3**

**[y,fs]=audioread('D:\文档\DSP\实验四的数据和系数\第一场雪加噪声.wav');%加入音频**

**load('D:\文档\DSP\实验四的数据和系数\h471.mat');%加入滤波器**

**hn=h471;**

**stem(hn);**

**y1=y';**

**y2=y1(1,1:end);**

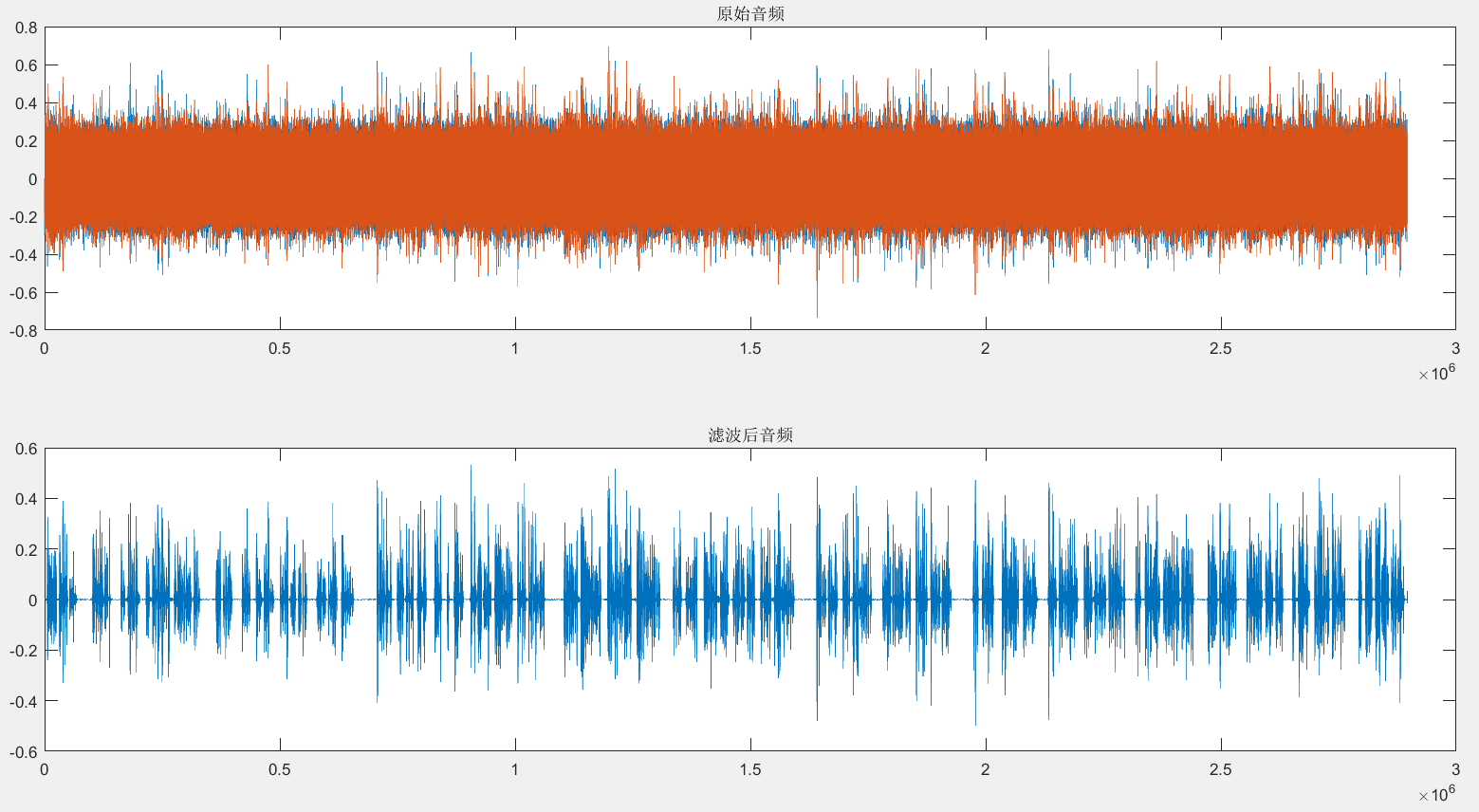
**z=conv(y2,hn);**

**subplot(2,1,2),plot(z),title('滤波后音频');**

**subplot(2,1,1),plot(y),title('原始音频');%绘制音频信号波形**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪 471.wav',z,fs);**

运行结果：



**结果分析：这次滤波后得到的音频文件完全没有了电流声。**

4，采用凯泽窗设计一个具有一下指标的FIR带通滤波器，，画出它的频率和相频响应。

利用wavread函数（或者audioread函数）将加噪声的音频文件读入Maatlab的工作空间，基于设计得到的FIR的带通滤波器，分别采用滤波和卷积的方法，，得到音频数据的滤波结果，并利用wavwrite函数将结果写入规定的音频文件，，绘制三个音频信号的波形。

代码如下：

**%4.4**

**fs=16000;**

**fl\_kaiser=[100 200 3200 3300];**

**fl\_mag=[0 1 0];**

**fl\_dev=[0.05 10^-2.5 0.05];**

**[fl\_n\_kaiser,fl\_wn,fl\_beta,fl\_ftype]=kaiserord(fl\_kaiser,fl\_mag,fl\_dev,fs);**

**figure(1);**

**h=fir1(50,fl\_wn,fl\_ftype,kaiser(50+1,fl\_beta));%生成滤波器**

**freqz(h,1);%观察滤波器的幅频响应时先把下面的注释掉**

**[x,fs]=audioread('D:\文档\DSP\实验四的数据和系数\第一场雪加噪声.wav');%加入音频**

**x1=x';**

**x2=x1(1,1:end);**

**y1=filter(h,1,x2);**

**y2=conv(h,x2);**

**figure(2);**

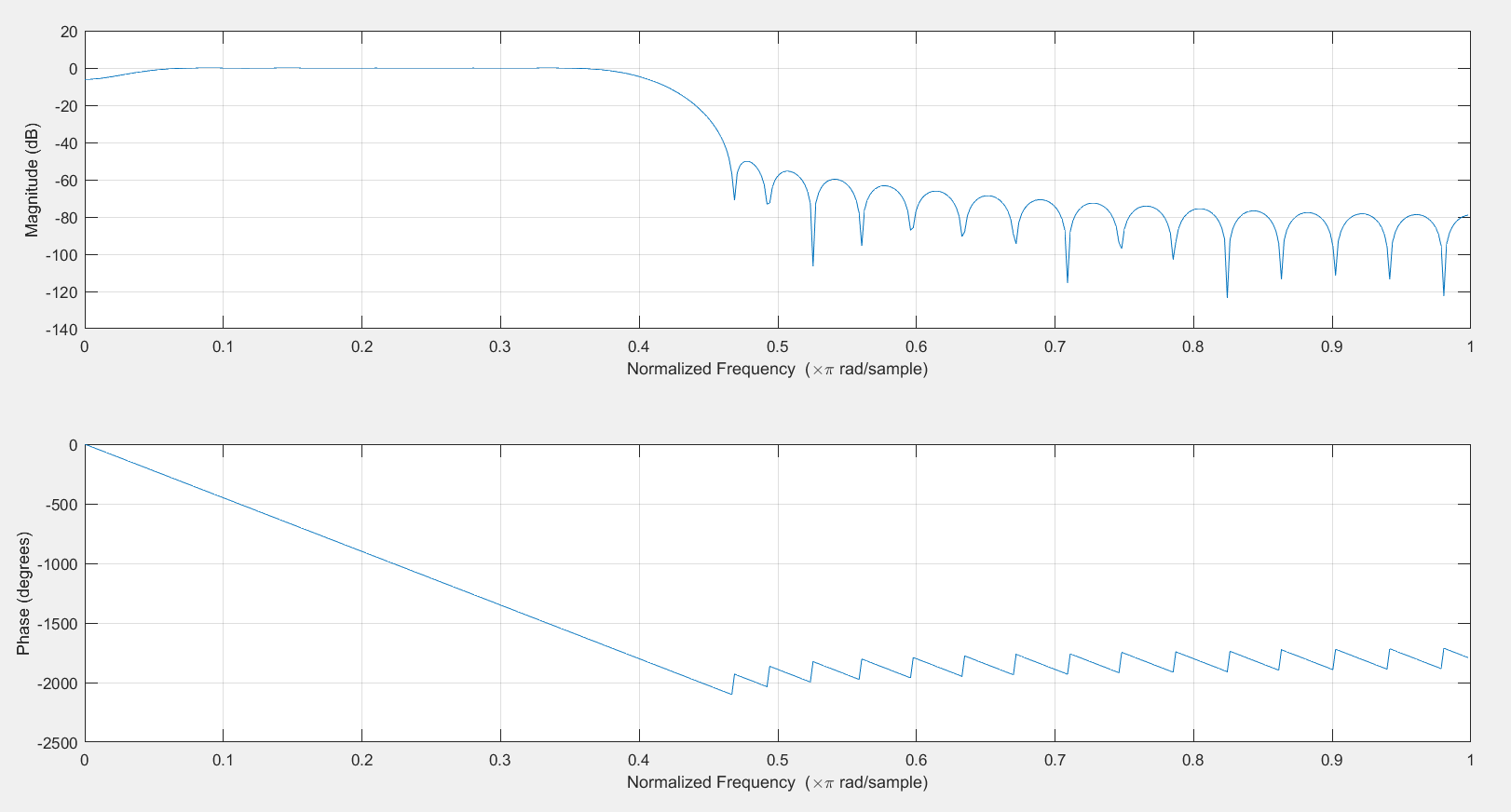
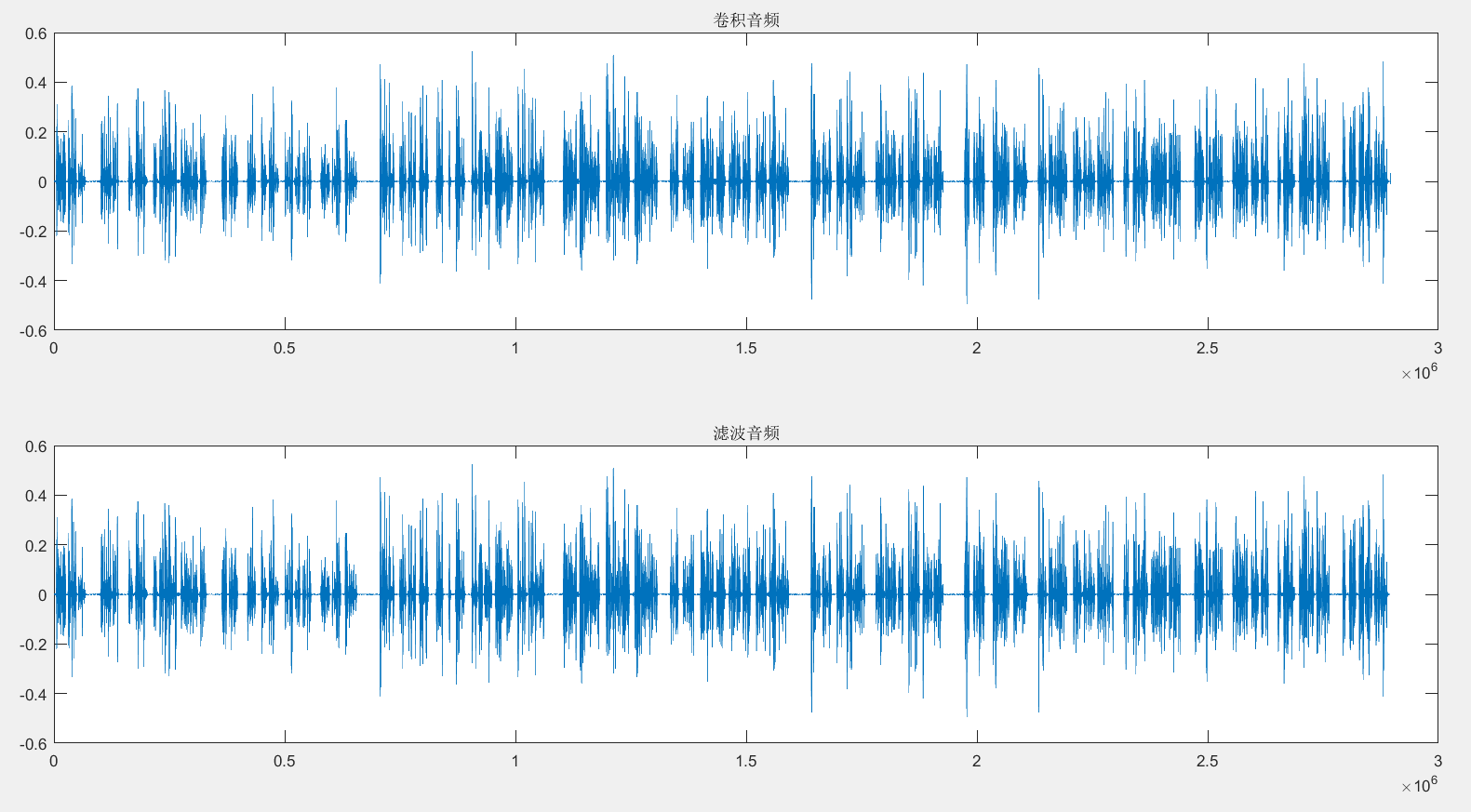
**subplot(2,1,2),plot(y1),title('滤波音频');**

**subplot(2,1,1),plot(y2),title('卷积音频');%绘制音频信号波形**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪filter滤波50.wav',y1,fs);**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪conv滤波50.wav',y2,fs);**

运行结果：



**结果分析：两种方式的滤波都将电流的噪声很好的过滤了出去，但两者没有明显区别。**

5，采用凯泽窗设计一个具有一下指标的FIR带通滤波器，，画出它的频率和相频响应。

利用wavread函数（或者audioread函数）将加噪声的音频文件读入Maatlab的工作空间，基于设计得到的FIR的带通滤波器，分别采用滤波和卷积的方法，，得到音频数据的滤波结果，并利用wavwrite函数将结果写入规定的音频文件，，绘制三个音频信号的波形。

代码如下：

**%4.5**

**fs=16000;**

**fl\_kaiser=[100 200 3200 3300];**

**fl\_mag=[0 1 0];**

**fl\_dev=[0.05 10^-2.5 0.05];**

**[fl\_n\_kaiser,fl\_wn,fl\_beta,fl\_ftype]=kaiserord(fl\_kaiser,fl\_mag,fl\_dev,fs);**

**h=fir1(fl\_n\_kaiser,fl\_wn,fl\_ftype,kaiser(fl\_n\_kaiser+1,fl\_beta));%生成滤波器**

**figure(1);**

**freqz(h,1);%观察滤波器的幅频响应时先把下面的注释掉**

**[x,fs]=audioread('D:\文档\DSP\实验四的数据和系数\第一场雪加噪声.wav');%加入音频**

**x1=x';**

**x2=x1(1,1:end);**

**y1=filter(h,1,x2);**

**y2=conv(h,x2);**

**figure(2);**

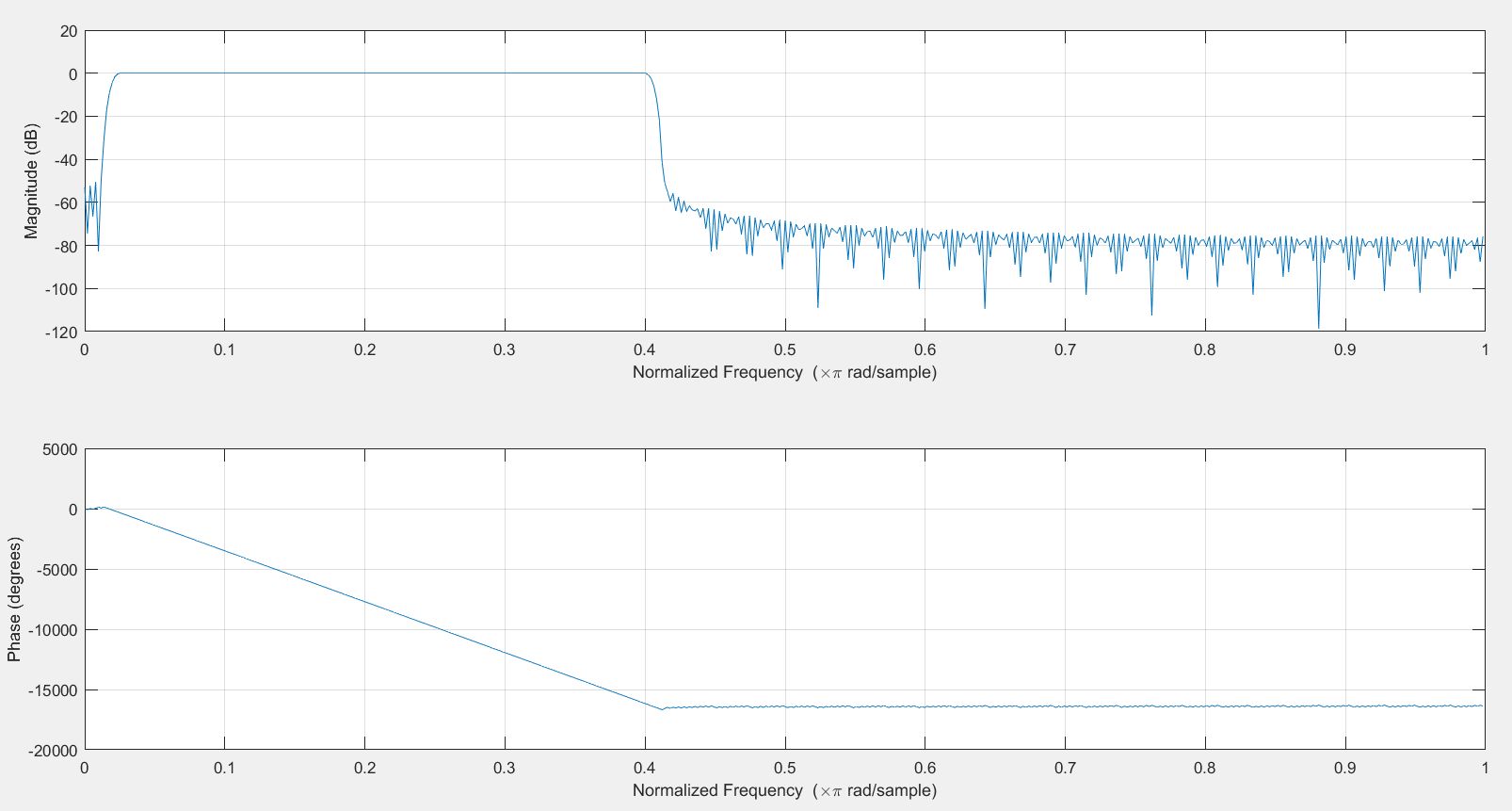
**subplot(2,1,2),plot(y1),title('滤波音频');**

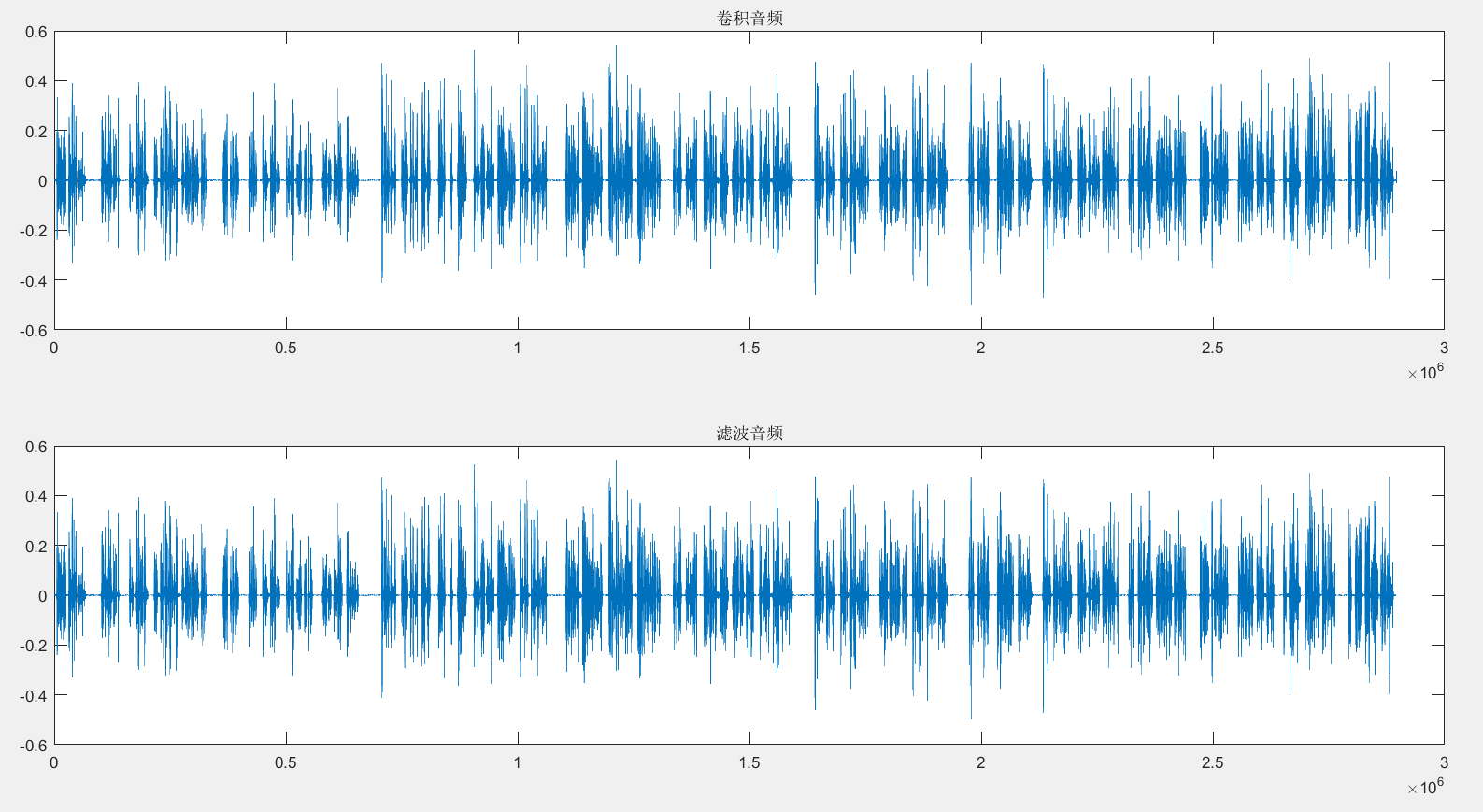
**subplot(2,1,1),plot(y2),title('卷积音频');%绘制音频信号波形**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪filter滤波(M).wav',y1,fs);**

**audiowrite('D:\文档\DSP\实验四的数据和系数\第一场雪conv滤波(M).wav',y2,fs);**

运行结果：





**结果分析：两种方式的滤波都将电流的噪声很好的过滤了出去，但两者没有明显区别。**