### 实验四 音频信号的数字化处理实现

##### 一、实验目的

1、增强对信号数字化处理的理解。

2、巩固IIR、FIR数字滤波器的滤波实现方法

3、掌握利用MATLAB完成音频信号数字滤波的基本方法。

4、体验并感受音频信号数字化处理前后的差异。

##### 二、理论依据

###### （一）数字化滤波

单位脉冲响应为的线性时不变离散时间系统，其输入与输出的关系可表述如下：

（1）

由于FIR数字滤波器的单位脉冲响应为有限长度，当输入信号长度也为有限长是，FIR数字滤波器可以采用上述关系式直接实现滤波。

对于IIR数字滤波器，由于其单位脉冲响应是无限长，因此需要从频域入手，假如的傅里叶变换存在，则有输入、输出的频域关系：

（2）

然后通过傅里叶反变换即可得到滤波结果。实际实现过程中，傅里叶变换一般是通过快速离散傅里叶变换来实现，而快速离散傅里叶变换是傅里叶变换的有限点的采样值，即是有限长序列。因此只要采样点数大于序列长度，就可以有效避免混叠失真。

IIR线性时不变系统的系统函数为：

对应的常系数线性差分方程为：

一般处理的信号都是从0时刻开始取值的有限长序列，因此IIR数字滤波器采用差分方程实现滤波时，起始部分的样点值无法实现滤波处理，通常采用的处理方式为让输出起始部分的值为0。

FIR线性时不变系统的系统函数为：

因此对应的常系数线性差分方程为：

由于FIR数字滤波器的单位脉冲响应为有限长度，因此可以利用关系式（1）、（2）实现滤波处理，也可以利用如关系式（4）的差分方程，采用快速傅里叶变换得到滤波的快速算法。

###### （二）音频数字化处理

（1）简易回音效果实现

回音，是一种效果，是声波传播后遇到障碍物反弹回来的声音。Echo跟Delay效果很类似，不同处在于Echo的音质会受到反弹物(如墙壁)材质、距离的影响；而Delay效果则纯粹将原音再拷贝，不影响音质。

假如音源为，简易回音效果实现输出。为衰减系数，为延时长度。

（2）立体声效果构造

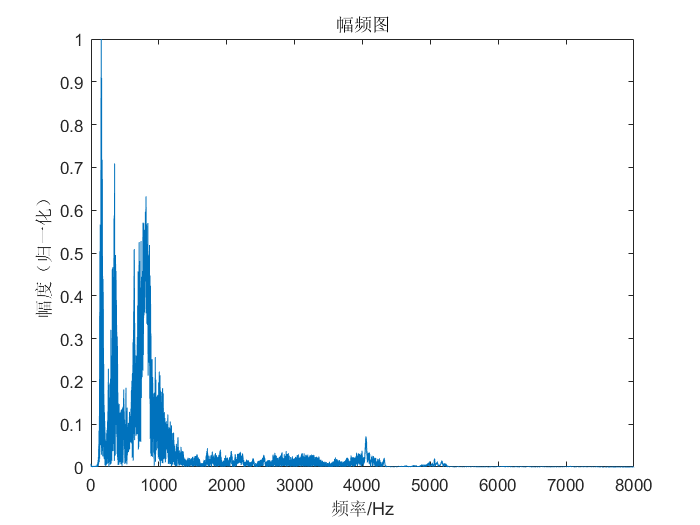
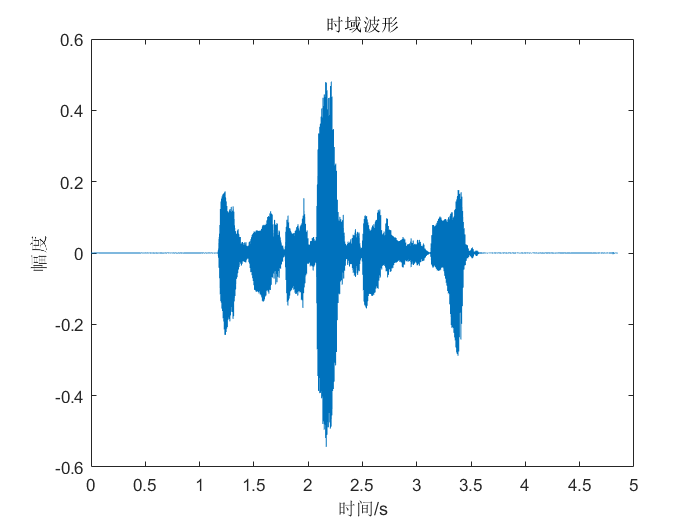
立体声能给听众以方位感和深度感，大大提高了听觉效果和声音品质。为了实现立体声效果，录制时通常需要多个摆放在不同位置的麦克风，播放时也需要两只或两只以上的扬声器。当只利用一个麦克风进行声音录制时，只能得到单声道音频，即便使用两只或两只以上的扬声器播放，也只能得到“平面化”的声音，而不会形成空间声像。利用人的空间听觉特性，如耳间声级差（Interaural Level Difference）、耳间时间差（Interaural Time Difference）和耳间相关性对空间声像定位，可以按照预期的听觉效果通过对多个声道信号进行相应处理，在听觉上形成空间感。对多声道信号进行的处理包括：扬声器排列法、分频法、移相法、延迟法等。

简易立体声效果的实现方式是将单声道音频转换为双声道音频，，并通过周期性地、交替地对两个声道上的声音利用时变增益系数，进行衰减和增强的方式，即可在感官上造成声源远离听者的一侧而运动到另一侧的感觉，从而形成声音方位感变化的立体声效果。

##### 三．实验内容

###### ①、

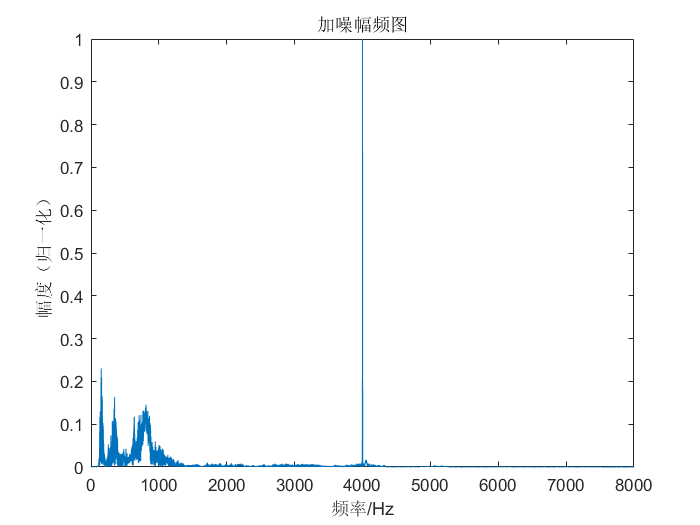
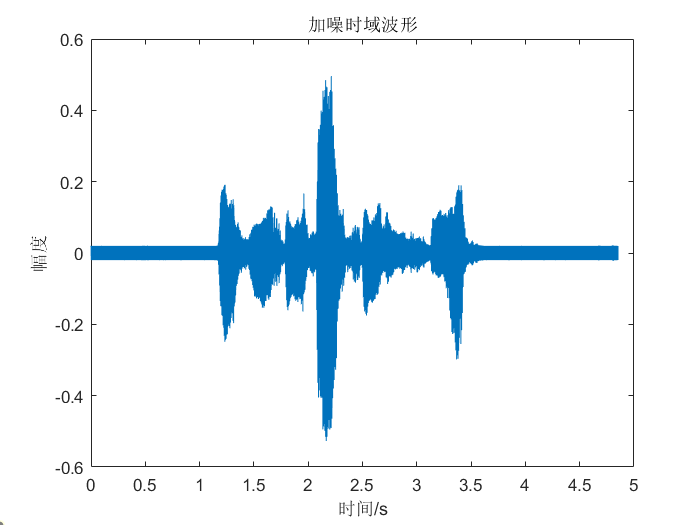
对给定的dspx1.wav文件，利用audioread函数读取音频信号数据流及采样频率信息，绘制出音频信号的时域波形（横坐标单位为“秒”）及幅频图（纵坐标归一化，横坐标范围0~8000Hz）。

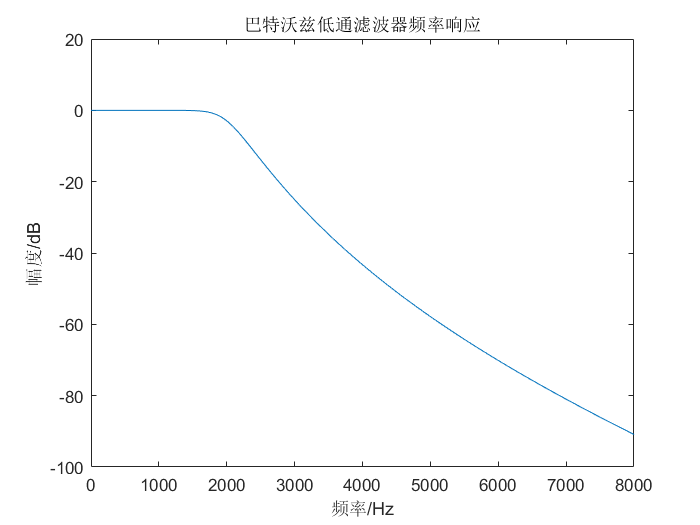


###### ②、

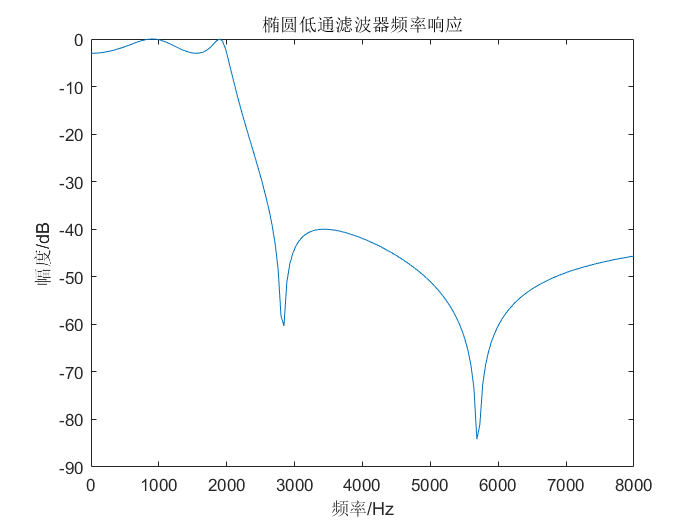
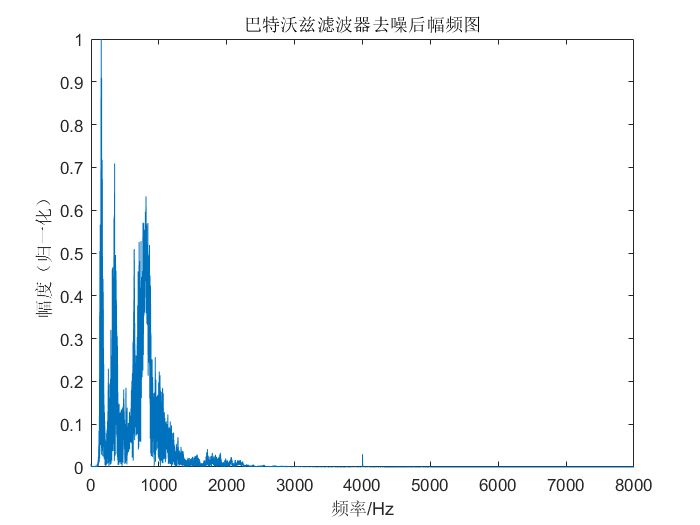
假定正弦噪声信号，，。以dspx1.wav的采样频率将噪声信号数字化并混入音频信号之中，得到加噪后的音频。请分别使用巴特沃兹滤波器、椭圆滤波器，设计出IIR数字滤波器，对音频进行去噪声处理。

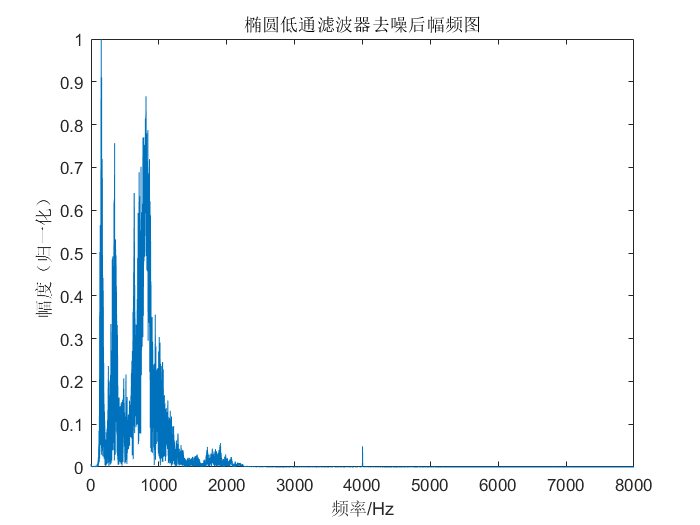
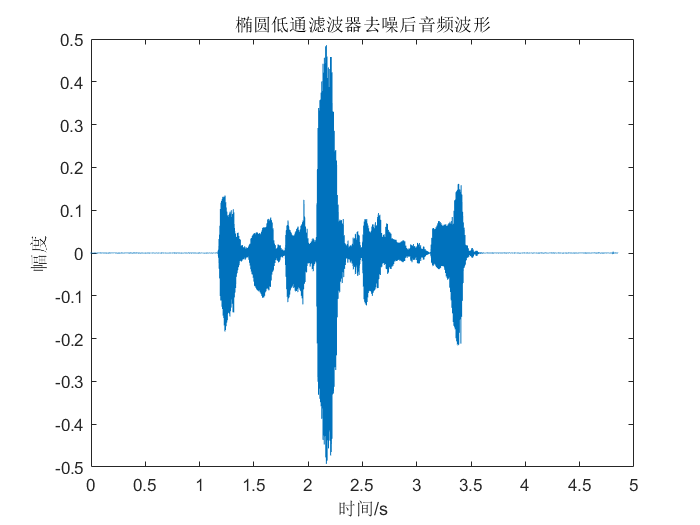
要求：设计的数字滤波器通带最大衰减，阻带最小衰减。注明所设计的滤波器类型（带通、带阻、低通）、以及设计的通带截止频率和阻带起始频率。绘制出加噪信号时域波形及幅频图、滤波器幅频特性曲线、以及去噪后的时域波形及幅频图进行对比。并利用sound函数播放去噪后的音频，体会数字滤波效果。



图表

描述已自动生成



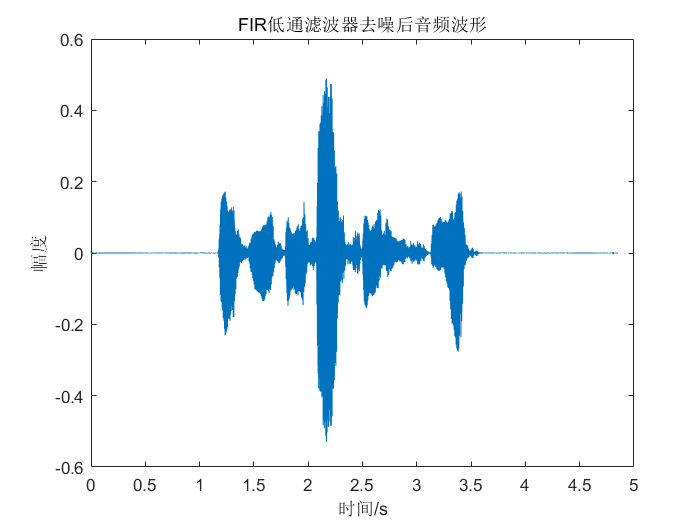
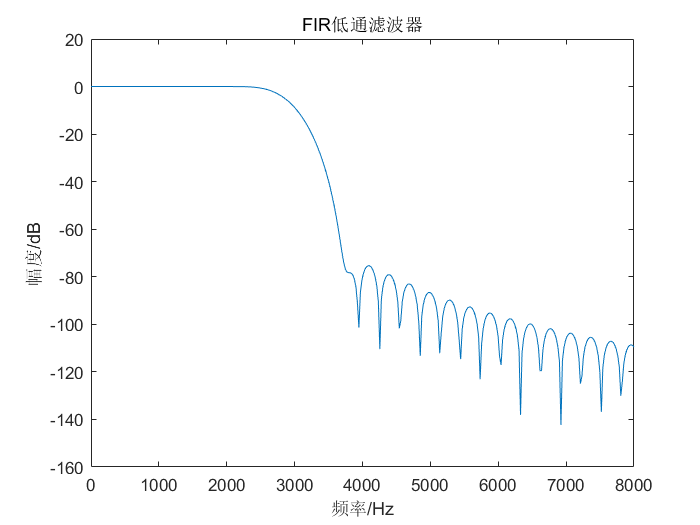


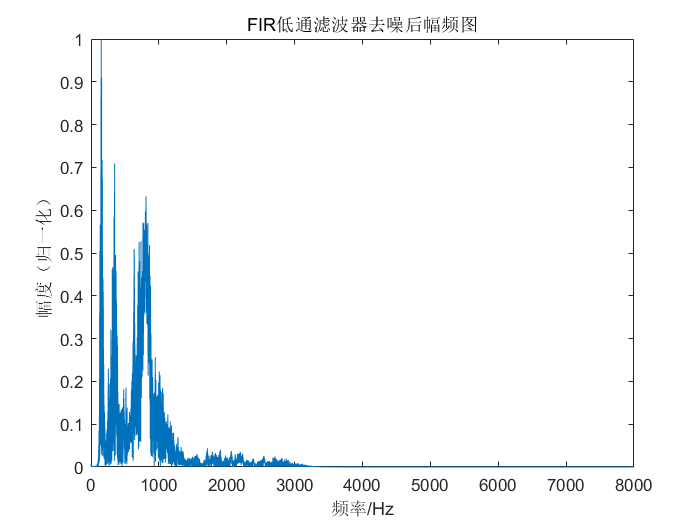
**注：**滤波器为巴特沃兹低通滤波器，通带截止频率为2000Hz，和阻带起始频率为3800Hz；滤波器为椭圆低通滤波器，通带截至频率为2000Hz，阻带起始频率为3800Hz。

**分析：**对比分析可知，加噪信号在4000Hz处出现一个尖峰，经过低通滤波器处理后，4000Hz处的尖峰已被滤除。使用sound函数试听结果，人耳感知与原音频信号的差别较小。

###### ③、

选取符合要求的窗函数法设计一个FIR滤波器，完成实验内容②。绘制FIR滤波器幅频特性曲线、以及去噪后的时域波形及幅频图。

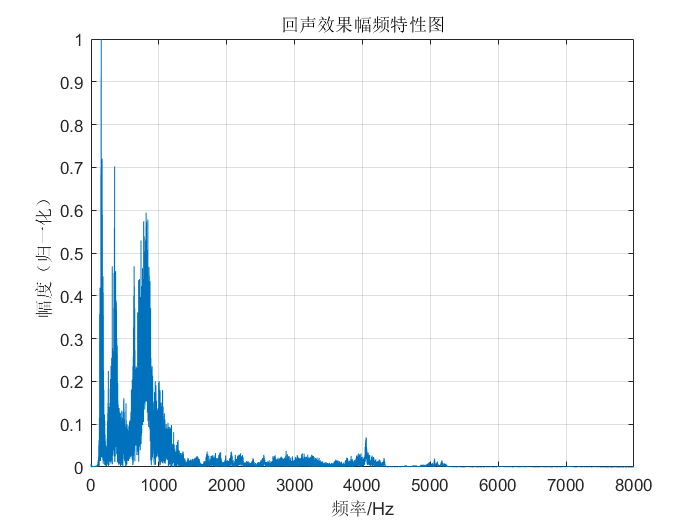
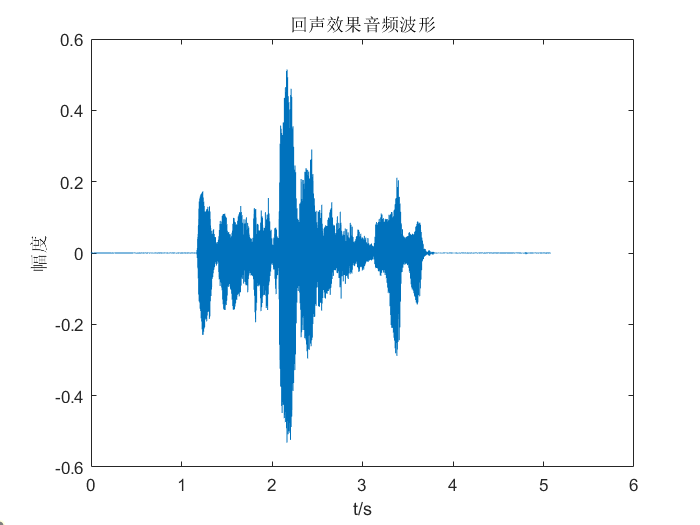




**注：**滤波器利用布莱克曼窗构造，窗的长度为149，滤波器的通带截至频率为2000Hz，阻带起始频率为3800Hz。

###### ④、

对dspx1.wav音频文件实现简易回音效果，要求个采样点。

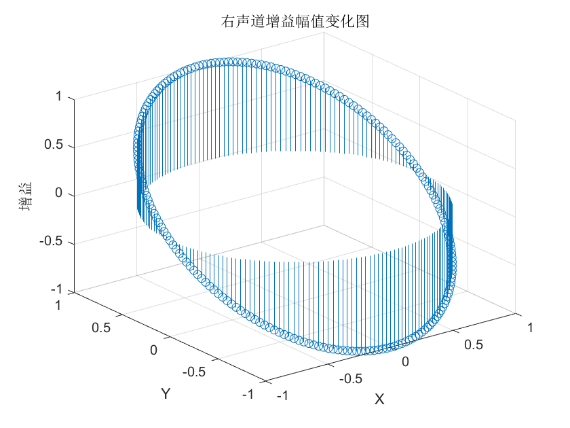


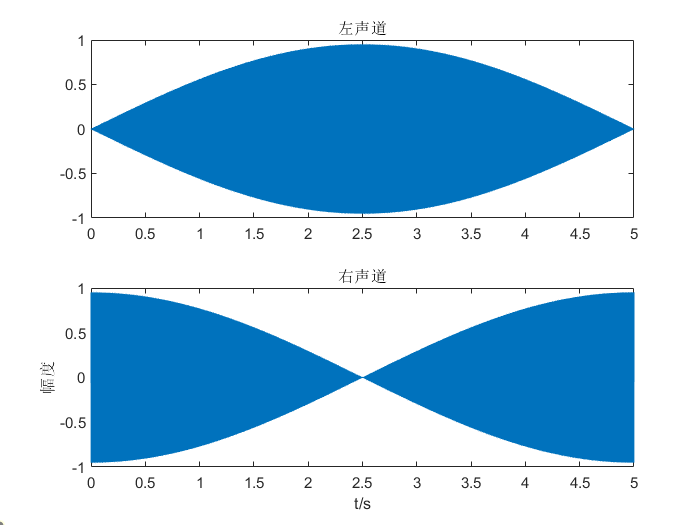
**注：**回音延迟点数为10000，强度为原始信号的0.5。

###### ⑤、

生成单一频率的正弦信号，幅度，，采样频率，持续时长为5秒。左声道时变增益系数和右声道时变增益系数在一个圆周内的幅度变化效果如下图所示，

|  |
| --- |
| C:\Users\Anomy\AppData\Local\Temp\ConnectorClipboard1130582526021015534\image16387826158730.png |

增益变化频率设置为，根据以上条件实现听者能感觉到声源在围绕自己进行圆周转动的效果。绘制出左、右声道的时域波形图。



###### ⑥、

**（选做者加分）**对dspx1.wav音频文件进行频移处理，实现简易的音调由男声向女声变换效果。

##### 四.实验代码

###### 实验①：

% 清除工作区变量和命令窗口

clc;clear;

% 读取音频文件

% y: 音频数据（振幅值）

% Fs: 采样频率（每秒采样点数）

[y, Fs] = audioread('dspx1.wav');

% 绘制时域波形

t = (0:length(y)-1) / Fs; % 将采样点序号转换为实际时间

figure;

plot(t, y);

title('时域波形');

xlabel('时间/s');

ylabel('幅度');

% 绘制幅频图

figure;

N = length(y); % 获取信号总长度

Y0 = fft(y); % 对信号进行快速傅里叶变换

f = (0:N-1)\*Fs/N; % 计算频率轴

Y = abs(Y0)/abs(max(Y0)); % 计算归一化幅度谱

plot(f, Y);

title('幅频图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]); % 限制显示频率范围为0-8000Hz

###### 实验②：

% 清空工作区和命令窗口

clc;clear;

% 读取音频文件并添加噪声

[y0,Fs] = audioread('dspx1.wav'); % 读取音频文件，y0为音频数据，Fs为采样频率

t = (0:length(y0)-1) / Fs; % 生成时间序列

f = 4000; % 设置噪声频率为4000Hz

A = 0.02; % 设置噪声幅度

Nt = (A\*cos(2\*pi\*f\*t))'; % 生成余弦噪声信号

y = y0 + Nt; % 将噪声信号添加到原始音频

%% 绘制加噪后的信号分析图

% 绘制时域波形

figure;

plot(t, y);

title('加噪时域波形');

xlabel('时间/s');

ylabel('幅度');

% 绘制幅频图

figure;

n = length(y); % 获取信号长度

Y0 = fft(y); % 对信号进行傅里叶变换

f = (0:n-1)\*Fs/n; % 计算频率轴

Y = abs(Y0)/abs(max(Y0)); % 计算幅度谱并归一化

plot(f, Y);

title('加噪幅频图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]); % 限制频率显示范围

%% 巴特沃兹低通滤波器设计与应用

% 设置滤波器参数

wp = 2000\*2\*pi/Fs; % 通带截止频率（2000Hz）

ws = 3800\*2\*pi/Fs; % 阻带截止频率（3800Hz）

rp = 3; % 通带最大衰减（dB）

rs = 40; % 阻带最小衰减（dB）

% 频率预畸变

wp = 2\*Fs\*tan(wp/2); % 将数字频率转换为模拟频率

ws = 2\*Fs\*tan(ws/2);

% 设计巴特沃兹滤波器

[N, wn] = buttord(wp,ws, rp, rs, 's'); % 计算滤波器阶数N和截止频率wn

[A,B] = butter(N, wn, 's'); % 设计巴特沃兹滤波器

[num1, den1] = bilinear(A, B,Fs); % 双线性变换，将模拟滤波器转换为数字滤波器

% 绘制滤波器频率响应

[h1,w] = freqz(num1, den1);

w = w/2/pi\*Fs;

figure;

plot(w,20\*log10(abs(h1)));

xlim([0,8000]);

xlabel('频率/Hz');

ylabel('幅度/dB');

title('巴特沃兹低通滤波器频率响应');

% 应用巴特沃兹滤波器并绘制结果

y1 = filter(num1,den1,y); % 对含噪声信号进行滤波

% 绘制滤波后的时域波形

figure;

plot(t,y1);

title('巴特沃兹滤波器去噪后音频波形');

xlabel('时间/s');

ylabel('幅度');

% 绘制滤波后的频谱

figure;

Y1 = fft(y1);

M1 = max(Y1);

f = (0:n-1)\*Fs/n;

Y1 = abs(Y1)/abs(M1);

plot(f, Y1);

title('巴特沃兹滤波器去噪后幅频图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]);

%% 椭圆低通滤波器设计与应用

% 设计椭圆滤波器（参数与巴特沃兹滤波器相同）

[N, wn] = ellipord(wp,ws, rp, rs, 's'); % 计算椭圆滤波器阶数和截止频率

[A,B] = ellip(N,rp,rs,wp,'s'); % 设计椭圆滤波器

[num2, den2] = bilinear(A, B,Fs); % 转换为数字滤波器

% 绘制滤波器频率响应

[h2,w] = freqz(num2, den2);

w = w/2/pi\*Fs;

figure;

plot(w,20\*log10(abs(h2)));

xlim([0,8000]);

xlabel('频率/Hz');

ylabel('幅度/dB');

title('椭圆低通滤波器频率响应');

% 应用椭圆滤波器并绘制结果

y2 = filter(num2,den2,y); % 对含噪声信号进行滤波

% 绘制滤波后的时域波形

figure;

plot(t,y2);

title('椭圆低通滤波器去噪后音频波形');

xlabel('时间/s');

ylabel('幅度');

% 绘制滤波后的频谱

figure;

Y2 = fft(y2);

M1 = max(Y2);

f = (0:n-1)\*Fs/n;

Y2 = abs(Y2)/abs(M1);

plot(f, Y2);

title('椭圆低通滤波器去噪后幅频图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]);

###### 实验③：

% 清除工作区和命令窗口

clc;clear;

% 读取音频文件

[y0,Fs]=audioread('dspx1.wav');

% 计算时间向量

t = (0:length(y0)-1) / Fs;

n = length(y0);

% 生成噪声信号

f=4000; % 噪声频率4000Hz

A=0.02; % 噪声幅度

Nt=(A\*cos(2\*pi\*f\*t))'; % 生成余弦噪声

y=y0+Nt; % 将噪声加入原始信号

% 设计FIR低通滤波器参数

wp=2000\*2\*pi/Fs; % 通带截止频率（弧度/采样）

ws=3800\*2\*pi/Fs; % 阻带截止频率（弧度/采样）

wide=ws-wp; % 过渡带宽度

N=ceil(12\*pi/wide)+1; % 计算滤波器阶数

wc=(wp+ws)/2; % 计算截止频率

% 使用布莱克曼窗设计FIR滤波器

h=fir1(N, wc/pi, blackman(N+1));

% 计算并绘制滤波器的频率响应

[H, w]=freqz(h,1,1000);

w=w\*Fs/(2\*pi);

plot(w,20\*log10(abs(H)));

xlim([0,8000]);

xlabel('频率/Hz');

ylabel('幅度/dB');

title('FIR低通滤波器');

% 对含噪声信号进行滤波

y3=filter(h,1,y);

% 绘制滤波后的时域波形

figure;

plot(t,y3);

title('FIR低通滤波器去噪后音频波形');

xlabel('时间/s');

ylabel('幅度');

% 计算并绘制滤波后信号的频谱

figure;

Y3 = fft(y3); % 进行傅里叶变换

M1=max(Y3); % 获取最大值用于归一化

f = (0:n-1)\*Fs/n; % 计算频率向量

Y3 = abs(Y3)/abs(M1); % 幅度归一化处理

% 绘制频谱图

plot(f, Y3);

title('FIR低通滤波器去噪后幅频图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]);

###### 实验④：

% 清除工作区和命令窗口

clc;clear;

% 读取音频文件

[y,Fs] = audioread('dspx1.wav'); % y为音频数据，Fs为采样频率

n = length(y); % 获取原始音频长度

% 设置回声参数

delay = 10000; % 设置延迟点数（控制回声延迟时间）

% 实际延迟时间 = delay/Fs 秒

% 创建回声信号

a = zeros(n+delay,1); % 创建一个比原始信号长delay个采样点的零向量

a(delay+1:n+delay) = 0.5.\*y; % 将原始信号延迟delay个点，并衰减到原来的0.5倍

% 扩展原始信号长度，使其与回声信号等长

y(n+delay,:) = 0; % 在原始信号末尾补零，使其长度与回声信号相同

% 将原始信号和回声信号叠加

y = y + a; % 将原始信号和回声信号相加

% 生成时间轴

t = (0:length(y)-1) / Fs; % 计算时间序列，单位为秒

% 播放带有回声效果的音频

sound(y, Fs); % 使用原始采样频率播放处理后的音频

% 绘制带回声效果的音频波形

figure;

plot(t, y);

title('回声效果音频波形');

xlabel('t/s');

ylabel('幅度');

% 添加幅频特性分析

figure;

Y = fft(y); % 对带回声信号进行FFT

M = max(Y); % 获取最大值用于归一化

f = (0:length(y)-1)\*Fs/length(y);% 计算频率向量

Y = abs(Y)/abs(M); % 幅度归一化处理

% 绘制频谱图

plot(f, Y);

title('回声效果幅频特性图');

xlabel('频率/Hz');

ylabel('幅度（归一化）');

xlim([0,8000]); % 限制频率显示范围

grid on; % 添加网格便于观察

###### 实验⑤：

% 清除工作区和命令窗口

clc;clear;

% 设置基本参数

A=1; % 信号幅度

f=800; % 信号频率(Hz)

Fs=10\*f; % 采样频率，设为信号频率的10倍，满足采样定理

N=5\*Fs; % 采样点数，采样5秒的数据

t=(0:N-1)/Fs; % 生成时间序列，范围为0到5秒

% 生成基本正弦信号

x=A\*sin(2\*pi\*f\*t); % 生成频率为800Hz的正弦波

% 生成立体声信号

s1=x.\*sin(0.1\*2\*pi\*t); % 左声道信号：原始信号与0.1Hz的正弦波相乘

s2=x.\*cos(0.1\*2\*pi\*t); % 右声道信号：原始信号与0.1Hz的余弦波相乘

% 播放立体声信号

sound([s1',s2'],Fs); % 将左右声道组合播放，注意需要转置为列向量

% 绘制时域波形

figure;

subplot(2,1,1); % 创建2行1列的子图，选择第1个

plot(t,s1); % 绘制左声道波形

title('左声道');

subplot(2,1,2); % 选择第2个子图

plot(t,s2); % 绘制右声道波形

title('右声道');

xlabel('t/s'); % x轴标签

ylabel('幅度'); % y轴标签

###### 实验⑥：

% 清除工作区变量和命令窗口

clc;clear;

% 读取音频文件

[y,Fs]=audioread('dspx1.wav'); % y为音频数据，Fs为采样率

% 生成时间轴

t = (0:length(y)-1) / Fs; % 根据采样率计算对应的时间点

% 准备频谱分析

N = length(y); % 获取信号长度

f = (0:N-1)\*Fs/N; % 计算频率轴，范围从0到采样率

% 进行傅里叶变换

Y = fft(y); % 对原始信号进行快速傅里叶变换

% 频谱搬移

X=circshift(Y,2000); % 将频谱整体移动2000个点

% 逆傅里叶变换

x=ifft(X); % 将移动后的频谱转换回时域

% 播放处理后的音频

sound(abs(x),Fs); % 使用原采样率播放处理后的信号

% 绘制频谱图

plot(f, abs(x)); % 绘制信号幅度谱

xlim([0,8000]); % 限制x轴显示范围为0-8000Hz

# 课程总结

数字信号处理（DSP）是研究数字信号获取、处理和分析的学科，广泛应用于通信、音频处理、医学成像、雷达、控制系统等多个领域。数字信号处理课程旨在帮助学生掌握基本理论、算法和应用技巧，培养在这一领域的综合能力。而数字信号处理实验课程则为理论知识提供了重要的实践支持，弥补了课本内容的局限，深化了我们对信号与系统、时域频域分析、数字滤波器、采样与重构等核心概念的理解。同时，实验课程还提高了我们使用MATLAB进行数据分析和编程的能力，对我们的学习具有重要意义。在实验过程中，老师的详细指导让我们的实验进展非常顺利。

关于实验课程的改进建议，我认为可以加入更多与实际应用相关的实验内容，如音频处理等，减少一些较为单一的实验项目，以提升实践的多样性和趣味性。