

基于 MATLAB 的音频 IIR 滤波系统

数字信号处理实验课程作业

1201003023 刘家彤

目录

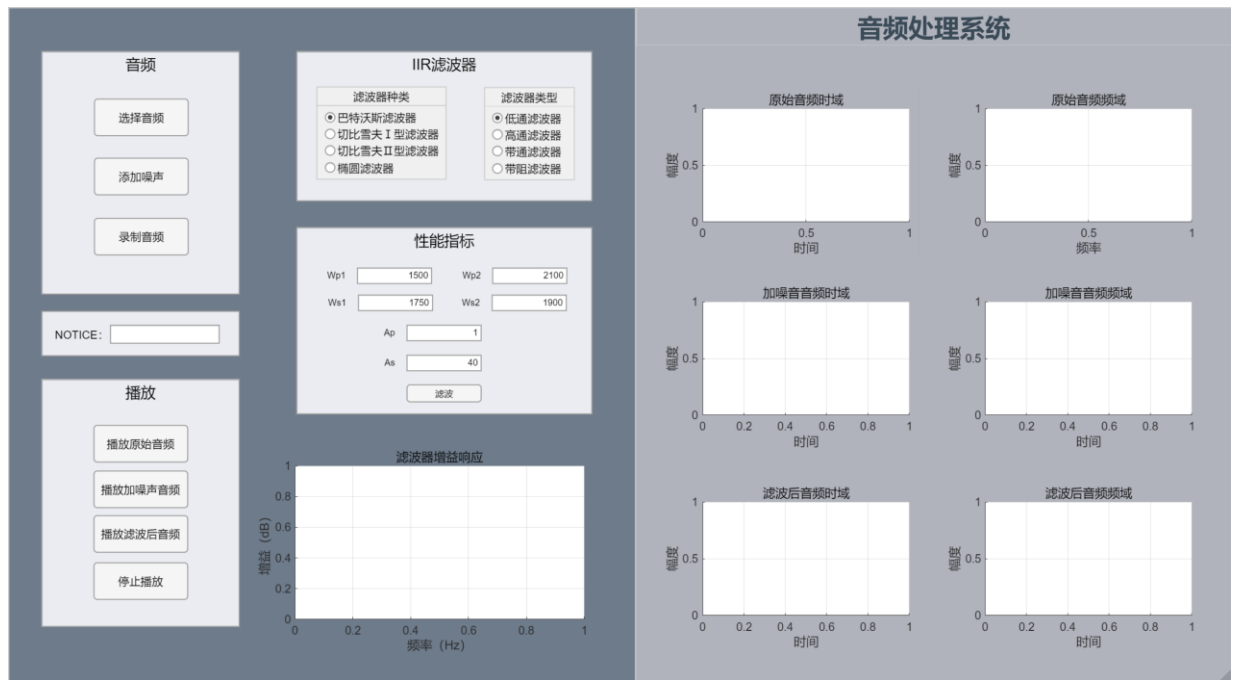
一、	整体介绍.....	1
二、	界面设计.....	1
三、	系统功能.....	2
a)	操作流程.....	2
b)	音频输入.....	3
•	上传音频文件.....	3
•	录制音频.....	3
•	添加噪声.....	4
c)	音频播放.....	4
d)	滤波器选择.....	5
e)	性能指标.....	5
f)	滤波.....	5
四、	运行结果.....	6
五、	不足.....	6

一、整体介绍

系统基于 MATLAB App Designer，通过设计实现添加噪音的音频的降噪处理，系统支持用户自主上传音频或录制一段 5s 的音频；支持用户选择四种 IIR 滤波器进行四类滤波处理；支持用户自行输入 IIR 滤波器性能指标；支持滤波前后音频播放；支持滤波前后时域、频域的图像输出。

二、界面设计

系统界面分为两部分，左半部分为用户操作区，实现音频的输入与播放，滤波器种类与类型的选择，滤波器性能指标的输入以及滤波器增益响应图像展示，各功能分块排列，便于用户进行使用；右半部分为图像输出区域，以二乘三排列实现原始音频、加噪音音频、滤波后音频时域与频域图像的展示，以使用户进行对比前后结果，也便于用户针对待处理音频输入针对的性能指标。



三、系统功能

a) 操作流程

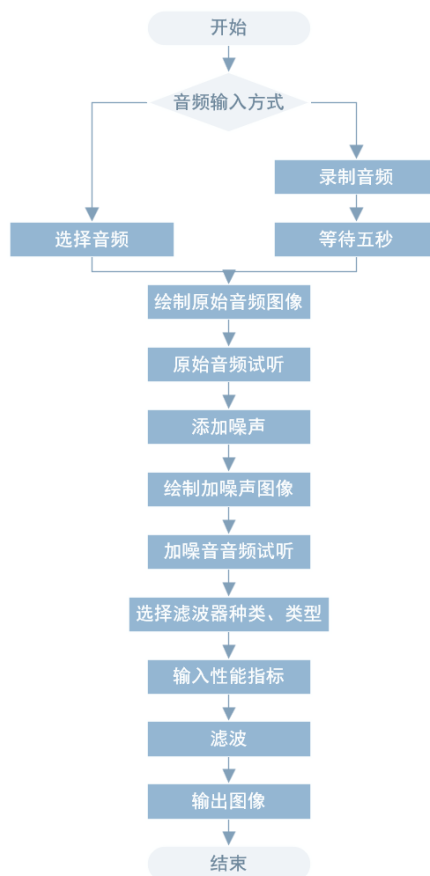


图 1 系统操作流程

b) 音频输入

- 上传音频文件

用户可以选择上传自己的音频文件，系统会自动筛选和用户文件夹中的 mp3 文件并进行提示，上传后右侧的作图区域自动绘制该音频文件的时域频域分布图。

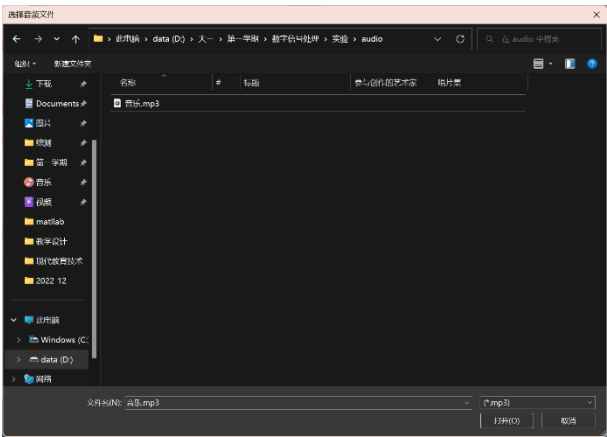


图 2 选择音频文件示例

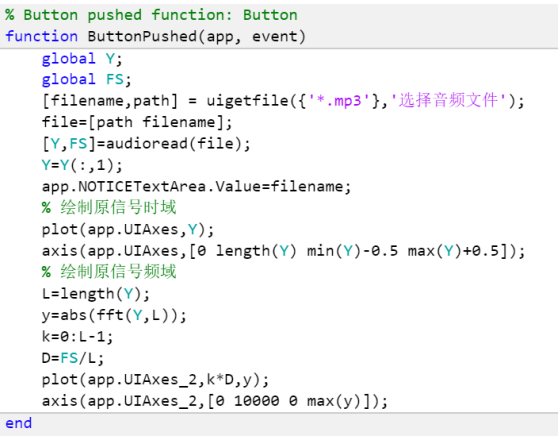


图 3 选择音频文件代码实现

- 录制音频

用户按下录制音频按键后系统开始录音，NOTICE 处提示用户开始录音，五秒后结束；五秒后提示录音结束，支持多次录音。录制完成录音后右侧的作图区域自动绘制该音频文件的时域频域分布图。



图 4 开始录音示例



图 5 结束录音示例

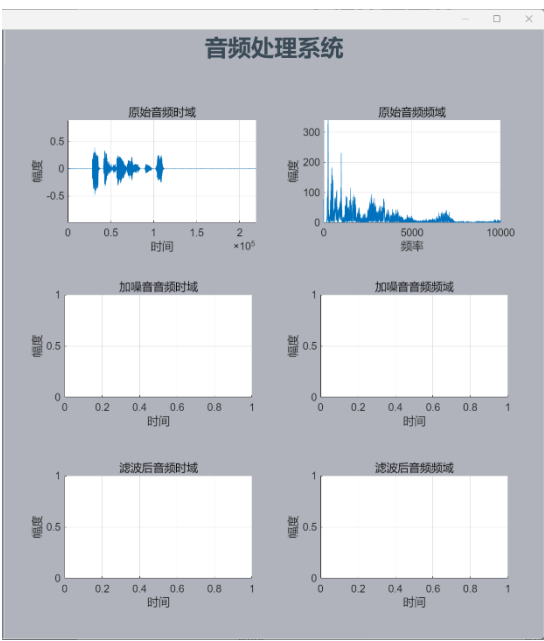


图 6 图像绘制

```

% Button pushed function: Button_16
function Button_16Pushed(app, event)
    global Y;
    global FS;
    R=audiorecorder(44100,16,1);
    FS=44100;
    record(R);
    app.NOTICETextArea.Value='开始录音，五秒后停止';
    pause(5);
    stop(R);
    app.NOTICETextArea.Value='录音结束';
    Y = getaudiodata(R);
    % 绘制原信号时域
    plot(app.UIAxes,Y);
    axis(app.UIAxes,[0 length(Y) min(Y)-0.5 max(Y)+0.5]);
    % 绘制原信号频域
    L=length(Y);
    y=abs(fft(Y,L));
    k=0:L-1;
    D=FS/L;
    plot(app.UIAxes_2,k*D,y);
    axis(app.UIAxes_2,[0 10000 0 max(y)]);
end

```

图 7 录音代码实现

- 添加噪声

在完成音频的上传后点击添加噪声即可为音频添加频率为正弦波函数 $0.01 \cdot \sin(3 \cdot \pi \cdot 1500 \cdot k / FS)$ ，添加噪声后右侧的作图区域自动绘制加噪音音频的时域频域分布图，如图可见在 X=2250 处有明显的噪声信号出现。

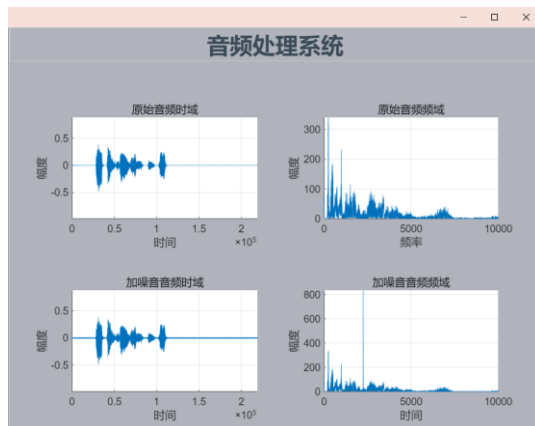


图 8 加噪音音频图像

```

% Button pushed function: Button_3
function Button_3Pushed(app, event)
    global Y;
    global FS;
    global YMix;
    L=length(Y);
    k=0:L-1;
    % 添加噪音
    noise=0.01*sin(3*pi*1500*k/FS);
    noise=noise';
    YMix=Y+noise;
    % 绘制时域
    plot(app.UIAxes_3, YMix);
    axis(app.UIAxes_3,[0 length(YMix) min(YMix)-0.5 max(YMix)+0.5]);
    % 绘制频域
    L=length(YMix);
    ymix=abs(fft(YMix,L));
    k=0:L-1;
    D=FS/L;
    plot(app.UIAxes_4,k*D,ymix);
    axis(app.UIAxes_4,[0 10000 0 max(ymix)]);
end

```

图 9 加噪音代码实现

c) 音频播放

音频播放分别支持原始音频的播放、加噪音音频的播放、滤波后音频的播放以及停止当前的播放。

```
% Button pushed function: Button_2
function Button_2Pushed(app, event)
    % 播放原始音频
    global Y;
    global FS;
    sound(Y,FS);
end

% Button pushed function: Button_15
function Button_15Pushed(app, event)
    % 停止播放
    clear sound;
end
```

```
% Button pushed function: Button_6
function Button_6Pushed(app, event)
    % 播放滤波后音频
    global Yf;
    global FS;
    sound(Yf,FS);
end

% Button pushed function: Button_5
function Button_5Pushed(app, event)
    % 播放加噪声音频
    global YMix;
    global FS;
    sound(YMix,FS);
end
```

d) 滤波器选择

用户通过单选按钮进行滤波器的选择，滤波器种类支持：巴特沃斯滤波器、切比雪夫 I、II 型滤波器、椭圆滤波器共四种；滤波器类型支持：低通滤波器、高通滤波器、带通滤波器、带阻滤波器四种。两者组合可实现 16 种滤波器。

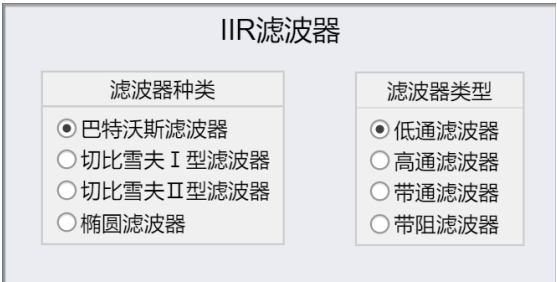


图 10 滤波器选择选项

e) 性能指标

MATLAB 提供了常用的设计模拟滤波器的函数，针对不同的音频信号，用户可以输入针对性的性能指标，分别是 W_p （通带截止频率）、 W_s （阻带截止频率）、 A_p （通带衰减）、 A_s （阻带衰减）。选择好滤波器类型并输入对应指标点击滤波按钮即可开始滤波。

f) 滤波

MATLAB 提供了常用的设计模拟滤波器的函数，这里以巴特沃斯低通滤波器为例进行示例：

借助 filter1、2 两个参数获取用户选择的滤波器类型，当满足 filter1、2=1 时即为巴特沃斯低通滤波器。借助 MATLAB 提供的 buttord 和 butter 函数，传入用户输入的指标，进行频率的转化，传入函数，设计对应的滤波器并进行加噪声音频的滤波，最后在绘图区绘制滤波后的时域频域图，并在左侧绘制滤波器的增益响应。整个过程中用户可以随意切换滤波器类型并修改滤波器指标。

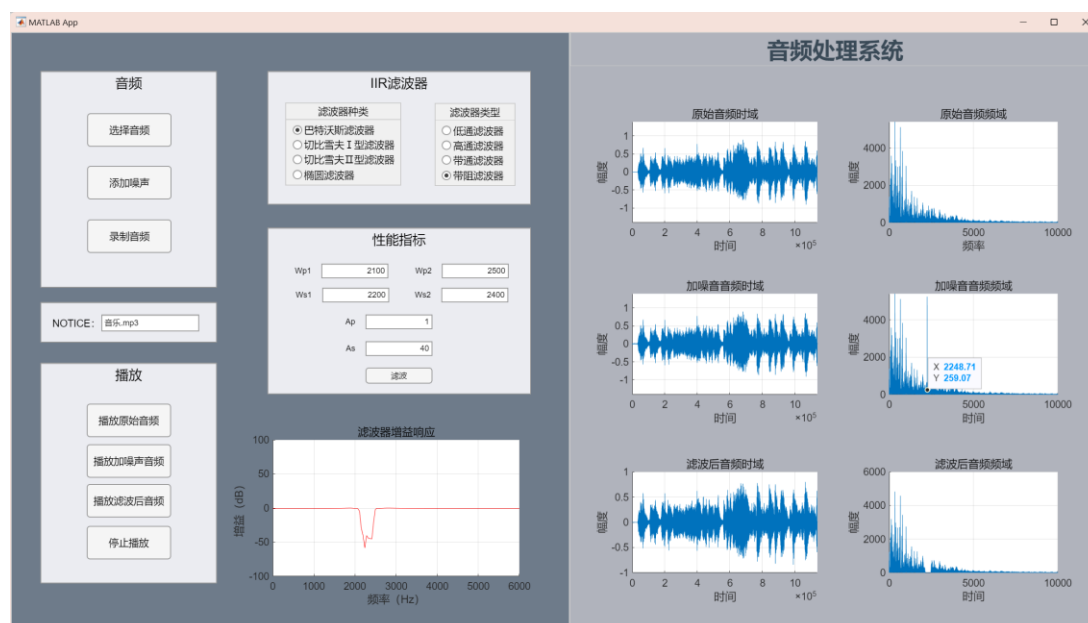
```
if filter1==1 % 巴特沃斯
    if filter2==1 % 低通
        [N,Wc]=buttord(wp1/pi,ws1/pi,Ap,As);
        [B,A]=butter(N,Wc);
        [H,W]=freqz(B,A);
        Yf=filter(B,A,YMix);
        plot(app.UIAxes_5,Yf);
        % 绘制频域
        L=length(Y);
        yf=abs(fft(Yf,L));
        k=0:L-1;
        D=FS/L;
        plot(app.UIAxes_6,k*D,yf);
        axis(app.UIAxes_6,[0 10000 0 6000]) ;|
        % 绘制滤波器增益响应
        [H,W]=freqz(B,A);
        dBH=20*log10(abs(H)/max(abs(H)));
        plot(app.UIAxes2,W/2/pi*FS,dBH,'r');
        axis(app.UIAxes2,[0,6000 -100 100]) ;
```

图 11 巴特沃斯低通滤波器代码实现

对于其他滤波器种类，MATLAB 同样给出对应的设计函数：对于切比雪夫滤波器，可以使用 cheb1ord、cheby1、cheb2ord、cheby2 实现；对于椭圆滤波器，可以使用 ellipord、ellip 实现；对于高通滤波器，在参数中加入‘high’即可实现；对于带通带阻滤波器，需要对用户输入的 Wp 和 Ws 进行合并传参实现。

四、 运行结果

例：用户上传音频，选择巴特沃斯带阻滤波器，Wp 设置为 2100-2500；Ws 设置为 2200-2400 进行滤波后可以看到位于 2250HZ 的噪声被过滤掉了。



五、 不足

在实际应用中还可以加入 FIR 滤波器的选项，满足更多的用户滤波需求；除此之外目前加入的噪声过于单一且固定，因此可以加入用户选择加入噪声种类的选项。