

# 南昌大学

NANCHANG UNIVERSITY

## 《信号与系统》

### 基于 MATLAB 的音频数字处理系统

(2023 年 2 月 25 日)



学 院：                      际銓书院

专业班级：                      2021 级人工智能实验班

学生姓名：           殷懿睿           学号：           5509121041          

指导教师： 朱莉

# 基于 MATLAB 的音频数字处理系统

## 摘要

本课题旨在运用信号与系统和数字信号处理的理论知识，对语音信号、图像信号等进行频谱分析，并对加噪声语音信号进行时域、频域分析和滤波设计。利用 MATLAB 软件进行编程，并通过调试和完善程序，在计算机上实现我们的目标。滤波器设计在数字信号处理中占有极其重要的地位。利用 MATLAB 信号处理工具箱，我们可以快速有效地设计各种数字滤波器。在本课题中，将采用窗函数法来设计 FIR 数字滤波器，并利用 MATLAB 作为辅助工具完成设计中的计算与图形的绘制。通过对所设计滤波器的仿真和频率特性分析，我们得知利用 MATLAB 信号处理工具箱可以有效快捷地设计 FIR 数字滤波器，过程简单方便，结果的各项性能指标均符合指定要求。除此之外，我们还将深入研究和探讨本课题涉及到的理论知识，以期更好地理解数字信号处理的相关概念和原理。通过本课题的学习，相信能够更深入地了解信号与系统及数字信号处理的理论知识，并能够运用所学知识进行实际应用。

为什么选择这个课题，是由于之前自己在算法中有一些关于 FFT 的积累，同时自己也对 DTFT 及衍生应用非常感兴趣，在上课的时候就被深深吸引了，然而受限于课时的限制，在课上没能学到特别多，就决定课下来摸索一下，正好也提高了自己对于离散傅里叶变换的理解，做了一个 MATLAB GUI 形式的音频数字处理系统，并且为了提高自己的代码能力和加深课上所学原理的应用，特意没选择使用表现比较优秀的深度学习模型而是根据基本性质进行编程，相关源码全部开源，可供参考：  
<https://github.com/JerryYin777/NCU-Signals-and-Systems-Project>

**关键字：** 音频处理   数字滤波器   信号变换   MATLAB GUI   模拟信号分析

# 一、引言

## 1.1 课题背景

随着互联网时代的飞速发展,音频信号依旧是当今时代通信行业发展的一大重要领域。音频模拟信号的分析也是各大领域研究的一个重点 [1]。在音频信号数字处理的过程也是研究的一个方向,将原始的音频信号进行处理,例如音乐行业调音师对音乐的细节调整,生物学研究生命发声的特性,计算机领域对音频信号的变声处理等,都离不开音频的数字信号处理。随着数字化技术的发展,许多场合所使用的音频设备都由数字化音频处理技术代替了模拟音频技术,数字化音频处理技术主要是通过数字滤波算法对所收集到的信号进行处理与变化来实现 [2]。

在工程领域中, MATLAB 是一种倍受程序开发人员青睐的语言。对于一些需要做大量数据运算处理的复杂应用以及某些复杂的频谱分析算法, MATLAB 显得游刃有余。因此,可以利用 MATLAB 来处理数字信号,对信号进行分析、滤波和处理,实现更加精准和有效的结果。此外,数字信号处理技术的不断创新和发展,也为 MATLAB 提供了更多的应用场景和发展机遇 [3]。

在数字信号处理中,除了数字滤波器外,还有许多其他重要的技术和工具。例如,数字信号处理中的离散傅里叶变换 (DFT) 和快速傅里叶变换 (FFT) 等技术,可以用于分析信号的频域特性,实现信号的滤波、去噪和特征提取等功能。此外,数字信号处理中的小波变换、自适应滤波和相关性分析等技术,也具有重要的应用和研究价值。这些技术和工具的应用,可以进一步提高数字信号处理的精度、效率和可靠性,为各行各业的发展带来更多的机遇和挑战 [4]。

## 1.2 研究意义

研究音频数字处理系统可以提供一种高效的工具,帮助研究者和工程师们处理音频数据:

1. 改善音频质量: 音频数字处理系统可以帮助人们改善音频质量,例如去噪、降噪、均衡、混响等处理,使音频更加清晰、自然。
2. 优化音频性能: 音频数字处理系统可以通过各种算法和技术对音频数据进行处理和优化,以提高音频的性能,例如减少失真、增强声音、调节频率等 [5]。
3. 开发音频应用: 音频数字处理系统可以为音频应用程序的开发提供支持。例如,可以将它们用于语音识别、自然语言处理、音频信号分析和合成等应用 [6]。
4. 促进音频技术的研究: 音频数字处理系统可以为音频技术的研究提供一个平台,例如声学建模、数字信号处理、音频编解码等,使得音频技术的研究更加高效、精确、

系统化 [7]。

### 1.3 研究内容

利用 MATLAB 的 GUI 图形化界面搭建工具和函数搭建基本的音频分析界面，便于后续的音频处理的选择和参数设定，来方便观察不同参数的影响。设计的基本的图形化界面如图1所示。工具栏主要包括文件——打开声音文件，录音，关闭三个选项，用于导入音频文件以便后续分析；叠加噪声——叠加白噪声，叠加单频噪声和多频噪声三个选项；滤波器设计——IIR 滤波器设计和 FIR 滤波器设计 [8]。

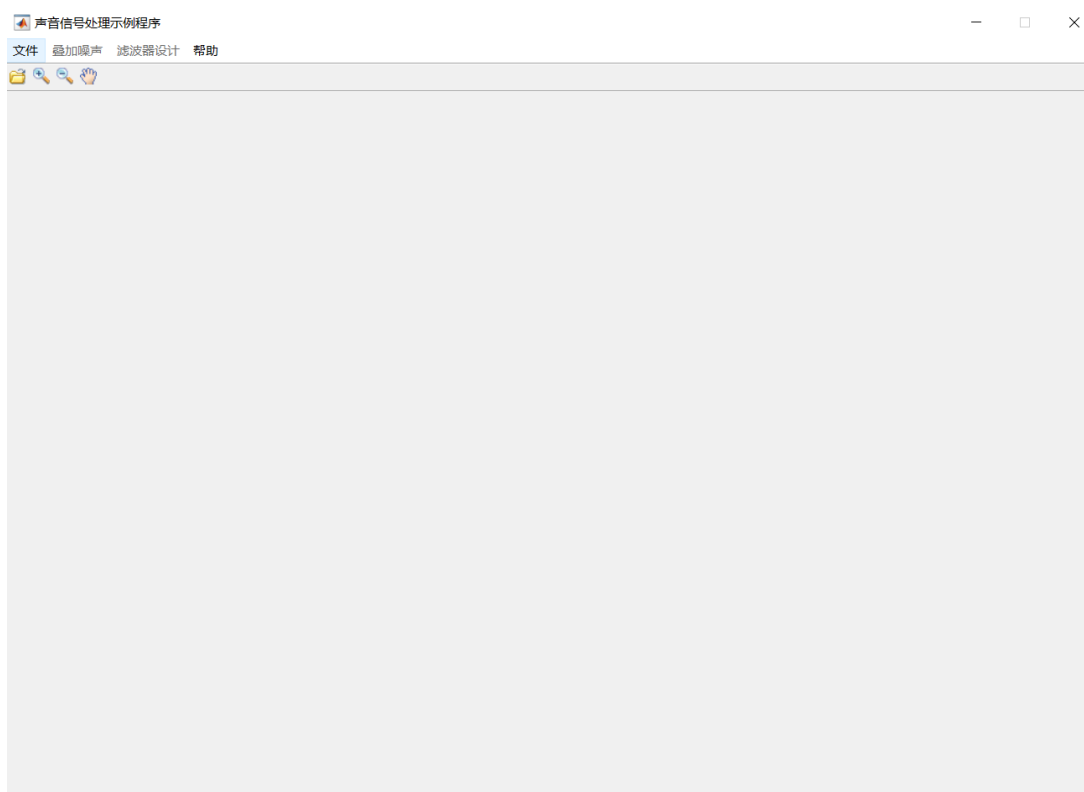


图 1 MATLAB GUI 音频分析界面

## 二、 设计原理

### 2.1 DFT 的基本原理

离散傅里叶变换 (Discrete Fourier Transform, DFT) 实质是有限长序列傅里叶变换的有限点离散采样，从而实现了频域离散化，使数字信号处理可以在频域采用数值运算的方法进行，这样就大大增加了数字信号处理的灵活性。DFT 有多种快速算法，统称为快速傅里叶变换 (Fast Fourier Transform, FFT)

设  $x(n)$  是一个长度为  $M$  的有限长序列，则  $x(n)$  的  $N$  点离散傅里叶变换为:

$$X(k) = \text{DFT}[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad k = 0, 1, 2, \dots, N-1$$

其中  $X(k)$  的离散傅里叶逆变换 (IDFT) 则定义为:

$$x(n) = \text{IDFT}[X(k)] = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} \quad n = 0, 1, 2, \dots, N-1$$

用 DFT 计算理想低通滤波器的频响曲线图2所示:

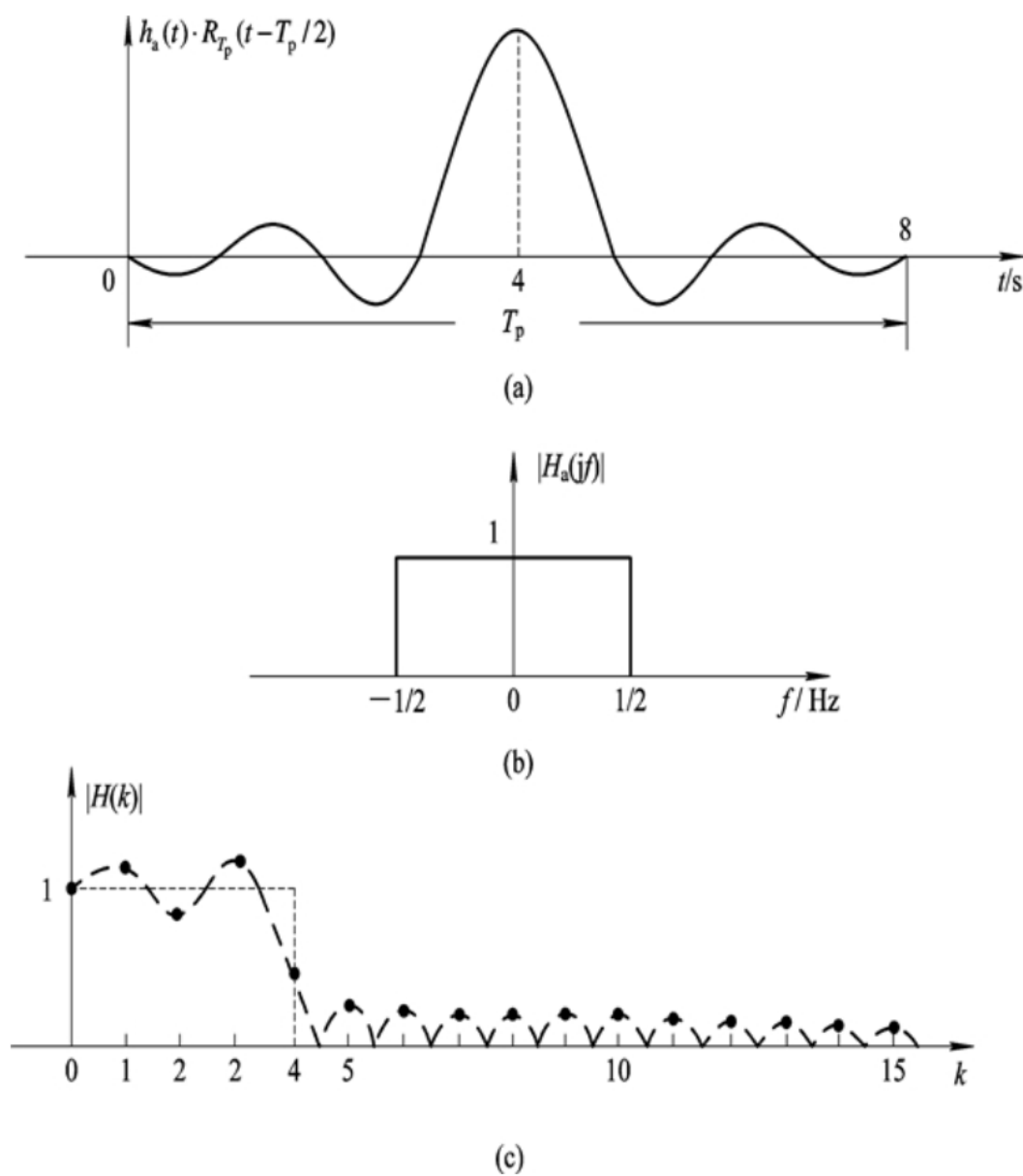


图2 DFT 计算理想低通滤波器的频响

FFT(Fast Fourier Transform) 是离散傅立叶变换的快速算法, 可以将一个信号变换到频域。有些信号在时域上是很难看出是什么特征, 但是如果变换到频域之后, 就很容易看出特征, 这就是很多信号分析采用 FFT 变换的原因。另外, FFT 可以将一个信号的频谱提取出来, 这在频谱分析方面也是经常用的。FFT 能够更高效的计算离散傅里叶变换, 因此常用于计算机编程的计算, 用来节省复杂信号的数字处理, 应用较广泛的就是数学软件 MATLAB 的 FFT 算法, 它集成了包括 fft, fftshift 等函数, 便于方便调用 [9]

## 2.2 IIR 滤波器

一般将网络结构分成两类, 一类称为有限长单位脉冲响应网络, 简称 FIR (Finite Impulse Response) 网络, 另一类称为无限长单位脉冲响应网络, 简称 IIR (Infinite Impulse Response) 网络 [10]。

IIR 网络结构存在输出对输入的反馈支路, 也就是说, 信号流图中存在反馈环路。这类网络的单位脉冲响应是无限长的, 其 N 阶差分方程为:

$$y(n) = \sum_{i=0}^M b_i x(n-i) + \sum_{i=1}^N a_i y(n-i)$$

其中设计时, 常用的数字滤波器一般属于选频滤波器。假设数字滤波器的频率响应函数  $H(e^{jw})$  用下式表示:  $H(e^{jw}) = |H(e^{jw})| e^{j\theta(w)}$  式中,  $|H(e^{jw})|$  称为幅频特性函数;  $\theta(w)$  称为相频特性函数。无限长单位脉冲响应 (IIR) 滤波器的系统函数为:

$$H(z) = \frac{\sum_{j=0}^M b_j z^{-j}}{1 + \sum_{k=1}^N a_k z^{-k}}$$

模拟滤波器的理论和设计方法也已经相当成熟, 有多种典型的模拟滤波器, 例如巴特沃斯 (Butterworth) 滤波器、切比雪夫 (Chebyshev) 滤波器、椭圆 (Ellipse) 滤波器、贝塞尔 (Bessel) 滤波器等。

## 2.3 FIR 滤波器

FIR 网络中一般不存在输出对输入的反馈支路, 因此差分方程描述为:

$$y(n) = \sum_{i=0}^M b_i x(n-i)$$

有限长单位脉冲响应 (FIR) 滤波器的系统函数为:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

FIR 滤波器的设计主要包括线性相位 FIR 数字滤波器设计, 利用窗函数法设计, 频率采样法设计, 等波纹最佳逼近法设计。其中窗函数的设计方式较为常见, 基本窗函数特性如图3所示:

| 窗函数类型                   | 旁瓣峰值 $\alpha_n$<br>/dB | 过渡带宽度 $B_t$ |            | 阻带最小衰减 $\alpha_s$<br>/dB |
|-------------------------|------------------------|-------------|------------|--------------------------|
|                         |                        | 近似值         | 精确值        |                          |
| 矩形窗                     | -13                    | $4\pi/N$    | $1.8\pi/N$ | -21                      |
| 三角窗                     | -25                    | $8\pi/N$    | $6.1\pi/N$ | -25                      |
| 汉宁窗                     | -31                    | $8\pi/N$    | $6.2\pi/N$ | -44                      |
| 哈明窗                     | -41                    | $8\pi/N$    | $6.6\pi/N$ | -53                      |
| 布莱克曼窗                   | -57                    | $12\pi/N$   | $11\pi/N$  | -74                      |
| 凯塞窗 ( $\beta = 7.865$ ) | -57                    |             | $10\pi/N$  | -80                      |

图3 窗函数特性总结

## 2.4 算法原型

- DFT (或 FFT) MATLAB 提供了用快速傅里叶变换算法 FFT 计算 DFT 的函数 `fft`，其调用格式为: `Xk = fft(xn, N)`; `Ifft` 函数计算 IDFT，其调用格式与 `fft` 函数相同，可参考 `help` 文件。
- Matlab 中利用巴特沃斯滤波器和双线性变换法设计实现 IIR 滤波器的基本实现，具体包含的函数 `buttord` ()，`buttap` () 和 `bilinear` () 函数。
- FIR 滤波器的实现利用窗函数法设计，并选用哈明窗设计，具体函数包括 `hamming` ()，`fir1` () 函数等。

## 三、系统方案

### 3.1 音频处理系统界面搭建

在图1的基础上建立如图5所示的 GUI 界面，核心代码如下：

```

1 function varargout = DSP(varargin)
2 %创建 DSP 音频输入 GUI 分析界面
3 gui_Singleton = 1;
4 gui_State = struct('gui_Name', mfilename, ...
5                   'gui_Singleton', gui_Singleton, ...
6                   'gui_OpeningFcn', @DSP_OpeningFcn, ...
7                   'gui_OutputFcn', @DSP_OutputFcn, ...
8                   'gui_LayoutFcn', [] , ...
9                   'gui_Callback', []);
10 if nargin && ischar(varargin{1})
11     gui_State.gui_Callback = str2func(varargin{1});
12 end
13
14 if nargout

```

```

15     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16 else
17     gui_mainfcn(gui_State, varargin{:});
18 end

```

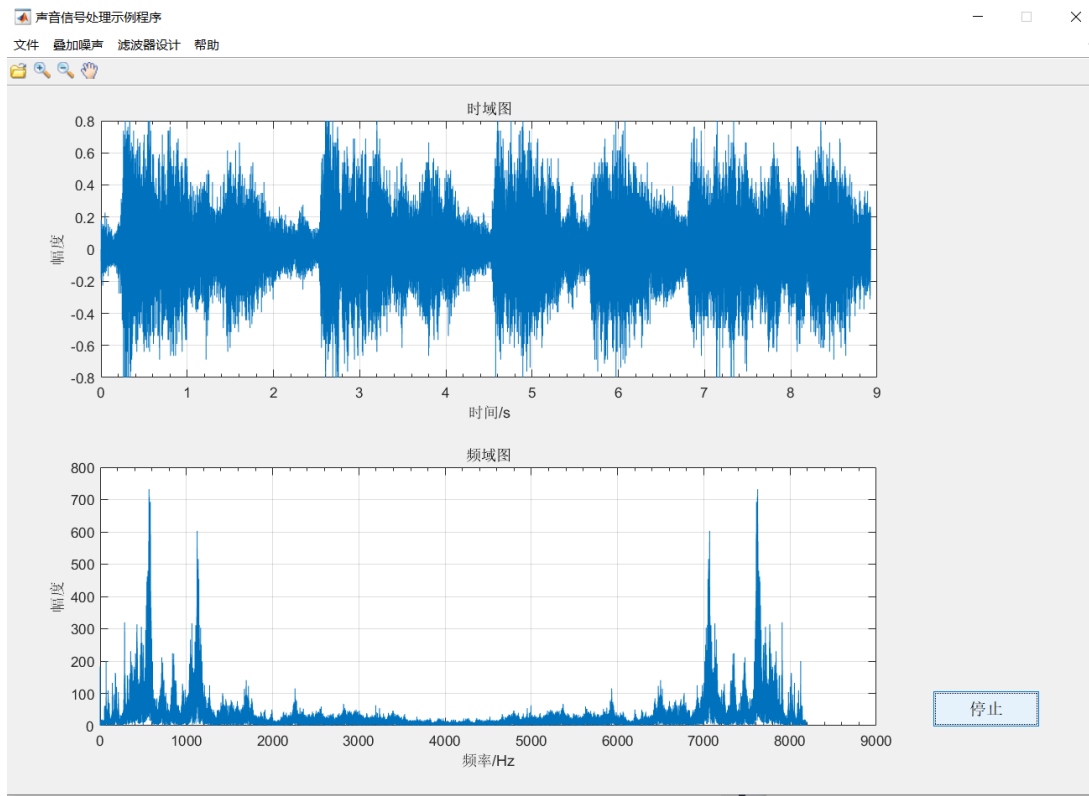


图 4 系统界面展示

### 3.2 音频的时域频域分析

导入音频文件或录音文件后，对音频信息进行时域分析和频域分析，具体核心代码如下：

```

1  %时域
2  reswav = IIR_filter(Wp,Ws,Rp,Rs,wav,fs);
3  assignin('base','resdata',reswav);
4  fftwav = abs(fft(reswav));
5  axes(handles.Time);
6  x = (0:length(reswav)-1)/fs;
7  handles.Line1 =plot(x,reswav);
8  guidata(hObject,handles);%保存值
9  set(handles.Time,'XMinorTick','on');
10 grid on;
11 xlabel('时间/s');
12 ylabel('幅度');

```



```

13 title('时域图');
14 %频域
15 axes(handles.Freq);
16 xf = (0:length(reswav)-1)*fs/length(fftwav);
17 handles.Line2 = plot(xf,fftwav);
18 guidata(hObject,handles);%保存值
19 set(handles.Freq,'XMinorTick','on');
20 grid on;
21 xlabel('频率/Hz');
22 ylabel('幅度');
23 title('频域图');

```

### 3.3 音频信号加噪声分析

对音频基本的时域和频域观察后，增加高斯白噪声核心代码如下：

```

1 wav = evalin('base','data');
2 fs = evalin('base','Fs');
3 reswav = awgn(wav,20);%添加信噪比为 10dB 的高斯噪声
4 assignin('base','resdata',reswav);
5 fftwav = abs(fft(reswav));
6 axes(handles.Time);
7 x = (0:length(reswav)-1)/fs;
8 handles.Line1 = plot(x,reswav);
9 guidata(hObject,handles);%保存值
10 set(handles.Time,'XMinorTick','on');
11 grid on;
12 xlabel('时间/s');
13 ylabel('幅度');
14 title('时域图');

```

### 3.4 音频信号经 IIR 和 FIR 滤波器分析

利用巴特沃斯滤波器和双线性变换法设计实现 IIR 滤波器，代码如下：

```

1 function resdata = IIR_filter(wp,ws,rp,rs,data,Fs)
2
3 Wp = 0.20*2*pi;Ws=0.25*2*pi;Rp=1;Rs=15; %数字性能指标
4 %Wp=wp*pi;Ws=ws*pi;Rp=rp;Rs=rs; Ts = 1/Fs;
5 Wp1 = (2/Ts)*tan(Wp/2); %将数字指标转换成模拟指标
6 Ws1 = (2/Ts)*tan(Ws/2);
7 [N,Wn] = buttord(Wp1,Ws1,Rp,Rs,'s'); %N 滤波器的最小阶数，Wn 截止频率
8 [Z,P,K] = buttap(N); %求模拟滤波器的系统函数，零极点和增益形式
9 [Bap,Aap] = zp2tf(Z,P,K); %变为多项式形式
10 [b,a] = lp2lp(Bap,Aap,Wn); %去归一化

```

```

11 [bz,az] = bilinear(b,a,Fs);      %双线性变换法实现 AF 到 DF 的转换
12
13 figure('NumberTitle', 'off', 'Name', 'IIR数字滤波器设计结果','menubar','none');
14 freqz(bz,az,Fs);      %滤波器的频率响应
15 resdata=filter(bz,az,data);      %数字滤波

```

利用窗函数法设计，并选用哈明窗设计 FIR 滤波器，代码如下：

```

1 %选哈明窗
2 function resdata = FIR_filter(wp,ws,rp,rs,data,Fs)
3
4 Wp=wp*1000*2/Fs;% 频率归一化
5 Ws=ws*1000*2/Fs;
6 wdel=Ws-Wp;% 过渡带宽
7 wn=0.5*(wp+ws);% 近似计算截止频率
8 N=ceil(6.6*pi/wdel);% 根据过渡带宽度求滤波器阶数
9 window=hamming(N+1);% 哈明窗
10 b=fir1(N,wn,window);% FIR 滤波器设计
11 figure('NumberTitle', 'off', 'Name', 'FIR数字滤波器设计结果','menubar','none');
12 freqz(b,1,512);% 查看滤波器幅频及相频特性
13 resdata = filter(b,1,data);%对信号 data 进行滤波

```

## 四、实验测试与结果分析

### 4.1 音频时域频域分析结果

导入一段长约 9 秒的音频到设计的音频数字处理系统中，获得的时域频域图如图5所示：

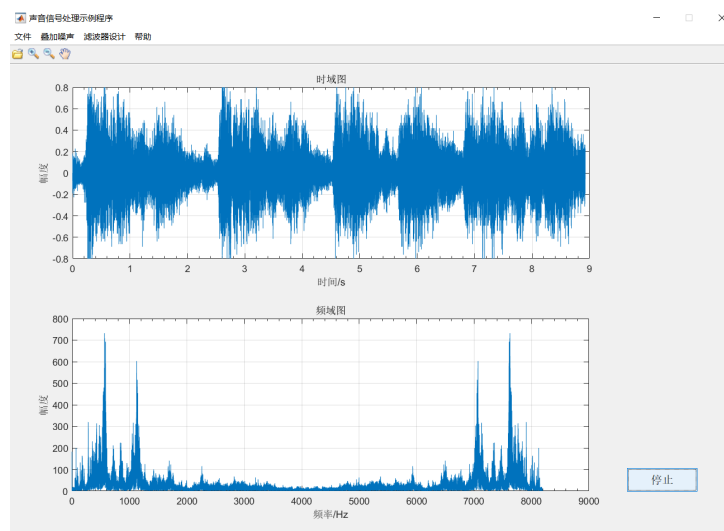


图 5 导入音频后时域频域图

通过分析可以看出，该段音频的低频和低频成分相对较多，中频成分占比较少。

## 4.2 音频加噪声分析结果

在源声音文件上添加高斯白噪声后的得到的结果如图6所示。可以看到高斯白噪声的添加使得音频频谱在各个频段普遍叠加上了一段信号，该信号是高斯白噪声的影响：

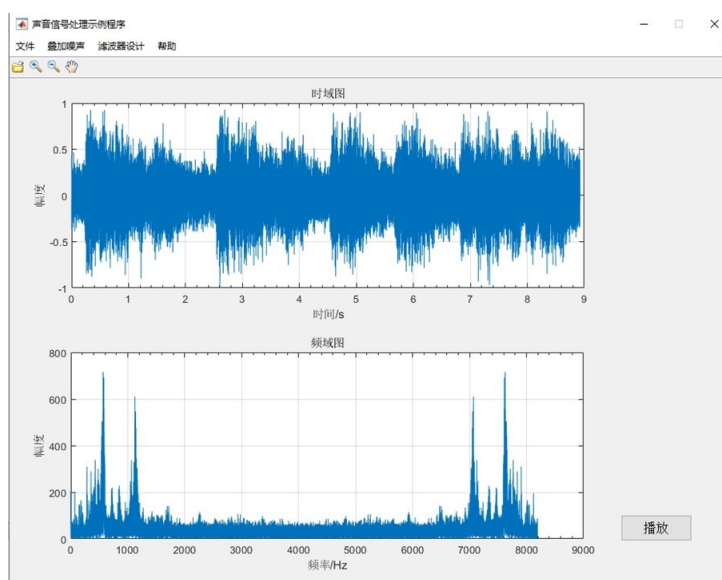


图 6 添加高斯白噪声后的音频波形图

在源声音文件上添加单频噪声得到的结果如图7所示：

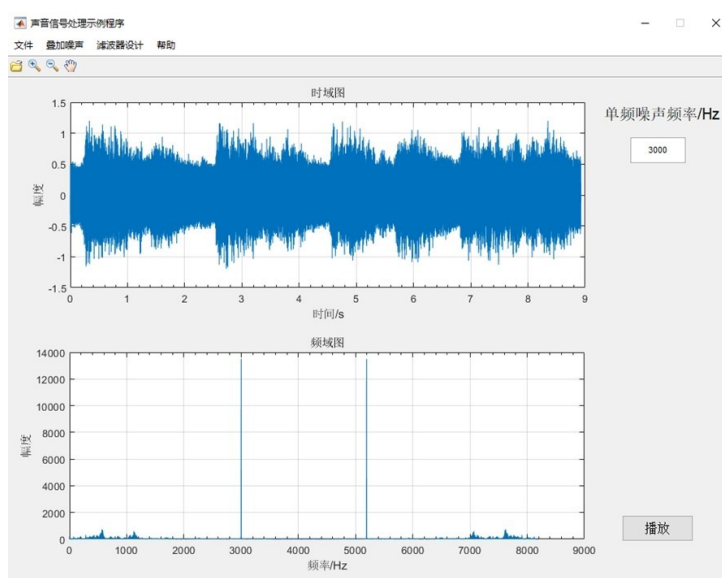


图 7 添加单频噪声后的音频波形图

在源声音文件上添加多频噪声得到的结果如图8所示：

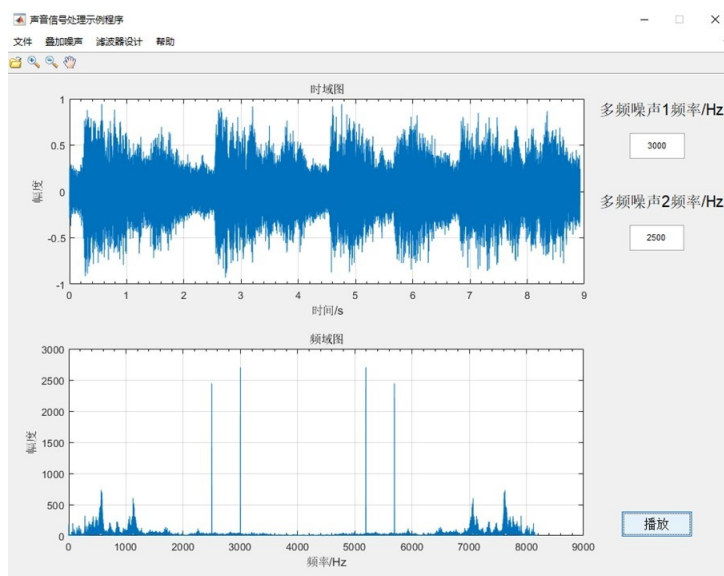


图 8 添加多频噪声后的音频波形图

### 4.3 音频经 IIR 和 FIR 滤波器分析结果

IIR 滤波器设置参数（通带频率  $W_p=0.2\text{kHz}$ ；阻带截至  $W_s=0.25\text{kHz}$ ；通带衰减  $R_p=1\text{dB}$ ；阻带衰减  $R_s=15\text{dB}$ ）后得到的滤波器特性如图9所示，经过该滤波器得到的音频特性如图10所示。可以看到，经过 IIR 滤波器后的滤去了很多幅值较低的中频信号，基本保留了低通信号。

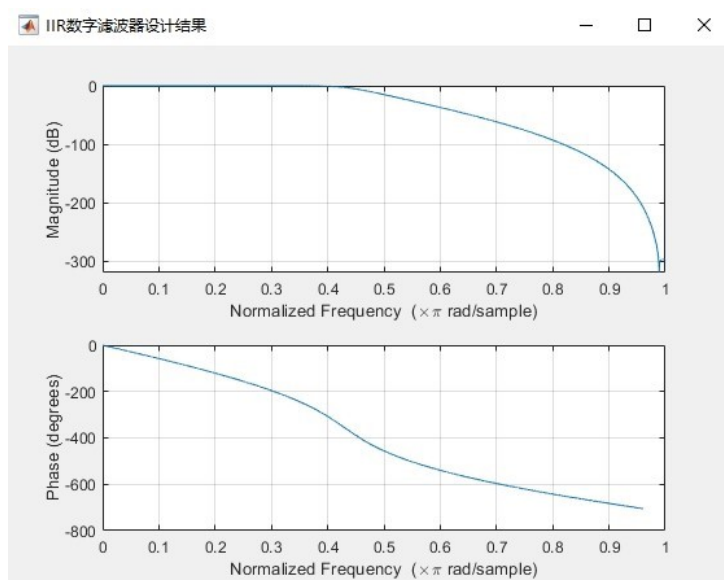


图 9 IIR 滤波器幅值相位特性曲线

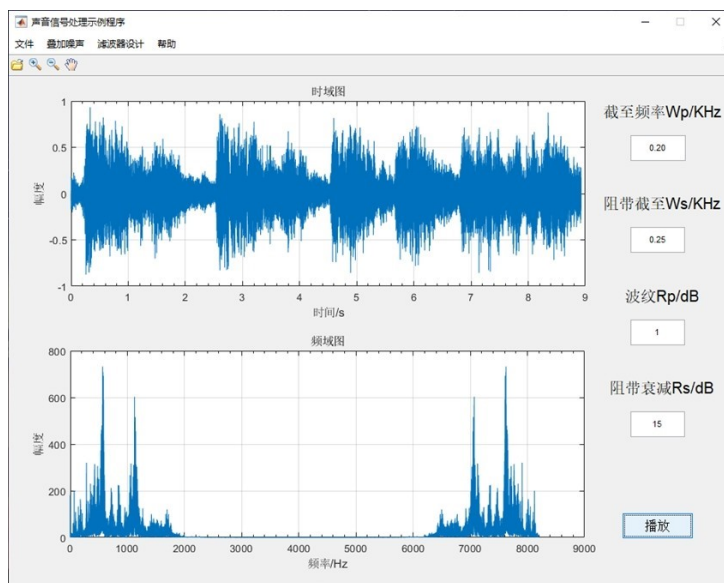


图 10 经过 IIR 滤波器得到的音频特性

FIR 滤波器设置参数（通带频率  $W_p=0.2\text{kHz}$ ；阻带截至  $W_s=0.25\text{kHz}$ ；阻带衰减  $R_s=50\text{dB}$ ）后得到的滤波器特性如图11所示，经过该滤波器得到的音频特性如图12所示：

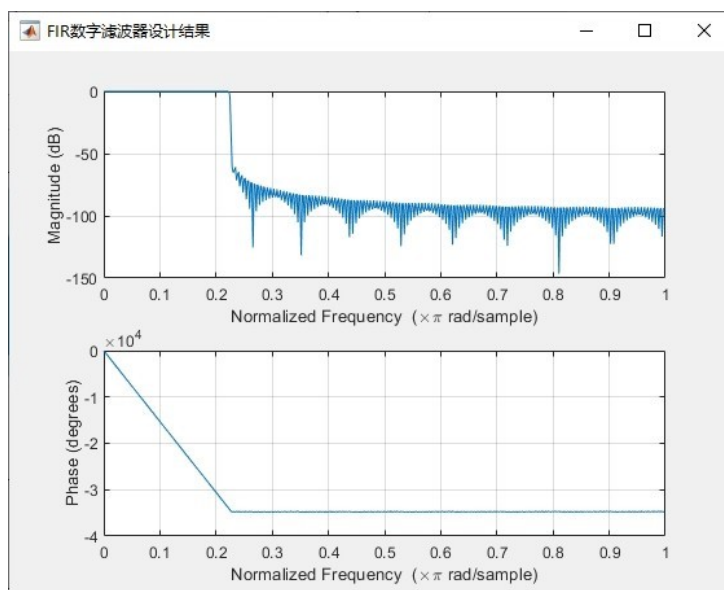


图 11 FIR 滤波器幅值相位特性曲线

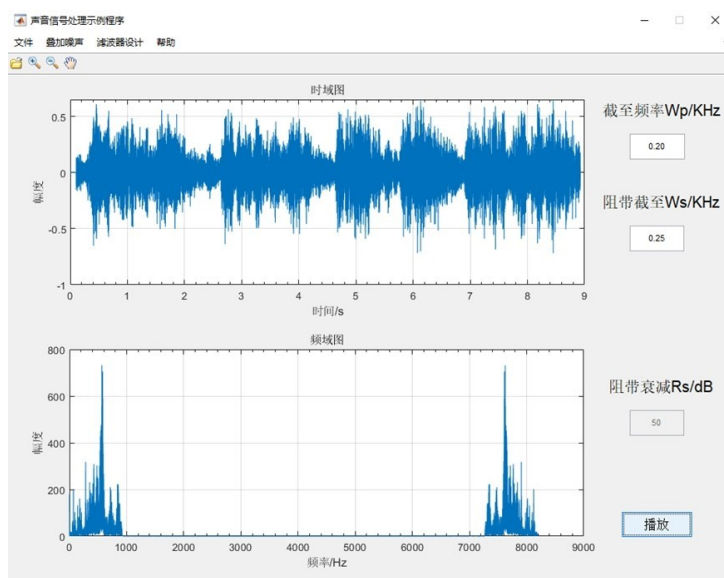


图 12 经过 FIR 滤波器得到的音频特性

## 五、设计总结及遇到的问题

### 5.1 设计总结

通过对音频数字信号处理系统的设计，对数字信号处理课上的学习的理论知识包括基本的离散傅里叶变换，快速傅里叶变换，IIR 滤波器和 FIR 滤波器的实现，以及信号处理常见的噪声问题进行实践，深刻了解了数字信号处理系统的过程步骤，并对理论知识有了新的独特的理解。

并且结合以前 MATLAB 课程的学习，通过 MATLAB 这一数学编程工具来实现一个简单的音频处理系统。通过亲身操作实践，同时还学到了很多课外知识，不仅仅局限于课本的理论，将实践与理论结合起来的时候，才是真的掌握了所学的知识。

课内课外我们都应该有一些思考和想法，作为人工智能实验班的的一名学生，应该严格要求自己，扩展自己的视野，多多实践才是王道，将软件技术和硬件技术原理结合在一起，真的是一件比较酷的事情！收获了不仅仅是熟练的 MATLAB 编程技巧、信号与系统及后置课程数字信号处理的进一步知识体系深化、更有的是满满的解决问题的成就感。

### 5.2 问题及解决方法

试验去噪效果时，一般噪声都不会特别明显，所以通常的做法是加噪声进行处理，于是我对音频进行了加噪声处理，效果会比较好。

双声道音频要改为单音频！起先用的是吉他音频信号，从加噪声到 FFT 变换，再到谱减法降噪，都没有出现问题。但当用自己手机的录音文件时却出现了无法 load 的问题，很困惑，不知道问题出现在哪里，起初以为只能 load MP3 格式的音频文件，但是

将其修改了仍没有用还是报错，后面发现是编码的问题，不同的手机有不同的音频编码格式，苹果的手机可以正常工作。

## 六、致谢

感谢朱莉老师 15 周的信号与系统课程的教诲，感谢您教会了我如何去和实践结合去解决问题，同时也应当积极关注前沿的动态，将死板的知识与自己的特长灵活结合，取长补短，创造新的价值；更感谢您不厌其烦地在课后为我解答问题，没有嫌弃我常常脑子真的转不过来；信号与系统真的特别难，很幸运能有您的帮助，打下坚实的基础，在风雨中走过在摇摇晃晃的独木桥。完整的代码开源到了 Github 上，如果您有教学的需求可以使用：<https://github.com/JerryYin777/NCU-Signals-and-Systems-Project>

## 参考文献

- [1] 曹亮. 基于 DSP 的音频信号处理系统设计 [J]. 电子技术与软件工程,2019(10):95.
- [2] 詹颖珩. 基于 MATLAB 的椭圆滤波器对音频信号的分离 [J]. 通讯世界,2018(06):273-274
- [3] 刘海波,汤群芳. 基于 MATLAB 的 IIR 滤波器的设计研究 [J]. 软件导刊,2008(11):187-188.
- [4] 周红鸥. 基于 DSP 的音频信号采集处理系统设计 [J]. 西南民族大学学报 (自然科学版),2011,37(S 1):100-103.
- [5] 冯志鸿.DSP 设计的数字音频信号处理 [J]. 电子技术与软件工程,2019(10):93.
- [6] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” arXiv preprint arXiv:1610.05492, 2016.
- [7] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10 700–10 714, 2019.
- [8] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, “Communication-efficient edge ai: Algorithms and systems,” arXiv preprint arXiv:2002.09668, 2020.
- [9] Y. Liu, J. Peng, J. Kang, A. M. Ilyasu, D. Niyato, and A. A. A. ElLatif, “A secure federated learning framework for 5g networks,” arXiv preprint arXiv:2005.05752, 2020.

- [10] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” in International Conference on Learning Representations, 2018. [Online]. Available: <https://openreview.net/forum?id=SkhQHMW0W>