

目录

音频数字处理系统设计	2
一 . 研究背景	2
二 . 实验环境	2
三 . 设计原理	2
1. 理论知识	2
2. 算法原型	5
四 . 系统方案	6
1. 音频处理系统界面搭建	6
2. 音频的时域频域分析	6
3. 音频信号加噪声分析	6
4. 音频信号经 IIR 和 FIR 滤波器分析	7
五 . 实验测试与结果分析	8
1. 音频时域频域分析结果	8
2. 音频加噪声分析结果	8
3. 音频经 IIR 和 FIR 滤波器分析结果	10
六 . 设计总结	12
七 . 致谢	12
八 . 参考文献	13

音频数字处理系统设计

通信 1701 胡成成 41724260

一． 研究背景

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。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。

同时，在数字信号处理课堂上已经学了很多数字信号处理的模型和算法，例如基本的 DFT，FFT，IIR 和 FIR 等方法的学习，将这些基本的处理方法和滤波器实现应用在音频的处理上更有助于我们对这些方法的理解。因此，此次大作业我选择制作一个音频数字处理的系统，可以实现基本的时域频域分析和噪声的一些添加处理，以及 IIR 和 FIR 滤波器的实现。

二． 实验环境

操作系统：Window 10；

设计软件：MATLAB R2019a；

三． 设计原理

1. 理论知识

1.1． DFT 的基本原理

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：

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

其中 $X(k)$ 的离散傅里叶逆变换(IDFT)则定义为：

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

例如：用 DFT 计算理想低通滤波器的频响曲线图 1 所示。

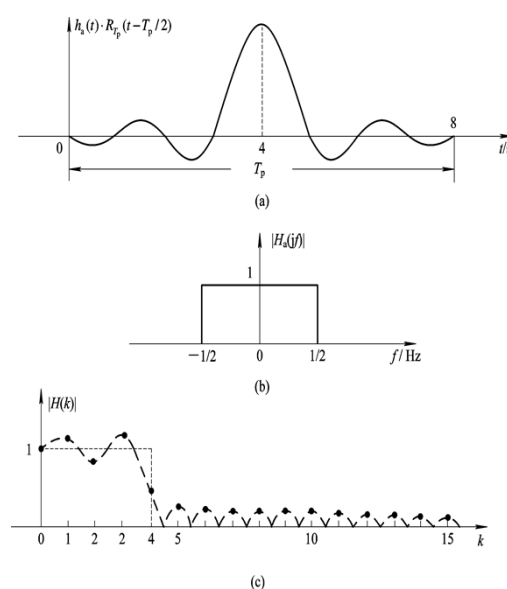


Figure 1 : DFT 计算理想低通滤波器的频响

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什么特征的，但是如果变换到频域之后，就很容易看出特征了。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 对 FFT 算法

的设计。

1.2 . 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。

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：

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

无限长单位脉冲响应(IIR)滤波器的系统函数为：

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

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 $H(e^{j\omega})$ 用下式表示： $H(e^{j\omega}) = |H(e^{j\omega})|e^{j\theta(\omega)}$ 。式中， $|H(e^{j\omega})|$ 称为幅频特性函数； $\theta(\omega)$ 称为相频特性函数。

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。课上我们主要是对巴特沃斯滤波器的设计进行了详细地学习。

1.3. 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 2 所示。

窗函数类型	旁瓣峰值 α_n /dB	过渡带宽度 B_t		阻带最小衰减 α_s /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

Figure 2：窗函数特性总结

2. 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: $X_k = \text{fft}(x_n, N)$ ；lfft 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（），butter（）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

四 . 系统方案

1. 音频处理系统界面搭建

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。设计出基本的图形化界面如图 3 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。

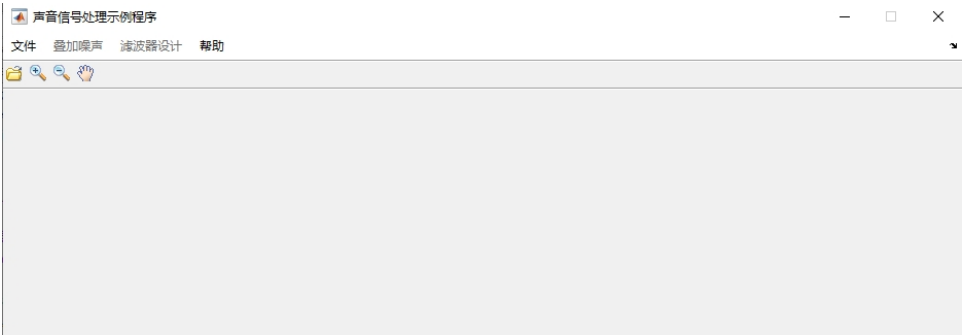


Figure 3: 音频信号处理图形化界面

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：左边是时域分析，右边是频域分析。

```
reswav = wav;
assignin('base','resdata',reswav);
fftwav = abs(fft(reswav));
axes(handles.Time);
x = (0:length(reswav)-1)/fs;
handles.Line1 = plot(x,reswav);
guidata(hObject,handles);%保存值
set(handles.Time,'XMinorTick','on');
grid on;
xlabel('时间/s');
ylabel('幅度');
title('时域图');

axes(handles.Freq);
xf = (0:length(reswav)-1)*fs/length(fftwav);
handles.Line2 = plot(xf,fftwav);
guidata(hObject,handles);%保存值
set(handles.Freq,'XMinorTick','on');
grid on;
xlabel('频率/Hz');
ylabel('幅度');
title('频域图');
assignin('base','flag',1);
```

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：

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

- 添加单频噪声和多频噪声核心代码如下：（左边单频噪声，右边多频噪声）

<pre>wav = evalin('base','data'); fs = evalin('base','Fs'); ts = 1/fs; t = 0:ts:(length(wav)-1)*ts; t = t'; fn = str2double(get(handles.E1,'String')); if(fn > fs/2) string =str2mat('参数错误'); msgbox(string); return; end; %Ps=sum(sum((wav-mean(mean(wav))).^2));%signal power single_noise = max(wav)*0.5*sin(2*pi*fn*t); reswav = wav + single_noise; assignin('base','resdata',reswav); fftwav = abs(fft(reswav)); axes(handles.Time); x = (0:length(reswav)-1)/fs; handles.Line1 =plot(x,reswav);</pre>	<pre>wav = evalin('base','data'); fs = evalin('base','Fs'); ts = 1/fs; t = 0:ts:(length(wav)-1)*ts; t = t'; fn1 = str2double(get(handles.E1,'String')); fn2 = str2double(get(handles.E2,'String')); if(fn1>fs/2 fn1>fs/2) string =str2mat('参数错误'); msgbox(string); return; end; %Ps=sum(sum((wav-mean(mean(wav))).^2));%signal power single_noise = max(wav)*0.1*sin(2*pi*fn1*t) + max(wav)*0.1*sin(2*pi*fn2*t); reswav = wav + single_noise; assignin('base','resdata',reswav); fftwav = abs(fft(reswav)); axes(handles.Time); x = (0:length(reswav)-1)/fs; handles.Line1 =plot(x,reswav);</pre>
---	--

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; %数字性能指标
5     Ts = 1/Fs;
6     Wp1 = (2/Ts)*tan(Wp/2); %将数字指标转换成模拟指标
7     Ws1 = (2/Ts)*tan(Ws/2);
8     [N,Wn] = buttord(Wp1,Ws1,Rp,Rs,'s'); %N滤波器的最小阶数，Wn截止频率
9     [Z,P,K] = buttap(N); %求模拟滤波器的系统函数，零极点和增益形式
10    [Bap,Aap] = zp2tf(Z,P,K); %变为多项式形式
11    [b,a] = lp2lp(Bap,Aap,Wn); %去归一化
12    [bz,az] = bilinear(b,a,Fs); %双线性变换法实现AF到DF的转换
13
14    figure('NumberTitle','off','Name','IIR数字滤波器设计结果','menubar','none');
15    freqz(bz,az,Fs); %滤波器的频率响应
16    resdata=filter(bz,az,data); %数字滤波
```

- 利用窗函数法设计，并选用哈密窗设计 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

```

五．实验测试与结果分析

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 4 所示。

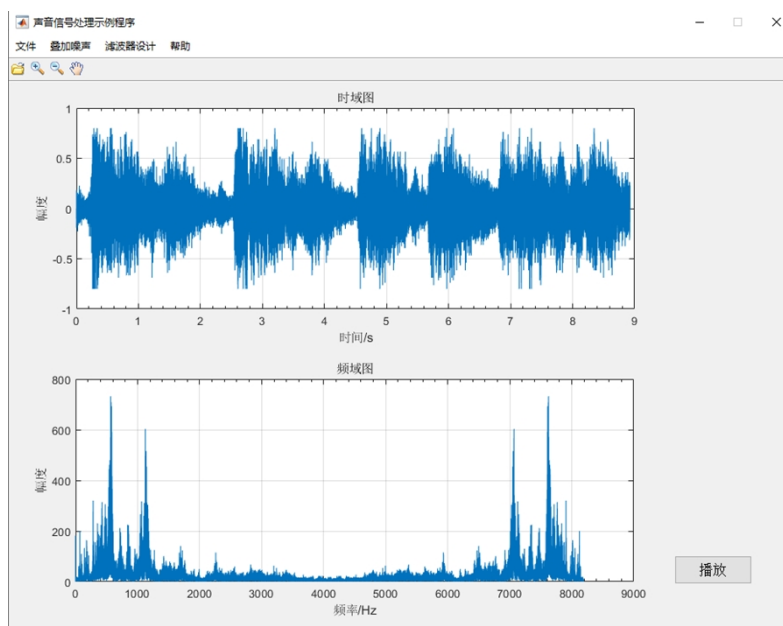


Figure 4: 导入音频后时域频域图

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。

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 5 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，该信号即使

高斯白噪声的影响。

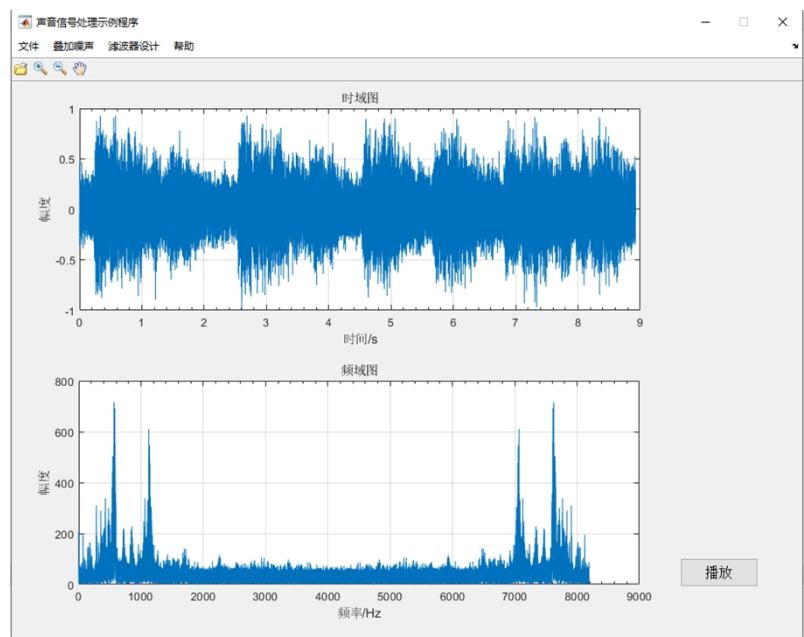


Figure 5: 添加高斯白噪声后的音频波形图

➤ 在源声音文件上添加单频噪声得到的结果如图 6 所示.

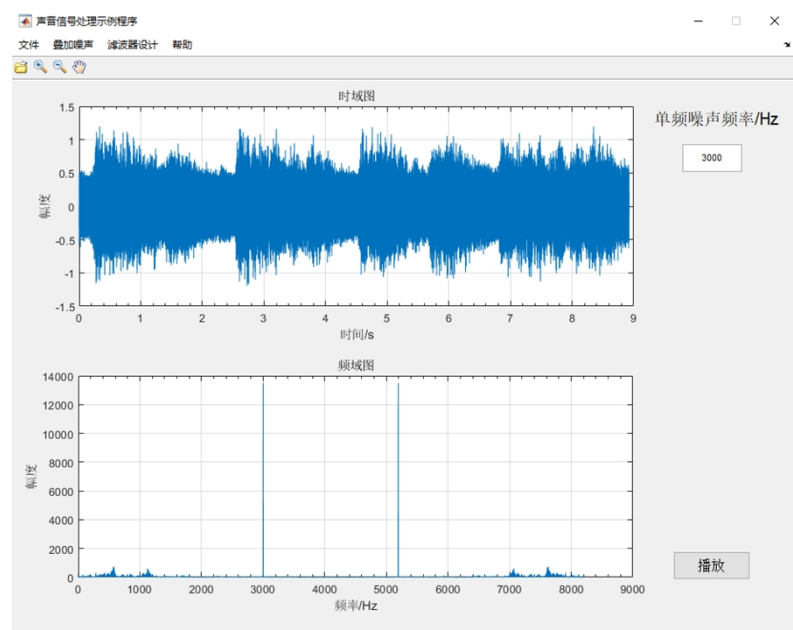


Figure 6: 添加单频噪声得到的音频波形

➤ 在源声音文件上添加多频噪声得到的结果如图 7 所示.

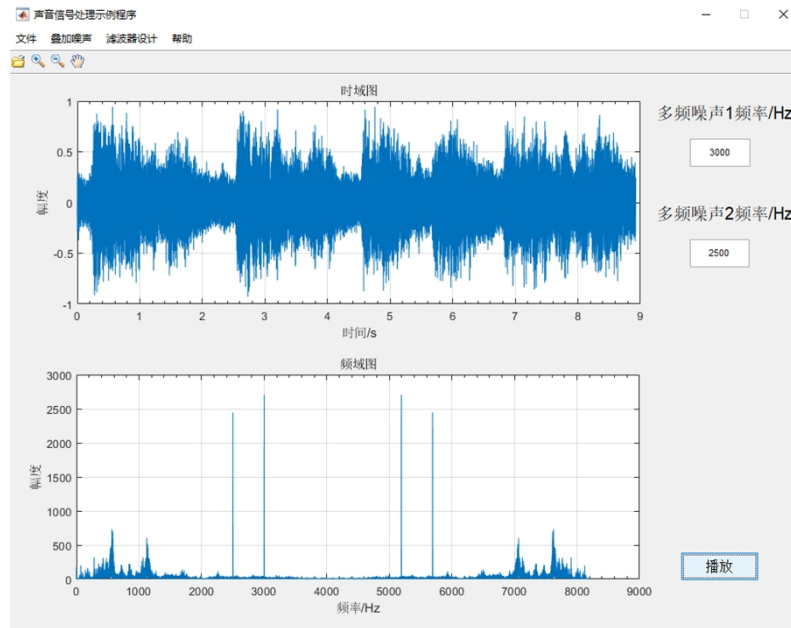


Figure 7: 添加多频噪声得到的音频波形

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 8 所示，经过该滤波器得到的音频特性如图 9 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。

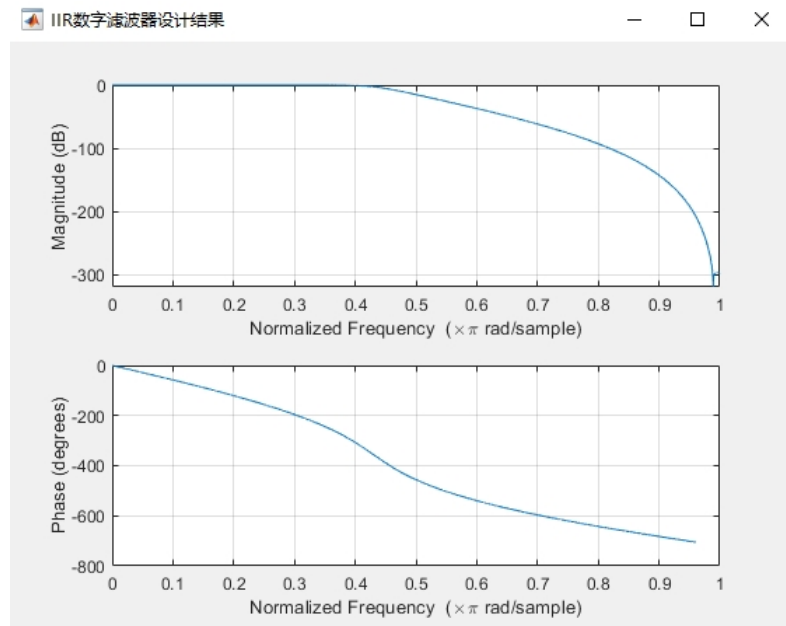


Figure 8: IIR 滤波器幅值相位特性曲线

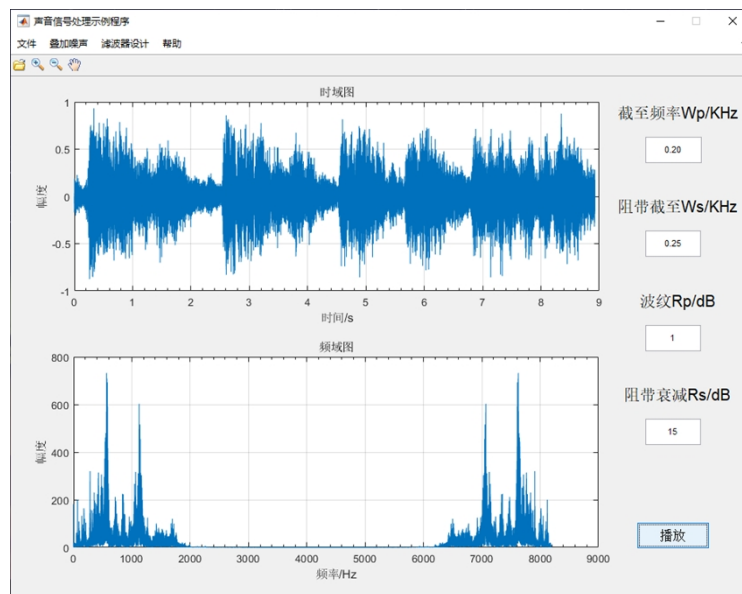


Figure 9: 经过 IIR 滤波器得到的音频特性

- 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 10 所示，经过该滤波器得到的音频特性如图 11 所示。

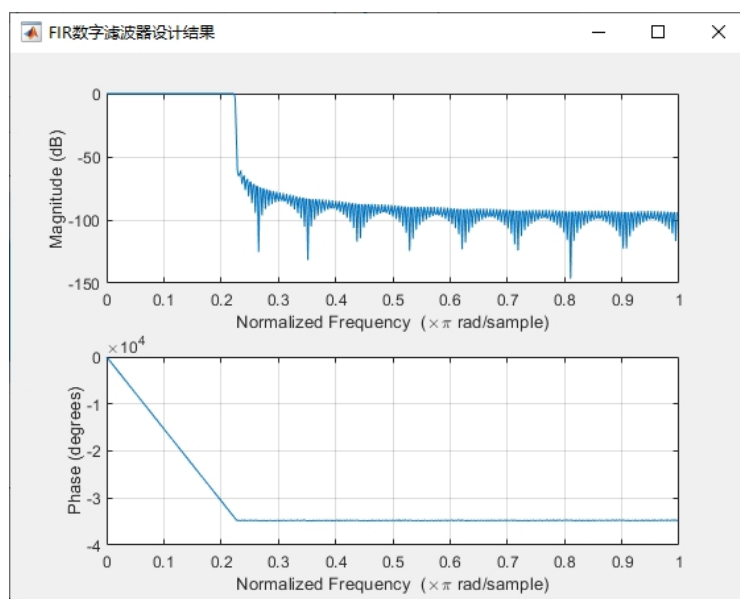


Figure 10: FIR 滤波器幅值相位特性曲线

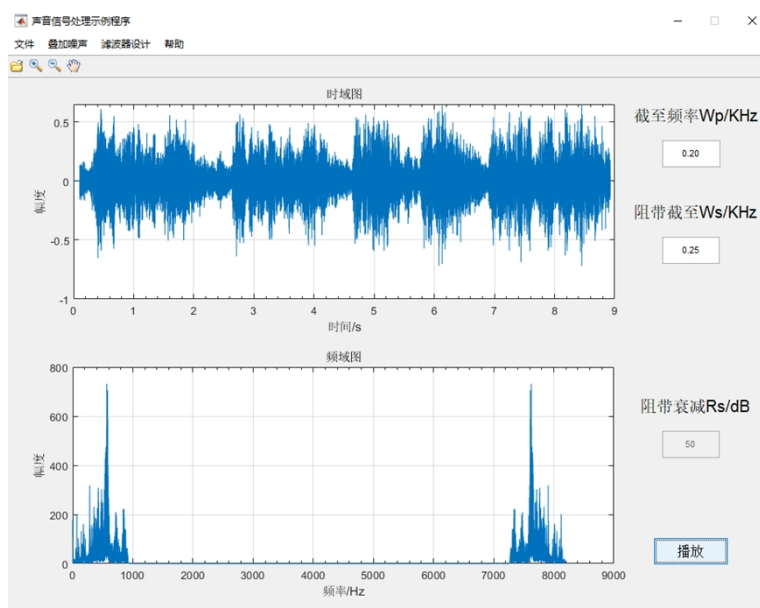


Figure 11: 经过 FIR 滤波器得到的音频特性

六．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,作为通信工程的一名大学生,应该严格要求自己,扩展自己的视野,多多实践才是王道。

七．致谢

感谢杨裕亮老师 12 周的数字信号处理课程的教诲;让我们知道作为通信人的一份子,不能满足于课本知识的学习,要更多地去实践,通过编程验证自己的想法。为此,本次设计的音频数字处理系统完整的代码已经上传的个人的博客地址,可供老师下载参考:<https://github.com/JackHCC/Audio-Digital-Processing>

八．参考文献

- [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(S1):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.