

Matlab语音信号滤波

陆毅



2023-2-5

华中科技大学

船舶与海洋工程学院

目录

[一、设计内容 2](#_Toc126518538)

[二、设计原理 2](#_Toc126518539)

[1、采样定理 2](#_Toc126518540)

[2、离散傅里叶变换 2](#_Toc126518541)

[3、滤波器设计 3](#_Toc126518542)

[三、设计过程 3](#_Toc126518543)

[1、采样语音信号，绘制时域波形图和频谱图 3](#_Toc126518544)

[2、对信号进行低通滤波，比较滤波前后的波形、频谱、听感 4](#_Toc126518545)

[3、对语音信号加入噪声，并进行滤波处理 5](#_Toc126518546)

[四、结论 7](#_Toc126518547)

[五、参考文献 8](#_Toc126518548)

# MATLAB语音信号滤波

陆毅 轮机2001班 U202010827

[摘要]按照一定的频率对音频信号进行采样，得出信号波形图；对采样后的信号进行傅里叶变换，得出信号频谱图。使用低通滤波器过滤高频噪声信号，对音频进行降噪。

## 一、设计内容

1. 采样语音信号，绘制时域波形图和频谱图
2. 用低通滤波器对信号进行滤波，比较滤波前后的波形、频谱、听感
3. 对语音信号加入噪声，并进行滤波处理

## 二、设计原理

### 1、采样定理

对连续时间进行数字处理，必须首先对信号进行采样，每隔时间T进行一次采样，每次采样时间为t，即可得到宽度为t、间隔为T的脉冲串，根据香农采样定理，为了避免出现混叠现象，采样的频率不宜过低，本次设计使用的采样频率为.

### 2、离散傅里叶变换

DFT：以时间为自变量的信号和以频率为自变量的频谱函数之间的一种变换关系。对于一个非周期的连续时间信号，其傅里叶变换得到连续的频谱：

FFT：一种快速计算DFT的方法。这种方法充分利用了DFT运算中的对称性和周期性，从而将DFT运算量从减少到。当N比较小时，FFT优势并不明显。但当N大于32开始，点数越大，FFT对运算量的改善越明显。比如当为1024时，FFT的运算效率比DFT提高了100倍。其基本原理是先将一个点时域序列的DFT分解为个1点序列的DFT，然后将这样计算出来的N个1点序列DFT的结果进行组合，得到最初的N点时域序列的DFT值。

### 3、滤波器设计

电容有“通高频、阻低频”的特性，即高频信号能很容易通过电容，而低频信号则很困难，电感则恰恰相反。我们可以利利用这一特性，在信号电路中使用电感或者电容将某些频率的信号过滤掉。

本次设计使用的是低通滤波器。对于最简单的一阶RC低通滤波器，其截止频率为

实际电路中通过级联多个RC环节可提高滤波器的阶数，以改善滤波性能。而在matlab仿真过程，低通滤波器只需选择合适的阶数及截止频率，即可利用函数完成滤波处理。

## 三、设计过程

### 1、采样语音信号，绘制时域波形图和频谱图

**采样：**

首先调用函数audioread对音频文件“test.mp3”进行采样，Fs为采样频率

[x,Fs]=audioread('test.mp3');

采样后的x包含的是音频左右两个声道的数字信号，我们取其中一个即可

x = x(:,1);

x = x';

N = length(x);

t = (0:N-1)/Fs; %时间=采样点数/采样频率

y = fft(x); %对信号进行傅里叶变换

f = Fs/N\*(0:round(N/2)-1); %频率=采样频率/采样点数（由对称性只取一半）

**绘图：**

subplot(2,1,1);

plot(t,x,'g');%绘制时域波形

xlabel('t/s');ylabel('Amplitude');

title('信号的波形');

grid;

subplot(2,1,2);

plot(f,abs(y(1:round(N/2))));

xlabel('f/Hz');ylabel('Amplitude');

title('信号的频谱');

xlim([0 1000]);

grid;

**结果：**

图片包含 图表

描述已自动生成

### 2、对信号进行低通滤波，比较滤波前后的波形、频谱、听感

设计阶数为10，截止频率为的低通滤波器：

1. function Hd = lowPass
2. Fs = 44100; % Sampling Frequency
3. N  = 10;     % Order
4. Fc = 1000;   % Cutoff Frequency
5. h  = fdesign.lowpass('N,F3dB', N, Fc, Fs);
6. Hd = design(h, 'butter');

对采样的语音信号x进行滤波：

h=lowPass*;*

xlp=filter(h,x)*;*

ylp=fft(xlp)*;*

**绘图：**

日程表

描述已自动生成

对比滤波前后频谱图可见高于的信号已基本被过滤，时域波形图也有一部分区域波形被削弱。

播放音频：

sound(xlp,Fs)；

可感受到音频中明亮的声音消失了，声音显得低沉一些。

### 3、对语音信号加入噪声，并进行滤波处理

首先生成10个之间的随机频率

noise\_f=100\*rand(1,10)+900\*ones(1,10);

将该随机频率的正弦信号叠加形成噪声信号

xx=0;

for k=1:length(noise\_f)

    xx=0.02\*sin(2\*pi\*noise\_f(k)\*t)+xx;

end

将噪声信号加入原音频信号，并进行傅里叶转换

xx=xx+x;

yy=fft(xx);

观察时域信号图和频域信号图：

日程表

描述已自动生成

显然在处出现了许多高峰值，即为噪声信号。

播放音频：

sound(xx,Fs)；

能听到明显噪声。

对信号进行滤波：

1. function Hd = noiseFilter
2. Fs = 44100;  % Sampling Frequency
3. N  = 20;   % Order
4. Fc = 750;  % Cutoff Frequency
5. h  = fdesign.lowpass('N,F3dB', N, Fc, Fs);
6. Hd = design(h, 'butter');
7. h=noiseFilter*;*
8. xxlp=filter(h,xx)*;*
9. yylp=fft(xxlp)*;*
10. subplot(211);
11. plot(f,abs(yy(1:round(N/2))));
12. xlabel('f/Hz');ylabel('Amplitude');
13. title('带噪声的信号频谱');
14. xlim([0 1000]);
15. grid;
16. subplot(212);
17. plot(f,abs(yylp(1:round(N/2))));
18. xlabel('f/Hz');ylabel('Amplitude');
19. title('降噪后的信号频谱');
20. xlim([0 1000]);
21. grid;

图表

低可信度描述已自动生成

可见，高频噪声信号已被消除，但同时也损失了原音频的高频信号。

播放音频：

sound(xxlp,Fs)；

可感受到噪声已被消除，但失去高频信号的音频听感也变得低沉。

## 四、结论

通过本次学习，学会了初步使用matlab对模拟信号的采样以及对数字信号的滤波处理，此次的噪声信号只是自己给定频率范围的，所以较好过滤，但在实际应用中会有更复杂的噪声信号，使得降噪变得更加复杂，对于信号滤波仍需加强学习，学习一些先进高效的滤波算法和硬件设计，在未来的实际应用中做出更好的效果。

本次课程设计的认识及建议：

个人感觉太简单了些，当然前提是如果我没有理解错题目要求的话。特别是信号滤波的内容，都是matlab自带的一些工具和函数，直接套用即可。

虽然我觉得简单但不代表我认为自己做得很好，只是感觉少了一种它很贴合课程内容的感觉，设计过程更多的是来自查阅课外资料。

并非评价好坏，因为其它课程的一些课设也让我有这种感觉，也许大学就是这样的，要培养学生自学技能和即学即用的能力。

对于教学建议我没有什么想说的，老师讲课讲得很清楚，该说的知识点也讲明白了，感谢老师的辛勤付出！

## 五、参考文献

[1]张文,蒋猛,陈智.基于MATLAB的语音信号滤波设计与实现[J].山西电子技术,2008,No.137(02):29-31.

[2]周玲.基于MATLAB的语音信号数字滤波处理[J].安庆师范学院学报(自然科学版),2011,17(03):46-49.

[3]瞿瑞杰.实现语音信号滤波的数字滤波器设计[J].设备管理与维修,2018,No.417(02):78-80.DOI:10.16621/j.cnki.issn1001-0599.2018.01D.45.