

MATLAB 的音频信号处理技术实现

1 、 MATLAB 处理音频信号的流程

分析和处理音频信号，首先要对声音信号进行采集，MATLAB 的数据采集工具箱提供了一整套命令和函数,通过调用这些函数和命令,可直接控制声卡进行数据采集[。Windows 自带的录音机程序也可驱动声卡来采集语音信号，并能保存为 WAV 格式文件，供 MATLAB 相关函数直接读取、写入或播放。本文以 WAV 格式音频信号作为分析处理的输入数据，用 MATLAB 处理音频信号的基本流程是：先将 WAV 格式音频信号经 wavread 函数转换成 MATLAB 列数组变量；再用 MATLAB 强大的运算能力进行数据分析和处理，如时域分析、频域分析、数字滤波、信号合成、信号变换、识别和增强等等；处理后的数据如是音频数据，则可用 wavwrite 转换成 WAV 格式文件或用 sound、wavplay 等函数直接回放。下面分别介绍 MATLAB 在音量标准化、声道分离合并与组合、数字滤波、数据转换等音频信号处理方面的技术实现。

2 、 MATLAB 处理音频信号的流程

分析和处理音频信号，首先要对声音信号进行采集，MATLAB 的数据采集工具箱提供了一整套命令和函数,通过调用这些函数和命令,可直接控制声卡进行数据采集[1]。Windows 自带的录音机程序也可驱动声卡来采集语音信号，并能保存为 WAV 格式文件，供 MATLAB 相关函数直接读取、写入或播放。本文以 WAV 格式音频信号作为分析处理的输入数据，用 MATLAB 处理音频信号的基本流程是：先将 WAV 格式音频信号经 wavread 函数转换成 MATLAB 列数组

变量；再用 MATLAB 强大的运算能力进行数据分析和处理，如时域分析、频域分析、数字滤波、信号合成、信号变换、识别和增强等等；处理后的数据如是音频数据，则可用 wavwrite 转换成 WAV 格式文件或用 sound、wavplay 等函数直接回放。下面分别介绍 MATLAB 在音量标准化、声道分离合并与组合、数字滤波、数据转换等音频信号处理方面的技术实现。

3、音量标准化

录制声音过程中需对声音电平进行量化处理，最理想的量化是最大电平对应最高量化比特，但实际却很难做到，常有音轻问题。利用 MATLAB 很容易实现音量标准化，即最大电平对应最高量化比特。基本步骤是：先用 wavread 函数将 WAV 文件转换成列数组变量；再求出数组变量的极值并对所有元素作归一化处理；最后用 wavwrite 函数还原成音量标准化的 WAV 文件。

录制声音过程中需对声音电平进行量化处理，最理想的量化是最大电平对应最高量化比特，但实际却很难做到，常有音轻问题。利用 MATLAB 很容易实现音量标准化，即最大电平对应最高量化比特。基本步骤是：先用 wavread 函数将 WAV 文件转换成列数组变量；再求出数组变量的极值并对所有元素作归一化处理；最后用 wavwrite 函数还原成音量标准化的 WAV 文件。

例 1：现以微软自带的“Windows XP 关机.wav”音频信号为例，先将其复制另存到文件名为 XPexit.wav 的 MATLAB 当前目录中，再通过音量标准化处理后保存为 XPquit.wav 文件。实现程序如下：

```
clear; close all; clc;

[Y,FS,NBITS]=wavread('XPexit.WAV');% 将 WAV 文件转换成变量
FS,NBITS,% 显示采样频率和量化比特
```

```
Ym=max(max(max(Y)),max(abs(min(Y))));% 找出双声道极值  
X=Y/Ym;% 归一化处理  
wavwrite(X,FS,NBITS,'XPquit.wav')% 将变量转换成 WAV 文件
```

试听可知标准化处理后音量稍大。

4 、声道分离合并与组合

立体声或双声道音频信号有左右两个声道,利用 MATLAB 实现双声道分离、两路声道合并和两个单声道组合成一个双声道等效果,实际上是利用了 MATLAB 的矩阵抽取、矩阵相加和矩阵重组运算。

例 2 : 现以例 1 生成的 XPquit.wav 为例,实现分离、合并和组合处理的程序如下:

```
clear; close all; clc;  
[x,FS,NBITS]=wavread('XPquit.WAV');% 将 WAV 文件转换成变量  
x1=x(:,1);% 抽取第 1 声道  
x2=x(:,2);% 抽取第 2 声道  
wavwrite(x1,FS,NBITS,'XPquit1.WAV');% 实现 1 声道分离  
wavwrite(x2,FS,NBITS,'XPquit2.WAV');% 实现 2 声道分离  
%如果合并位置不对前面补 0 %声道长度不对后面补 0  
x12=x1+x2;% 两路单声道列向量矩阵变量合并  
x12m=max(max(x12),abs(min(x12))),% 找出极值  
y12=x12./x12m;% 归一化处理  
wavwrite(y12,FS,NBITS,'XPquit12.WAV');% 实现两路声道合并  
%如果组合位置不对前面补 0--声道长度不对后面补 0  
x3=[x1,x2];% 两路单声道变量组合
```

```
wavwrite(x3,FS,NBITS,'XPquit3.WAV');% 实现两路声道组合
```

可以试听声道分离、合并与组合的效果，也可对各文件大小进行比较。

5、数字滤波

数字滤波是常用的音频处理技术。可根据技术指标，先利用 FDATool 工具，设计一个数字滤波器[2]，再用 Filter 或 Filter2 函数即可实现滤波处理。调用的 Filter 函数格式是： $Y = \text{filter}(B,A,X)$ 。其中，B 和 A 是滤波器传输函数的分子和分母系数，X 是输入变量，Y 是实现滤波后的输出变量。如果处理立体声音频信号，可分开处理，但用 FIR 滤波器时调用 Filter2 函数更方便。

例 3：现以例 2 生成的 XPquit12.wav 为例，实现数字滤波的程序如下：

```
clear; close all; clc;

[X,FS,NBITS]=wavread('XPquit12.WAV');% 将 WAV 文件转换成变量
%利用 FDATool 设计一个 LowpassButterworth 滤波器
%指标 FS=22050Hz Fp=1000Hz Ap=1dB Fs=3000Hz As=20dB
B=[0.0062,0.0187,0.0187,0.0062];% 分子系数
A=[1,-2.1706,1.6517,-0.4312];% 分母系数
Y=filter(B,A,X);% 实现数字滤波
t=(0:length(X)-1)/FS;% 计算数据时刻
subplot(2,2,1);plot(t,X);% 绘制原波形图
title('原信号波形图');% 加标题
subplot(2,2,3);plot(t,Y);% 绘制滤波波形图
title('滤波后波形图');% 加标题
xf=fft(X);% 作傅里叶变换求原频谱
yf=fft(Y);% 作傅里叶变换求滤波后频谱
```

```

fm=3000*length(xf)/FS;% 确定绘频谱图的上限频率

f=(0:fm)*FS/length(xf);% 确定绘频谱图的频率刻度

subplot(2,2,2);plot(f,abs(xf(1:length(f))));% 绘制原波形频谱图

title('原信号频谱图');% 加标题

subplot(2,2,4);plot(f,abs(yf(1:length(f))));% 绘制滤波后频谱图

title('滤波后信号频谱图');% 加标题

wavwrite(Y,FS,NBITS,'XPquitFilter.WAV');% 写成 WAV 文件

```

程序运行结果如图 1 所示。由图可知，滤波对波形影响不大，但对高频有较大衰减。试听会感觉到处理后的声音比较沉闷。

6、数据转换

数据转换是指改变音频格式中的采样频率或量化位数。转换原理是：先用矩阵插值或抽取技术实现变量变换，如果是抽取数据还需在变换前作滤波处理使之满足采样定理；变量变换完成后再用 Wavwrite 函数重新定义量化位数和采样频率即可实现数据转换。数据转换过程中，要注意采样频率与原始采样频率及插值或抽取系数的关系。MATLAB 实现插值或抽取的函数有 decimate、interp 和 resample，具体应用可参考有关文献[3]。这里以 2 倍抽取为例，将例 3 中经过滤波后产生的 XPquitFilter.WAV 文件进行数据转换处理。具体程序如下：

```

clear; close all; clc;

[x,FS,NBITS]=wavread('XPquitFilter.WAV');% 将 WAV 文件转换成变量

N=length(x);% 计算数据点数

% 不是偶数点化成偶数点

```

```

if mod(N,2)==0; N=N; else x(N)=[]; N=N-1; end;

% 原信号波形频谱分析

tx=(0:N-1)/FS;% 计算原信号数据点时刻

subplot(3,2,1);plot(tx,x);% 绘制原信号波形

title('原信号波形图');% 加标题

xf=fft(x);% 求原信号频谱

fx=(0:N/2)*FS/N;% 确定频谱图频率刻度

subplot(3,2,2);plot(fx,abs(xf(1:N/2+1)));% 绘制原信号频谱

title('原信号频谱图');% 加标题

% 实现数据抽取

k=[1:N/2];% 确定抽取位置

y=x(2*k);% 实现抽取后的数据

M=length(y);% 计算抽取后数据点数

% 抽取数据在原采样频率 FS 下的波形频谱分析

ty=(0:M-1)/FS;% 计算数据点时刻

subplot(3,2,3);plot(ty,y);% 绘制信号波形图

title('原采样率下新波形图');% 加标题

yf=fft(y);% 求频谱

fy=(0:M/2)*FS/M;% 确定频谱图频率刻度

subplot(3,2,4);plot(fy,abs(yf(1:M/2+1)));% 绘制频谱图

title('原采样率下新频谱图');% 加标题

% 抽取数据在 FS/2 采样频率下的波形频谱分析

tz=(0:M-1)/(FS/2);% 计算数据点时刻

subplot(3,2,5);plot(tz,y);% 绘制信号波形图

title('新采样率下新波形图');% 加标题

fz=(0:M/2)*(FS/2)/M;% 确定频谱图频率刻度

```

```

subplot(3,2,6);plot(fz,abs(yf(1:M/2+1)));% 绘制频谱图

title('新采样率下新频谱图');% 加标题

% 实现数据转换

wavwrite(y,FS/2,NBITS,'XPquit16B.WAV');% 音频格式 PCM 11025
Hz 16 位

wavwrite(y,FS/2,NBITS/2,'XPquit8B.WAV');% 音频格式 PCM 1102
5Hz 8 位

```

运行程序，在得到的图形窗口中，执行 Edit/Axes Properties...命令，再把各分图下 X 标签中的 Limits 设为 0、0.01 和 0、1000，得到 0—0.01 秒的波形和 0—1000Hz 的频谱如图 2 所示。由图可知，在满足采样定律条件下，实现数据抽取，在原采样率下波形变密、频谱变宽且幅度减半，但在新采样率下波形和频谱都很好。通过试听输出文件还可感受处理效果。

7、结束语

MATLAB 提供了许多专用工具箱，灵活利用这些工具箱和函数，可以实现很多信号处理任务。同时，MATLAB 还支持用户对其函数进行二次开发，以满足不同要求。在信号处理过程中，MATLAB 兼顾了专用工具软件的简单性和计算机程序语言的灵活性，特别是处理效果的可视性和可感知，有利于理解信号处理的本质，有利于激发学习和研究兴趣，也有利于培养 MATLAB 软件的操作技能。