

# 基于 Matlab 的声音信号采集与分析处理

陈家焱 陈冬娇 张达响

( 东华理工学院机电系 江西 南昌 330013 )

**摘要** :介绍了利用 Matlab 软件采集与处理声音信号的方法 ,并结合蛋壳破损声音检测的应用实例 ,深入地分析了声音信号采集与处理的 Matlab 实现。

**关键词** :声音信号采集 ;Matlab ;频谱分析

**中图分类号** :TP391.42

**文献标识码** :A

## Collecting and Processing of Sound Signal with Matlab

CHEN Jia-yan ,CHEN Dong-jiao ,ZHANG Da-xiang

( Mechanical & Electrical Engineering Department , East China Institute of Technology , Nanchang 330013 , China )

**Abstract** :The method of collecting and processing of sound signal with Matlab software is introduced in this paper , then , a case of using sound signal to detect crack eggs is combined for deeply analyzing its application .

**Key words** :collecting of sound signal ; Matlab ; analyzing of frequency

## 0 引 言

声音信号的采集与分析处理在工程应用中是经常需要解决的问题 ,如何实时采集声音信号并对其分析处理 ,找出声音信号的特征在科学研究中是一项非常有意义的工作。

Matlab 是一种功能强大、效率高、交互性好的数值计算和可视化计算机高级语言 ,它将数值分析、信号处理和图形显示有机地融合为一体 ,形成了一个极其方便、用户界面友好的操作环境。Matlab 的不断发展 ,其功能越来越强大 ,使其广泛应用于数字信号处理、数值图像处理、仿真、自动控制、小波分析和神经网络等领域。本文实现声音信号的采集与分析处理是基于 Matlab 软件完成的。

## 1 声音信号的采集

Matlab 函数库中的 `analoginput()` 函数可实现对模拟信号的采集 ,这种模拟信号可以是声音信号、传感器信号等。通过定义 `ai = analoginput( 'ADAPTOR' )` ,可建立模拟信号采集的对象 ,其中 ADAPTOR 表示设备名称 ,若是采集声音信号 ,则设置 `ADAPTOR = 'win-`

`sound'` ,即建立了声音信号采集的对象。

对建立的声音信号采集对象要进行声音采集的属性参数设置。`Addchanne(ai , 1)` 设置采集声音的通道数 ,1 表示单声道 [ 1 2 ] 则表示立体声道 ,同时还可对声道进行更深层次设置 ,如设置每个声道的名称、每个声道的输入信号值期望范围和 A/D 转换的电压范围等 ;`set(ai , 'SampleRate' , value)` 设置声音信号采集的采样频率 ,value 的取值根据实际情况进行确定 ,但必须满足采样定理 ,通常的取值有 `Value = 8000、16000、44100` 等 ;`set(ai , 'SamplesPerTrigger' , value)` 设置采集声音信号的长度 ,可用秒作单位也可以用采集样本个数作单位 ;`set(ai , 'TriggerRepeat' , value)` 设置连续采集的次数 ,若 `value = 0` 表示只采集一次 ,`value = 1` 表示采集两次声音信号 ;`set(ai , 'TriggerDelay' , value)` 设置延长的时间长短 ,表示触发完成后再延长一段时间后采集声音信号 ,当然 value 也可以取负值 ,表示触发提前采集 ;`set(ai , 'TriggerType' , 'value')` 设置声音信号采集的触发方式 ,触发方式主要有三种 :立即触发、手工触发和软件触发 ,软件触发又可分为信号上升沿触发、下降沿触发等 ,需根据不同的实际情况选择合适的触发方式 ,`set(ai , 'TriggerConditionValue' , value)`

设置声音信号采集的触发临界值,只要满足触发条件,就可立即对声音信号进行采集;se(ai,'TimeOut',value)设置超时等待的时间长短。

执行 start(ai)函数打开声音信号采集的对象,一旦满足触发条件就立即采集声音信号,函数 getdata(ai)可持续进行采集声音信号数据,执行这个函数能够得到声音信号的时域特征值,delete(ai)函数删除声音信号采集的对象。依次执行上面的函数就能够实现对声音信号的采集。

## 2 声音信号的分析处理

在采集得到的声音信号中,有些声音信号是研究中不需要的,必须剔除,可通过滤波器对所采集的声音信号进行滤波,以提高研究的精确性与针对性。运用滤波器对输入的声音信号进行滤波,首先必须设计滤波器。滤波器的种类很多,一般分为模拟滤波器和数字滤波器两大类,根据滤波器的功能,可将滤波器分为低通滤波器(LPF)、高通滤波器(HPF)、带通滤波器(BPF)和带阻滤波器(BSF)。

在 Matlab 中可以设计各种各样的滤波器,这要根据所要处理的信号特征来选择合适的滤波器。设计的滤波器通常返回滤波器的分子系数相量和分母系数相量(按照降幂排列),利用这两个参数,就可以调用 Matlab 内的滤波函数,Matlab 中主要的滤波函数有 filter()、filtfilt()、fftfilter() 和 filter2()。例如  $y = \text{filter}(b, a, x)$  计算输入信号  $x$  的滤波输出,向量  $b$  和  $a$  分别是所采用的滤波器的分子系数向量和滤波器的分母系数向量。

这里得到的是声音信号是其时域特征,提取声音信号的时域特征量很多,以语音信号处理为例说明,在语言信号的时域处理方法中把短时平均幅度和能量、短时平均过零率、短时自相关函数和短时平均幅度差函数作为它的时域特征量,但通常分析声音信号是看它的频域特征,从它的频域特征图上能够提取出更多有效的特征量。因此,对声音信号进行频谱分析、研究其在频域内的某些特征与频率变化的关系更有实际意义,一般来说,功率谱分析方法可分为经典谱分析和现代谱分析。经典功率谱分析函数主要有 Periodogram()、Psd()、Csd()和 Pwelch()等,现代功率谱分析研究方法中主要有 Pyulear()、Arburg()、Pburg()、pmtm()和 pmusic()等,可根据不同的实际情况选择合适的分析函数。

## 3 应用实例

笔者在研究鸭蛋破损检测过程中,运用了 Matlab

软件编制程序实现实时采集敲击鸭蛋蛋壳的声音信号,并对敲击蛋的声音信号进行分析处理,提取出敲击蛋壳声音信号的特征变量,利用得到的特征变量再作进一步分析处理,可实现将其中的损壳蛋剔除。下面是声音采集与分析处理程序。

```
ai = analoginpu('winsound');
c1 = addchannel(ai,1);
se(ai,'SampleRate',22050);
--- ;//设置采集声音的属性参数
start(ai)
try
clear data,time;
[ data,time]= getdata(ai);
catch
time=0;data=0;
disp('A timeout occured');
end
figure
plo(time,data,'b');
title('时域图');
xlab('Sample(64)');
ylab('幅值');
grid
--- ;%滤波处理
--- ;%功率谱分析处理
figure
plo(s,'b');
title('频域图');
xlab('频率值(HZ)');
ylab('功率谱值');
grid;
stop(ai);
delete(ai);
```

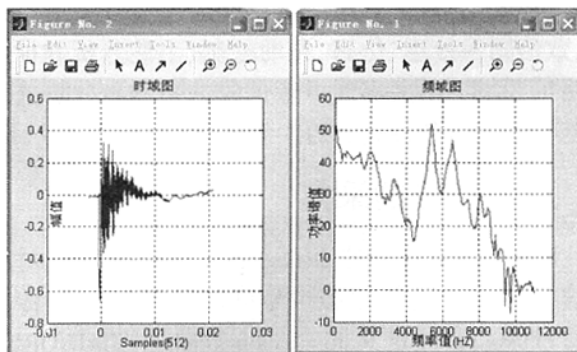


图 1 声音信号的时域图

图 2 声音信号的频域图

程序主要由三部分组成,声音采集的初始化部分、声音信号的处理部分(包括滤波与频谱分析)和处理结果的输出部分,图 1 是采集声音信号的时域图,图 2 是声音信号经过滤波后再进行频 (下转第 96 页)

```

memcpy(( DWORD * )PidMapAddress + 6 ,&stime ,sizeof( SYS-
TEMTIME ));
memcpy( &workpid , ( DWORD * ) PidMapAddress , sizeof
( DWORD ));
memcpy( &mtime ,( DWORD * )PidMapAddress + 1 ,sizeof
( SYSTEMTIME ));
if( !ReleaseMutex( PidFileMutex )) exit( -1 );
hMutex = OpenMutex( MUTEX_ALL_ACCESS , FALSE , "
workprocess" );
if( !hMutex ){CreateProcess( NULL ,"d:\ \ c + + \ \ de-
bug \ \ work. exe" , NULL , NULL , FALSE , CREATE_NEW_CON-
SOLE , NULL , NULL ,&si ,&pi );continue ;}
CloseHandle( hMutex );
if( mtime.wMinute == stime.wMinute + 1 ) mtime.wSecond =
mtime.wSecond + 60 ;
if( stime.wMinute == mtime.wMinute + 1 ) stime.wSecond =
stime.wSecond + 60 ;
if(( mtime.wSecond-stime.wSecond < 10 ) &&( mtime.wSec-
ond-stime.wSecond > -10 )) continue ;
if( !( hProcess = OpenProcess( PROCESS_ALL_ACCESS ,false ,
workpid))) exit( -1 );
TerminateProcess( hProcess ,0 );
CloseHandle( hProcess );
CreateProcess( NULL ,"d:\ \ c + + \ \ debug \ \ work.
exe" , NULL , NULL , FALSE , CREATE_NEW_CONSOLE , NULL ,
NULL ,&si ,&pi );continue ;
}
return 0 ;}

```

在这个程序中,首先通过建立互斥量“ watchpro-  
cess”标志了程序的运行,限制系统中监视程序实例的  
唯一性。其后打开内存映射文件“ Processinfo”记录进  
程的 PID 值和系统时间,利用互斥量“ Processinfo-

tex”保证了“对共享内存“ Processinfo”访问的同步性。  
在程序的主循环中每隔 8 秒做这些工作,利用互斥量  
“ workprocess”检查工作进程是否运行,如果没有运行,  
立刻启动工作进程;写系统当前时间到共享内存中,  
并检查共享内存中由工作进程写入的时间与计算机  
的当前时间之差是否在 10 秒之内,如果超出则判断  
工作进程出错,重启工作进程。

在工作进程的运行中,同样要建立互斥量  
“ workprocess”标志程序的运行,使用共享内存“ Pro-  
cessinfo”和互斥量“ Processinfo mutex”记录时间,检查监  
视进程是否正常运行。为了不影响工作进程的其它  
执行功能,可以另开一个线程专门做这些定期检查的  
工作,也可以使用系统定时器定时完成检查工作(这  
种方法仅限于 Windows Application 程序中)。这些代  
码与监视进程的相似,此处不再写出。

### 3 结束语

为了使常驻内存软件能够长期稳定地在系统中  
运行,必须采取一些特殊的措施监管软件运行。本文  
提出一种软件的方法保证常驻内存运行的稳定性。  
目前在电力能量管理系统软件中使用这种方法,取得  
了令人满意的结果。

#### 参考文献:

- [1] A L Williams. Windows 2000 系统编程[M]. 北京:机械工  
业出版社,2001.
- [2] Jeffrey Richter. Windows 核心编程[M]. 北京:机械工业出  
版社,2000.
- [3] 张红军,耿随心,陈春林. Visual C++ 编程详解[M]. 北  
京:科学出版社,2002.

(上接第 92 页)谱分析得到的频域图。

### 4 结束语

在利用 Matlab 软件实现对声音信号的采集与分  
析处理过程中,可得到以下一些结论(1)运用 Matlab  
软件进行声音信号的采集可加快研究的进程,避免了  
应用其它工具的繁琐性(2)Matlab 软件采集声音信  
号可根据需要灵活对属性参数进行修改,测试不同的  
试验条件来获得最佳的试验效果,在对声音信号进行  
功率谱处理时,也可以选择不同的功率谱分析函数来  
进行,并对不同的函数进行比较,从中分析得到有效  
的特征量。

利用 Matlab 编写程序实现声音信号的采集、设备

的控制、数据的分析以及结果的显示等功能,构建这  
样的数据采集分析系统具有精度高、实时性好、性价  
比高、人机界面好等优点。

#### 参考文献:

- [1] 陈亚勇. MATLAB 信号处理详解[M]. 北京:人民邮电出  
版社,2001.
- [2] 薛年喜. Matlab 在数字信号处理中的应用[M]. 北京:清  
华大学出版社,2003.
- [3] 周金萍. MATLAB 6 实践与提高[M]. 北京:中国电力出  
版社,2002.
- [4] 易克初. 语音信号处理[M]. 北京:国防工业出版社,  
2002.