百度文库

搜索



   基于matlab的语音信号的采集和处理

电子科技大学

课 程 设 计 报 告

课程名称： 信号与系统

设计名称： 语音信号的采集和处理

姓 名： 肖燕平

学 号: \*\*\*\*\*\*\*\*\*\*\*\*\*

班 级： 通信九班

\*\*\*\* \*\*\*\*\*\*\*\*\*

起止日期： 2012.12.15-2012.12.20

# 基 于 MATLAB 的 语音信号的采集和 处理

摘要：本文介绍了一种基于 matlab 的语音信号的采集和处理设计实现方案。 声音是由物体的振动产生，以声波的形式在介质中传播，介质主要可分为固 体， 液体以及气体。 声波振动内耳的听小骨， 这些振动被转化为微小的电子脑波， 它就是我们觉察到的声音。 内耳采用的原理与麦克风捕获声波或扬声器的发音一 样，它是移动的机械部分与气压波之间的关系。在国际标准中，人声的频率范围 是 300Hz~3400Hz ，不同的人或乐器产生的声音频率不一致，通过对声音信号的 研究能够更好的处理声音信号的处理以及传输。 Matlab 作为一款主要面对科学 计算、可视化以及交互式程序设计的高科技计算软件，能够很好的完成对声音信 号的分析和处理，快速的得出声音信号的时域图以及频域图。

关键字：声音 | 、 频率 、 时域图 、 频域图

1 绪 论

1.1 课题研究目的及意义

掌握语音信号采集的方法

掌握一种语音信号基音周期提取方法

了解 Matlab 的编程方法

1.2 本课题的设计要求 及设计 方案 概述

一：使用 wavrecord 录入自己的语音信号 ，使用 save 函数进行保存后使用 wavplay

函数进行 播放。

二：使用 plot 再 画出 该 语音信号的时域波形 ，对原始波形进行用 fft 函数傅里叶

变换后，使用 plot 画出其 频谱 。

三：设计切比雪夫的低通，高通，带通 滤波器 对原始信号进行滤波。

四： 画出滤波后的信号时域、频域图

五：考虑到国际标准人声的频率范围在 300Hz~3400Hz ，于是 给原始语音信号加入

3800hz 的正弦 高频噪声， 再 分析语音信号的特点 。

六：设计低通 计滤波器将高频噪声滤除 。

2 设计过程

2.1 本课题的设计要求

1 ，使用 wavrecord 录入自己的语音信号 ，保存到一个数组后，再使用 save 函数进行保存文件，后使用 wavplay 函数进行 播放。程序实现如下：

fs=8000; % 采样频率为 8000hz

y=wavrecord(3\*fs,fs); % 录入 3s 的声音

wavplay(y,fs); % 播放已录入的声音信号

file='voice.mat'; % 创建一个数据文件，且赋给变量 file

save('voice.mat','y'); % 将语音数据保存到数据文件 voice.mat 中

load voice.mat; % 读出 voicet.mat 的语音数据

2 ，使用 plot 再画出该语音信号的时域波形

程序代码为：

subplot(211); % 对图形窗口进行分割

plot(y);

xlabel('time n');

ylabel('amplitude');

title(' 原始语音信号 '); % 标出横纵坐标和图像名称

再对原始波形进行用 fft 函数傅里叶变换后得到向量，使用 plot 画出其频谱。 因为直接用 fft 得出的数据与频率不是对应的， 由于 fftshift 可以将 fft 的直流分量移到频谱中心，即 让正半轴部分和负半轴部分的图像分别关于各自的中心对称。 故再使用 fftshift 对 fft 进行变换。

程序代码为：

a=fftshift(fft(y,l)/l ) ; % 对语音信号进行离散时间傅里叶变换，且将 fft 的直流分量移 到频谱中心

fd=fs/l;

fx=fd\*(-l/2:l/2-1); % 将横轴变为频率轴

fy=abs(a); % 将纵轴变为频率幅度轴

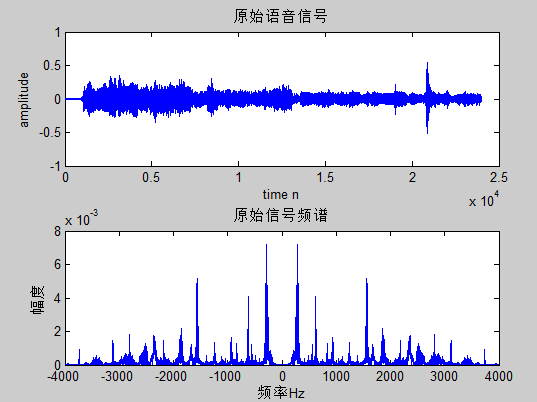
subplot(21 2 );

plot(fx,fy);

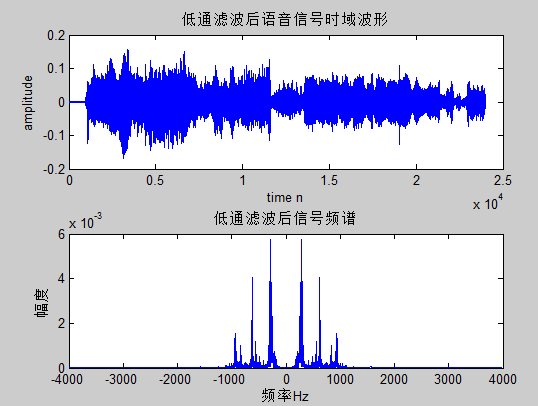
xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 原始信号频谱 ');

如图所示： 

3 ，设计切比雪夫 滤波器 对原始信号进行滤波时， 利用 cheb1ord 计算滤波器阶数： [N, Wn] = cheb1ord (Wp, Ws, Rp, Rs) ；利用 cheby1 计算滤波器系数： [b,a] = cheby1 (N,Wn) 。 其中设计了通带截止频率为 1000hz, 阻带截止频率为 1200hz 的低通滤波器，如图所示



程序为：

fph=1000;

fsh=1200;

rp=2;

rs=15;

omegaph=fph/(fs/2); % 归一化

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'low');

yl=filter(BZ,AZ,y);

subplot(211);

plot(yl);

xlabel('time n');

ylabel('amplitude');

title(' 低通滤波后语音信号时域波形 ');

al=fftshift(fft(yl,l))/l; % 对低通滤波后信号进行傅里叶变换

fdl=fs/l;

fxl=fdl\*(-l/2:l/2-1);

fyl=abs(al);

subplot(212);

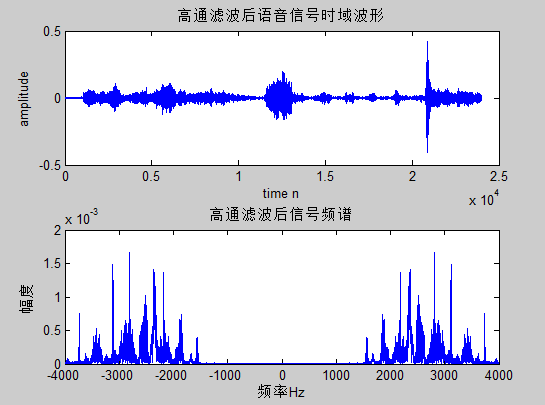
plot(fxl,fyl);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 低通滤波后信号频谱 ');

设计了通带截止频率为 2000hz, 阻带截止频率为 1600hz 的高通滤波器，如图所示



fph=2000;

fsh=1600;

rp=2;

rs=15;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'high'); % 计算切比雪夫高通滤波器系数

yh=filter(BZ,AZ,y);

subplot(211);

plot(yh);

xlabel('time n');

ylabel('amplitude');

title(' 高通滤波后语音信号时域波形 ');

ah=fftshift(fft(yh,l))/l;

fdh=fs/l;

fxh=fdh\*(-l/2:l/2-1);

fyh=abs(ah); subplot(212);

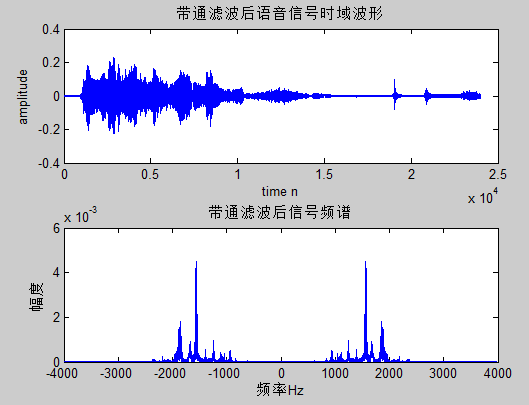
plot(fxh,fyh);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 高通滤波后信号频谱 ');

设计了截止频率 Wp ＝ [1000Hz 2000Hz] ； 阻带截止频率 Ws ＝ [800Hz 2500Hz] ； 的带 通滤波器，如图所示



程序为：

fph=[1000,2000];

fsh=[800,2500];

rp=3;

rs=20;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'bandpass');

yb=filter(BZ,AZ,y);

subplot(211);plot(yb);

xlabel('time n');

ylabel('amplitude');

title(' 带通滤波后语音信号时域波形 ');

ab=fftshift(fft(yb,l))/l;

fdb=fs/l;

fxb=fdb\*(-l/2:l/2-1);

fyb=abs(ab);

subplot(212);plot(fxb,fyb);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 带通滤波后信号频谱 ');

4 ，使用 wavplay 函数播放滤波前后的语音信号

程序为：

wavplay(y,fs); % 播放原始语音信号

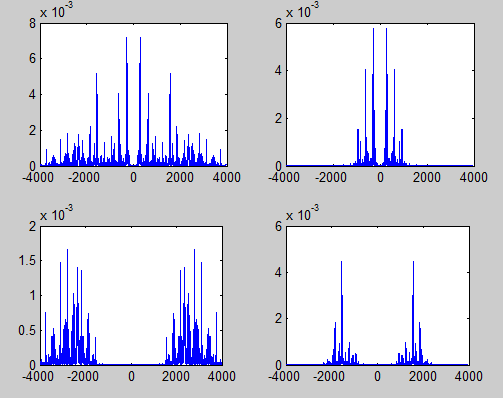
wavplay(yl,fs); % 播放低通滤波后语音信号

wavplay(yh,fs); % 播放高通滤波后语音信号

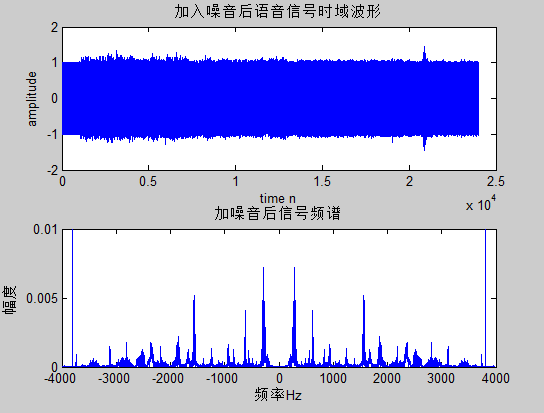
wavplay(yb,fs); % 播放带通滤波后语音信号

可以明显地听到经过低通后声音明显较先前低沉但音量减小，如下一、二图所示，低通滤波后信号高频部分明显大幅衰，即信号只剩余低音部分；经过高通滤波后声音较原始信号刺耳但音量减小，如下一、三图所示，高通滤波后信号低频部分明显大幅衰，剩下高频部分，即信号只剩余高音部分；经过带通滤波后声音较原始信号无明显区别但音量明显减小，如下一、四图所示，带通滤波后信号低频和高频部分明显大幅衰，剩下中频部分，即信号只剩余中音部分；

如图所示，分别为原始信号，低滤波后信号，高通滤波后信号，带通滤波后信号频谱



5 ，考虑到国际标准人声的频率范围在 300Hz~3400Hz ，于是 给原始语音信号加入 频率为 3800hz 的正弦 高频噪声 ，如图所示，为加入高频噪声后的信号时域波形和频谱图，由图可以看出，加入信号后的时域波形明显幅度加大，使用 wavplay 函数播放也可明显听到明显‘滴’声，频谱图中也可看到较原信号而言，加噪声后频谱在 3800hz 和 -3800hz 处增加了一高峰。



程序为：

Au=1;

t=0:1/fs:(l-1)/fs;

d=Au\*sin(2\*pi\*3800\*t); % 构造频率为 3800hz 的正弦信号，且取其一段

d=d';

yy=y+d; % 将所得噪音信号加入原始信号中

subplot(211);

plot(yy); % 绘制加入噪音后波形

xlabel('time n');

ylabel('amplitude');

title(' 加入噪音后语音信号时域波形 ');

ay=fftshift(fft(yy,l))/l;

fdy=fs/l;

fxy=fdy\*(-l/2:l/2-1);

fyy=abs(ay);

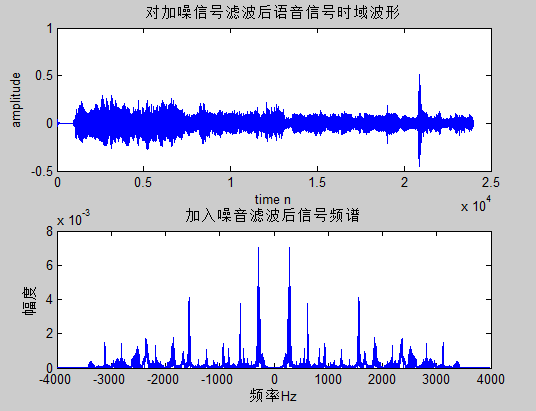
subplot(211);

plot(yy); % 绘制加入噪音后频谱

xlabel('time n');

ylabel('amplitude');

title(' 加入噪音后语音信号时域波形 ');

6 ，设计阻带截止频率为 3600 ，通带截止频率为 3400 的切比雪夫低通 计滤波器将高频噪声滤除 。所得信号时域波形及频谱图如下： 

程序为：

fph=3400;

fsh=3600;

rp=2;

rs=60;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'low');

yyy=filter(BZ,AZ,yy);

subplot(211);

plot(yyy);

xlabel('time n');

ylabel('amplitude');

title(' 对加噪信号滤波后语音信号时域波形 ');

ayy=fftshift(fft(yyy,l)/l);

fdyy=fs/l;

fxyy=fdyy\*(-l/2:l/2-1);

fyyy=abs(ayy);

subplot(212);plot(fxyy,fyyy);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 加入噪音滤波后信号频谱 ');

3 本课题的设计原理

1 、 采样定理： 在进行模拟与数字信号的转换过程中，当采样大于最高频率的 2 倍 时，则采样之后的数字信号完整的保留了原始信号中的信息，一般实际 应用中保证采样频率为信号最高频率的 5~10 倍。

2 、采样频率： 采样频率是指计算机每秒钟采样多少个声音样本， 是描述声音文件 的音质、音频、衡量声卡、声音文件的质量标准。采样频率越高，即采 样的时间间隔越短，则在单位时间内计算机得到的声音样本数据越多，

对声音波形的表示也越准确。

3 、 采样位数与采样频率： 采样位数即采样值或取样值 , 用来衡量声音波动变化的参数 , 是指 声卡在采集和播放声音文件时所使用数字声音信号的二进制位数。 采样频率是指录音设备在一秒钟内对声音信号的采样次数 , 采样频率越高声 音的还原就越真实越自然。 采样位数和采样率对于音频接口来说是最为 重要的两个指标。无论采样频率如何 , 理论上来说采样的位数决定了音频数据最大的力度范围。采样位数越多则捕捉到的信号越精确。

4 、利用 MATLAB 对语音信号进行分析 和处理： 采集语音信号后，利用 MATLAB 软件平台进行频谱分析；并对所采集的语音信号加入干扰噪声，对加入噪声的信号进行频谱分析， 设计合适的滤波器滤除噪声，恢复原信号。

4 总结和心得体会

在为期一周的设计中 我遇到不少的问题 ，主要有以下两点：

1 ，刚开始时对 fft 函数的本质不是很了解，导致画出的波形有一半是人声无法发出的高频。后来经过复习信号与系统的课本，与同学讨论后才发现这部分看起来是高频的成分实际是 -w, 因为直接用 fft 得出的数据与频率不是对应的， 由于 fftshift 可以将 fft 的直流分量移到频谱中心，即 让正半轴部分和负半轴部分的图像分别关于各自的中心对称。 故再使用 fftshift 对 fft 进行变换。

2 , 加入噪声时发现直接加入频率为 5500hz 的正弦信号后所得信号频率不再是实际的 5500hz, 再次翻阅书本才回忆起采样的原则是 Ws>2Wm ，若采样频率为 fs=8000, 则 5000hz 频谱将会出现重叠现象，信号无法恢复，即所得噪音信号频率改变，考虑到国际标准人声的频率范围在 300Hz~3400Hz ，于是改为 给原始语音信号加入 频率为 3800hz 的正弦 高频噪声 。

通过这一个星期的课程设计， 同时 我学到了很多的东西 , 不仅巩固了我以前所 学过的知识 , 还让我学到很多在书本上所没有学到过的知识。 同时进一步加深了对语音信号的了解和熟练了对 Matlab 的使用 。 也深深地体会到作为工科女独有的骄傲， 因为以前都是基于课本上所学的理论知识， 很少将所学知识运用于实际， 而通过这次课程设计之后 更加感觉到只是会做题并不代表真正理解了课本的知识，必须要在实际中才能真正地理解 。 同时，这次设计也让我感受到了书本的知识也很重要，做任何一个工程，再小都需要足够的理论知识作为垫脚石。

5 参考文献

【 1 】 Alan.V.Oppenheim ， Signal and System ，电子工业出版社， 2009.

【 2 】 数学实验讲义，电子科技大学出版社 ， 2010.

【 3 】 刘树棠， 《信号与系统》计算机练习 —— 利用 MATLAB ，西安交通大学出版

社， 2006

【 4 】 谢云荪，数学实验，科学出版社， 1999.

【 5 】 苏金明， MATLAB 实用教程，电子工业出版社， 2005.

【 6 】 徐全智，概率论与数理统计，高等教育出版社 ,2004.

【 7 】 杨克昌，计算机程序设计典型例题精解，国防科技大学出版社

程序清单

fs=8000; % 采样频率为 8000hz

y=wavrecord(3\*fs,fs); % 录入 3s 的声音

wavplay(y,fs); % 播放已录入的声音信号

file='voice.mat'; % 创建一个数据文件，且赋给变量 file

save('voice.mat','y'); % 将语音数据保存到数据文件 voice.mat 中

load voice.mat; % 读出 voicet.mat 的语音数据

subplot(211); % 对图形窗口进行分割

plot(y);

xlabel('time n');

ylabel('amplitude');

title(' 原始语音信号 '); % 标出横纵坐标和图像名称

a=fftshift(fft(y,l)/l ) ; % 对语音信号进行离散时间傅里叶变换，且将 fft 的直流分量移到频谱中

fd=fs/l;

fx=fd\*(-l/2:l/2-1); % 将横轴变为频率轴

fy=abs(a); % 将纵轴变为频率幅度轴

subplot(21 2 );

plot(fx,fy); % 画出原始信号频谱图

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 原始信号频谱 ');

fph=1000; % 低通滤波程序开始

fsh=1200;

rp=2;

rs=15;

omegaph=fph/(fs/2); % 归一化

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs); % 计算滤波器阶数

[BZ,AZ]=cheby1(N,rp,wn,'low'); % 计算滤波器系数

yl=filter(BZ,AZ,y); % 对原始信号低通滤波

subplot(211);

plot(yl);

xlabel('time n');

ylabel('amplitude');

title(' 低通滤波后语音信号时域波形 ');

al=fftshift(fft(yl,l)/l); % 对低通滤波后信号进行傅里叶变换

fdl=fs/l;

fxl=fdl\*(-l/2:l/2-1);

fyl=abs(al);

subplot(212);

plot(fxl,fyl);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 低通滤波后信号频谱 ');

fph=2000; % 高通滤波程序开始

fsh=1600;

rp=2;

rs=15;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'high'); % 计算切比雪夫高通滤波器系数

yh=filter(BZ,AZ,y);

subplot(211);

plot(yh);

xlabel('time n');

ylabel('amplitude');

title(' 高通滤波后语音信号时域波形 ');

ah=fftshift(fft(yh,l)/l);

fdh=fs/l;

fxh=fdh\*(-l/2:l/2-1);

fyh=abs(ah); subplot(212);

plot(fxh,fyh);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 高通滤波后信号频谱 ');

fph=[1000,2000]; % 带通滤波程序开始

fsh=[800,2500];

rp=3;

rs=20;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'bandpass');

yb=filter(BZ,AZ,y);

subplot(211);plot(yb);

xlabel('time n');

ylabel('amplitude');

title(' 带通滤波后语音信号时域波形 ');

ab=fftshift(fft(yb,l)/l);

fdb=fs/l;

fxb=fdb\*(-l/2:l/2-1);

fyb=abs(ab);

subplot(212);plot(fxb,fyb);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 带通滤波后信号频谱 ');

wavplay(y,fs); % 播放原始语音信号

wavplay(yl,fs); % 播放低通滤波后语音信号

wavplay(yh,fs); % 播放高通滤波后语音信号

wavplay(yb,fs); % 播放带通滤波后语音信号

Au=1;

t=0:1/fs:(l-1)/fs;

d=Au\*sin(2\*pi\*3800\*t); % 构造频率为 3800hz 的正弦信号，且取其一段

d=d';

yy=y+d; % 将所得噪音信号加入原始信号中

subplot(211);

plot(yy); % 绘制加入噪音后波形

xlabel('time n');

ylabel('amplitude');

title(' 加入噪音后语音信号时域波形 ');

ay=fftshift(fft(yy,l)/l);

fdy=fs/l;

fxy=fdy\*(-l/2:l/2-1);

fyy=abs(ay);

subplot(211);

plot(yy); % 绘制加入噪音后频谱

xlabel('time n');

ylabel('amplitude');

title(' 加入噪音后语音信号时域波形 ');

fph=3400; % 对加入噪音后信号进行低通滤波，滤除噪音

fsh=3600;

rp=2;

rs=60;

omegaph=fph/(fs/2);

omegash=fsh/(fs/2);

[N,wn]=cheb1ord(omegaph,omegash,rp,rs);

[BZ,AZ]=cheby1(N,rp,wn,'low');

yyy=filter(BZ,AZ,yy);

subplot(211);

plot(yyy);

xlabel('time n');

ylabel('amplitude');

title(' 对加噪信号滤波后语音信号时域波形 ');

ayy=fftshift(fft(yyy,l)/l);

fdyy=fs/l;

fxyy=fdyy\*(-l/2:l/2-1);

fyyy=abs(ayy);

subplot(212);plot(fxyy,fyyy);

xlabel(' 频率 Hz');

ylabel(' 幅度 ');

title(' 加入噪音滤波后信号频谱 ');



版权说明：本文档由用户提供并上传，收益归属内容提供方，若内容存在侵权，请进行举报或认领

相关推荐

* 基于matlab的语音信号的采集和处理
* 基于matlab的语音信号的采集与处理
* (最新整理)MATLAB语音信号采集与处理(DOC)
* matlab语音信号采集与处理6,基于MATLAB的语音信号的采集与处理详解
* 基于MATLAB的语音信号的处理

猜你想看

* 基于MATLAB语言实现语音信号的采集与分析
* 课程设计基于MATLAB的语音信号录制采集和分析的程序设计
* 语音信号处理基础(三)——基于MATLAB的语音采集与读
* 基于MATLAB的语音信号采集和分析系统的可视化设计
* MATLAB课程设计---基于MATLAB的语音信号处理

相关好店

飞翔网络科技有限公司

「互联网」

在行创意

「互联网」

A小陈96

「互联网」

一品素材

「互联网」

吾爱网络项目

「互联网」

工具

收藏

领福利

下载文档

分享

领福利

举报

取消

分享

复制链接

QQ好友

新浪微博

QQ空间

可通过复制链接分享到微信等

收藏成功，可至个人中心查看

立即查看