

实验二 语音信号的常用处理方法

学号：2110688 姓名：史文天 专业：信息安全

实验要求

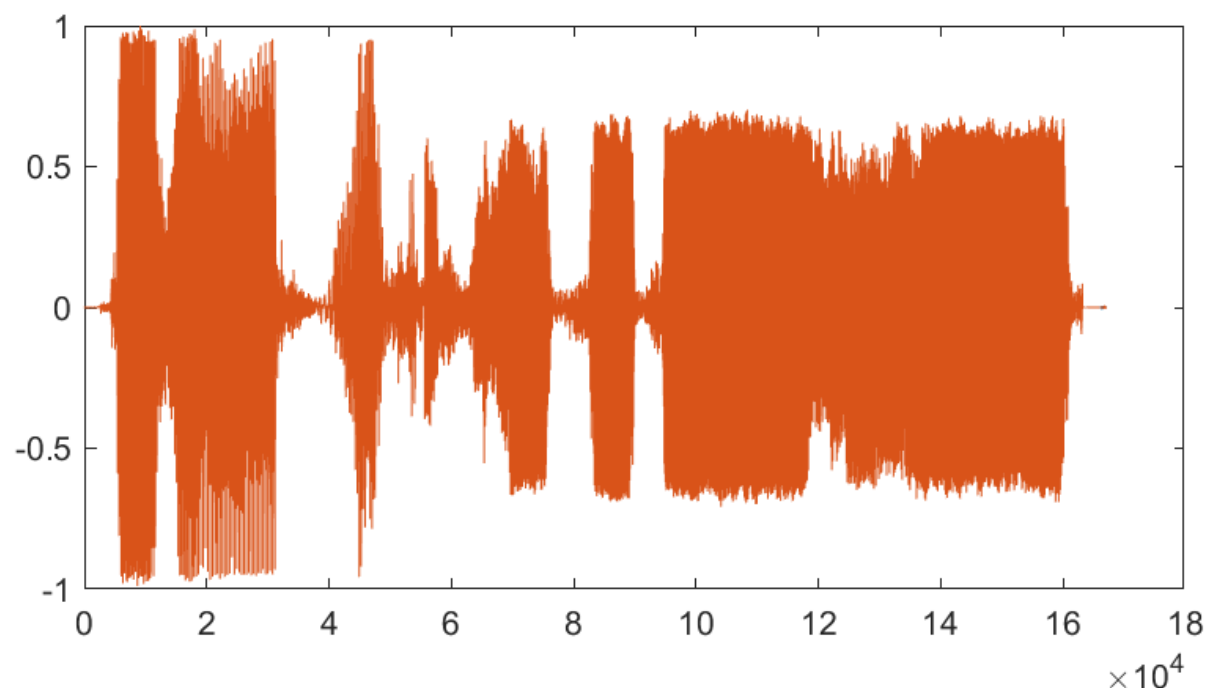
- 1、学习慕课：2.2语音信号处理基础
- 2、FFT
- 3、DWT
- 4、DCT

在matlab中调试完成课堂上的例题，练习使用常用的语音信号处理方法。

原语音信号图像

使用以下代码读取并展示语音信号的图像

```
[x, fs] = audioread('sound.wav');  
plot(x);
```



FFT（快速傅里叶变换）

介绍

1. $Y = \text{fft}(y)$;
2. $Y = \text{fft}(y, N)$;

式中， y 是序列， Y 是序列的快速傅里叶变换。 y 可以是一向量或矩阵，若 y 为向量，则 Y 是 y 的 FFT，并且与 y 具有相同的长度。若 y 为一矩阵，则 Y 是对矩阵的每一列向量进行 FFT。

说明

1. 函数fft 返回值的数据结构具有对称性

根据采样定理，fft 能分辨的最高频率为采样频率的一半（即Nyquist 频率），函数fft 返回值是以Nyquist 频率为轴对称的，Y 的前一半与后一半是复数共轭关系。

2. 幅值

作FFT 分析时，幅值大小与输入点数有关，要得到真实的幅值大小，只要将变换后的结果乘以2 除以N 即可（但此时零频—直流分量—的幅值为实际值的2 倍）。对此的解释是：Y 除以N 得到双边谱，再乘以2 得到单边谱（零频在双边谱中本没有被一分为二，而转化为单边谱过程中所有幅值均乘以2，所以零频被放大了）。

3. 基频

若分析数据时长为T，则分析结果的基频就是 $f_0=1/T$ ，分析结果的频率序列为 $[0:N-1]*f_0$

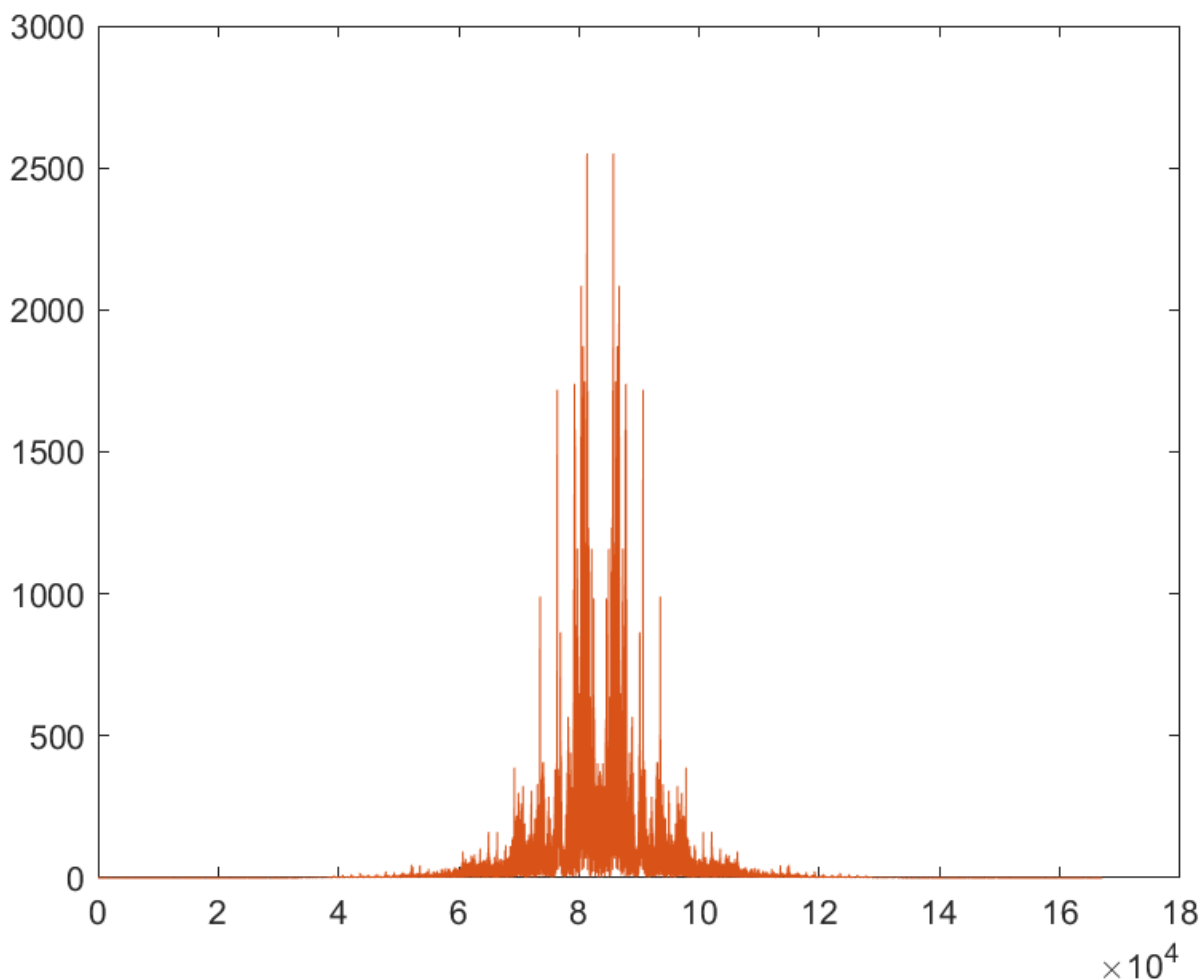
4. 执行N 点FFT

在调用格式2 中，函数执行N 点FFT。若y 为向量且长度小于N，则函数将y 补零至长度N，若向量y 的长度大于N，则函数截断y 使之长度为N。

实验结果

使用以下代码读取并对语音信号做FFT处理

```
[x,f]=audioread ('sound.wav');  
fx=fft(x);  
plot(abs(fftshift(fx)));
```



DWT（离散小波变换）

介绍

小波变换是20世纪80年代中后期逐渐发展起来的一种数学分析方法，他一出现就受到数学界和工程界的广泛重视。1984年法国科学家J.Molet在分析地震波的局部特性时，首先用小波变换对信号进行分析，并提出小波这一术语。

小波，小的波形，小是指其具有衰减性，波是指其具有波动性，即小波的振幅具有振幅正负相间的震荡形式。小波理论采用多分辨率思想，非均匀的划分时频空间，它使信号仍能在一组正交基上进行分解，为非平稳信号的分析提供了新途径。

小波就是在函数空间的一个满足条件的函数或者信号。小波分析能够对函数和信号进行任意指定点处的任意精细结构的分析，同时，这也决定了小波分析在对非平稳信号进行时频分析时，具有对时频同时局部化的能力。

连续小波的时频窗时时频平面上一个可变的矩形，他的时频窗的面积与小波的母函数有关，这一点决定了小波变换在信号的时频分析中的特殊作用。

特点

小波变换的时频关系受到不确定性原理的制约。还有恒Q性质，Q为母小波的品质因数。

$Q = \text{带宽} / \text{中心频率}$ 。

恒Q性质是小波变换的一个重要性质，也是小波变换区别于其他类型的变换，且被广泛应用的一个重要原因。当用较小的a对信号做高频分析时，实际上使用高频小波对信号做细致观察；而用较大的a对信号做低频分析时，实际上使用低频小波对信号做概貌观察。小波分析师傅里叶分析的发展和拓展，区别是

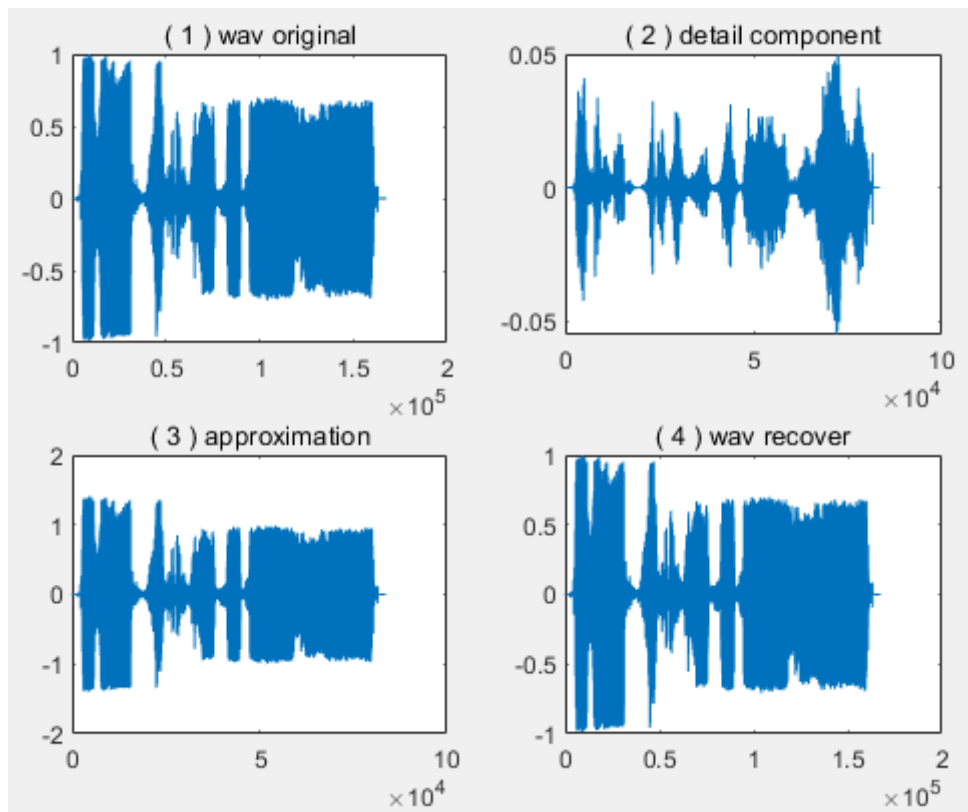
1. 傅里叶变换用到的基本函数具有唯一性，小波分析用到的函数具有不唯一性，同样一个问题用不同的小波函数进行分析，有事结果相差甚远。
2. 在频域中，傅里叶变换具有较好的局部化能力，特别是对于那些频率成分比较简单的确定信号，傅里叶变换可以很容易的把信号表示成各种频率成分叠加和的形式；但在时域中，傅里叶变换没有局部化能力，无法从信号的傅里叶变换中看出原信号在任一时间点附近的形态。
3. 若用信号通过滤波器来解释，小波变换与短时傅里叶变换的不同之处在于，对短时傅里叶变换来说，带通滤波器的带宽与中心频率无关；相反，小波变换带通滤波器的带宽则正比于中心频率，即小波变换对应的滤波器有一个恒定的相对带宽。

实验结果

一级小波分解（DWT）

这里小波基采用Daubechies-4小波，然后我们分别给出原始语音信号、一级分解的细节分量、一级分解的近似分量，和最后一级分解重构的结果。

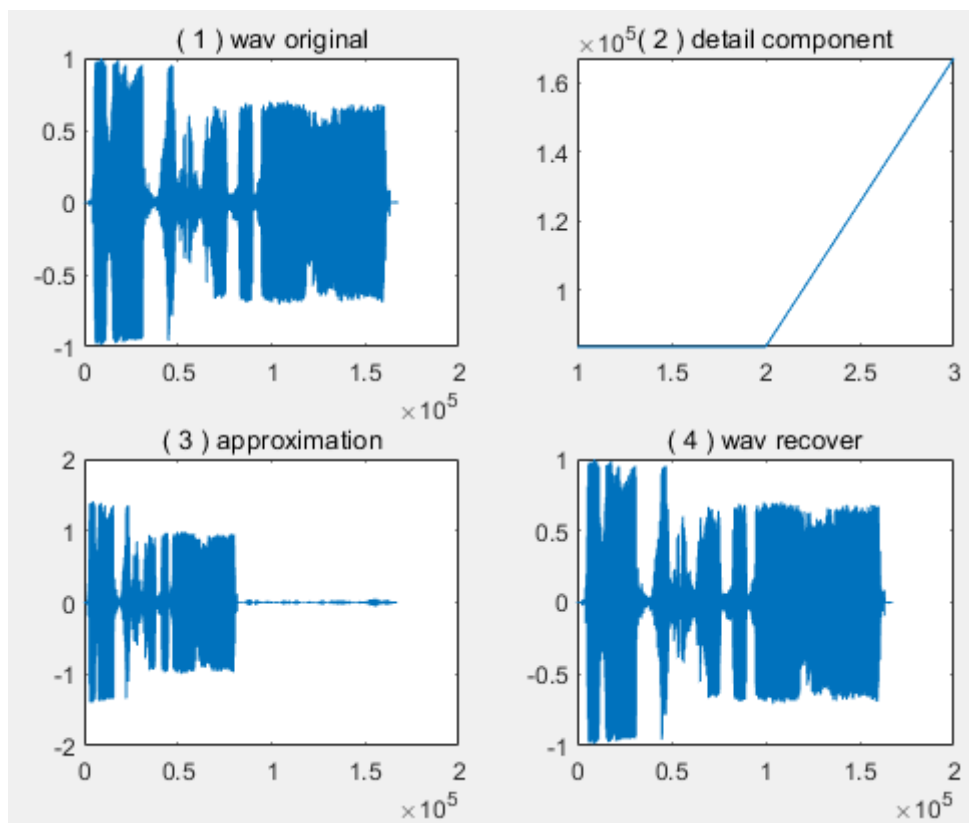
```
[a,fs]=audioread ("sound.wav") ;
[ca1,cd1]=dwt(a(:,1),'db4') ;
a0=idwt(ca1,cd1,'db4',length(a(:,1))) ;
%绘图
subplot (2,2,1) ; plot ( a ( : , 1)) ; %原始波形
subplot (2,2,2) ; plot ( cd1 ) ; %细节分量
subplot (2,2,3) ; plot ( ca1 ) ; %近似分量
subplot (2,2,4) ; plot ( a0 ) ; %一级分解的重构结果
axes_handle = get ( gcf,'children') ;
axes ( axes_handle (4) ) ; title('( 1 ) wav original') ;
axes ( axes_handle (3) ) ; title('( 2 ) detail component') ;
axes ( axes_handle (2) ) ; title('( 3 ) approximation') ;
axes ( axes_handle (1) ) ; title('( 4 ) wav recover') ;
```



一级小波分解 (wavedec)

这里小波基也采用Daubechies-4 小波，然后我们分别给出原始语音信号、一级分解的细节分量、一级分解的近似分量，和最后一级分解重构的结果。

```
[a,fs]=audioread ("sound.wav") ;
[ca1,cd1]=wavedec(a(:,1),1,'db4') ;
a0=waverec(ca1,cd1,'db4') ;
%绘图
subplot (2,2,1) ; plot ( a ( : , 1)) ; %原始波形
subplot (2,2,2) ; plot ( cd1 ) ; %细节分量
subplot (2,2,3) ; plot ( ca1 ) ; %近似分量
subplot (2,2,4) ; plot ( a0 ) ; %一级分解的重 构 结 果
axes_handle = get ( gcf,'children') ;
axes ( axes_handle (4) ) ; title('( 1 ) wav original') ;
axes ( axes_handle (3) ) ; title('( 2 ) detail component') ;
axes ( axes_handle (2) ) ; title('( 3 ) approximation') ;
axes ( axes_handle (1) ) ; title('( 4 ) wav recover') ;
```

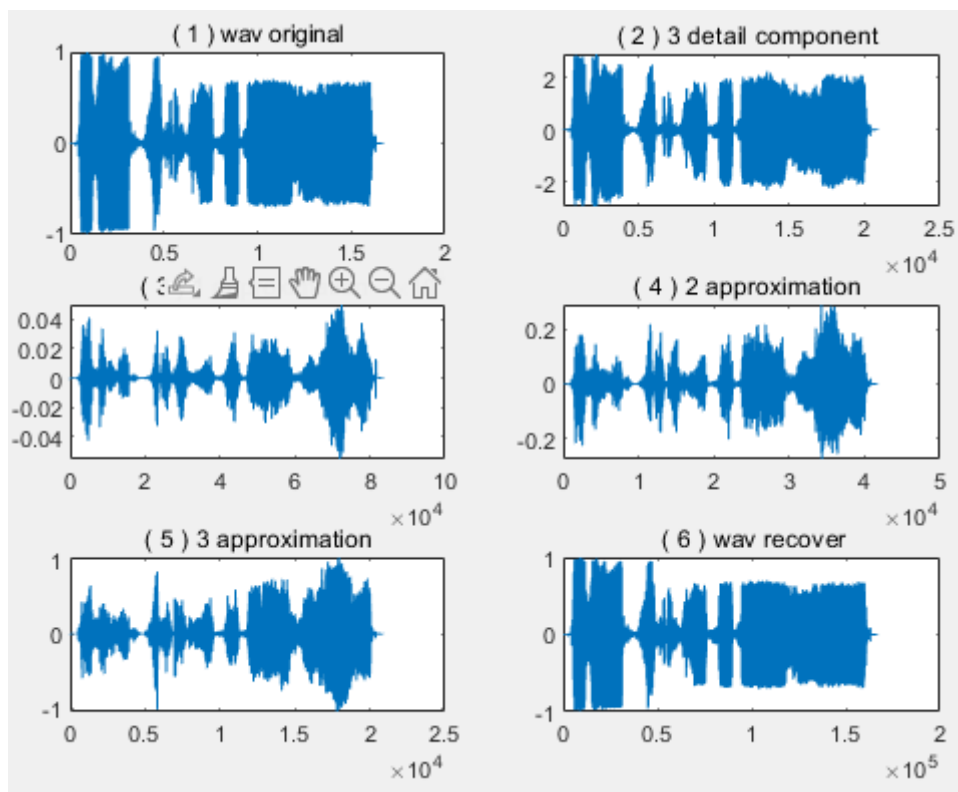


三级小波分解 (wavedec)

三级小波分解比之前更为复杂，小波基仍然采用Daubechies-4 小波，同时给出原始语音信号、三级分解的细节分量、一级分解的近似分量、二级分解的近似分量、三级分解的近似分量，最后是三级分解重构的结果。

```
[a,fs]=audioread('sound.wav') ;
[c,l]=wavedec(a(:,2),3,'db4') ;
ca3=appcoef(c,l,'db4',3) ;
cd3=detcoef(c,l,3) ;
cd2=detcoef(c,l,2) ;
cd1=detcoef(c,l,1) ;
a0=waverec(c,l,'db4') ;

subplot(3,2,1) ; plot(a(:,2)) ;
subplot(3,2,2) ; plot(ca3) ; % 三级分解近似分量
subplot(3,2,3) ; plot(cd1) ; % 一级分解细节分量
subplot(3,2,4) ; plot(cd2) ; % 二级分解细节分量
subplot(3,2,5) ; plot(cd3) ; % 三级分解细节分量
subplot(3,2,6) ; plot(a0) ; % 重构结果
axes_handle = get(gcf,'children') ;
axes(axes_handle(6)) ; title('(1) wav original') ;
axes(axes_handle(5)) ; title('(2) 3 detail component') ;
axes(axes_handle(4)) ; title('(3) 1 approximation') ;
axes(axes_handle(3)) ; title('(4) 2 approximation') ;
axes(axes_handle(2)) ; title('(5) 3 approximation') ;
axes(axes_handle(1)) ; title('(6) wav recover') ;
```



离散余弦变换

介绍

DCT 全称为Discrete Cosine Transform，即离散余弦变换。DCT 变换属于傅里叶变换的一种，常用于对信号和图像（包括图片和视频）进行有损数据压缩。

实验结果

```
[a,fs]=audioread('sound.wav') ;
dct_a=dct ( a ( : , 1 ) ) ;
a0=idct ( dct_a ) ;

subplot (3 ,1 ,1) ; plot ( a ( : , 1 ) ) ; %原始波形
subplot (3 ,1 ,2) ; plot ( dct_a ) ; %dct 处理后的波形
subplot (3 ,1 ,3) ; plot ( a0 ) ; %重构得到的结果

axes_handle = get ( gcf , 'children' ) ;
axes ( axes_handle (3) ) ; title ( '( 1 ) wav original' ) ;
axes ( axes_handle (2) ) ; title ( '( 2 ) wav dct' ) ;
axes ( axes_handle (1) ) ; title ( '( 3 ) wav recover' ) ;
```

