

信息隐藏技术实验报告

实验二：语音信号的常用处理方法

2113662 张丛

实验内容

快速傅里叶变换

原理

实验

离散小波变换

原理

实验

离散余弦变换

原理

实验

小结

实验内容

1. FFT(快速傅里叶变换)
2. DWT(离散小波变换)
3. DCT(离散余弦变换)

快速傅里叶变换

原理

FFT是一种算法，用于将信号从时域转换为频域。

它可以将信号分解为一系列正弦和余弦函数的频率成分，从而帮助我们理解信号的频率结构。

FFT在很多领域中都有广泛的应用，包括音频处理、图像处理和通信系统等。

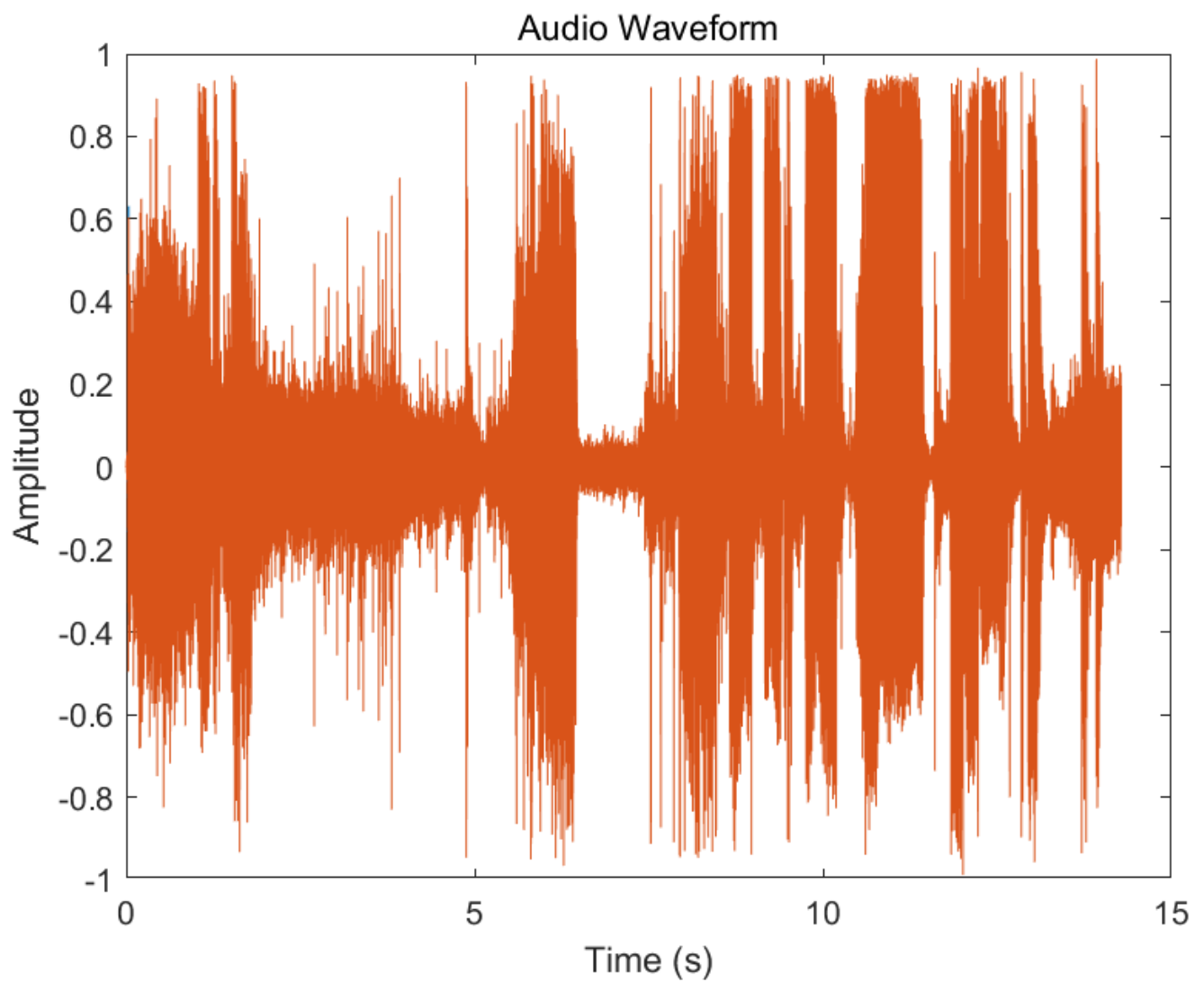
实验

查看音频波形：

```
%读取音频文件
[audioData, sampleRate] = audioread('sheep.wav');

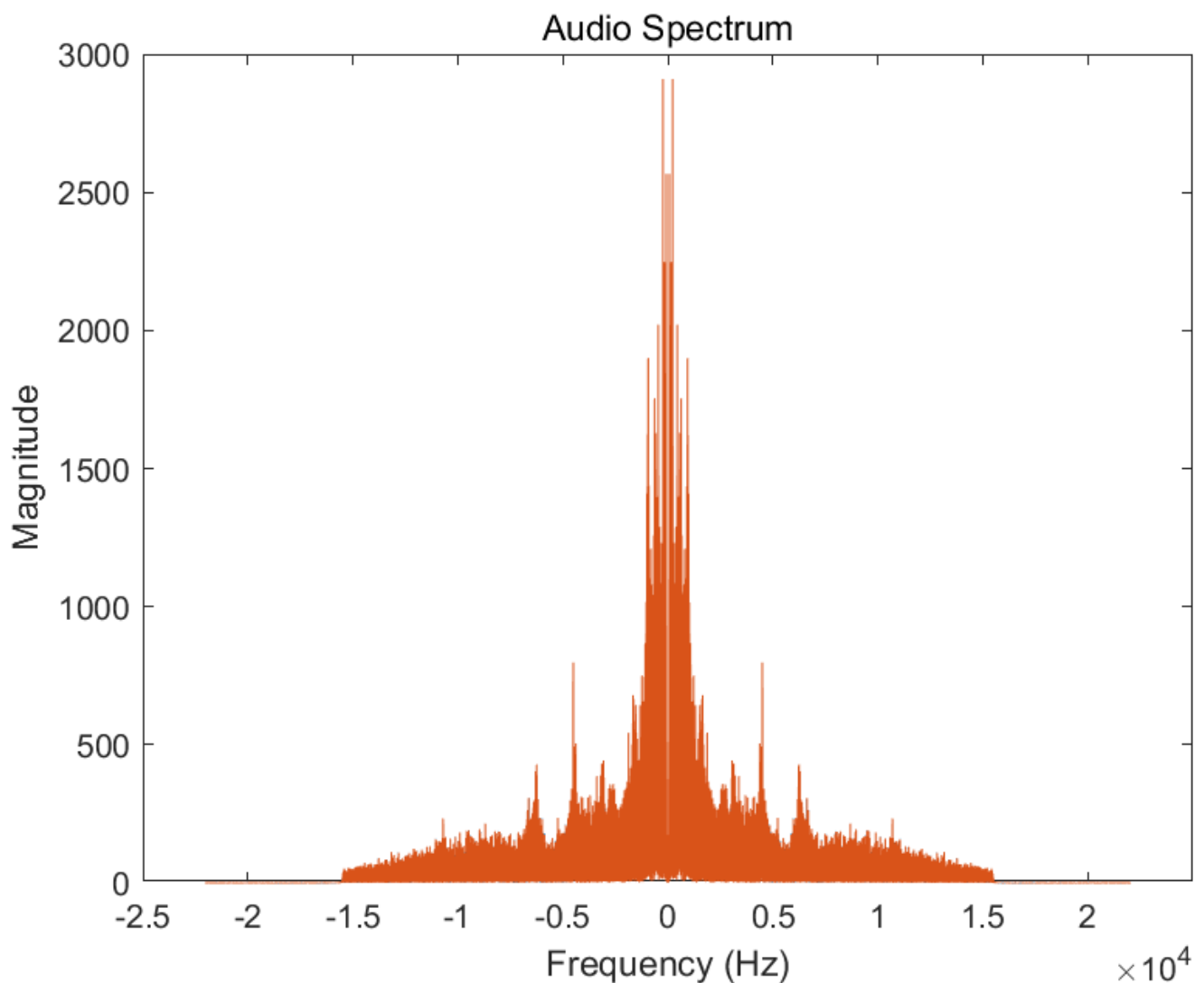
%播放音频
sound(audioData, sampleRate);

%波形
time = (0:length(audioData)-1) / sampleRate;
plot(time, audioData);
xlabel('Time (s)');
ylabel('Amplitude');
title('Audio waveform');
```



快速傅里叶变换 (FFT):

```
[x, f] = audioread('sheep.wav');  
N = length(x); % 信号的长度  
x = fft(x); % 进行傅里叶变换  
frequencies = linspace(-f/2, f/2, N); % 计算频率轴  
plot(frequencies, fftshift(abs(x))); % 绘制频谱  
xlabel('Frequency (Hz)');  
ylabel('Magnitude');  
title('Audio Spectrum');
```



其中：

`fft` 函数对音频数据 `x` 进行快速傅里叶变换（FFT），以将信号从时域转换为频域。

`fftshift` 函数将频谱数据移动，以使频谱的零频率位于频率轴的中心。

离散小波变换

原理

DWT是一种将信号分解为不同尺度的小波基函数的技术。

与FFT不同，DWT可以提供更好的时间和频率局部化特性，因此在某些应用中更受欢迎。它常用于信号压缩、噪声消除、特征提取等领域。

实验

一级小波分解：

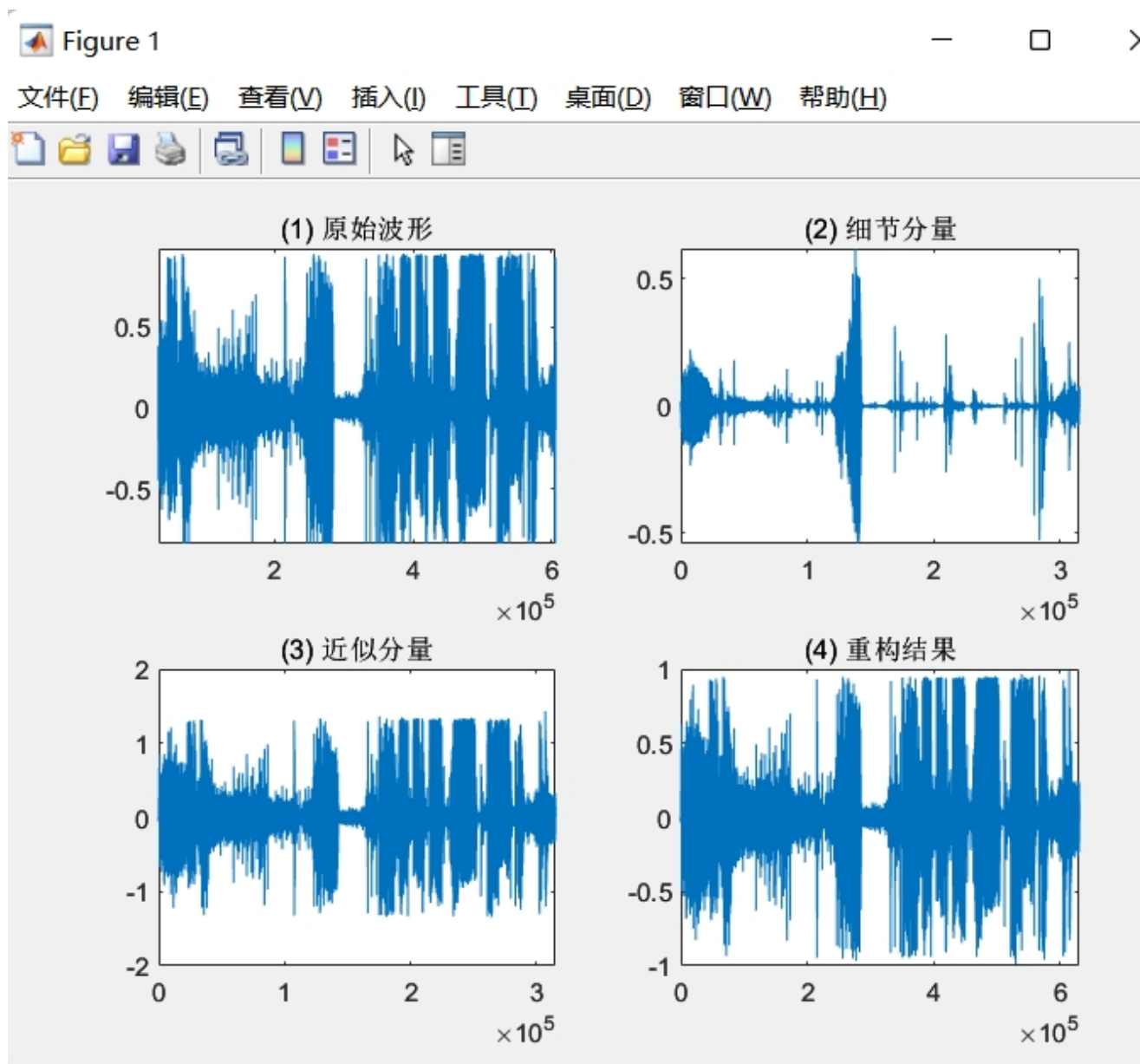
```
% 读取音频文件
[a, fs] = audioread("sheep.wav");

% 进行小波变换
[ca1, cd1] = dwt(a(:, 1), 'db4');
```

```
% 逆小波变换并重构原始信号
a0 = idwt(ca1, cd1, 'db4', length(a(:, 1)));
```

```
% 绘图
```

```
figure;
subplot(2, 2, 1); plot(a(:, 1)); title('(1) 原始波形');
subplot(2, 2, 2); plot(cd1); title('(2) 细节分量');
subplot(2, 2, 3); plot(ca1); title('(3) 近似分量');
subplot(2, 2, 4); plot(a0); title('(4) 重构结果');
```



二级小波分解:

```
% 读取音频文件
```

```
[a, fs] = audioread("sheep.wav");
```

```
% 进行二级小波分解
```

```
[C, L] = wavedec(a(:, 1), 2, 'db4');
```

```
% 提取二级分解的近似系数和细节系数
```

```
ca2 = appcoef(C, L, 'db4', 2);
```

```
cd2 = detcoef(C, L, 2);
```

```

ca1 = appcoef(C, L, 'db4', 1);
cd1 = detcoef(C, L, 1);

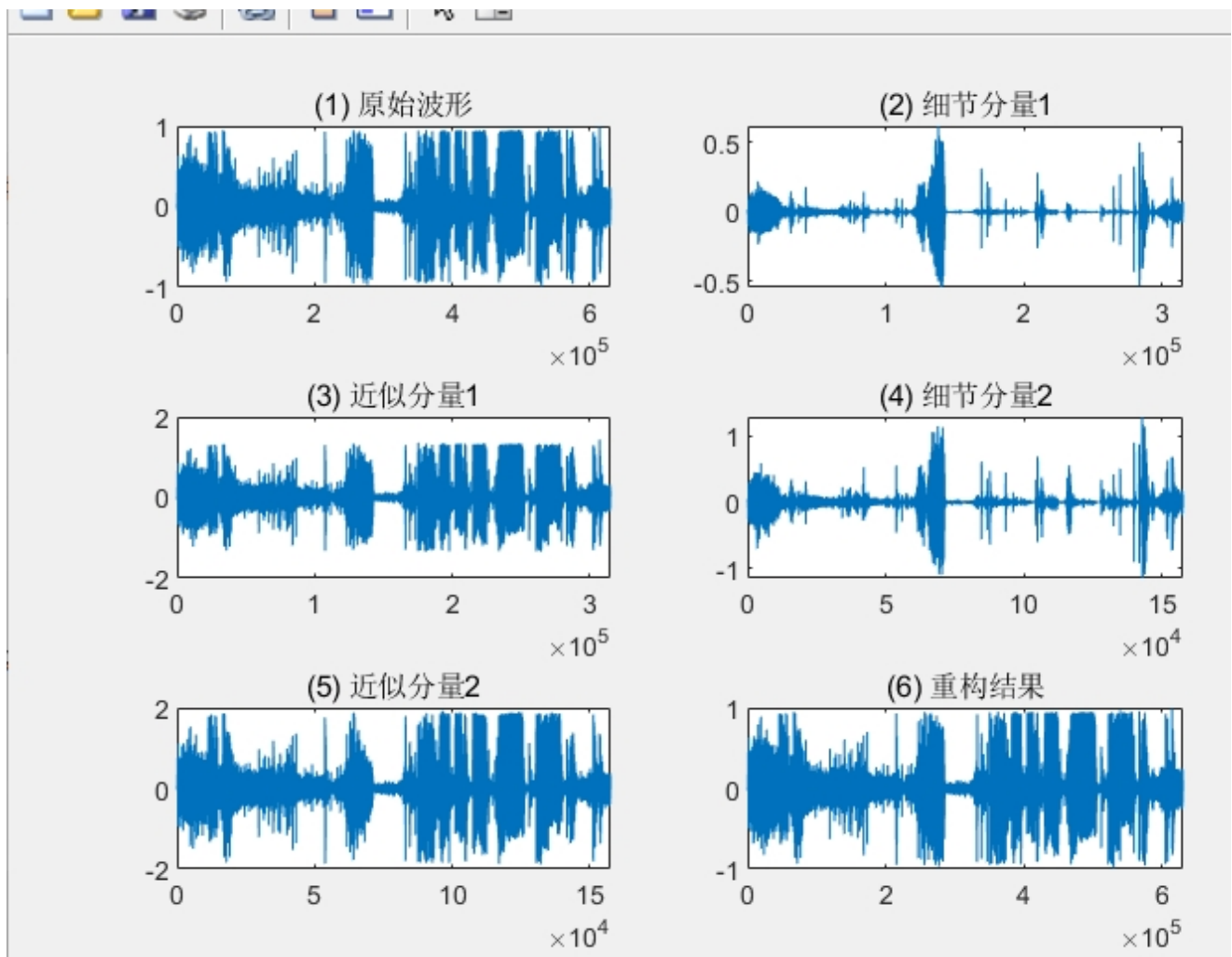
% 逆小波变换并重构原始信号
a0 = waverec(C, L, 'db4');

% 绘图
figure;

% 定义子图标题
titles = {'原始波形', '细节分量1', '近似分量1', '细节分量2', '近似分量2', '重构结果'};

for i = 1:6
    % 绘制子图
    subplot(3, 2, i);
    switch i
        case 1
            plot(a(:, 1)); % 原始波形
        case 2
            plot(cd1); % 细节分量1
        case 3
            plot(ca1); % 近似分量1
        case 4
            plot(cd2); % 细节分量2
        case 5
            plot(ca2); % 近似分量2
        case 6
            plot(a0); % 重构结果
    end
    title(['(', num2str(i), ') ', titles{i}]); % 添加标题
end

```



三级小波分解:

% 读取音频文件

```
[a, fs] = audioread('sheep.wav');
```

% 进行三级小波分解

```
[c, l] = wavedec(a(:, 2), 3, 'db4');
```

% 提取分解结果中的近似系数和细节系数

```
ca3 = appcoef(c, l, 'db4', 3);
```

```
cd3 = detcoef(c, l, 3);
```

```
cd2 = detcoef(c, l, 2);
```

```
cd1 = detcoef(c, l, 1);
```

% 逆小波变换并重构原始信号

```
a0 = waverec(c, l, 'db4');
```

% 绘图

```
figure;
```

% 定义子图标题

```
titles = {'原始信号', '三级分解近似分量', '一级分解细节分量', ...  
          '二级分解细节分量', '三级分解细节分量', '重构结果'};
```

```
for i = 1:6
```

```
    % 绘制子图
```

```
    subplot(3, 2, i);
```

```
    switch i
```

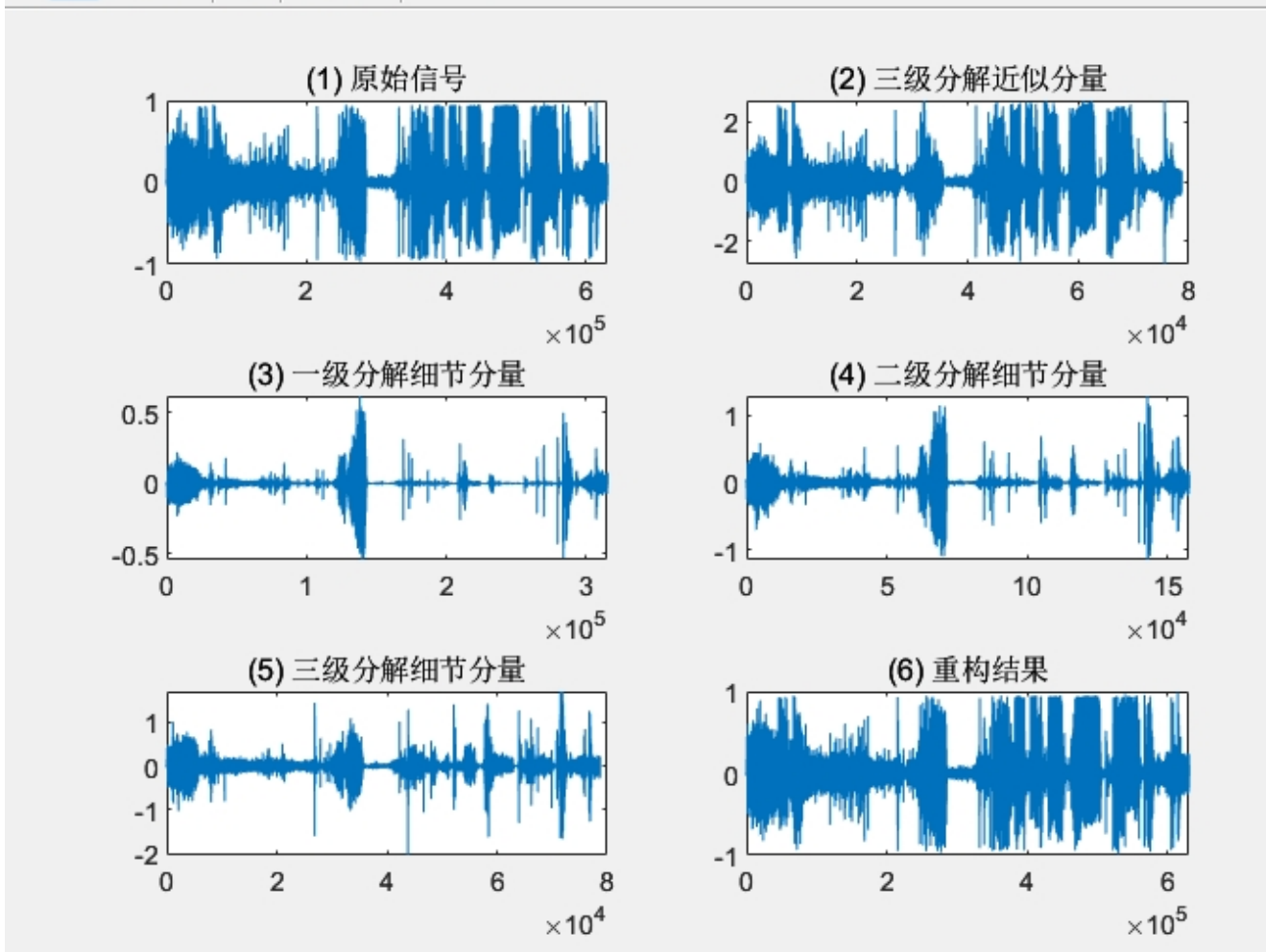
```
        case 1
```

```

        plot(a(:, 2)); % 原始信号
    case 2
        plot(ca3); % 三级分解近似分量
    case 3
        plot(cd1); % 一级分解细节分量
    case 4
        plot(cd2); % 二级分解细节分量
    case 5
        plot(cd3); % 三级分解细节分量
    case 6
        plot(a0); % 重构结果
end
title(['(', num2str(i), ') ', titles{i}]); % 添加标题
end

```

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)



离散余弦变换

原理

DCT是一种将信号分解为一系列余弦函数的技术。与FFT类似，它也用于将信号从时域转换为频域。

DCT在图像和视频压缩中特别常见，例如JPEG图像压缩和MPEG视频压缩，因为它能够提供更好的能量集中性，从而更有效地编码信号的信息。

实验

```
% 读取音频文件
[a, fs] = audioread('sheep.wav');

% 对第一列音频数据进行离散余弦变换 (DCT)
dct_a = dct(a(:, 1));

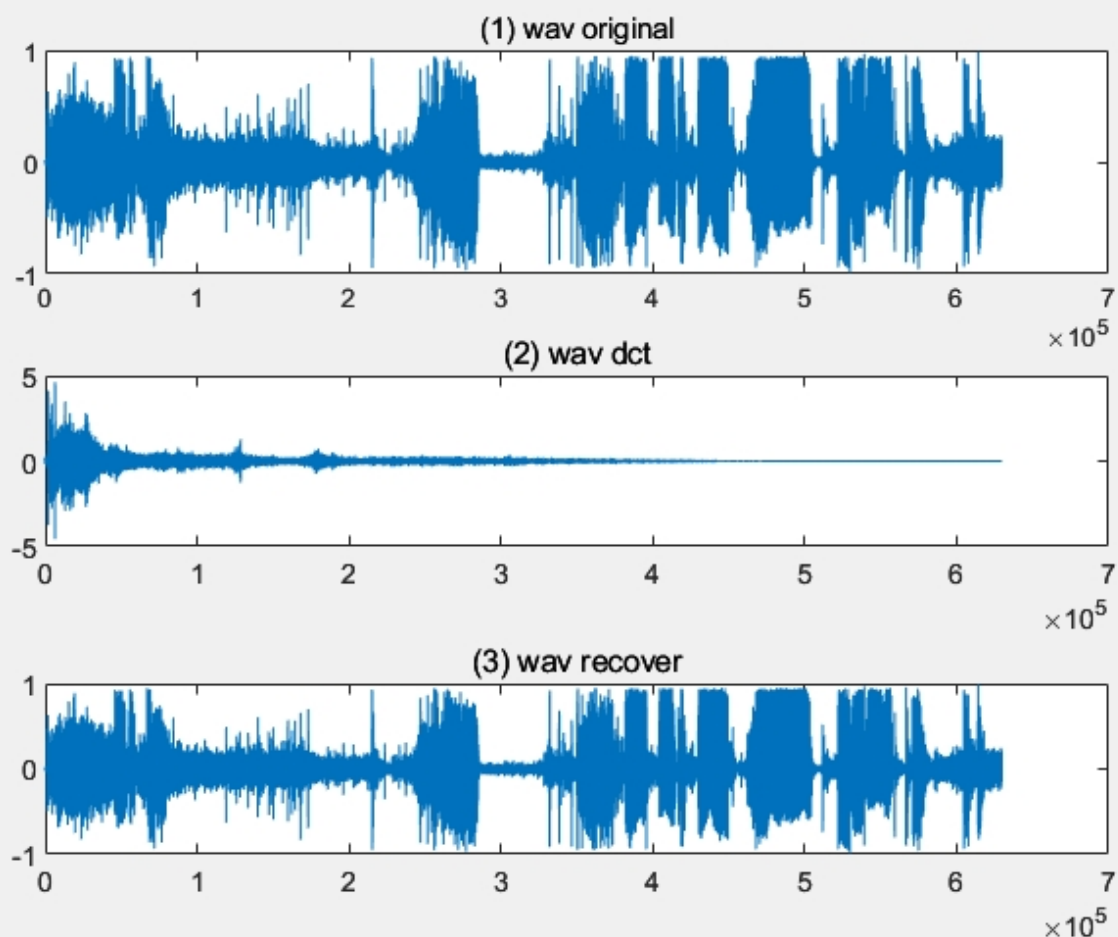
% 逆离散余弦变换 (IDCT) 以重构原始信号
a0 = idct(dct_a);

% 绘图
figure;

% 定义子图标题
titles = {'原始波形', 'DCT处理后的波形', '重构得到的结果'};

for i = 1:3
    % 绘制子图
    subplot(3, 1, i);
    switch i
        case 1
            plot(a(:, 1)); % 原始波形
        case 2
            plot(dct_a); % DCT处理后的波形
        case 3
            plot(a0); % 重构得到的结果
    end
    title(['(', num2str(i), ') ', titles{i}]); % 添加标题
end

% 设置子图标题
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');
```

小结

这次实验主要使用了语音信号的常用处理方法，包括傅立叶变换（FFT）、离散小波变换（DWT）和离散余弦变换（DCT）。

通过这些方法，可以从不同的角度对语音信号进行分析和处理，提取出不同的特征信息。这些方法在语音识别、音频处理等领域都有着重要的应用，对于理解 and 处理语音信号具有重要意义。

通过本次实验，我们对语音信号的常用处理方法有了更深入的了解，并且学会了如何使用 MATLAB 进行实际操作。