

# 信息隐藏技术第一次实验 —— 语音信号处理

学号：2111033

姓名：艾明旭

专业：信息安全

## 一、常用语音处理算法简介

### 1、傅立叶变换与短时傅立叶变换

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}$$

$$STFT_x(t, f) = \int [x(t')g^*(t' - t)]e^{-j2\pi ft'} dt'$$

$g^*(t)$  为窗函数

### 2、小波变换

$$CWT_x(\tau, a) = \frac{1}{\sqrt{|a|}} \int x(t)h^*\left(\frac{t-\tau}{a}\right)dt$$

$$x(t) = \frac{1}{C_H} \iint \frac{1}{a^2} CWT_x(\tau, a) \frac{1}{\sqrt{|a|}} h\left(\frac{t-\tau}{a}\right) da db$$

$h(t)$  为小波母函数

$$\frac{1}{\sqrt{|a|}} h\left(\frac{t-\tau}{a}\right) \text{ 为小波的平移与伸缩}$$

### 3、离散余弦变换

$$\text{正变换} \quad T(u) = \sum_{x=0}^{N-1} f(x)g(x,u) \quad u = 0,1,2,\dots,N-1$$

$$g(x,0) = \frac{1}{\sqrt{N}}$$

正变换核

$$g(x,u) = \sqrt{\frac{2}{N}} \cos \frac{(2x+1)u\pi}{2N}$$

$$\text{DCT变换的系数} \quad C(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x)$$

$$C(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N} \\ u = 1,2,\dots,N-1$$

$$\text{反变换} \quad f(x) = \frac{1}{\sqrt{N}} C(0) + \sqrt{\frac{2}{N}} \sum_{u=1}^{N-1} C(u) \cos \frac{(2x+1)u\pi}{2N}$$

$$x = 0,1,\dots,N-1$$

## 二、代码实现与解释

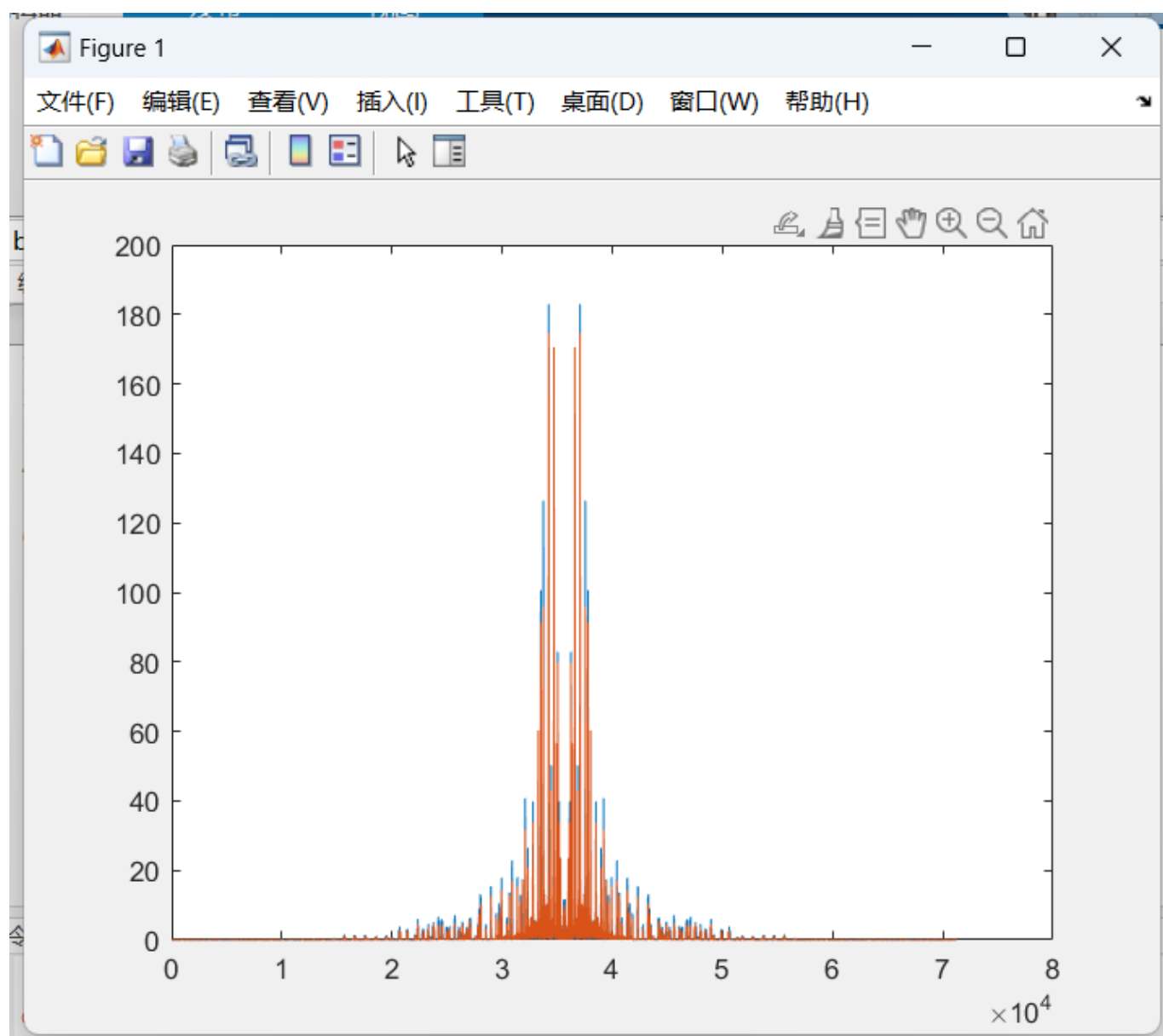
在本次实验中所有的代码均用MATLAB来实现，具体代码如下所示：

### 1、快速离散傅里叶变换

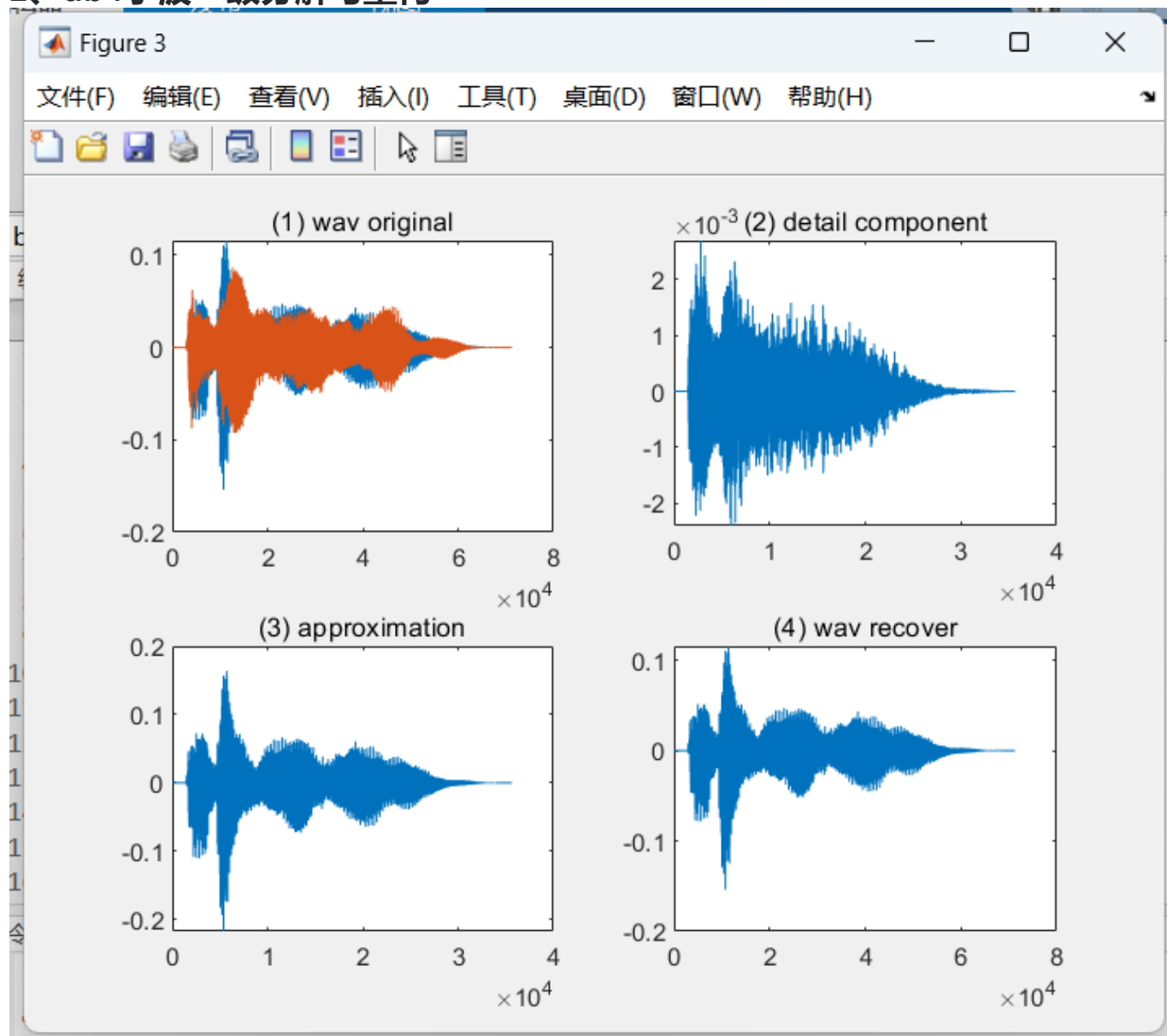
```
wav_name = 'tada.wav';
wav = audioread(wav_name);
% plot(wav);
f_wav = fft(wav);
plot(abs(fftshift(f_wav)));
% plot(fftshift(f_wav));
```

利用 `fft()` 函数进行快速傅里叶变换，并利用 `plot()` 函数将其展示

结果如图：



## 2、db4小波一级分解与重构



```

wav_name = 'tada.wav';
[wav,fs] = audioread(wav_name); % 获得语音信号
len = length(wav); % 语音信号大小
wav_vec = zeros(len,1); % 预分配内存

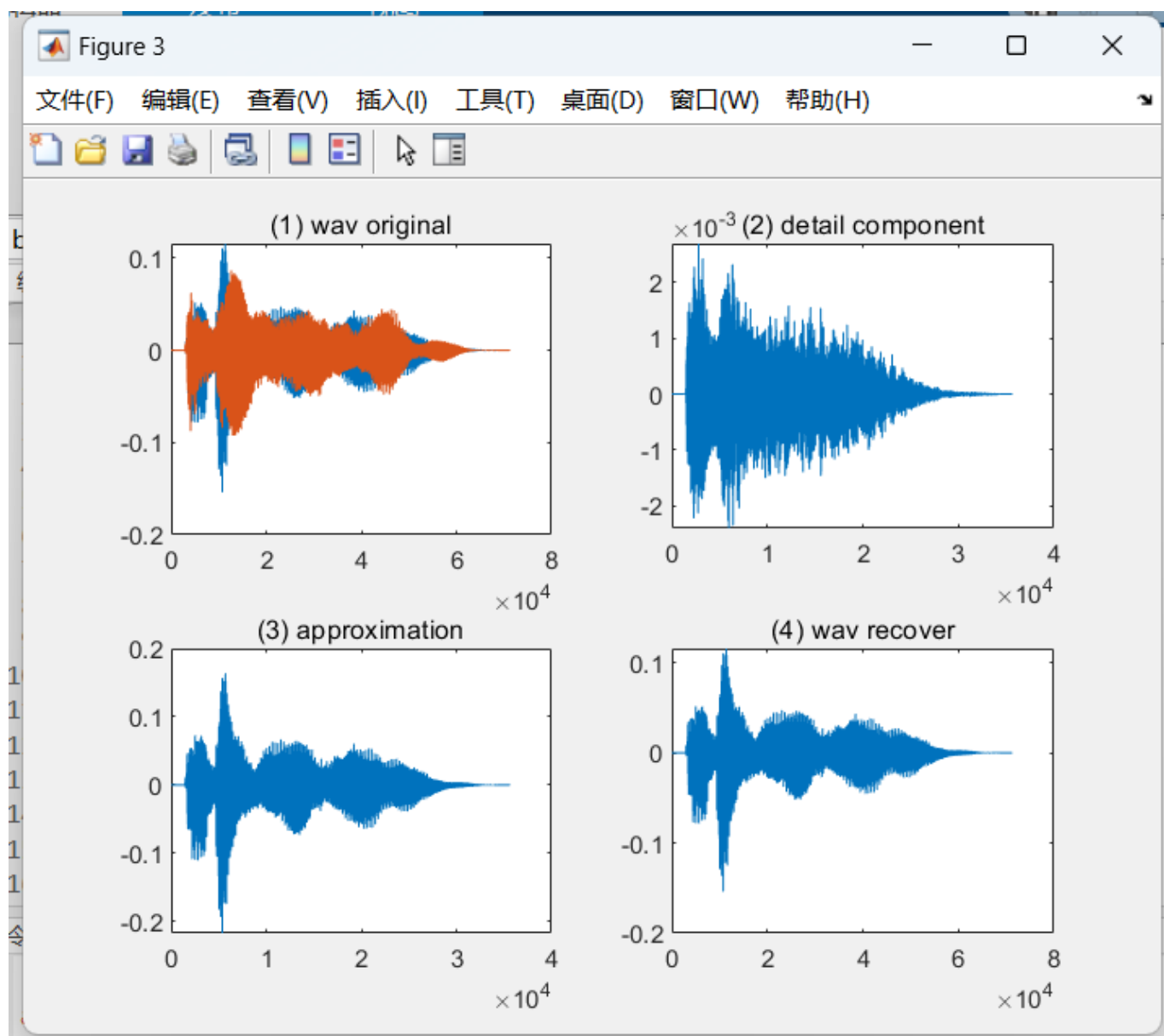
for i = 1 : len % 语音信号转向量
    wav_vec(i) = wav(i);
end

[ca1,cd1] = dwt(wav_vec,'db4'); % 小波基选用 Daubechies-4 小波
wav0 = idwt(ca1,cd1,'db4',len); % 逆 dwt
% wav0 = waverec(ca1,cd1,'db4'); % 逆 wavedec
figure
subplot(2,2,1),plot(wav);
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(wav0);
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');

```

利用 `dwt()` 函数经 `db4` 小波进行分解，之后利用 `idwt()` 函数重构

结果如图：



### 3、db4小波三级分解与重构

```

wav_name = 'tada.wav';
[wav,fs] = audioread(wav_name); % 获得语音信号
len = length(wav); % 语音信号大小
wav_vec = zeros(len,1); % 预分配内存

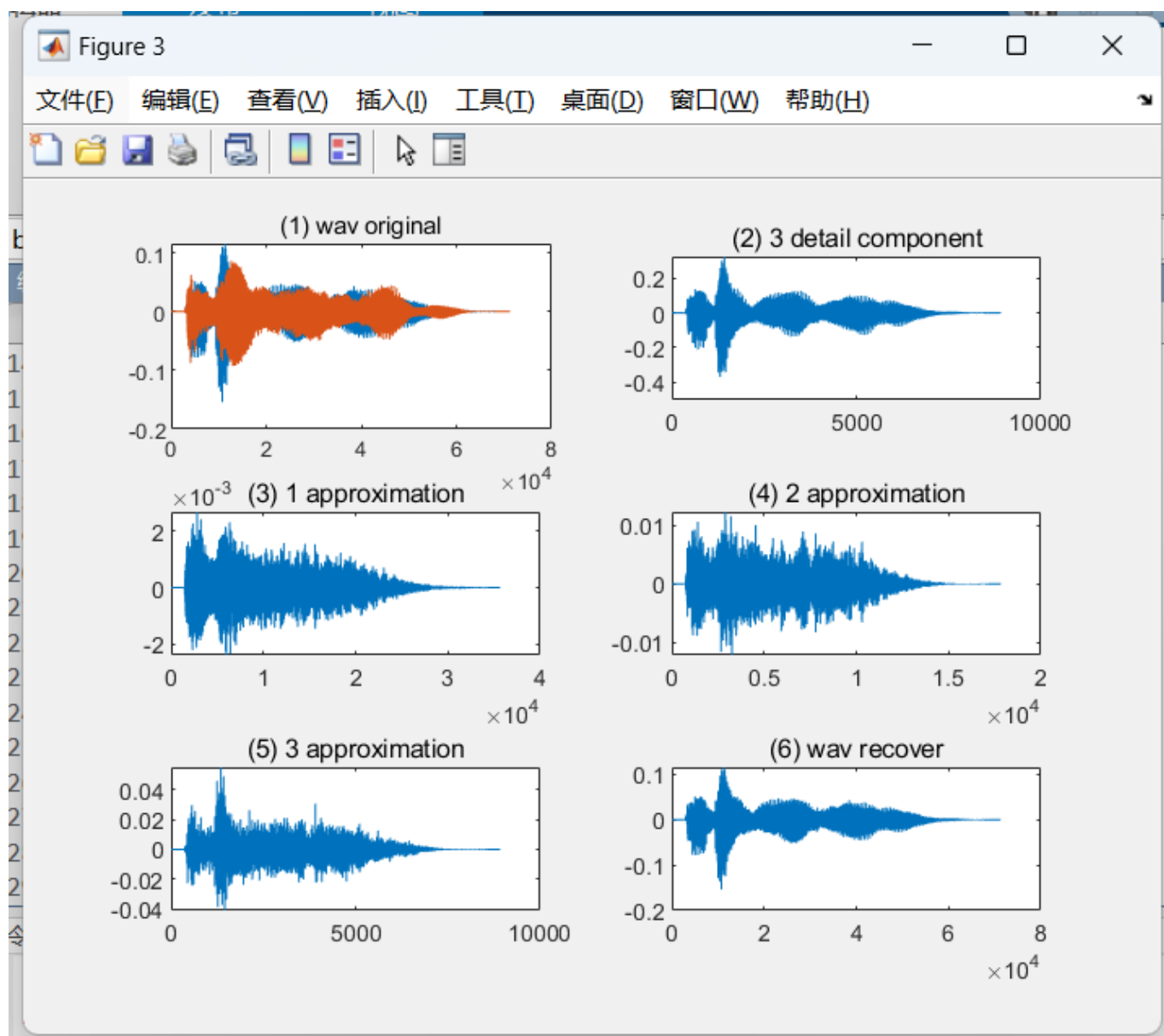
for i = 1 : len % 语音信号转向量
    wav_vec(i) = wav(i);
end

[c,l] = wavedec(wav_vec,3,'db4'); % 小波基选用 Daubechies-4 小波
ca3 = appcoef(c,l,'db4',3); % 三级分解近似分量
cd3 = detcoef(c,l,3); % 三级分解细节分量
cd2 = detcoef(c,l,2); % 二级分解细节分量
cd1 = detcoef(c,l,1); % 一级分解细节分量
wav0 = waverec(c,l,'db4'); % 逆 dwt
figure
subplot(3,2,1),plot(wav);
subplot(3,2,2),plot(ca3); % 三级分解近似分量
subplot(3,2,3),plot(cd1); % 一级分解近似分量
subplot(3,2,4),plot(cd2); % 二级分解近似分量
subplot(3,2,5),plot(cd3); % 三级分解近似分量
subplot(3,2,6),plot(wav0); % 三级重构
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');

```

利用 `wavedec()` 函数经 db4 小波进行分解，之后利用 `waverec()` 函数重构

结果如图：



#### 4、离散余弦变换



```

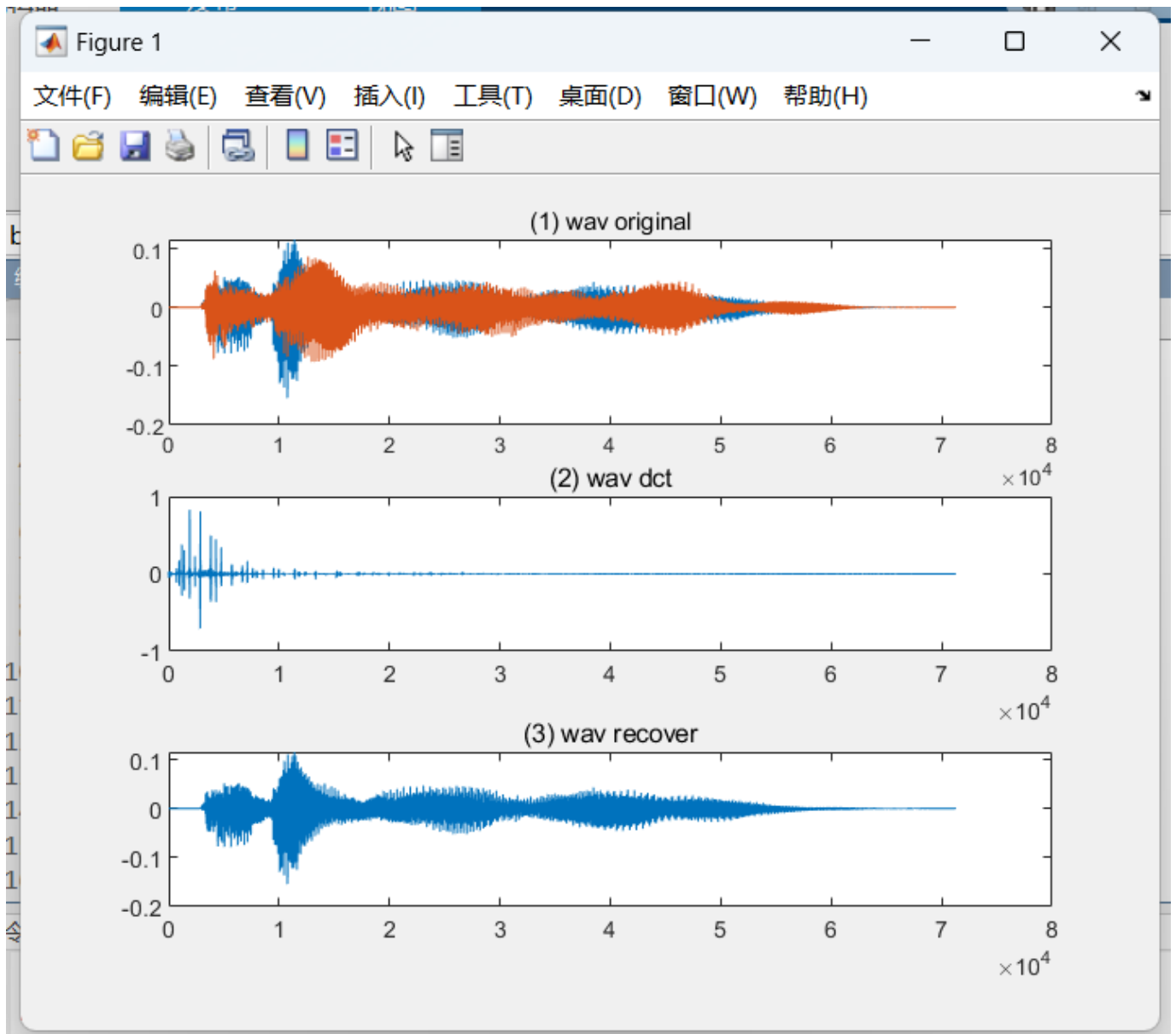
wav_name = 'tada.wav';
[wav,fs] = audioread(wav_name); % 获得语音信号
len = length(wav); % 语音信号大小
wav_vec = zeros(len,1); % 预分配内存

for i = 1 : len % 语音信号转向量
    wav_vec(i) = wav(i);
end

da = dct(wav_vec); % dct
wav0 = idct(da); % 逆 dct
figure
subplot(3,1,1),plot(wav);
subplot(3,1,2),plot(da); % dct
subplot(3,1,3),plot(wav0); % 重构
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');

```

利用 `dct()` 函数经 db4 小波进行分解，之后利用 `idct()` 函数重构



### 三、实验结果分析

#### 1、快速离散傅里叶变换

图一为双声道原音频。 图二为快速傅里叶变换得到的音频。 图三为还原的双声道音频。

#### 2、db4小波一级分解与重构

下图为db4小波一级分解与重构：

1. 图一是原始语音信号（单音频）
2. 图二是一级分解的细节分量，是高频部分，分解后数据长度变为一半。

3. 图三是一级分解的近似分量，是低频部分，分解后数据长度缩减一半
4. 图四是一级分解重构的结果

### 3、db4小波三级分解与重构

下图为db4小波一级分解与重构：

1. 图一是原始语音信号
2. 图二是三级分解的细节分量，也就是高频部分。数据长度较原数据缩短一半。
3. 图三是一级分解的近似分量，也就是高频部分。数据长度较一级数据缩短一半。
4. 图四是二级分解的近似分量，也就是高频部分。数据长度较二级数据缩短一半。
5. 图五是三级分解的近似分量，也就是低频部分。数据长度较二级数据缩短一半。
6. 图六是三级分解重构的结果

### 4、离散余弦变换

下图为DCT变换的结果：

1. 图一是原始语音信号
2. 图二是DCT变换后的结果
3. 图三是重构的结果

## 四、总结与展望

---

在本次实验中，首先学习了一些普遍的常用语音处理方法以及一些公式，比如傅立叶变换、小波变换等，然后通过 MATLAB 中复现用快速傅立叶变换、db4小波一级分解与重构、db4小波三级分解与重构、离散余弦变换来对语音信号进行处理，对所学到的理论知识进行相应的应用，对 MATLAB 的应用也更加的熟练，最后期待自己未来更好的发展，能在本学期学习信息隐藏技术这门课程的过程当中有更大的收获，也能在未来关于这门课程的研究上有所成就。