

语音信号处理实验报告

实验一

- 1 用 Matlab 读取一段语音（自己录制一段，最好其中含有汉语四种声调变化，该段语音作为本课实验原始材料），绘制原始语音波形图。
2. 用 Matlab 计算这段语音的短时平均过零率、短时平均能量和短时平均幅度，并将多个波形同步显示绘图。
3. 观察各波形在不同音情况下的参数特点，并归纳总结其中的规律。

```
clc clc;
[x, fs]=wavread('benpao.wav');
figure
plot(x);
axis([0 length(x) min(x) max(x)]);
title('原始语音波形')
xlabel('时间')

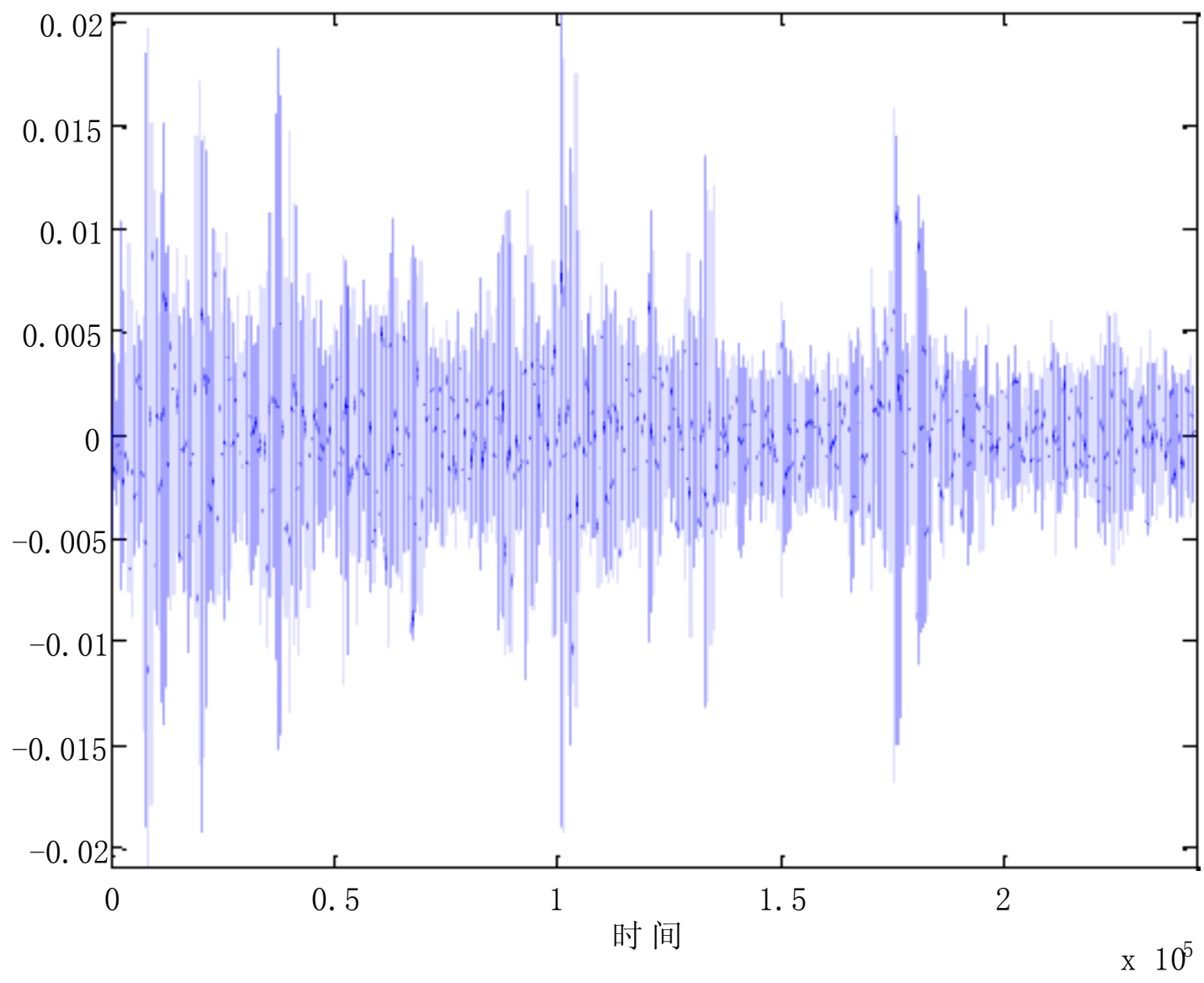
f=enframe(x, 300, 100);
[m, n]=size(f);
for i=1:m
energy(i)=sum(f(i, 1:n).^2);
mn(i)=sum(abs(f(i, 1:n)));
end
figure
plot(energy);
axis([0 length(energy) min(energy) max(energy)]);
title('短时能量')
figure
plot(mn);
axis([0 length(mn) min(mn) max(mn)]);
title('短时幅度')
lingd=zeros(m);
for x=1:m
    for y=1:n-1
        temp=f(x, y)*f(x, y+1) ;
        if temp<= 0
            lingd(x)=lingd(x)+1;
        end
    end
end
%temp1=num(x, 1)/300;
%count(x)=temp1;
end
figure
plot(lingd);
```

```
%axis([0 length(lingd) min(lingd) max(lingd)]);  
title(' 短时过零率')
```

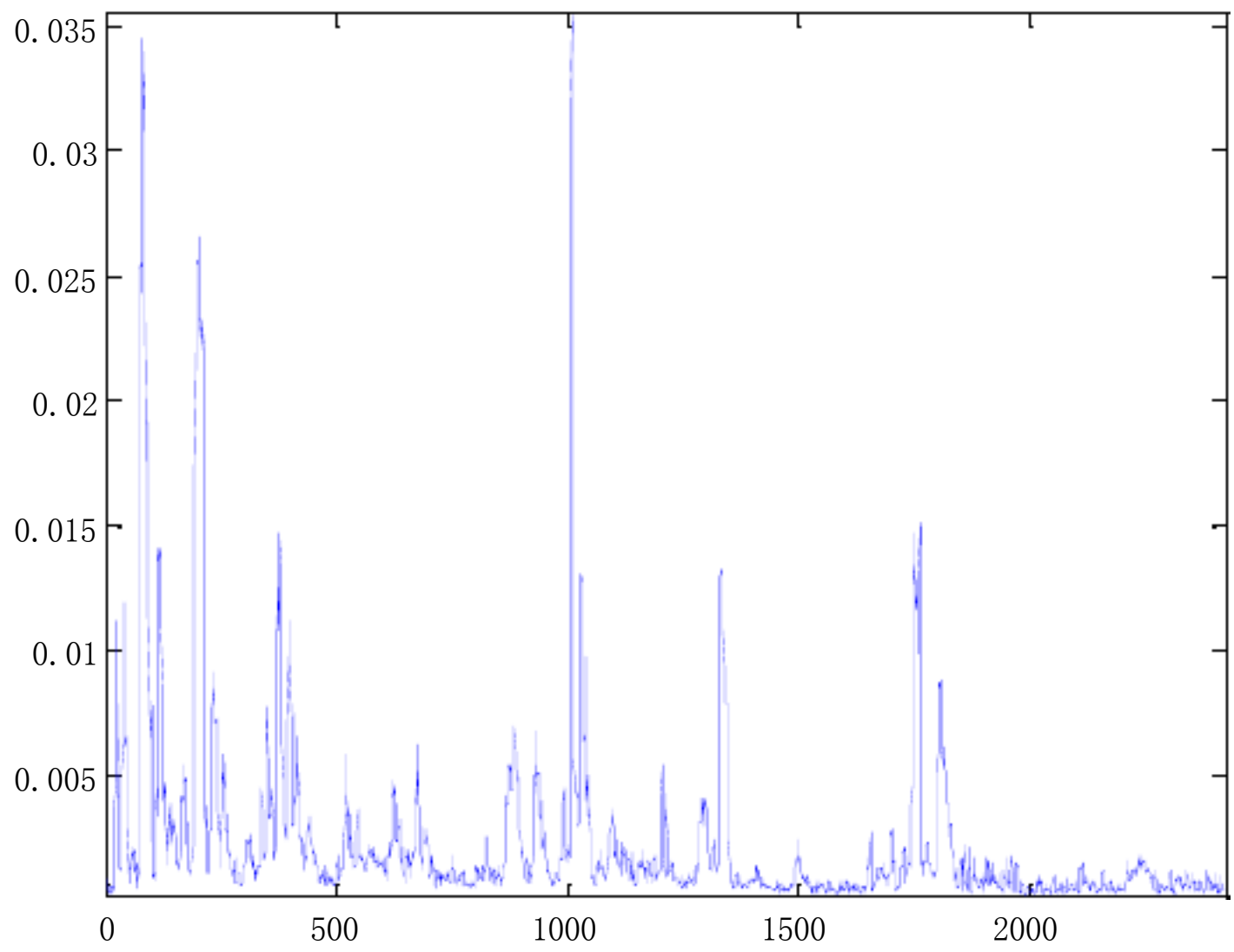
子函数:

```
function f=enframe(x,win,inc) %定义函数。  
nx=length(x(:)); %x(:) 的作用是把x给弄成一个向量, x为一行, 则变成一列, 如果为  
矩阵, 则按每一列的顺序排成一列。得出的nx 为序列的数据个数。  
nwin=length(win);  
if (nwin==1)  
len=win; %如果win 中就一个数, 则len 就=该数, 此例中为256 个点。即每帧长  
else  
len=nwin; %如果有多个数, 则len= 个数。  
end  
if (nargin<3) %nargin 返回的是函数输入的个数, 如果中间有变量, 返回的是负值。  
inc=len; %也就是说, 如果函数enframe 的输入只有两个的话, 系统就自动赋inc  
end  
nf=fix((nx-len+inc)/inc);  
%这个比较关键, nf 为分帧的组数  
f=zeros(nf, len); %构成以组数为行, 帧长为列的矩阵。  
indf=inc*(0:(nf-1)).'; %indf 为一列nf 个数据, 即0到nf-1 的inc 倍, 即分好了每  
帧起点。  
inds=(1:len); %构成了长度为len 的一行。  
f(:)=x(indf(:,ones(1, len))+inds(ones(nf, 1), :));  
if (nwin>1) %nwin 大于1的情况就不说了。  
w=win(:)';  
f=f.*w(ones(nf, 1), :);  
end
```

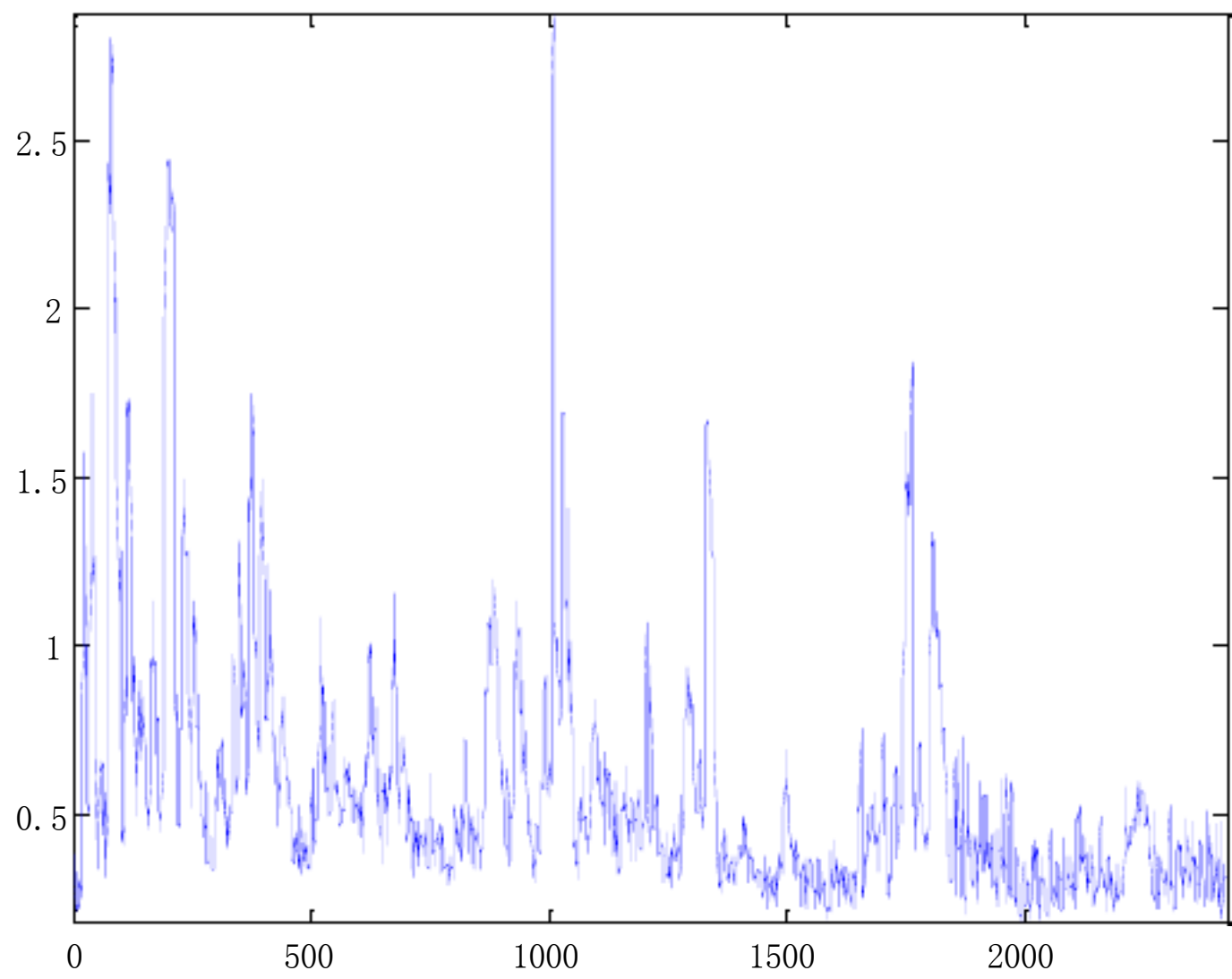
原始语音波形



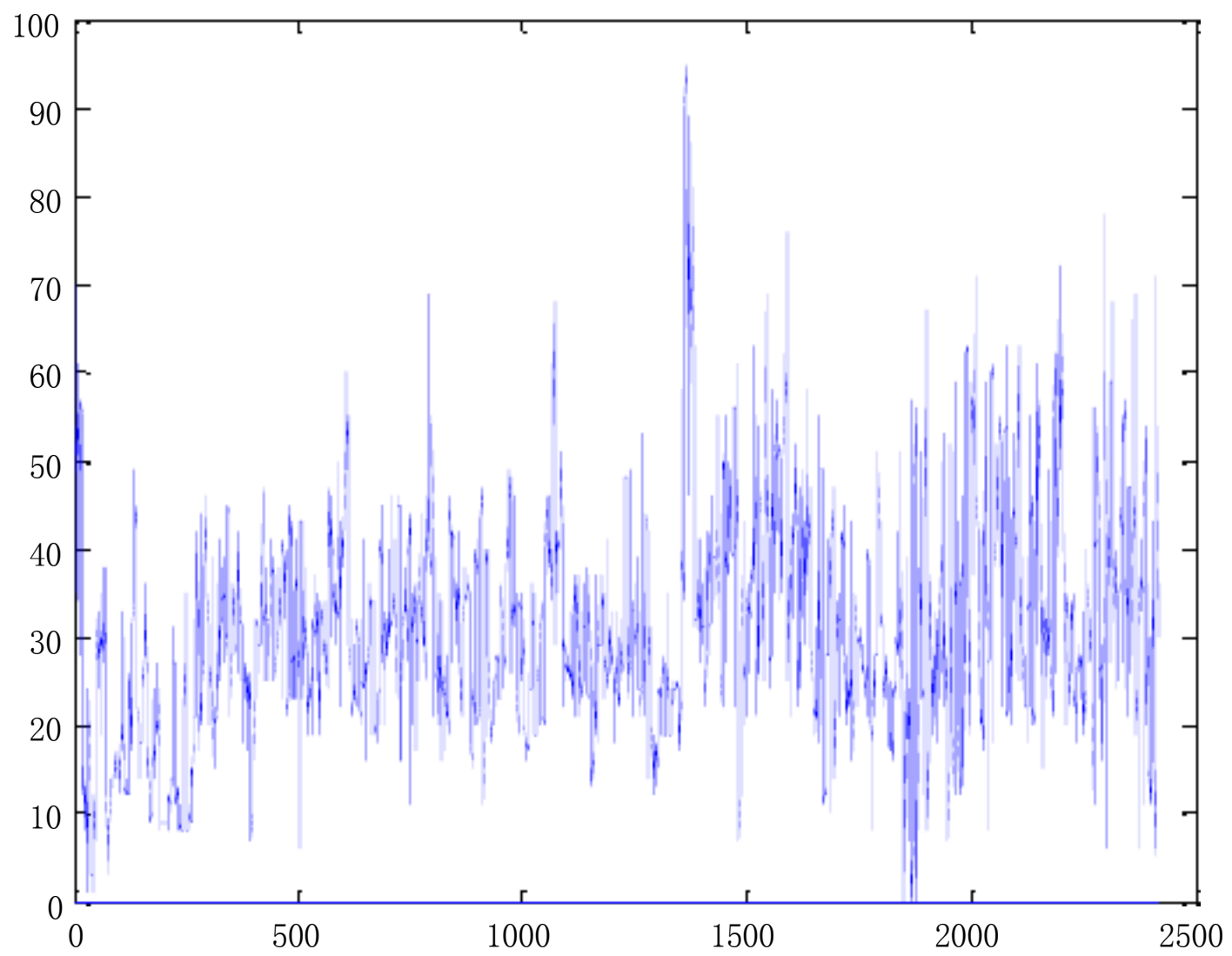
短时能量



短时幅度



短时过零率



结论:

浊音能量集中与较低频段中，所以具有较低的过零率；相反，清音能量集中于较高频率中，所以有较高的过零率。

浊音短时的平均幅度比清音的要高，所以可以从图中区别浊音和清音。

可以从边缘区别声母和韵母的分界、无声和有声的分界。

实验二要求：

- 1.分别选定浊音段和清音段一帧语音信号，计算其自相关函数和平均幅度差函数，并同步绘制两参数波形，总结在不同音情况下，这两参数表现的特点。
- 2.利用三电平中心削波自相关法实现清、浊音判定，并提取一段话音的基频参数，绘制基频曲线，并将该参数曲线合并到前面提取的短时过零率和短时能量参数曲线图中，以示比较。
- 3.加入 3 点或 5 点中值平滑处理后再绘制上述基频曲线与未平滑曲线比较，观测完整一段话音的基频包络变化情况。

```
[Y, Fs, bits]=wavread( 'benpao.wav' );
```

```
%浊音，取13270--13510 个点
```

```
%短时自相关函数
```

```
temp=Y(13271:13510);
```

```
Rn1=zeros(1, 240);
```

```
for nn=[1:240],
```

```
    for ii=[1:240-nn],
```

```
        Rn1(nn) =Rn1(nn)+ temp(ii)*temp(nn+ii);
```

```
    end
```

```
end
```

```
figure(1)
```

```
subplot(4, 1, 1);
```

```
jj=[1:240];
```

```
plot(jj, Rn1, 'b');title( '浊音-自相关函数' )
```

```
grid
```

```
%短时平均幅度差函数
```

```
Yn1=zeros(1, 240);
```

```
for nn=[1:240],
```

```
    for ii=[1:240-nn],
```

```
        Yn1(nn) =Yn1(nn)+ abs(temp(ii)-temp(nn+ii));
```

```
    end
```

```
end
```

```
figure(1)
```

```
    subplot(4, 1, 2)
```

```
jj=[1:240];
```

```
plot(jj, Yn1, 'b');title( '浊音-短时平均幅度差函数' )
```

```
grid
```

```
%清音，取12120--12360 个点
```

```

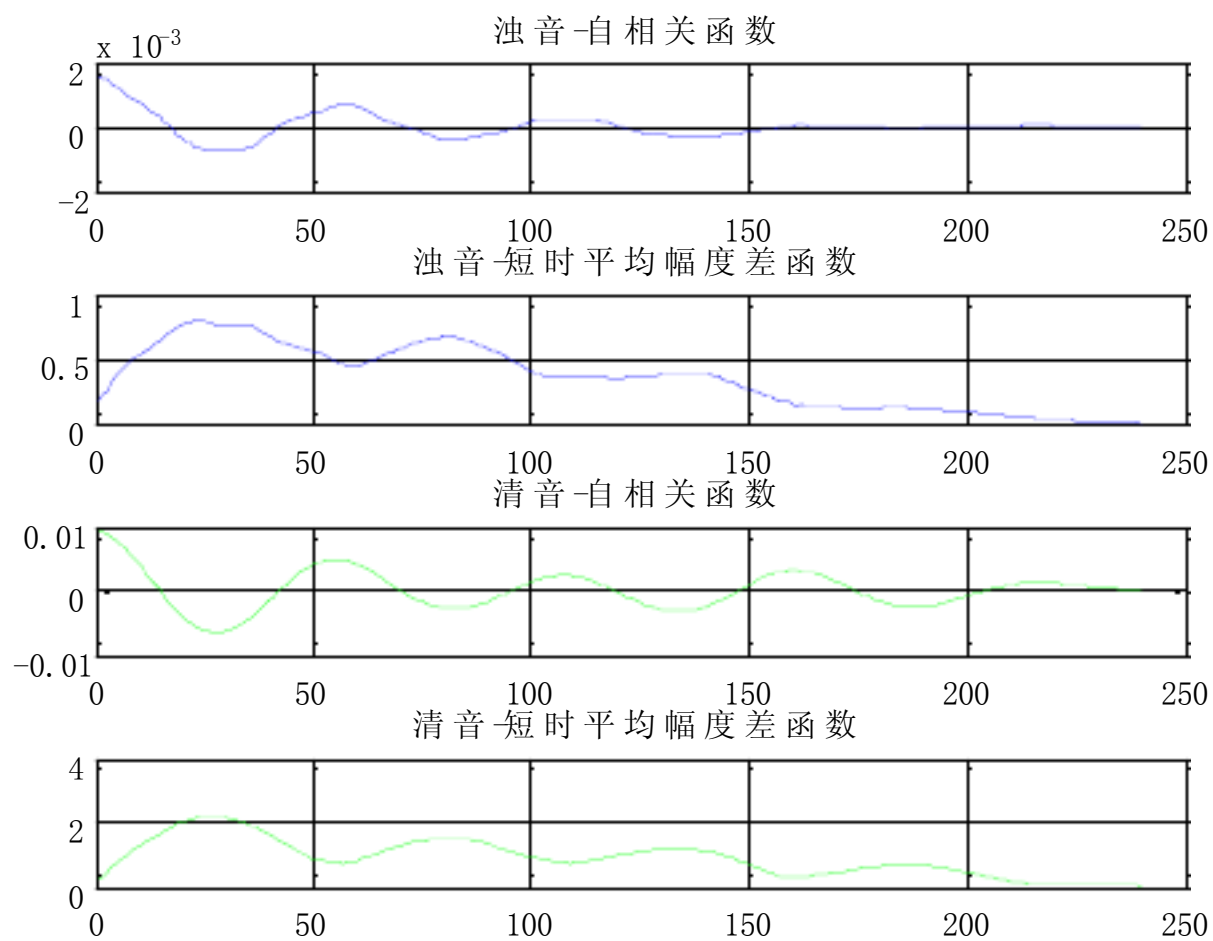
%短时自相关函数
temp=Y(12121:12360);
Rn2=zeros(1,240);
for nn=[1:240],
    for ii=[1:240-nn],

        Rn2(nn) =Rn2(nn)+ temp(ii)*temp(nn+ii);
    end
end
figure(1)
subplot(4,1,3)
jj=[1:240];
plot(jj, Rn2, 'g');title('清音-自相关函数')
grid

%短时平均幅度差函数
Yn2=zeros(1,240);
for nn=[1:240],
    for ii=[1:240-nn],

        Yn2(nn) =Yn2(nn)+ abs(temp(ii)-temp(nn+ii));
    end
end
figure(1)
subplot(4,1,4)
jj=[1:240];
plot(jj, Yn2, 'g');title('清音-短时平均幅度差函数')
grid

```



```

fid=fopen( 'benpao.txt' , 'rt' );
[a, count]=fscanf(fid, '%f' , [1, inf]);
L=length(a);
m=max(a);
for i=1:L
a(i)=a(i)/m;
end
m=max(a);
n=min(a);
ht=(m+n)/2;
for i=1:L;
a(i)=a(i)-ht;
end
figure(1);
subplot(2,1,1);
plot(a, 'k' );
axis([0,1711,-1,1]);
title( '中心削波前语音波形' );
xlabel( '样点数' );
ylabel( '幅度' );
coeff=0.7;
th0=max(a)*coeff;
for k=1:L ;
    if a(k)>=th0
        a(k)=a(k)-th0;
    elseif a(k)<=(-th0);
        a(k)=a(k)+th0;
    end
end

```

```

        else
            a(k)=0;
        end
    end
end
m=max(a);
for i=1:L;
a(i)=a(i)/m;
end
subplot(2,1,2);
plot(a, 'k');
axis([0,1711,-1,1]);
title('中心削波后语音波形');
xlabel('样点数');
ylabel('幅度');
fclose(fid);
fid=fopen('sara.txt','rt');
[b,count]=fscanf(fid,'%f',[1,inf]);

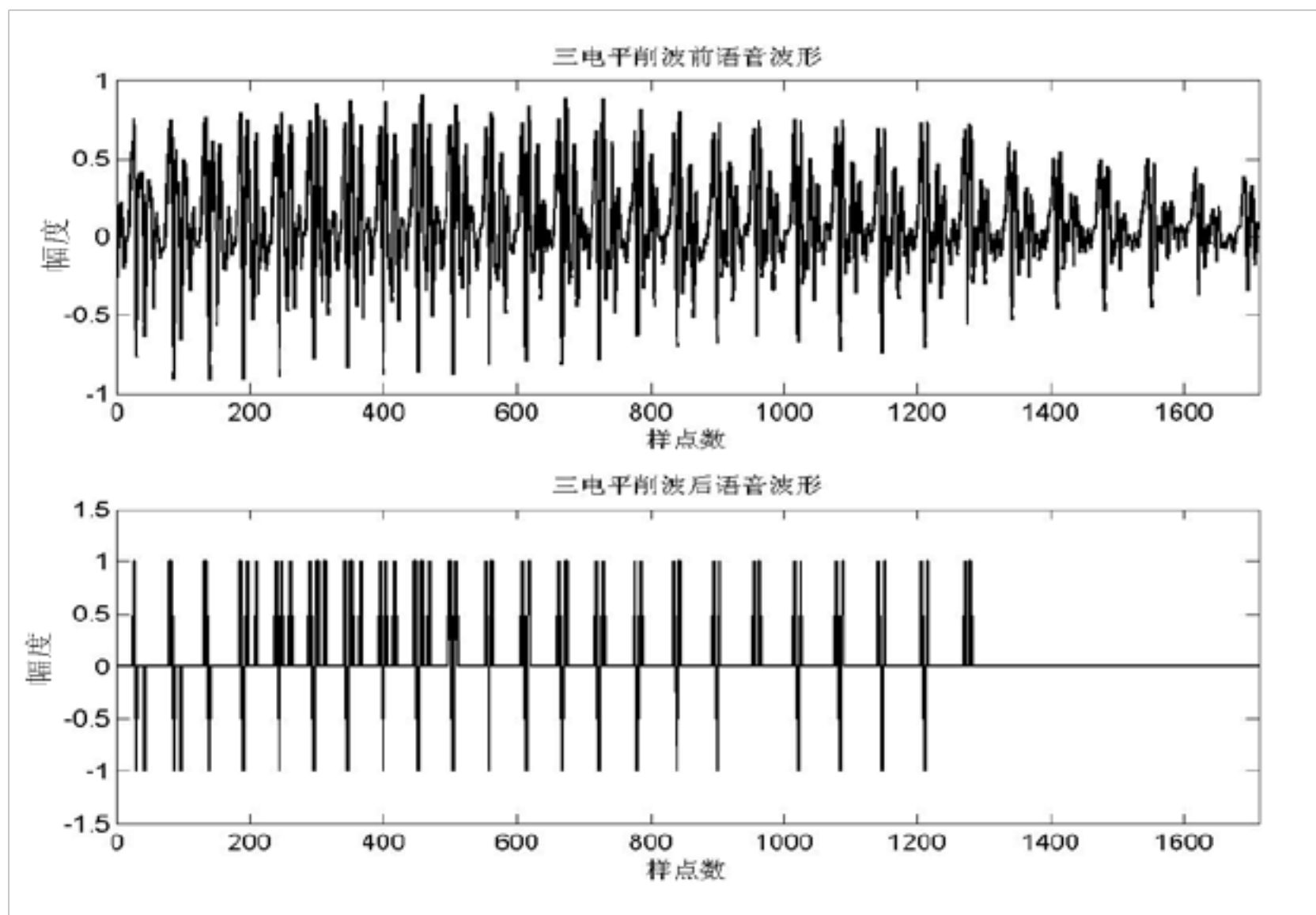
fclose(fid);
N=320;
A=[];
for k=1:320
sum=0;
for m=1:N-(k-1)
sum=sum+b(m)*b(m+k-1);
end
A(k)=sum;
end
for k=1:320
    B(k)=A(k)/A(1);
end
figure(2);
subplot(2,1,1);
plot(B, 'k');
title('中心削波前修正自相关');
xlabel('延时k');
ylabel('幅度');
axis([0,320,-1,1]);
N=160;
A=[];
for k=1:160;
sum=0;
for m=1:N;
sum=sum+a(m)*a(m+k-1);

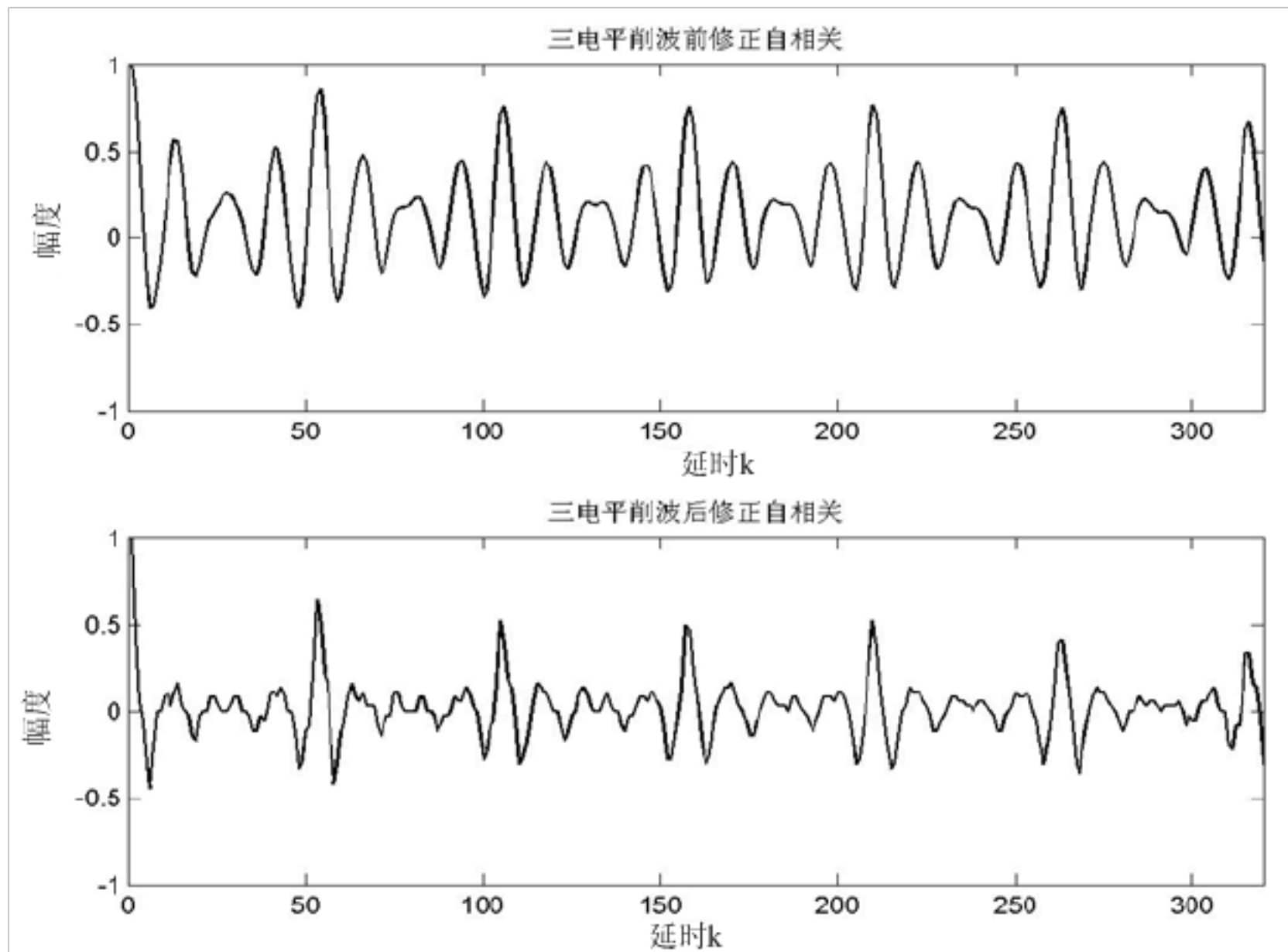
```



```
end
A(k)=sum;
end
for k=1:320
    C(k)=A(k)/A(1);
end
subplot(2,1,2);
plot(C, 'k' );
title( '中心削波后修正自相关' );
xlabel( '延时k' );
ylabel( '幅度' );
axis([0,320,-1,1]);
```

结果:





主函数

```
[x, fs]=wavread( 'benpao.wav' );
zhouqi0=medfilt1(x, 5);
zhouqi2=linsmooth(zhouqi0, 5);
w=[];
w=x;
w1=w-zhouqi2;
w1=medfilt1(w1, 5);
w1=linsmooth(w1, 5);
plot(zhouqi2);
title( '5点中值平滑处理后波形' );
axis([0, length(x), min(zhouqi2), max(zhouqi2)]);
```

子函数

```
function [y]=linsmooth(x, n, wintype)
if nargin<3
    wintype= 'hann' ;
end
if nargin<2
    n=3;
end
win=hann(n);
win=win/sum(win);
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/377152051022010002>