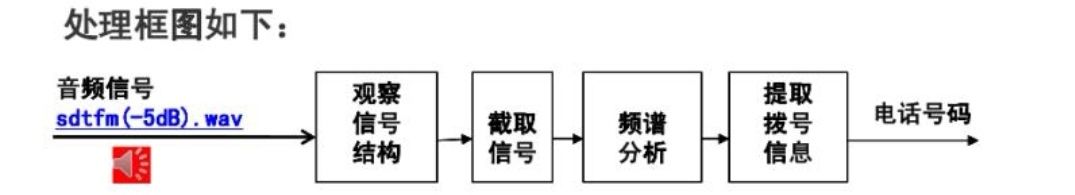
# 一、问题描述及要求

在音频信号中混杂了一段电话拨号音，其采用DTMF编码，要求通过信号分离及频谱分析，了解信号频谱成分，从频谱信息中提取电话号码。

处理框图：



# 二、问题分析及建模

## 相关术语解释：

DTMF：双音多频，由高频群和低频群组成，高低频群各包含4个频率。一个高频信号和一个低频信号叠加组成一个组合信号，代表一个数字。DTMF信号有16个编码。利用DTMF信令可选择呼叫相应的[对讲机](https://baike.baidu.com/item/%E5%AF%B9%E8%AE%B2%E6%9C%BA/641751" \t "_blank)。

1.所有音频都位于人的可听范围内，因此按键下去时人可以听到。  
2.8个频率中没有一个频率是其他任意一个频率的倍数。  
3.任意两个频率的组合，相加或相减都不等于其他任意一个频率。



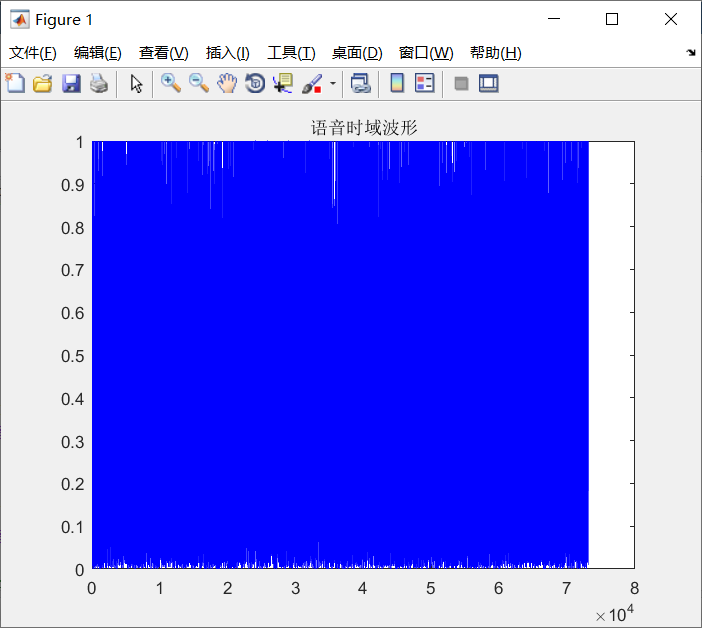
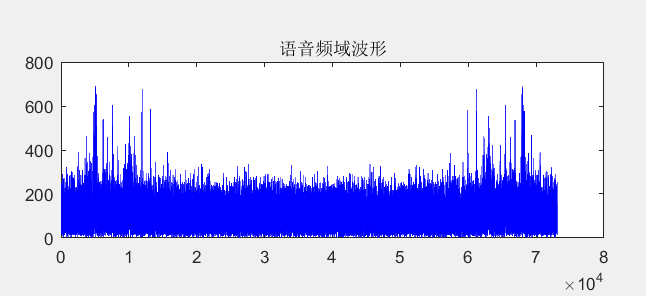
滤波器：滤波器是由电容、电感和电阻组成的滤波电路。滤波器可以对电源线中特定频率的频点或该频点以外的频率进行有效滤除，得到一个特定频率的电源信号，或消除一个特定频率后的电源信号。

短时能量：由于语音信号的能量随时间而变化，清音和浊音之间的能量差别相当显著。因此，对短时能量和短时平均幅度进行分析，可以描述语音的这种特征变化情况。

Goertzel算法：格兹尔算法( Goertzel algorithm )是数字信号处理的一种运算技巧，此运算技巧提供一个有效率的方式来估计部分区域的离散傅立叶转换，广泛的运用在数字电话中的的双音多频信号(每个拨号的数字键由两个频率的音所组成，一个低频，一个高频)，此算法在1958年被杰拉德·格策尔( Gerald Goertzel )所提出。

## 设计思路：

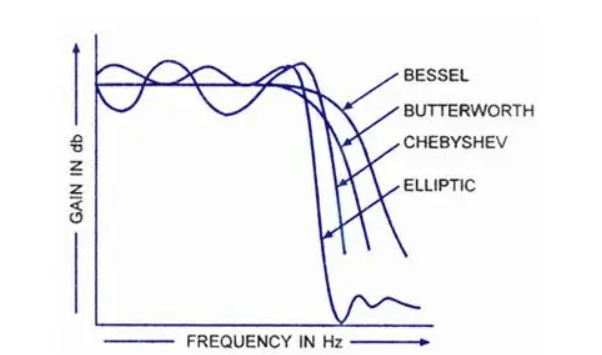
首先利用matlab的audioread函数，将需要处理的音频文件进行采样，这里采取的fs为自带采样频率，为8192Hz，利用audioinfo函数，获取音频基本信息。fft函数进行快速傅里叶变换，将未处理前音频信号的时域与频域特征显示。

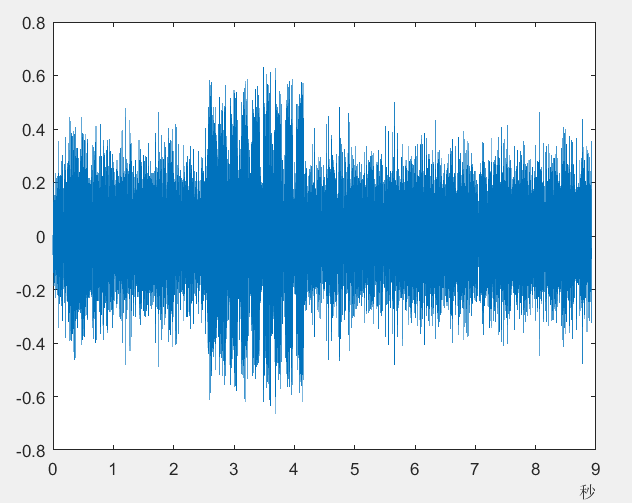
 

如图所示，可以发现音频信号时域混乱且杂乱无章，不能分辨出播音信号，需要进行滤波处理。

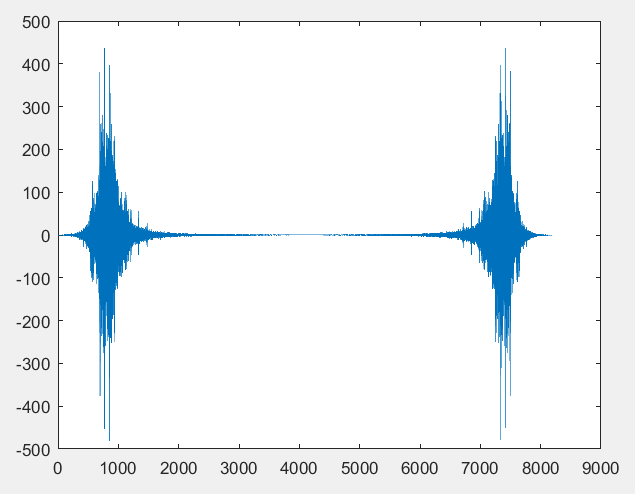
通过DTMF理论知识可知，其由一有限范围低频组信号和一高频组信号叠加组成，可以通过构造滤波器来去除其他无关信号。查阅资料，

Matlab可以构造贝塞尔，巴特沃斯，切比雪夫与椭圆滤波器等滤波函数。由于巴特沃斯滤波器带外衰减速度较慢，而该音频文件背景噪音较大，故需要采用椭圆滤波器，使得阶段陡峭，阻隔其他无关频率信号。



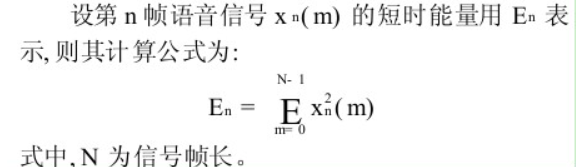


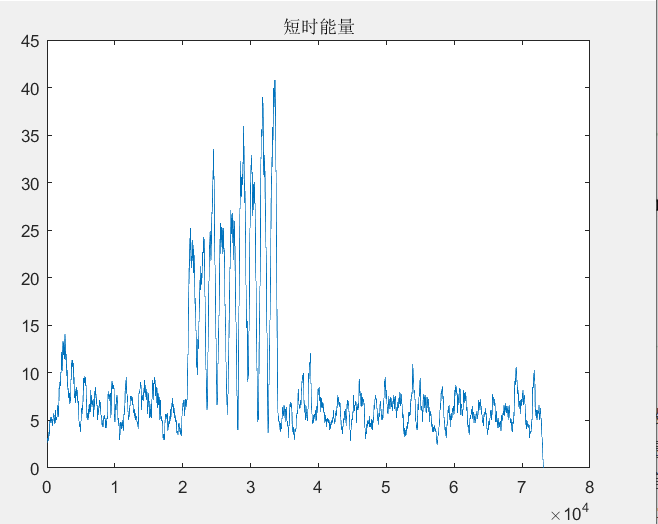
低通滤波后时域信号



低通滤波后频域信号

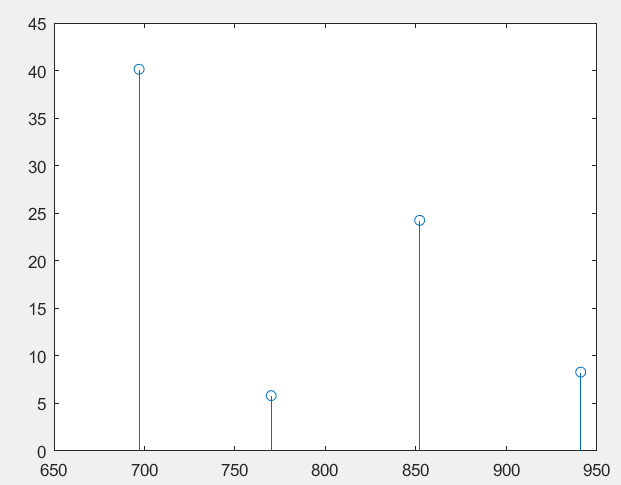
由于时域部分仍不显著，无法分辨出播音信号起止位置，需要采取短时能量法。语音信号一般可分为无声段、清音段和浊音段。无声段是背景噪声段,平均能量最低;浊音段为声带振动发出对应的语音信号段,平均能量最高;清音段是空气在口腔中的摩擦、冲击或爆破而发出的语音信号段,平均能量居于前两者之间。清音段和无声段的波形特点有明显的不同,无声段信号变化较为缓慢, 而清音段信号在幅度上变化剧烈,穿越零电平次数也多。经验表明,通常清音段过零率最大。端点检测就是首先判断/有声0还是/无声0,如果有声,则还要判断是/清音0还是/浊音0。为正确地实现端点检测,一般综合利用短时能量。





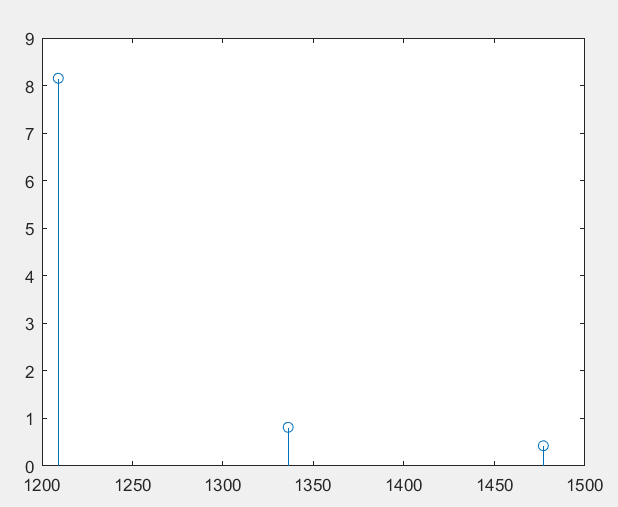
很明显，图像具有9个高峰，显然有9个拨号。

利用findpeaks函数，找出波峰所在位置帧数的横坐标，利用Goertzel算法来进行低频信号匹配。



如图，697Hz时分量最大，则第一个音的低频组为697Hz。

同理，对原信号进行高频率波，其原理与低频相同



该图为第一个音频的高频组分量为1209Hz，和其低频组分量叠加后可以发现解调后的音为1.

# Matlab仿真结果及代码

low = [697 770 852 941];

high = [1209 1336 1477];

tele = [1,2,3; 4,5,6; 7,8,9; 11,0,12]; % '\*' = 11; '#' = 12

[yy,fs]=audioread('sdtmf(-5dB).wav');%读入音频文件

audio1=audioinfo('sdtmf(-5dB).wav');

t=0:seconds(1/fs):seconds(audio1.Duration);

t=t(1:end-1);

grid on; %网格线

hold on; %设置图像保持

YY=fft(yy);%对语音进行傅里叶变换

f=(0:audio1.TotalSamples-1)\*fs/audio1.TotalSamples;

subplot(2,1,1);

figure(1);

plot(abs(yy),'b');%语音时域波形

title('语音时域波形');

figure(2);

subplot(2,1,2);

plot(abs(YY),'b');%语音频域波形

title('语音频域波形');

%制作一个低通滤波器（巴特沃斯滤波器）

wp =690 / (fs/2); %通带截止频率,并对其归一化

ws = 950 / (fs/2); %阻带截止频率,并对其归一化

alpha\_p = 3; %通带允许最大衰减为 3 db

alpha\_s = 10; %阻带允许最小衰减为 10 db

%获得转移函数系数

[ b, a ] = butter(2, [wp ws], 'bandpass');

%滤波

s1 = filter(b,a,yy);

w1 = fft(s1);

figure(3);

plot(t,s1);

figure(4);

plot(f,w1);

N=seconds(0.05)/seconds(audio1.Duration)\*audio1.TotalSamples;

%分帧

a=zeros(1,audio1.TotalSamples);

for i=1:audio1.TotalSamples

for j=i:i+N

if j~=1&&j<audio1.TotalSamples

a(i)=a(i)+s1(j)^2;

end

end

end

figure(5);

plot(1:audio1.TotalSamples,a);

title('短时能量');

[p,location2] = findpeaks(a,'minpeakheight',25);

location2=[21134,23032,24546,25561,27099,28976,30163,31757,33545];

freq\_indices = round(low/fs\*N) + 1;

for i=1:length(location2)

figure(5+i);

v=goertzel(s1(location2(i):location2(i)+N-1),freq\_indices);

[M,index1]=max(abs(v));

low\_f(i)=low(index1);

stem(low,abs(v));

end

%制作高通滤波器（同样使用巴特沃斯滤波器）

wp2 =1200/ (fs/2); %通带截止频率,并对其归一化

ws2 = 1500 / (fs/2); %阻带截止频率,并对其归一化

alpha\_p = 3; %通带允许最大衰减为 3 db

alpha\_s = 10; %阻带允许最小衰减为 10 db

%获得转移函数系数

[ b2, a2 ] = butter(2, [wp2 ws2], 'bandpass');

%滤波

s2 = filter(b2,a2,yy);

w2 = fft(s2);

figure(15);

plot(t,s2);

figure(16);

plot(f,w2);

N=seconds(0.05)/seconds(audio1.Duration)\*audio1.TotalSamples;

%分帧

a2=zeros(1,audio1.TotalSamples);

for i=1:audio1.TotalSamples

for j=i:i+N

if j~=1&&j<audio1.TotalSamples

a2(i)=a2(i)+s2(j)^2;

end

end

end

figure(17);

plot(1:audio1.TotalSamples,a2);

title('短时能量');

[p2,location2] = findpeaks(a2,'minpeakheight',25);

location2=[21134,23032,24546,25561,27099,28976,30163,31757,33545];

freq\_indices = round(high/fs\*N) + 1;

for i=1:length(location2)

figure(20+i);

v2=goertzel(s1(location2(i):location2(i)+N-1),freq\_indices);

[M,index2]=max(abs(v2));

high\_f(i)=low(index2);

stem(high,abs(v2));

end

# 结果分析及结论

通过分析，可以得出此音频的结果为137025689