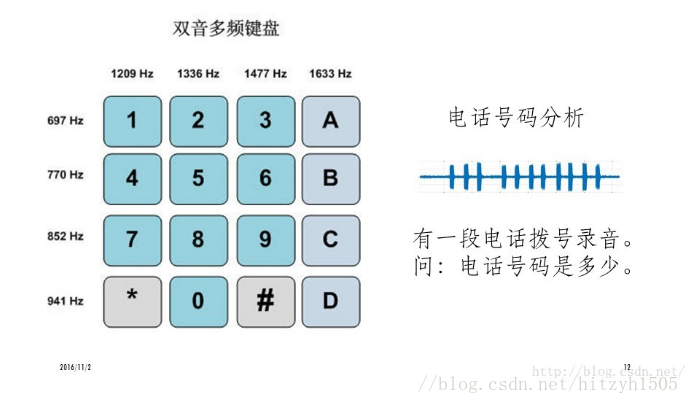
**Practice4**

DTMF即双音多频。双音多频的拨号键盘是4×4的矩阵，每一行代表一个高频，每一列代表一个低频。用户每按一个键就发送一个高频和低频的正弦信号组合，比如在键盘上按“1”键，电话机就会发送一个697 Hz和1209 Hz的组合信号给电话交换机。每个按键的按键音包含着两个频率的信号，在生成时即可将两个频率的信号进行加和来生成DTMF信号。



**生成DTMF代码**

fs=8000;%采样率

tk=0.5;%the time of the key按键的声音持续时间

t=0:1/fs:(tk-1/fs);%时间点

z=zeros(1,4000);%按键与按键之间空白时间

%--------------------生成每个频率的信号----------------------

s697=sin(2\*pi\*697\*t);

s770=sin(2\*pi\*770\*t);

s852=sin(2\*pi\*852\*t);

s941=sin(2\*pi\*941\*t);

s1209=sin(2\*pi\*1209\*t);

s1336=sin(2\*pi\*1336\*t);

s1477=sin(2\*pi\*1477\*t);

s1633=sin(2\*pi\*1633\*t);

%---------------------生成每个按键的多频-------------------

p1=s697+s1209;

p2=s697+s1336;

p3=s697+s1477;

pa=s697+s1633;

p4=s770+s1209;

p5=s770+s1336;

p6=s770+s1477;

pb=s770+s1633;

p7=s852+s1209;

p8=s852+s1336;

p9=s852+s1477;

pc=s852+s1633;

pxing=s941+s1209;%星号\*\*

p0=s941+s1336;

pjing=s941+s1477;%#号

pd=s941+s1633;

%------------------生成声音序列----------------------------

sound=[p1,z,p6,z,p0,z,p1,z,p0,z,p1,z,p9,z,p9,z,p0,z,p4,z,p4,z,];

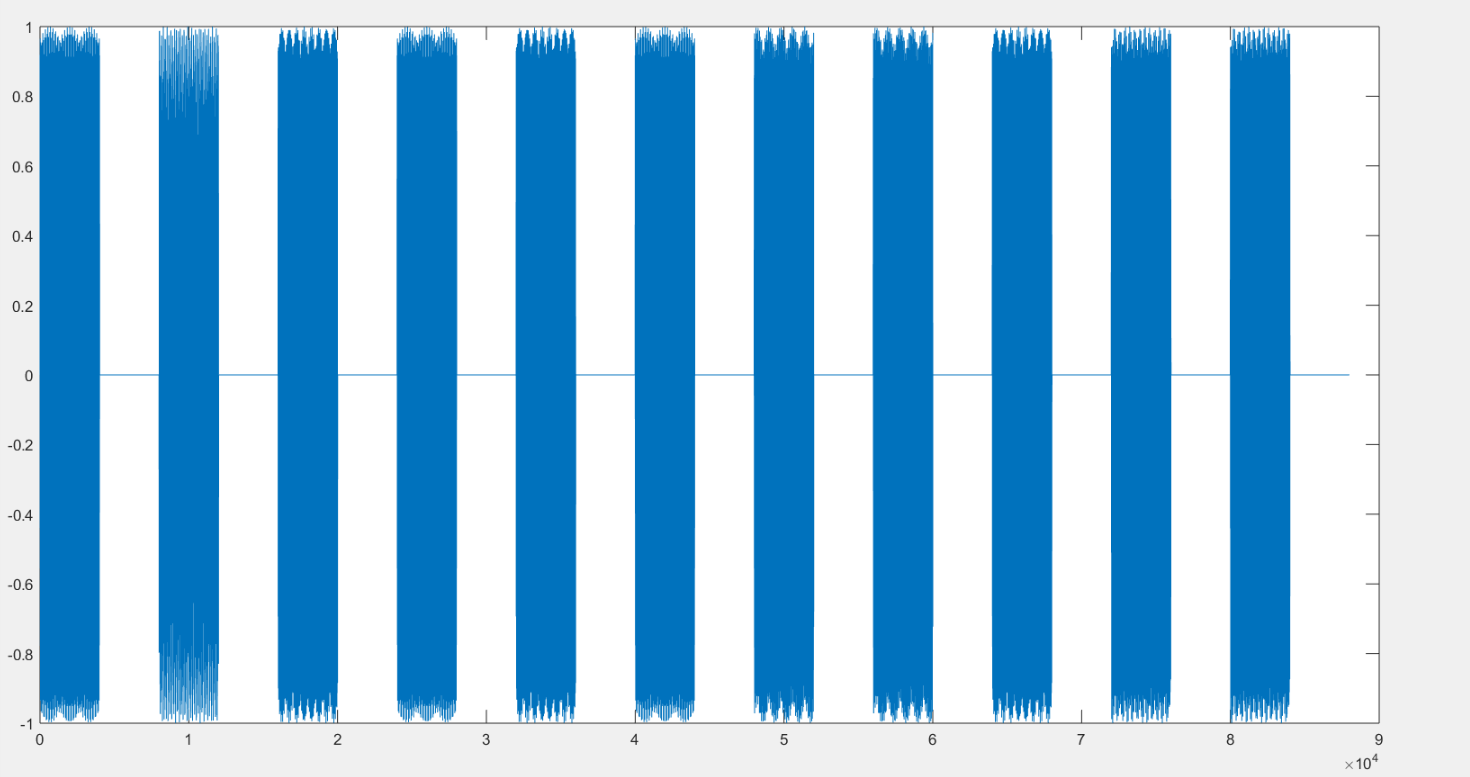
sound=sound/max(abs(sound));%归一化声音序列

figure(1);

plot(sound);

%写出音频

audiowrite('my\_phone\_number\_sound\_test.wav',sound,8000);

**产生波形如图**

在解调DTMF时，可以将一段声音信号根据按键的持续长度截断成几个小段，对每个小段进行fft变换，求出该段的频率分量。由于一个按键的声音包含两个频率，所以将fft得到的频率与表中的频率进行对比来判断按键类型。在写代码时将fft结果转换为单变谱，由于dtmf的行频率与列频率正好以1000Hz为界，可以将fft后的单边谱的1000Hz以下的分量与列频率进行比较，将1000Hz以上的频率与hang频率进行比较，由两者的结果来判断按键类型。

**代码**

[h,Fs]=audioread('my\_phone\_number\_sound\_test.wav');%读出信号，采样率和采样位数。

h=h(:,1);

keytime=0.5;%按键持续时间

zerotime=0.5;%中间为零的时间

n=keytime\*Fs;

n1=zerotime\*Fs;

for i=1:11

y=h(1+n1\*(i-1)+n\*(i-1):n+n1\*(i-1)+n\*(i-1));

H=fft(y);%快速傅里叶变换

h\_d=abs(H/n);

h\_d1= h\_d(1:n/2+1);

h\_d1(2:end-1)=2\*h\_d1(2:end-1);

f=Fs\*(0:n/2)/n;

h\_d2=h\_d1(1:n/8);

[M,I] = max(h\_d2);

I=Fs\*I/n;

h\_d3=h\_d1(n/8:n/4-1);

[A,B] = max(h\_d3);

B=Fs\*B/n+1000;

if I>=690&&I<=700&&B>=1200&&B<=1300

disp(1);

end

if I>=690&&I<=700&&B>=1300&&B<=1400

disp(2);

end

if I>=690&&I<=700&&B>=1400&&B<=1500

disp(3);

end

if I>=690&&I<=700&&B>=1600&&B<=1700

disp(A);

end

if I>=700&&I<=800&&B>=1200&&B<=1300

disp(4);

end

if I>=700&&I<=800&&B>=1300&&B<=1400

disp(5);

end

if I>=700&&I<=800&&B>=1400&&B<=1500

disp(6);

end

if I>=700&&I<=800&&B>=1600&&B<=1700

disp(B);

end

if I>=800&&I<=900&&B>=1200&&B<=1300

disp(7);

end

if I>=800&&I<=900&&B>=1300&&B<=1400

disp(8);

end

if I>=800&&I<=900&&B>=1400&&B<=1500

disp(9);

end

if I>=800&&I<=900&&B>=1600&&B<=1700

disp(C);

end

if I>=900&&I<=1000&&B>=1200&&B<=1300

disp('\*');

end

if I>=900&&I<=1000&&B>=1300&&B<=1400

disp(0);

end

if I>=900&&I<=1000&&B>=1400&&B<=1500

disp('#');

end

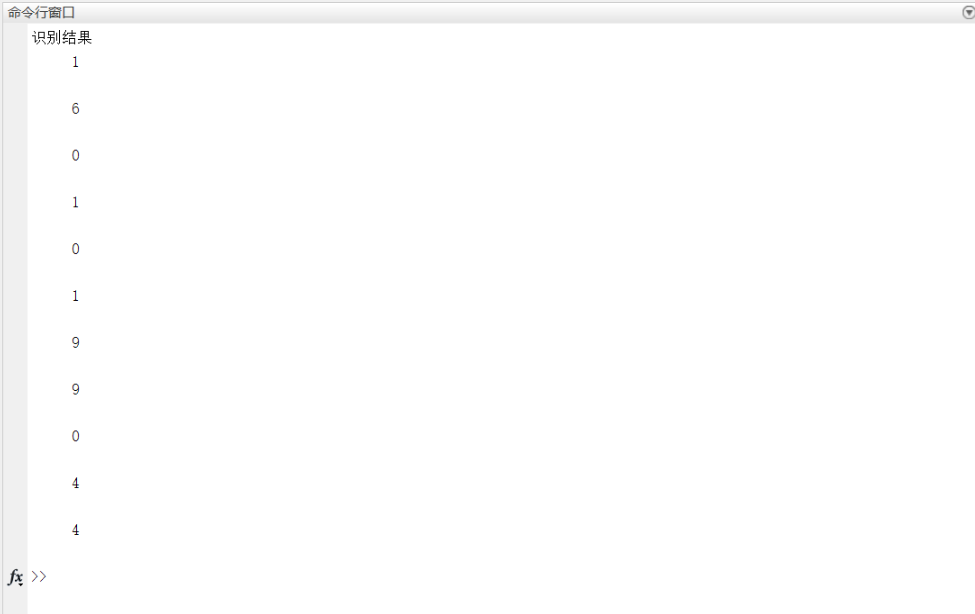
if I>=900&&I<=1000&&B>=1600&&B<=1700

disp(D);

end

end

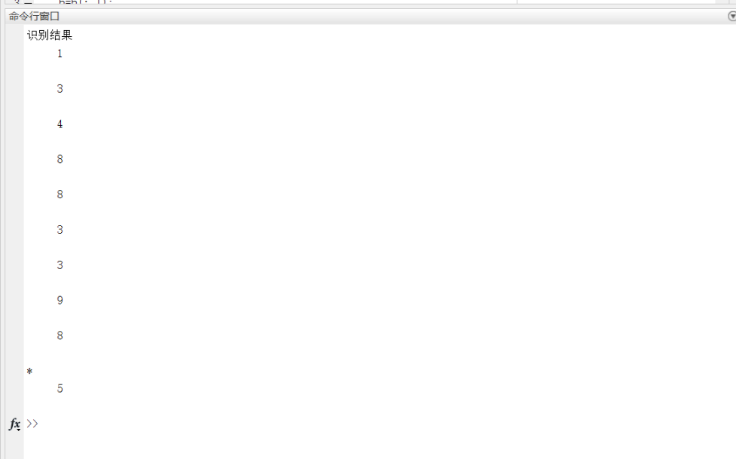
**判断结果**

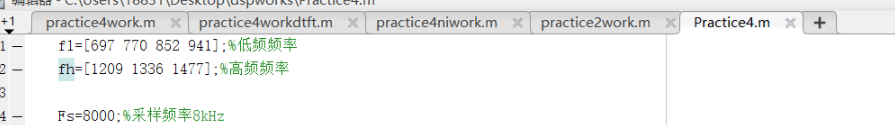
由于我产生dtmf的按键持续时间为0.5秒，中间无声音时间为0.5秒，所以代码中的keytime=0.5即按键持续时间为0.5秒，zerotime=0.5即中间为零的时间为0.5秒。

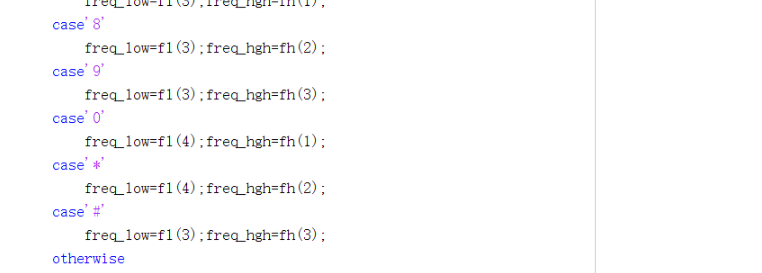
与本人学号相同。

老师的dtmf产生按键持续时间为0.5秒，中间无声音为0.125秒，所以代码中的keytime=0.5即按键持续时间为0.5秒，zerotime=1/8即中间为零的时间为1/8秒。

注意，识别结果中存在星号是因为老师在生成wav文件时即dtmf过程中将0和\*的频率弄混，这一点可以从例程中看出。



例程中的错误



与标准dtmf表对比可以发现\*和0的频率错误。