

信号处理综合作业题

-电话号码破译



组员：

精 81 杜列高 2018080048

精 82 张银龙 2018080056

精 83 周凯兴 2018080052

一、任务分析

根据一段手机按键的录音（音频文件 dialWav.wav），利用测试课程第 2 章所学的信号处理和频谱分析相关知识，基于 MATLAB 平台识别按键数字组合。具体要求：

***求该录音对应的按键数字组合。**

- 1) 得到音频中的所有按键及正确顺序，没有错位、重复。
- 2) 在 MATLAB 中分析完整的音频，不得人工将 wav 文件切分成多个。
- 3) 绘制识别出的各个按键的频谱图。
- 4) 提供完整、友好的程序交互界面。
- 5) 提供完整实现代码和实验报告。

根据完成程度进行评分，在以上要求基础上实现更完整的测量过程可加分，加分项包括但不限于：

- 1) 实时解析麦克风输入的音频；
- 2) 实现完整的测量过程，包括滤除噪声、短时频谱分析等。

二、背景理论

（1）按键的双音多频信号

电话按键声音遵循“双音多频信号”（Dual-Tone Multi-Frequency, DTMF）规则，即一个按键声中包含 2 个主要频率。具体规则参考表（1）

表（1） 双音多频信号表

低频/Hz	高频/Hz			
	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	x	0	#	D

（2）噪声滤除

按键音频信号的频率在 697-1633Hz，故可采用带通滤波器，对频率 $f > 1633\text{Hz}$ 及 $f < 697\text{Hz}$ 的信号进行滤除

（3）信号分帧

声音信号是随时间变化的非平稳随机过程，因此对于声音的分析一般都是短时分析。这是因为声音虽然是时变的但是具有短时相关性。声音的短时分析一般需要分帧加窗来处理。

（4）频谱分析

频谱分析是将信号分解成不同频率的分量的过程，在声学环境中有两种截然不同的方法，取决于结果是在具有恒定分辨率的线性频率标度上（以 Hz 为单位）还是在具有恒定百分比分辨率的对数频率标度上需要。时域和频域之间的基本联系，即傅里叶变换，最容易用线性时间和频率尺度来解释，尤其是现在用来计算它的实用版本 FFT（快速傅立叶变换）。

（5）数字识别

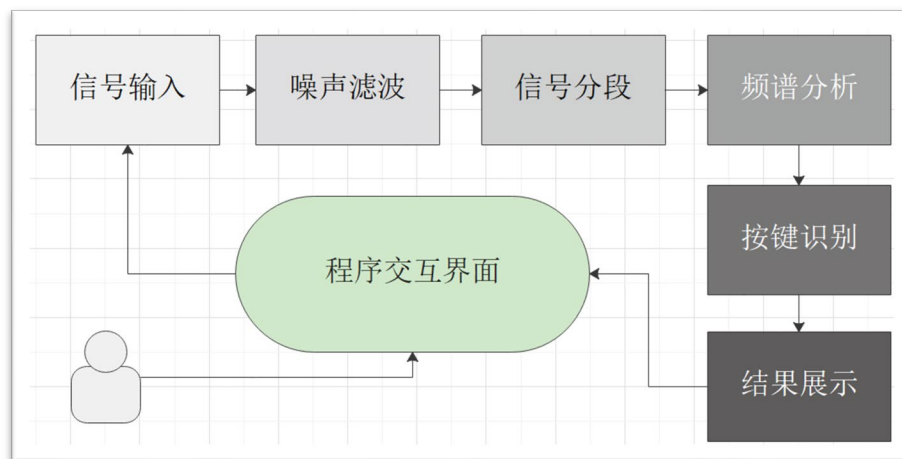
通过寻找幅频谱中的特征谱线，与双音多频表进行比对，确定按键音对应的数字。

（6）交互界面

为了设计友好的交互界面，我们选择利用 MATLAB 里的 GUI 设计。GUI 设计出现的时间很短，但是好的设计的通用性在它的到来之前没有改变。有一些关于设计的指导原则，但是总结它们的最好方法是集中于以下基本原则：简单性、一致性和熟悉性。在这个实验室中，有必要以有序的方式展示关键图形，这些图表给出了音频输入、处理和识别的一般概念。

三、系统设计

根据上述背景理论，我们设计了如下的破译系统



图（1） 系统设计流程图

四、系统原理

（1）噪声滤波设计

在滤波设计上，我们开始是先用了比较简单的方式将杂波滤除：我们观察了原始波形，发现杂波幅值均低于 0.016，因此我们直接将低于这一值的信号全置为零。

（2）信号分帧

通过滤波后的信号只有在按键处有幅值，而在其他部分幅值均为零。因此我们想着利用这一特点来分离信号。当信号从无幅值到有幅值的瞬间，我们记录这一时间点为按键信号开始的声音，当信号从有幅值到无幅值的瞬间，我们记录这一时间点为按键信号结束的声音。

（3）频谱分析

通过如下代码将每一段的按键音频信号转化为关键的频率，高频和低频。

```

TD_signal_leng = length(digi{ind});           %length of time domain sign
n2 = 2^nextpow2(TD_signal_leng);             % length of signal in power of two
tranx = fft(digi{ind},n2);                   %FFT
half_tranx = tranx(1:n2/2);                  %illiminate mirror part
freq_period = fs*(0:(n2/2)-1)/n2;            %create freq label
  
```

然后再将得到的 freq_period 与 half_tranx 绘图（前者为 Y 轴，后者为 X 轴），从而得到每一按键信号的频谱图。频谱图结果附在后面结果分析。

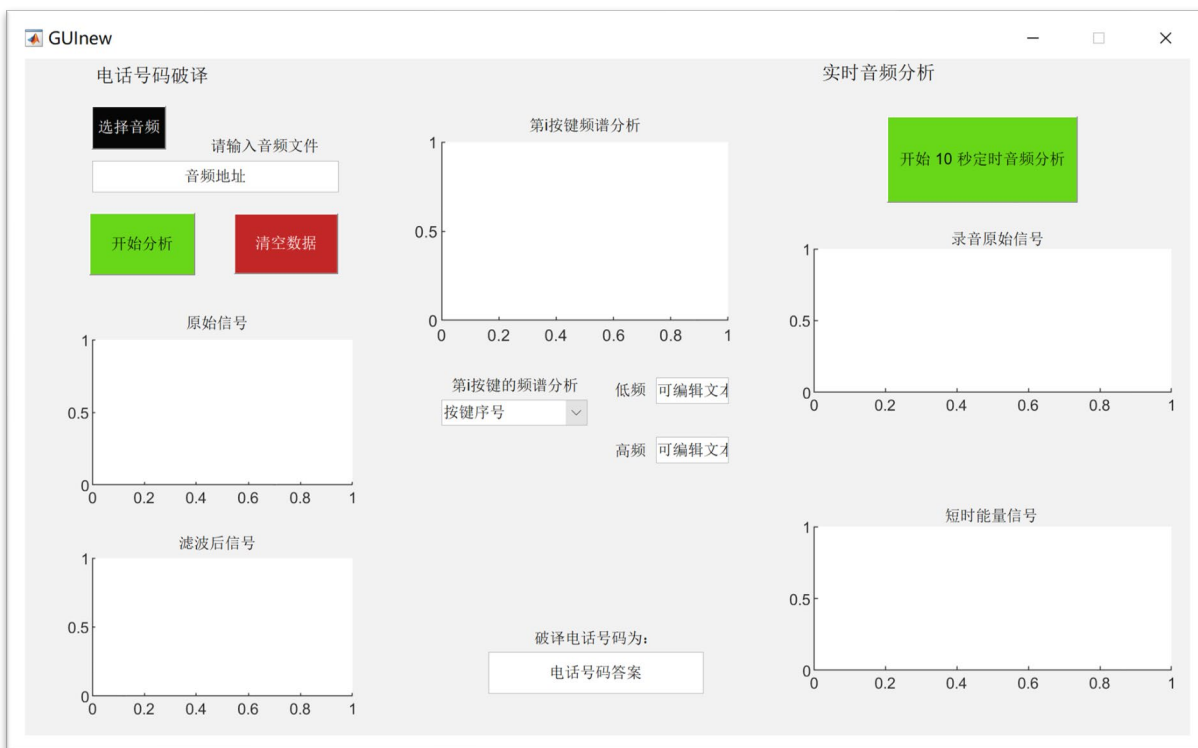
(4) 数字识别

现在拥有了每个按键信号的高频和低频信息，我们需要对应表（1）中的规则来确定按键音对应的数字。如下列程序所示，我们通过 for 循环和选择语句以及分离区间的方法来分类我们得到的双频音所对应的数字和号码。

```
for i = 1:length(input)
if input(1,i)<=700 && input(2,i)<=1260
    q(i) = '1';
elseif input(1,i)<700 && input(2,i)>1260 && input(2,i)<1390
    q(i) = '2';
elseif input(1,i)<700 && input(2,i)<1530 && input(2,i)>1390
    q(i) = '3';
elseif input(1,i)<700 && input(2,i)>1530
    q(i) = 'A';
elseif input(1,i)>700 && input(1,i)<800 && input(2,i)<1260
    q(i) = '4';
elseif input(1,i)>700 && input(1,i)<800 && input(2,i)>1260 && input(2,i)<1390
    q(i) = '5';
elseif input(1,i)>700 && input(1,i)<800 && input(2,i)<1530 && input(2,i)>1390
    q(i) = '6';
elseif input(1,i)>700 && input(1,i)<800 && input(2,i)>1530
    q(i) = 'B';
elseif input(1,i)<900 && input(1,i)>800 && input(2,i)<1260
    q(i) = '7';
elseif input(1,i)<900 && input(1,i)>800 && input(2,i)>1260 && input(2,i)<1390
    q(i) = '8';
elseif input(1,i)<900 && input(1,i)>800 && input(2,i)<1530 && input(2,i)>1390
    q(i) = '9';
elseif input(1,i)<900 && input(1,i)>800 && input(2,i)>1530
    q(i) = 'C';
elseif input(1,i)>900 && input(2,i)<1260
    q(i) = '*';
elseif input(1,i)>900 && input(2,i)>1260 && input(2,i)<1390
    q(i) = '0';
elseif input(1,i)>900 && input(2,i)<1530 && input(2,i)>1390
    q(i) = '#';
elseif input(1,i)>900 && input(2,i)>1530
    q(i) = 'D';
```

(5) 交互界面

GUI 交互界面的设计如下图所示



图（2） GUI 界面设计

GUI 交互界面主要由两个部分组成：电话号码破译部分（左中）和实时音频分析部分（右）。

电话号码破一部分，我们将最重要的几个环节的图像显示出来了，分别是原始信号、滤波后信号、以及每一个按键的频谱图。设计频谱图展示时，考虑到一共有十个频谱图，若将每个频谱图都画在界面上，空间会比较拥挤。因此我们选择了利用下拉框来分别显示十个频谱图，这样也可以把频谱图的大小放大许多。

界面的左上端，我们设计了导入音频的按钮和输入音频文件地址的可编辑文本，方便用户选择导入的音频文件。然后有两个按钮：开始分析和清空数据。点击开始分析后，程序将会读取导入的音频文件，并开始执行 MainRun 函数，对音频进行滤波、分帧、频谱分析、数字识别，最后再将识别的答案显示在中下端的答案窗口。结束分析后可以点击清空数据来清空文本和图形的数据。

实时音频分析的一部分，我们采用 10 秒定时的方式进行能量分析。点击绿色按钮“开始十秒定时音频分析”后，程序开始录音，并且实时记录下原始信号和短时能量图。如果录下的信号有多音双频信号，那么答案框将会输出这一信号所对应的数字后字母，然后自动停止录音。当然，前面的清空数据在这里也有效。

(6) 实时信号分析

我们设计了一个实时音频信号分析系统。按下按绿色按键，开始录音并且同时画出音频的图像，每 0.01 秒更新图像，持续 10 秒自动结束。在此期间音频会经过滤波、短时能量分析，并且若信号是电话号码的双音多频信号，会在文字框输出所对应的号码。

主要运行函数为 `[num] = LiveRun(y,fs,axes2)`。其原理与上面的 `MainRun` 函数相似。区别在于我们运用的更严格的滤波系统以及加了个短时能量分析。

我们采用了 Matlab 中的 `filter` 函数和 `butter` 函数。`butter` 函数是求 Butterworth 数字滤波器的系数，在求出系数后对信号进行滤波时用 `filter` 函数。由于按键音频信号的频率在 697-1633Hz，我们采用带通滤波器：

$$[B,A] = \text{BUTTER}(N,Wn)$$

其中 N 是滤波器的阶数。 Wn 的确定跟你的采样频率 F_s 有关。得到滤波系数之后，就可以直接带入 `filter` 函数：

$$\text{signal_1} = \text{filter}(B,A,x)$$

从而得到滤波后的信号。

然后我们接着分析信号的短时能量：

$$\text{framesignal} = w(n) * \text{signal}((i-1) * \text{framelength} + 1 : i * \text{framelength}), \\ 1 \leq i \leq \text{framenumbers}$$

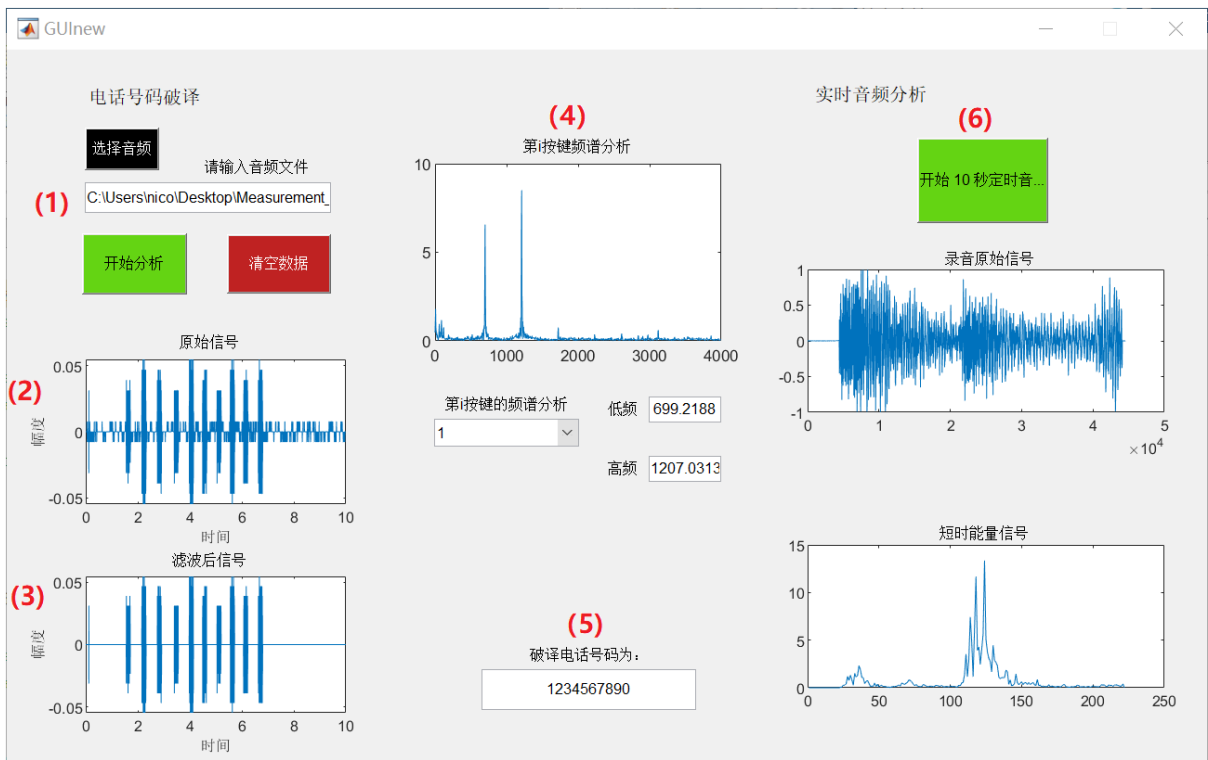
式中， $w(n)$ 为窗函数，本次实验我们采用汉明窗； framelength 为帧长； framenumbers 为分帧后的总帧数。

计算第 i 帧信号 `framesignal` 的短时能量公式为

$$E(i) = E(i) + \text{framesignal}(j)^2, \quad 1 \leq j \leq \text{framelength}$$

五、 结果分析

项目代码完成后，程序界面具有以下格式：



图（3） 运行 GUI 程序后的结果图

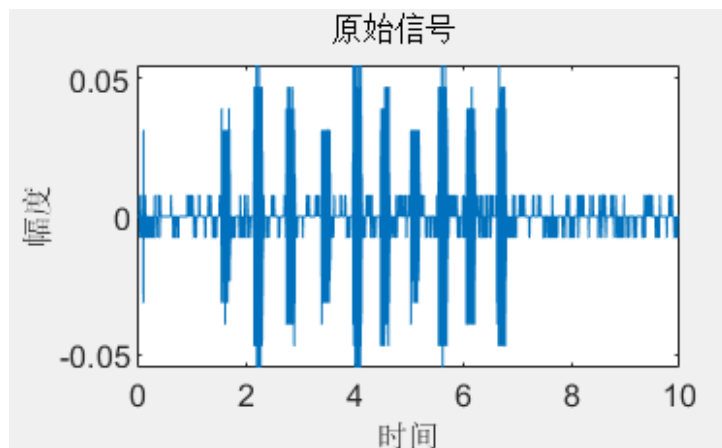
1) 输入音频文件与控制端：

此部分允许用户选择任何.wav 或.m4a 文件来处理它。显示的结果可以清除，以便处理其他文件。

2) 原始信号图：

第一张图显示了声音文件的快速傅立叶变换。y 轴表示信号的幅度，x 轴表示其时间。在这种情况下，要显示的第一幅度最有可能是干扰或录音设备启动时的声音。

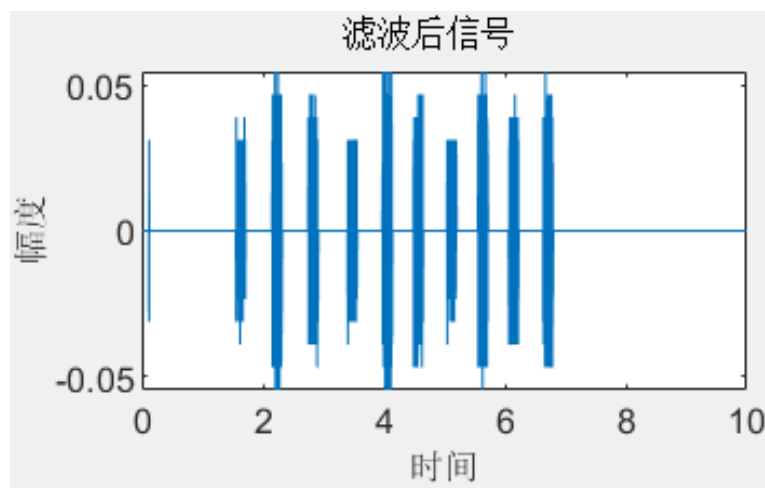
下面显示的十个大幅度表示给定文件中每个数字的声音，图中较小的幅度表示背景干扰。



图（4） 原始信号图

3) 滤波后信号图:

滤波后信号的图型已经没有底噪了，可以看到，整个图像只剩下按键的信号。

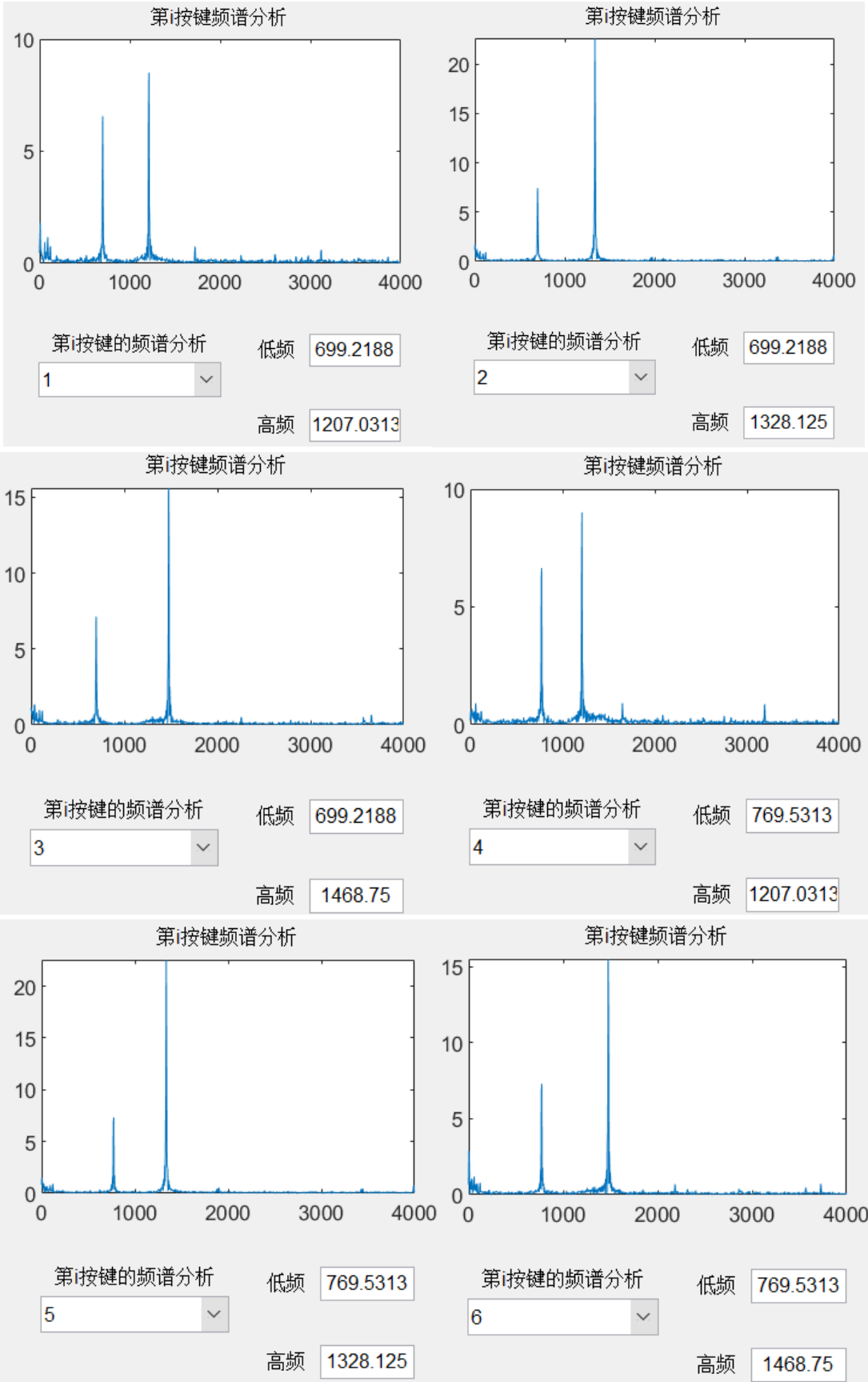


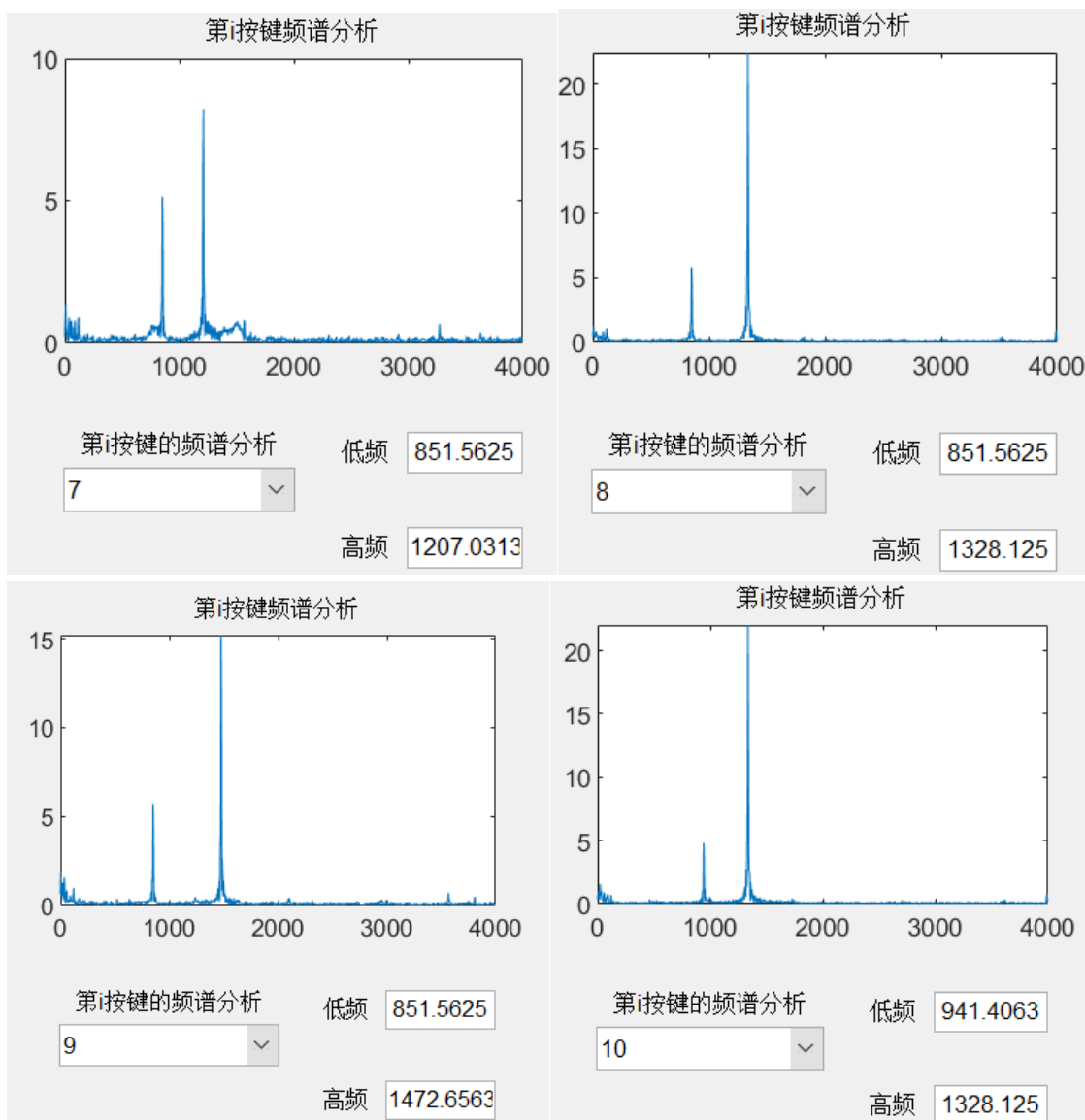
图（5） 原始信号经过初步滤波后的图

4) 频谱分析:

每一个按键信号的频谱图都有很明显的两个关键频率。下图所示图像的横坐标为频率，单位为 Hz。中坐标为幅值。

列表的顺序（图（6）从 1 到 10）表示按键信号的顺序（图（5）从左到右）。





图（6） 每个按键信号的频谱图

5) 破译电话号码:

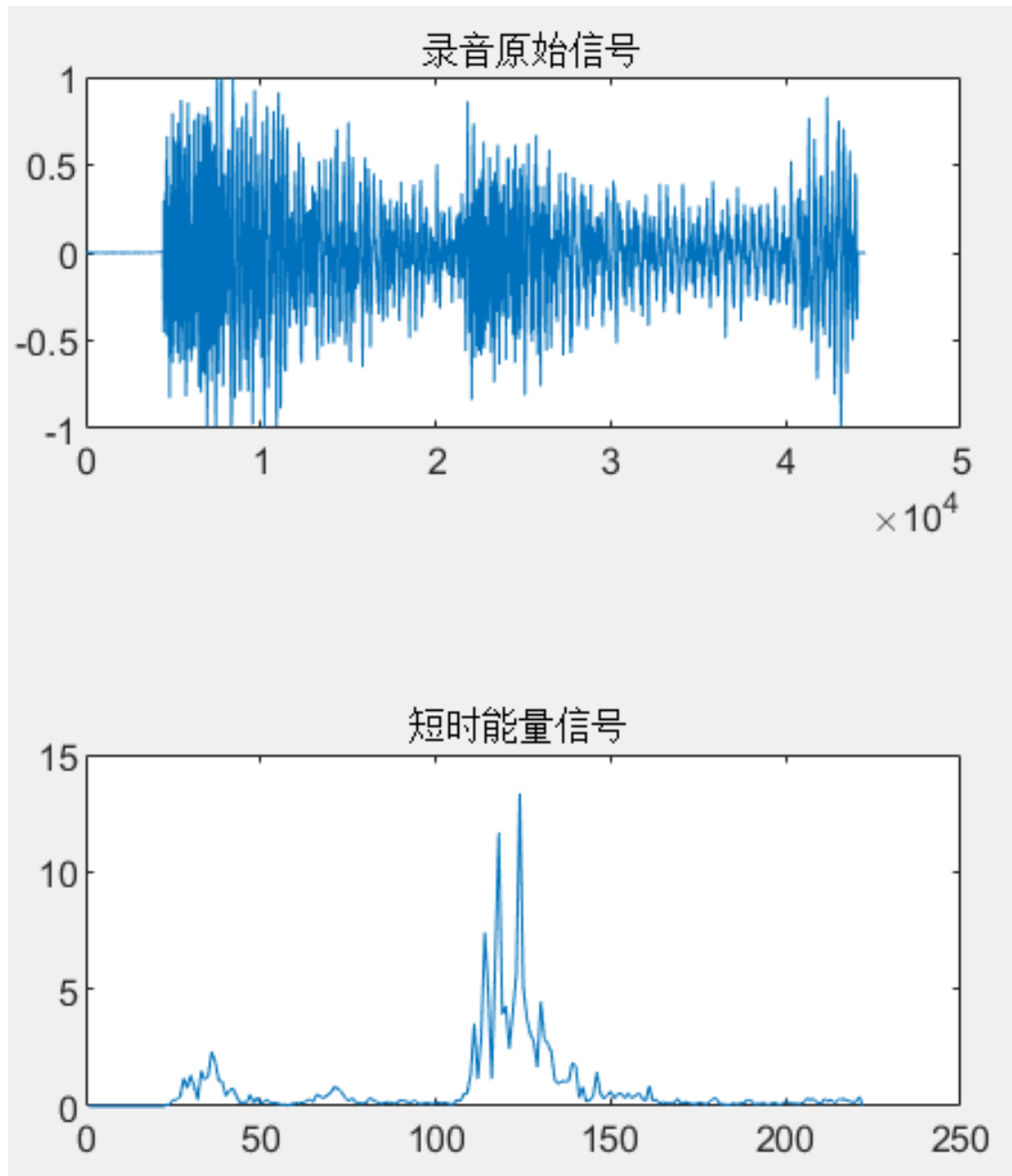
最后，利用 `tonumber.m` 文件中的条件函数，将频谱分析得到的低频和高频信号与双频信号表进行比较。于是显示最终结果

破译电话号码为：
1234567890

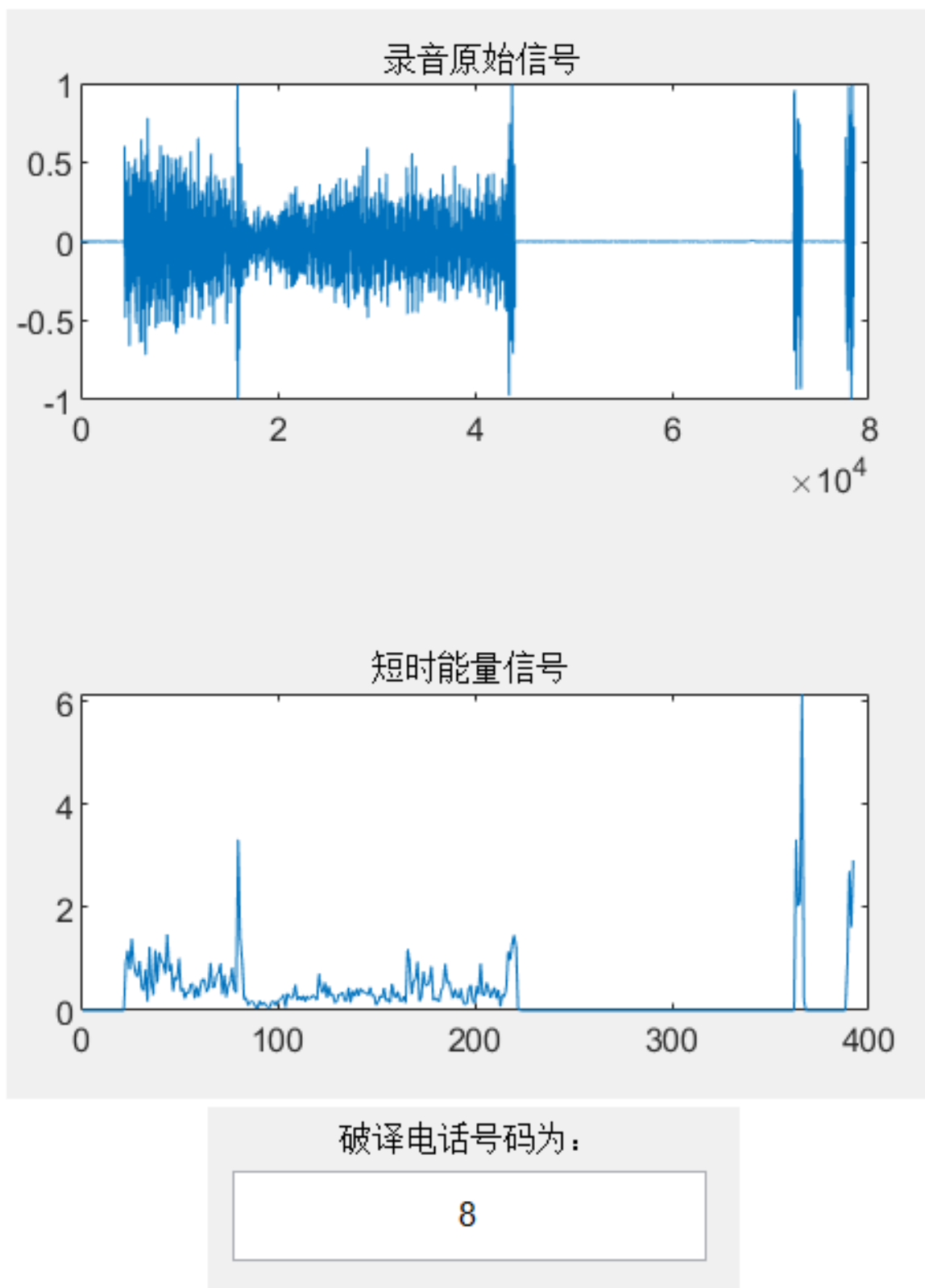
图（7） 破译号码结果显示

6) 实时音频分析:

实时频谱分析部分能够记录 10 秒的现场音频，计算信号的快速傅立叶变换，并在答案框中显示电话数字（如果存在的话）。由于现场录音处理的声音种类更多，所以使用更一般的滤波形式（如 Dfilter.m 所示）。



图（8） 讲话时实时音频原信号与能量分析



图（9） 按电话按键“8”的时候的能量图像与破译结果显示

六、项目总结

通过本次综合作业，我们最大的收获就是对 FFT 的应用以及 MATLAB 的使用更加熟悉了。尤其是 MATLAB 的 GUI 设计的功能，我们之前从来没有接触过。我们发现我们需要设计交互界面，因此恶补了 GUI 设计的知识。当然不只是 GUI 的设计，还有过滤和短时分析都是查询网上资料，通过学习再创作更新的。我们也认识到，FFT 是多么强大的功能，以及他可以在如此有趣的领域里应用到。值得一提的是，创造力和逻辑思维对于解决现实生活中的场景来说，与理解傅立叶级数和傅立叶变换背后的原理的能力同等重要。与队友讨论、合作的这种群组模式有助于改善学习过程和思维方式，互相提意见、建议和相互补充知识是非常高效的过程。最后，感谢老师和助教的耐心指导，使我们获得如此大的收获。

七、分工明细

张银龙：编写设计信号滤波、分帧、数字识别代码，合作撰写报告

周凯兴：编写设计 GUI 界面，短时能量分析，改善代码，合作撰写报告。

杜烈高：编写实时音频分析，检查代码，研究理论，合作撰写报告。