信号处理原理第八次作业

黄家晖 2014011330

1 DTMF 按键识别

1.1 理论依据

根据参考资料 [1] 中的讲述，在识别 DTMF 时，最优的时域序列长度为 205，经过了如此 FFT 变换之后，既满足可分辨条件和采样定理，又能保证离散频域中的与准确值的误差比较小。

对于 DTMF 所需识别某一频率 f，其对应的 DFT 频域下标为:

$$k = 205 \times \frac{f}{f_s}$$

其中真实选取的整数 k 值应最接近等式右面的值。

因此无论是 fft 算法还是 Goertzel 算法，只需要检测相应下标的 DFT 的模是否大于一定阈值即可，如果不考虑识别速度，也可以计算最大值，从而提高准确率。

1.2 实验描述

实验中录制了一份 8000 采样率、长度约 2 秒的 DTMF 电话录音，保存在 ring.wav 文件中供读取。该电话录音中是电话机从 0 至 D 总共 16 个按键依次的拨号序列。

识别过程中，两种算法选择的阈值均为 10。

1.3 实验结果

实验中两种算法均能以 100% 的正确率识别所有电话按键录音。程序输出如下:

```
Recog Result = 1, Real Result = 1
Recog Result = 2, Real Result = 2
Recog Result = 3, Real Result = 3
Recog Result = 4, Real Result = 4
```

```
Recog Result = 5, Real Result = 5
Recog Result = 6, Real Result = 6
Recog Result = 7, Real Result = 7
Recog Result = 8, Real Result = 8
Recog Result = 9, Real Result = 9
Recog Result = 0, Real Result = 0
Recog Result = *, Real Result = *
Recog Result = #, Real Result = #
Recog Result = A, Real Result = A
Recog Result = B, Real Result = B
Recog Result = C, Real Result = C
Recog Result = D, Real Result = D
```

令二者总共识别 1000 次 16 个按键，测量五次取平均值。其中，FFT 花费时间为 0.88 秒，Goertzel 算法所花费时间为 0.76 秒，由于 Goertzel 算法仅计算相应下标的 DFT 取值，因此效率更高一些。

2 卷积计算方法比较

2.1 实验描述

实验中实现了公式法、普通 FFT 法、Overlap-Add 法和 Overlap-Save 法。在分析计算效率时，x 序列和 h 序列均采用随机产生的向量作为输入，控制变量得到实验结果。

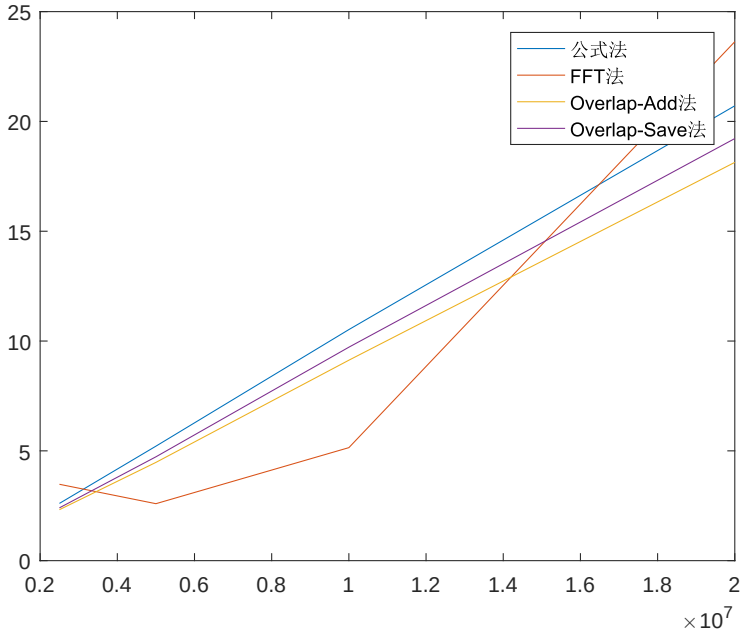
2.2 实验结果

程序输出如下:

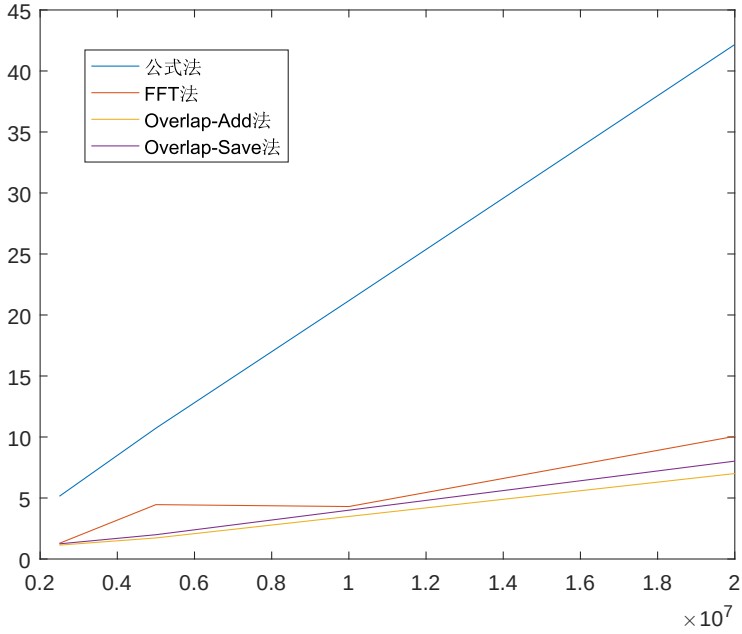
```
+ Now testing sequence of length 5000000.
        Naive Convolution takes 10.722000 secs.
        FFT Convolution takes 4.456000 secs.
        Overlap Add Convolution takes 1.727000 secs.
        Overlap Save Convolution takes 1.991000 secs.
+ Now testing sequence of length 10000000.
        Naive Convolution takes 21.172000 secs.
        FFT Convolution takes 4.301000 secs.
        Overlap Add Convolution takes 3.487000 secs.
        Overlap Save Convolution takes 4.000000 secs.
......
```

以下图中横轴均为序列长度，纵轴为时间（单位：秒)。

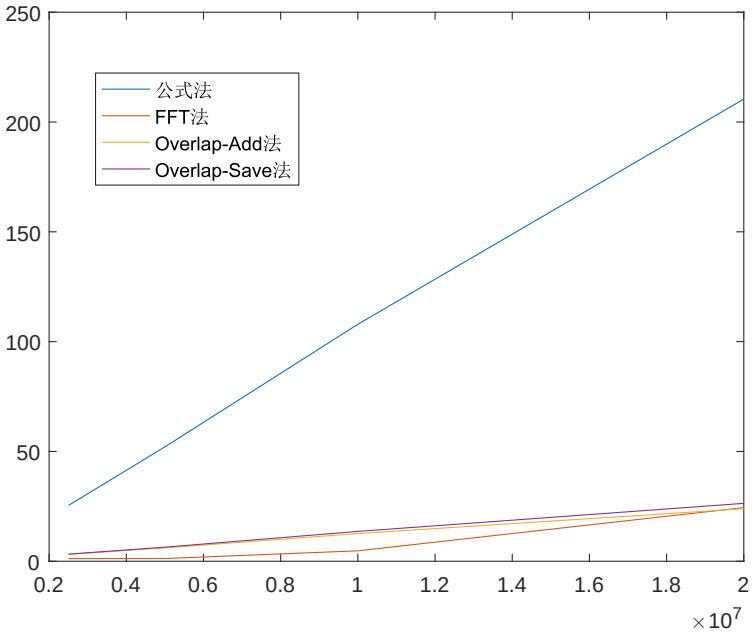
当卷积核的大小（h 序列）为 50 时，四种方法效果比较：



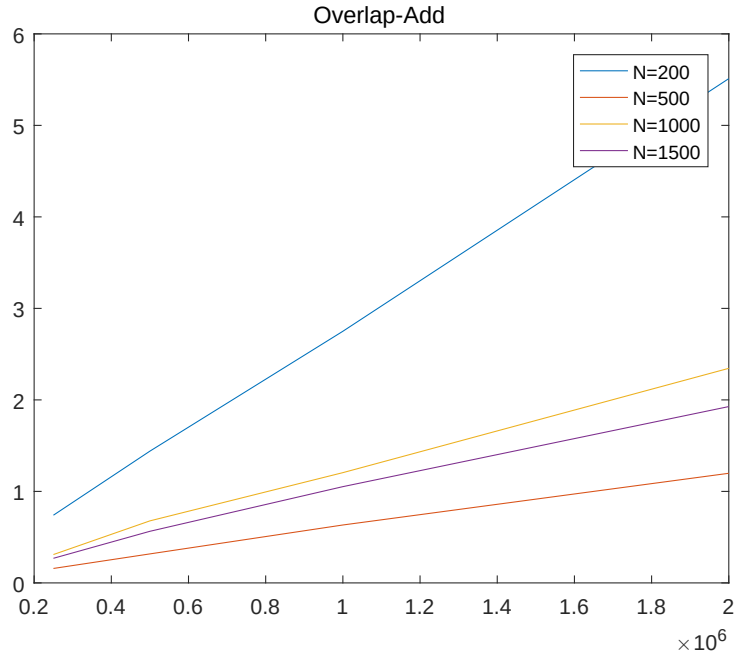
当卷积核的大小为 100 时，四种方法的比较：

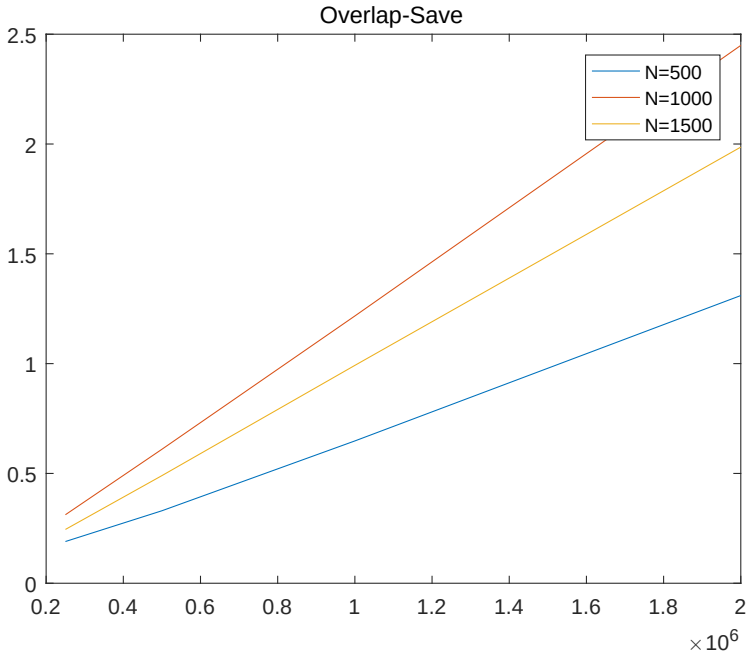


当卷积核的大小为 500 时，四种方法的比较：



变化 Overlap 系列方法的 block 大小，进行计算效率对比，卷积核大小为 500：





可以看出使用公式法计算的卷积复杂度随着卷积核的大小线性增长（实际复杂度为平方，这里只改变一次项），且花费时间相比 FFT 法和 Overlap 系列的方法在卷积核和序列长度很大的情况下要长得多。

而对比 FFT 法和 Overlap 法，其效率取决于所选择的 block 大小与卷积核大小的相对关系，如果 block 大小选取恰当，Overlap 算法的性能要好于单纯的 FFT 方法，且由于其可以实时计算卷积结果，更多被人们采用。

而 Overlap-Save 和 Overlap-Add 方法原理相近，计算复杂度也相近，因此计算时间相差不多。

3 语音信号的频分复用

3.1 实验描述

本实验的所有代码均在 main 中实现，包括编码和解码的过程。

实验中采用了 3 段 wav 音频，包含了一段音乐（音调不超过 4000Hz）、一段英文对话和一段中文对话，时长为 4 秒左右，为了方便均截取其前三秒的部分。

编码阶段，首先将音频做变换到数字频域，并将三段数字频域分别排在 0-4000Hz, 4000-8000Hz, 8000-12000Hz 的频带范围内进行传输，最终将传输的离散时域信号保存在 mixed.wav 中，新音频的采样率为 24000Hz。

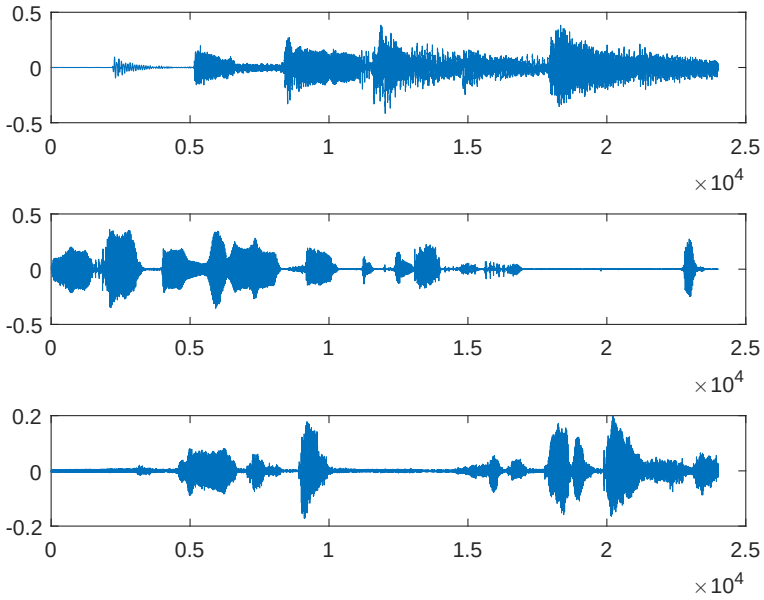
解码阶段，程序读入 mixed.wav 文件，将频域中的上述分量分离重组，即得到解码结果。

3.2 实验结果

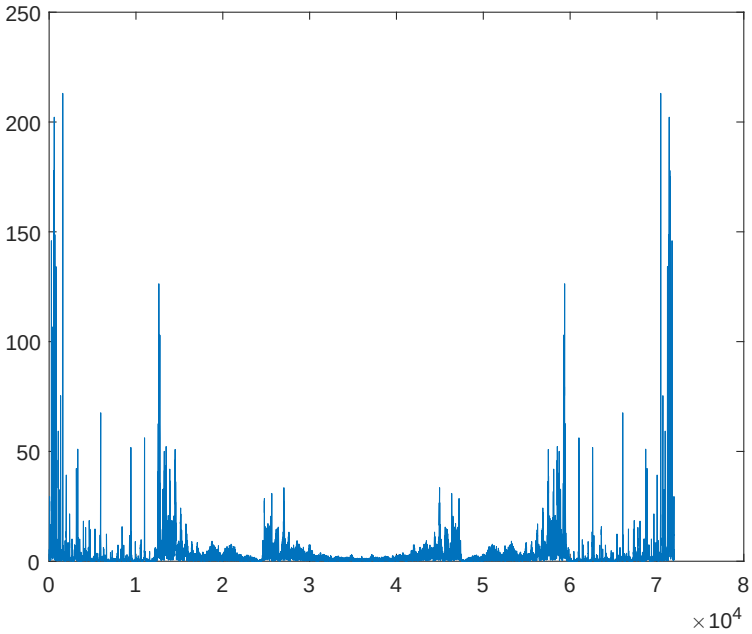
程序输出如下：

```
Loading and modulating audio files...
Audio transmitted (in mixed.wav)
Demodulate finished.
```

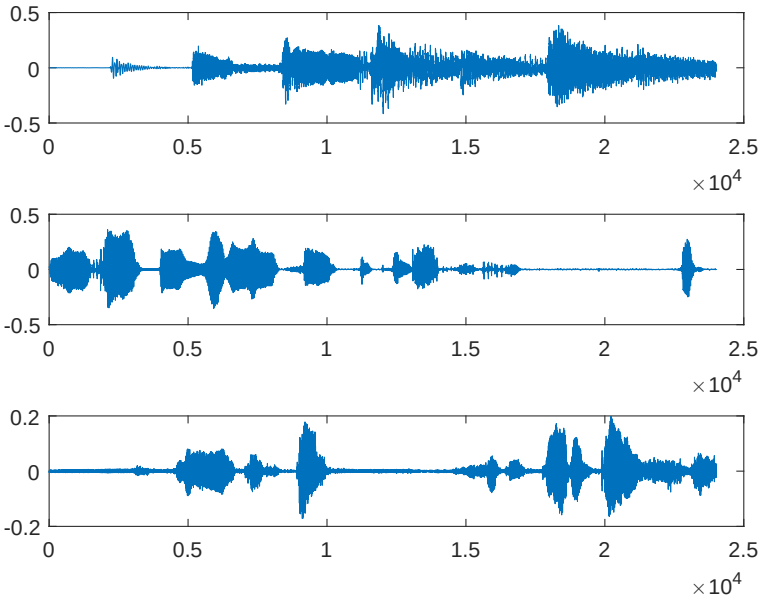
输入三个信号的时域谱如下：



经过编码之后传输的频域谱（取模之后）如下：



经过重新拼合与 IFFT 变换，还原之后的时域谱如下：



可以发现，编码与解码之后的振幅基本吻合，试听之后也基本和原有音频一致。

4 参考代码和资料

本实验中所有参考的代码和资料已经在 matlab 程序中说明，现在重新列出：

1. Goertzel 算法原理和实现参考：

   http://download.csdn.net/detail/skeletonwang/2913563

2. Overlap-Save 算法实现参考：

   https://cn.mathworks.com/matlabcentral/fileexchange/41337-overlap-save-method/content/Overlap_Save_Method.m