武汉大学教学实验报告

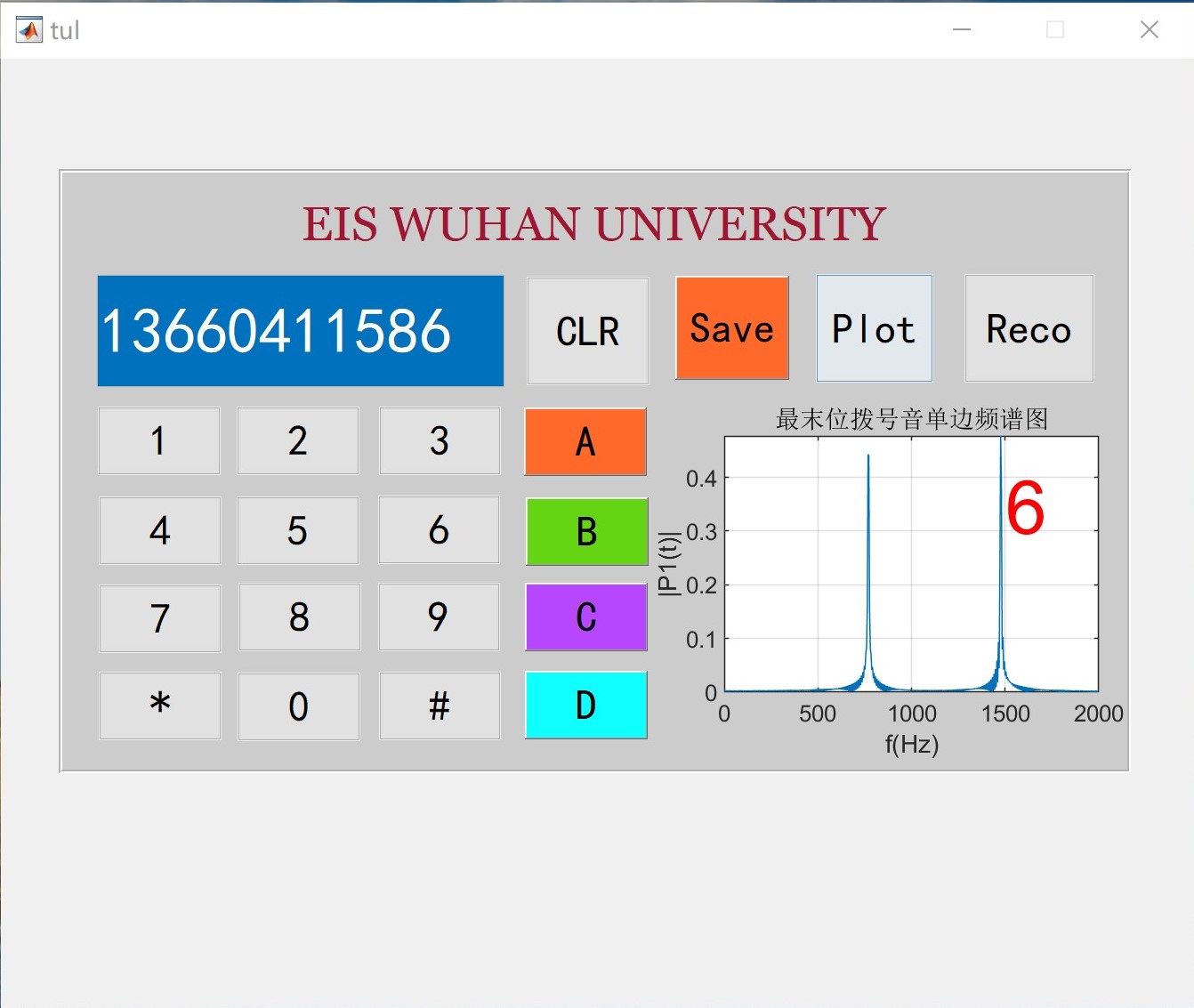
电子信息学院 电子信息工程 专业 2019 年 9 月 29 日

实验名称 电话拨号音的合成与识别 指导教师 黄根春

姓名 李昊 年级 2017级 学号 2017301200060 成绩

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 实验目的   1. 基于电话通信系统中使用的双音多频（DTMF）技术，利用MATLAB软件以及FFT算法，实现一个能够合成拨号音、识别拨号音的程序； 2. 进一步地，利用MATLAB中的图形用户界面GUI制作简单直观的模拟界面，并加入号码存储等家用电话中实用的功能，从而对生活中的电话拨号音系统进行简单的实验仿真。   2 实验原理  双音多频DTMF（Dual Tone Multi-Frequency）信号，是用两个特定的单音频率信号的组合来代表数字或功能。在DTMF电话机中有16个按键，其中10个数字键0 - 9 ，6个功能键\*、#、A、B、C、D。其中12个按键是我们比较熟悉的按键，另外由第4列确定的按键作为功能键留为今后他用。  根据 CCITT 建议，国际上采用697Hz、770Hz、852Hz、94lHz低频群及1209Hz、1336Hz、1477Hz、1633Hz高频群。从低频群和高频群任意各抽出一种频率进行组合，共有16种组合，代表16种不同的数字键或功能键，每个按键唯一地由一组行频和列频组成，如表2.1所示。  表**2.1 DTMF频率按键对应表**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | (Hz)  (Hz) | 1209 | 1336 | 1477 | 1633 | | 697 | 1 | 2 | 3 | A | | 770 | 4 | 5 | 6 | B | | 852 | 7 | 8 | 9 | C | | 941 | \* | 0 | # | D |   3 实验内容  使用MATLAB的GUI工具，fft()函数等，设计一个模拟电话拨号音系统，要求实现以下功能：   1. 实现电话拨号盘，按下按键时响起该键对应的DTMF信号； 2. 实现电话号码显示窗，能够显示当前输入的号码； 3. 将输入各号码的DTMF信号首尾连接，形成完整拨号音； 4. 实现拨号音识别功能，读入一段拨号音，能够识别出对应的号码； 5. 实现频谱图显示功能，处理一段DTMF信号，产生其归一化的单边频谱图； 6. 参考家用电话，实现其他的功能，如号码存储等。   4 实验操作过程  4.1 图形电话拨号面板的制作  根据需要实现的功能，使用GUI设计工具制作图形界面，保存为tul.fig，界面说明详见附录A.1的说明书。  4.2 拨号音的产生与合成  现在对上节制作的图形界面上的各控件的动作和变化进行设置，即对 tul.m 文件进行编辑。其主要的功能是使对应的按键，按照表2.1的对应关系产生相应的拨号音，完成对应行频及列频的叠加输出。此外，出于图形界面的需要，还要使按键的号码数字显示在拨号显示窗口中。  在MATLAB中，可用sin()函数直接产生一定频率的信号，而DTMF信号可以使用两个不同频率的正弦波信号叠加得到，如式（4.1）所示，其中，是从低频群，高频群中选取的某个频率，是音频抽样频率。  **（4.1）**  鉴于CCITT对DTMF信号规定的指标，每个数字信号取1000个采样点模拟按键信号，并且每两个数字之间用100个0来模拟静音，以便区别连续的两个按键信号。另外，音频抽样频率取8192Hz。  对于保留的两个功能键\*、#，按照现行键盘式拨号电话的习惯，将\*键作为删除键，#键作为确认键。删除键的作用是将前面拨错的号码删除退回，表现为将显示窗口已经显示的错误号码退回一位数字，并且将连续拨号音信号的存储单元 NUM 中退回一位拨号音信号和静音信号。删除可以进行连续的操作。确认键的作用是将之前输入的拨号音进行保存，以用于后续的拨号音识别部分。  详细的代码实现见附录A.2。  4.3 拨号音的检测识别  接收到的拨号音为若干个连续的DTMF信号，首先需要取出单个号码对应的DTMF信号。根据4.2节中对信号长度的规定，每1100个点取前1000个点，便可取出单个号码的DTMF信号。  每个号码的DTMF信号由两个不同频率的正弦波叠加而成，它理论上的傅里叶变换只含有两个不同频点的冲激。因此，可以在傅里叶变换序列中定位到低频群和高频群的频率区间，寻找区间中的峰值，然后再寻找峰值最接近的频点，即可恢复出与。然后通过查表的方式寻找到频率确定的号码。  由表2.1的频率数据，低频群和高频群间有250Hz左右的频差，低频、高频群内各频率的间隔在70–150Hz之间。在进行傅里叶变换的时候，为了减少计算量的同时保证频率之间可以分辨，需要取合适的点数进行变换。本实验取N = 2048，此时频谱分辨率（变换序列中两点间距代表的频差）可用（4.2）计算，变换序列上任意点对应的离散频率可用式（4.3）计算。  (Hz) **（4.2）**  **（4.3）**  详细的代码实现见附录A.3。  4.4 绘制单个号码的DTMF信号频谱  在上一节进行傅里叶变换的基础上，取单边频谱进行归一化，确定频谱分辨率后，即可绘制出单个号码的DTMF信号频谱。详细的代码实现见附录A.4。  4.5 程序的其他功能  本程序还有其他的一些功能，如清除全部输入，存储任意长度号码等，由于这些功能不涉及信号处理的知识，故不做详细介绍。具体实现的功能见附录A.1的说明书。  5 实验数据与结果  本实验主要的成果便是tul.m这个GUI程序，该包含了实验要求的所有功能，限于篇幅不能一一展示。下面的内容是程序部分主要功能的示意图，其他的功能可以参考附录A.1的说明书了解，或是操作实际的程序体验。  5.1 单个数字的DTMF信号频谱图  图**5.1** 数字“0”对应的DTMF信号单边频谱图  图5.1是数字“0”对应的DTMF信号单边频谱图，可见频谱呈现清晰的两个峰，峰值分别出现在950Hz与1350Hz附近，符合数字“0”规定的频率（941Hz，1336Hz）。其他按键产生的信号频谱图也类似，区别在于峰值出现的位置。无限长正弦信号的理论频谱是一个冲激函数，这里的频谱有一定宽度，原因是DTMF信号是有限长的正弦序列，而且在进行2048点傅里叶变换时，在原有信号序列后补零这一操作也一定程度影响了频谱，不过不影响峰值的辨认。  5.2 号码存储与来电识别  程序利用DTMF信号中保留的4个字母功能键A、B、C、D，设置了存储号码的功能，处于保存模式下时按下功能键即可保存当前显示在号码输入框中的号码，同时保存对应的拨号音；保存模式之外，按下功能键可以调出之前保存的号码，拨号音也会恢复，可以继续编辑，正常拨号；程序中设置了“Reco”按键用于来电识别，按键调用detect.m，detect.m从文件“N0.wav”中读取拨号音，按照4.3节的方法还原出号码，将号码显示在程序的号码输入框中，这段拨号音也会覆盖程序原来的拨号音。具体的使用效果可以参考附录A.1的说明书，或是操作实际的程序体验。  6 实验效果分析与总结  本实验是一个综合性实验，综合了此前三次实验实践过的FFT、GUI设计、频谱图绘制等内容，同时也类似于一个课程设计，需要我们手动去拓展程序的某些功能，如号码存储，来电识别等，从而设计出一个比较完善的，能够模拟生活中使用的电话的电话拨号音合成与识别系统。在实验的过程中，我首先着眼于总体需要实现的功能，规划GUI上的元素和各自摆放的位置；之后参考实验指导书和网上的教程，将拨号音合成、修改号码输入框、保存音频、识别拨号音等各模块的代码放在对应的回调函数中，形成完整的程序。  这次的实验还是一个和生活应用紧密联系的实验，我不仅学到了拨号时的音频是如何合成的，也对信号处理在生活中的运用有了更深刻的理解。 |
| 教师评语 |
| 指导教师 年 月 日 |

附录

A.1 本实验设计的拨号音合成、识别程序使用说明书

图**A.1** 程序界面

1. 基本拨号盘功能

左侧的4\*3数字键盘，连同蓝色号码显示框以及显示框旁边的“CLR”键都属于基本的拨号盘，使用方法与电话、手机的拨号盘一致。4\*3的键盘中，按数字键时会产生对应频率的DTMF信号，DTMF信号播放一遍后插入到拨号音的末尾，同时显示的号码末尾出现按下的数字；按“\*”键和“#”键时也会响起对应的DTMF音频，但不会插入到拨号音中：“\*”键的功能是删除输入号码中最末一位，同时也会删除拨号音的对应部分，而“#”键的功能是确认号码，按下后完整的拨号音会响起，随后该拨号音会保存在“dialtone.wav”文件中，并且程序中的号码将会清空。“CLR”键的功能是清空输入的号码和拨号音。

1. 号码存储功能

本程序提供A，B，C，D四个存储位置，对应键盘上的四个特殊功能键“A”，“B”，“C”，“D”，支持存储任意位数的号码。要存储号码，首先需要按下上方的“Save”开关，开关按键按下表示进入保存模式，可以存储号码；之后选择四个功能键中的任何一个，当前显示的号码，连同对应的拨号音会保存到本地，如选择“A”键，号码和拨号音会存储到“numberA.mat”文件中，选择其他键同理。注意保存时会覆盖原来存放在该位置的号码。要调出存储的号码，首先要退出保存模式，将“Save”开关关闭，表现为开关按键弹起；之后选择想要调出的号码保存的位置对应的功能键，按下按键，保存的号码便会恢复到号码显示框中，拨号音也会一并恢复。“A”，“B”，“C”，“D”四个功能键有对应的DTMF信号，按下按键时也会有DTMF音频响起。

1. 频谱显示功能

程序窗体上方的“Plot”按键，功能是显示当前拨号号码最后一位的DTMF信号的频谱图，程序会截取当前拨号音的最末1100个点，取出最后一位的DTMF信号并进行FFT，具体原理详见4.3节。

1. 拨号音识别功能

程序窗体上方的“Reco”按键，功能是识别本地wav格式的拨号音，显示在号码显示框中，并用文件中的拨号音替换程序原有的拨号音，保证号码与拨号音一致。目前，暂时只能在“detect.m”中手动更改路径来选择识别的文件，默认识别的文件是“N0.wav”，之后的版本会加入在图形界面中选择文件的功能。

A.2 拨号音的产生与合成 核心部分实现

下面的程序段是按键“0”的回调函数，实现了数字0对应的拨号音的产生于合成，其他按键的拨号音与此类似。

% --- Executes on button press in button0.

% 拨号盘数字0，在电话号码显示区域加入数字，产生对应的DTMF信号并发声

function button0\_Callback(hObject, eventdata, handles)

Fs = 8192;

Ts = 1 / Fs;

t = (1:1000) \* Ts;

lf = [697 770 852 941];

hf = [1209 1336 1477 1633];

n = strcat(get(handles.edit1, 'string'), '0');

set(handles.edit1, 'string', n);

d = sin(2\*pi\*lf(4)\*t) + sin(2\*pi\*hf(2)\*t);

space = zeros(1, 100);

global NUM;

phone = [NUM, d];

NUM = [phone, space];

sound(d, Fs);

A.3 拨号音的检测识别 核心部分实现

下面的程序代码是“detect.m”文件的内容，主程序中按键“Reco”的功能便是调用该函数实现。

function [number, audio] = detect()

N = 2048;

[audio, Fs] = audioread('N0.wav'); % 默认读取N0.wav文件，也可手动设置识别的目标音频

sound(audio, Fs);

Ts = 1 / Fs;

t = (1:1000) \* Ts;

L = length(audio);

l = L / 1100;

number = '';

for i = 1:l

j = (i - 1) \* 1100 + 1; % 每1100点取前1000个点，对应的是每个号码的DTMF信号

d = audio(j:j+999);

f = fft(d, N);

f = abs(f / N); % 频谱归一化

num(1) = 170 + find(f(170:238) == max(f(170:238))); % 找出低频区的峰值

num(2) = 300 + find(f(300:412) == max(f(300:412))); % 找出高频区的峰值

% 判断具体属于哪个频率

if (num(1) < 180)

row = 1;

elseif (num(1) < 200)

row = 2;

elseif (num(1) < 220)

row = 3;

else

row = 4;

end

if (num(2) < 320)

col = 1;

elseif (num(2) < 340)

col = 2;

else

col = 3;

end

% 按照频率与号码的对应关系，将DTMF信号最终还原为号码

switch row

case 1

digit = col;

case 2

digit = 3 + col;

case 3

digit = 6 + col;

otherwise

digit = 0;

end

number = strcat(number, string(digit));

A.4 绘制单个数字的DTMF信号频谱图 核心部分实现

下面的程序段是按键“Plot”的回调函数。

% --- Executes on button press in buttonPlot.

% Plot按键，绘制最末一位号码的归一化单边频谱图

function buttonPlot\_Callback(hObject, eventdata, handles)

N = 2048; % FFT变换点数

Fs = 8192; % 原信号采样频率

global NUM; % 保存拨号音的全局变量

if length(NUM) > 0

num = get(handles.edit1, 'string');

digit = num(end);

lastNUM = NUM(end-1099:end-100); % 取出最后一个号码对应的DTMF信号

f = fft(lastNUM, N);

f = abs(f / N);

fs = f(1: (N/2 + 1)); % 取左半边，得到单边频谱

fs(2:end-1) = 2 \* fs(2:end-1); % 除去直流分量外，其他频率幅值乘2，以符合幅值归一化的要求

freq = Fs \* (0: (N/2)) / N; % 生成频谱图的横坐标

axes(handles.axes1); % 频谱图输出到程序中的图框

% figure(1); % 频谱图输出到单独的窗口

freqRange = 500;

plot(freq(1:freqRange), fs(1:freqRange));

xlabel('f(Hz)');

ylabel('|P1(t)|');

text(1500, 0.35, digit, 'Color', 'red', 'FontSize', 28); % 在频谱图中加上标签

title('最末位拨号音单边频谱图');

grid;

end