

2024-2025 学年第二学期

《信号处理实验》实验报告



专业班级 2023211119

姓 名 XXXXXX

学 号 2023210XXX

报告日期 2025 年 05 月 31 日

# 实验 DTMF 信号的产生与检测

## 一、实验目的

- (1) 了解 DTMF 信号产生与检测的方法。
- (2) 熟悉 DTMF 信号在工程实践中的应用。

## 二、实验原理

### 2.1 原理一：DTMF 信号产生方法原理

DTMF 信号是由两个不同频率正弦波叠加而成的信号。每个按键对应一个特定的频率对。当按键被按下时，产生对应的两个频率信号。

DTMF 频率对照情况如 表 1 所示。

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

表 1 DTMF 电话按键频率对照表（纵轴为低频，横轴为高频）

输出信号的生成公式如下：

$$x(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t)$$

DTMF 信号的生成有以下几种方式：

- (1) 查表法：存储周期，之后直接重复；
- (2) 函数法：通过正弦函数直接生成波形；
- (3) 滤波法：通过二阶滤波器生成信号。

在滤波法中，我们可以使用二阶 IIR 滤波器对高频和低频分别滤波，其公式如下：

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

### 2.2 原理二：DTMF 信号识别方法原理

- (1) 滤波器组法：

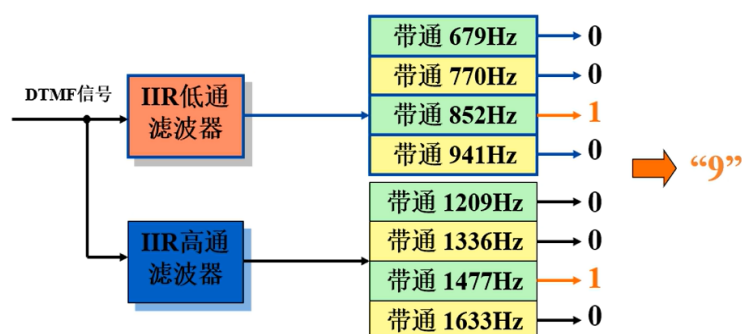


图1 滤波器组法实现 DTMF 信号解码的原理

该方法的主要原理是滤波，随后比较到各个频点能量大小。

(2) 基于 Goertzel 算法：

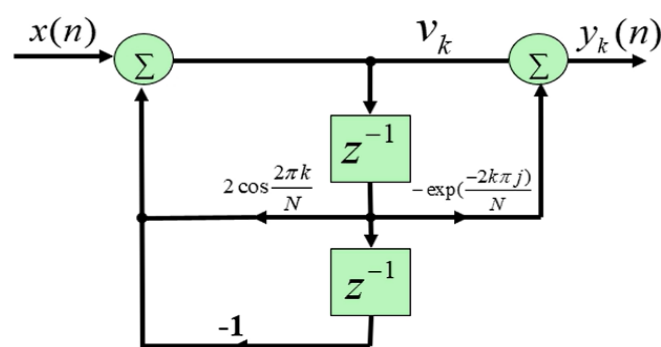


图2 Goertzel 算法信号流程图

Goertzel 算法： $O(N)$ ，每个频率点需要进行  $N$  次计算

FFT 算法： $O(N \log N)$ ，计算所有频率点

当只需要少量频率点( $k \ll N$ )时，Goertzel 算法的效率比 FFT 高

### 三、实验步骤

#### 3.1 码上使用过程

在学习本次任务的过程中，我首先通过知识问答 Agent 了解了 Goertzel 算法，它帮助我了解到其核心也是一套迭代算法，并且帮助我很快理解了信号流程图。

在真正开始编写代码的过程中，我使用了代码解读工具快速了解了老师提供的代码，并且并且清楚自己要实现代码的具体内容。

#### 3.2 Matlab 实验过程

(1) DTMF 波形生成

1. 函数法

下面的代码实现了定义各个按键对应函数表达式的功能，对于任意的电话

号码，只需要按照数序播放下面的函数即可。

```
fs=8000;%采样率
t=(0:399)/fs;
fc1=697; fc2=770; fc3=852; fc4=941;%低频组
fr1=1209; fr2=1336; fr3=1477;%高频组
num0 = (sin(fc4*2*pi*t)+sin(fr2*2*pi*t))/2;% 数字 0
num1 = (sin(fc1*2*pi*t)+sin(fr1*2*pi*t))/2;% 数字 1
num2 = (sin(fc1*2*pi*t)+sin(fr2*2*pi*t))/2;% 数字 2
num3 = (sin(fc1*2*pi*t)+sin(fr3*2*pi*t))/2;% 数字 3
num4 = (sin(fc2*2*pi*t)+sin(fr1*2*pi*t))/2;% 数字 4
num5 = (sin(fc2*2*pi*t)+sin(fr2*2*pi*t))/2;% 数字 5
num6 = (sin(fc2*2*pi*t)+sin(fr3*2*pi*t))/2;% 数字 6
num7 = (sin(fc3*2*pi*t)+sin(fr1*2*pi*t))/2;% 数字 7
num8 = (sin(fc3*2*pi*t)+sin(fr2*2*pi*t))/2;% 数字 8
num9 = (sin(fc3*2*pi*t)+sin(fr3*2*pi*t))/2;% 数字 9
numStar = (sin(fc4*2*pi*t)+sin(fr1*2*pi*t))/2;% 符号*
numJin = (sin(fc4*2*pi*t)+sin(fr3*2*pi*t))/2;% 符号#
blk=zeros(size(num1));%拨号间隔时间
```

## 2. 滤波法

下面的代码实现了主要滤波器的相关功能，通过滤波就可以得到 16 个按键各自的波形。

```
lf=[697 770 852 941];
hf=[1209 1336 1477 1633];
fs=8000;
lw=2*pi*lf/fs;
hw=2*pi*hf/fs;
n = 0:1:204;
x = [1,zeros(1,204)];
% j,k 可以取 1~4,用于遍历 16 个按键
lb = [0, sin(lw(j))]; % 分子系数
la = [1, -2*cos(lw(j)), 1]; % 分母系数
ly = filter(lb,la,x);
hb = [0, sin(hw(k))];
ha = [1, -2*cos(hw(k)), 1];
hy = filter(hb,ha,x);
r = 0.5*ly+0.5*hy;
```

其结果如 图 3 所示

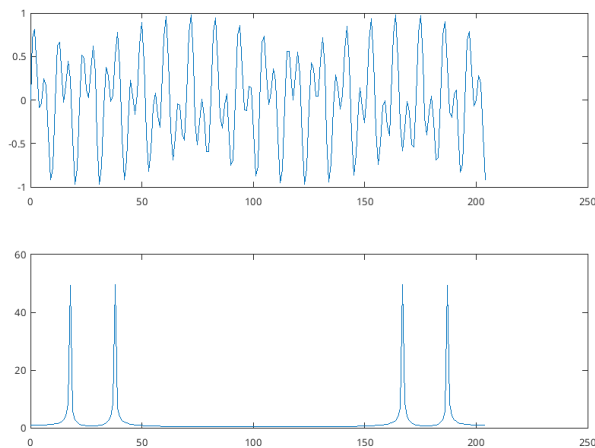


图3 滤波法实现 DTMF 信号生成（16 个波形之一）

## （2）DTMF 信号解析

### 1. 滤波器组法

分别实现高频和低频的滤波器即可，代码如下

```
f_L=[fc1 fc2 fc3 fc4]; % 低频组
f_H=[fr1 fr2 fr3]; % 高频组
f = [fc1 fc2 fc3 fc4 fr1 fr2 fr3];

% 设计带通滤波器（低频组）
N = 400; % 滤波器阶数
Bandwidth = 70;
B_L = zeros(4,N+1); % 用于存放低频组滤波器系数
for i=1:4
    Wo=f_L(i);
    wc1 = ( Wo - Bandwidth / 2 ) * 2 * pi / fs;
    wc2 = ( Wo + Bandwidth / 2 ) * 2 * pi / fs;
    B_L(i,:) = fir2(N, [0,wc1/pi, wc2/pi,1], [0 1 1 0]);
    % freqz(B_L(i,:),1);
end
freqz(B_L(i,:),1);

% 设计带通滤波器（高频组）
N = 200; % 滤波器阶数
Bandwidth = 110; %
B_H = zeros(3,N+1); % 用于存放高频组滤波器系数
for i=1:3
    Wo=f_H(i);
    wc1 = ( Wo - Bandwidth / 2 ) * 2 * pi / fs;
    wc2 = ( Wo + Bandwidth / 2 ) * 2 * pi / fs;
    B_H(i,:) = fir2(N,[0,wc1/pi,wc2/pi,1],[0,1,1,0]);
    % freqz(B_H(i,:),1);
end
```

```
freqz(B_H(i,:),1);
```

## 2. Goertzel 算法

这里 Matlab 提供了内置的 `goertzel` 函数，直接计算各个声音在各个频率点的 DFT，然后选出其中最高的两个，即是需要的高频和低频分量。

## 四、实验结果与分析

本实验所有代码均会在 附录 中完整展示

本实验共有三个 Matlab 代码，其中滤波器生成单个信号的结果已经在 图 3 中展示，这里补充一个生成的 wav 文件的波形展示如 图 4

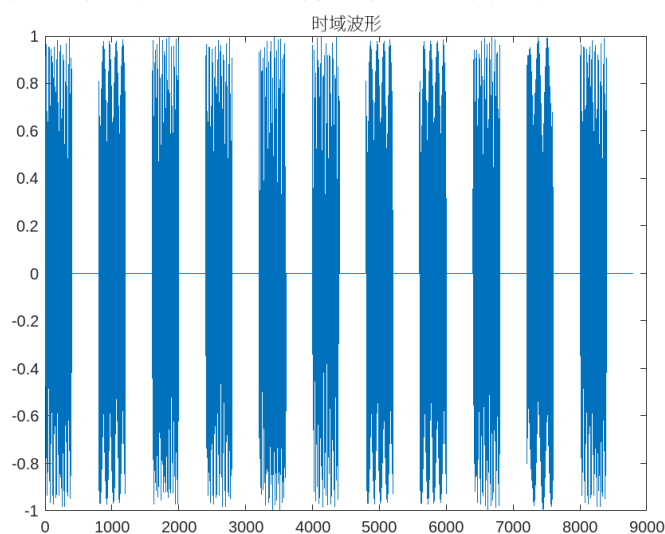


图 4 时域波形展示

在上图中，每一个数字之间存在一个间隔，用于生成分立的信号。

下面展示解码相关的实验结果，下面将从单音频识别和真实场景两部分出发展示相关实验结果并进行分析。

首先是滤波器法的单音频结果：

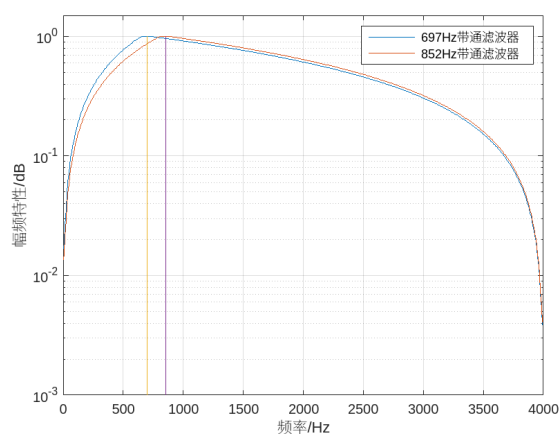


图 5 数字 “7”的滤波器组法识别结果

从图中可以看出，最后数字 “7”对应的恰好是 697Hz 和 852Hz 的滤波器，

也就说明对应的音频在上述两个频率的能量最高，由此就验证了滤波器组方法原理的正确性。

之后是真实电话号码的解码（使用 Goertzel 算法）

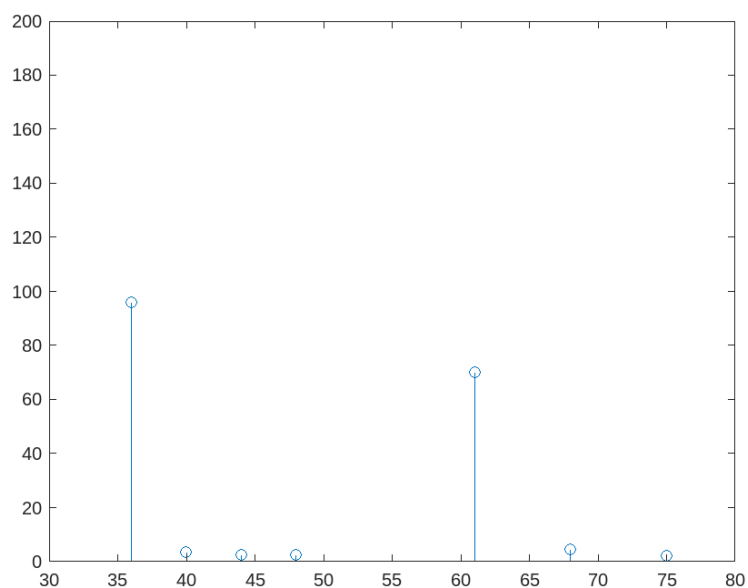


图 6 电话号码其中某一数字的识别结果

如 图 6 所示，通过 Goertzel 算法，我们可以明显看出其能够分辨各个 DTMF 频点

## 五、码上使用心得体会

码上平台提供的代码和指导虽然准确，但在面对复杂实验或特殊需求时，还是需要进一步优化代码的逻辑和实验设计，虽然其可以帮助生成代码和报告框架，但对于实验的深入分析和创新部分，仍然需要我们结合自己的理解和实验观察进行扩展与讨论，避免过度依赖生成内容。

在今后的实验中，建议更清晰地明确实验的目标与实际问题，注重代码在各个平台上的兼容性，将理论课程与实验课程结合起来。

## 六、附录

### 6.1 dtmf\_gen\_F\_cover.m

```
% dtmf_gen_F_cover.m
```

```
lf=[697 770 852 941];  
hf=[1209 1336 1477 1633];  
fs=8000;  
lw=2*pi*lf/fs;  
hw=2*pi*hf/fs;  
n = 0:1:204;  
x = [1,zeros(1,204)];
```

```

%K = [18 20 22 24 31 34 38 42];
%ww = 2*cos(2*pi*K/205);
%%
% 滤波法产生 DTMF 信号

for j = 1:4
    for k = 1:4
        lb = [0, sin(lw(j))]; % 分子系数
        la = [1, -2*cos(lw(j)), 1]; % 分母系数
        ly = filter(lb,la,x);
        hb = [0, sin(hw(k))];
        ha = [1, -2*cos(hw(k)), 1];
        hy = filter(hb,ha,x);

        r = 0.5*ly+0.5*hy;

        subplot(2,1,1);
        plot(n,r)
        Y = fft(r,205);
        subplot(2,1,2);
        plot(n,abs(Y))
        pause(2);
    end
end

```

## 6.2 dtmf\_det\_F\_cover.m

```

% dtmf_det_F_cover.m

clear;
clc;
fs=8000; %采样率
t=(0:399)/fs; %100ms 的按键
fc1=697; fc2=770; fc3=852; fc4=941; %低频组
fr1=1209; fr2=1336; fr3=1477; %高频组
%%

num0 = (sin(fc4*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 0
num1 = (sin(fc1*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 1
num2 = (sin(fc1*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 2
num3 = (sin(fc1*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 3
num4 = (sin(fc2*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 4
num5 = (sin(fc2*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 5
num6 = (sin(fc2*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 6
num7 = (sin(fc3*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 7
num8 = (sin(fc3*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 8
num9 = (sin(fc3*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 9
numStar = (sin(fc4*2*pi*t)+sin(fr1*2*pi*t))/2; % 符号*
numJin = (sin(fc4*2*pi*t)+sin(fr3*2*pi*t))/2; % 符号#

```



```

blk=zeros(size(num1)); %拨号间隔时间

f_L=[fc1 fc2 fc3 fc4]; % 低频组
f_H=[fr1 fr2 fr3]; % 高频组
f = [fc1 fc2 fc3 fc4 fr1 fr2 fr3];
%%

% 设计带通滤波器（低频组）
N = 400; % 滤波器阶数
Bandwidth = 70;
B_L = zeros(4,N+1); %用于存放低频组滤波器系数
for i=1:4
    Wo=f_L(i);
    wc1 = ( Wo - Bandwidth / 2 ) * 2 * pi / fs;
    wc2 = ( Wo + Bandwidth / 2 ) * 2 * pi / fs;
    B_L(i,:) = fir2(N, [0,wc1/pi, wc2/pi,1], [0 1 1 0]);
    % freqz(B_L(i,:),1);
end

freqz(B_L(i,:),1);

%%

% 设计带通滤波器（高频组）
N = 200; %滤波器阶数
Bandwidth = 110; %
B_H = zeros(3,N+1); %用于存放高频组滤波器系数
for i=1:3
    Wo=f_H(i);
    wc1 = ( Wo - Bandwidth / 2 ) * 2 * pi / fs;
    wc2 = ( Wo + Bandwidth / 2 ) * 2 * pi / fs;
    B_H(i,:) = fir2(N,[0,wc1/pi,wc2/pi,1],[0,1,1,0]);
    % freqz(B_H(i,:),1);
end

freqz(B_H(i,:),1);
%%

DialNum = num7; %按键
sound(DialNum);
plot(DialNum);
title('时域波形');
%%

% 计算当前信号与各个频点的距离（低频组）
Diatance_L = zeros(1,4);
for i=1:4

```

```

% 滤波处理
Out = filter(B_L(i,:), 1, DialNum); % 使用第 i 个低频滤波器处 DialNum

% 能量计算
Diatance_L(1,i) = sum(Out.^2); % 计算滤波后信号的能量
end
[maxnum_lo, index_lo] = max(Diatance_L(1,:));
%%

% 计算当前信号与各个频点的距离（高频组）
Diatance_H = zeros(1,3);
for i=1:3
    Out = filter(B_H(i,:),1,DialNum);
    Diatance_H(1,i) = max(abs(fft(Out)));
end
[maxnum_hi,index_hi]=max(Diatance_H(1,:));
%%

% 判断按键值
if index_lo == 1 && index_hi == 1
    keynum = 1;
elseif index_lo == 1 && index_hi == 2
    keynum = 2;
elseif index_lo == 1 && index_hi == 3
    keynum = 3;
elseif index_lo == 2 && index_hi == 1
    keynum = 4;
elseif index_lo == 2 && index_hi == 2
    keynum = 5;
elseif index_lo == 2 && index_hi == 3
    keynum = 6;
elseif index_lo == 3 && index_hi == 1
    keynum = 7;
elseif index_lo == 3 && index_hi == 2
    keynum = 8;
elseif index_lo == 3 && index_hi == 3
    keynum = 9;
elseif index_lo == 4 && index_hi == 2
    keynum = 0;
elseif index_lo == 4 && index_hi == 1
    keynum = '*';
elseif index_lo == 4 && index_hi == 3
    keynum = '#';
end
%%

% 比较不同带通滤波器的幅频特性
[H_L1,W_L1]=freqz(B_L(1,:),1); % fc1=697Hz
[H_L3,W_L3]=freqz(B_L(3,:),1); % fc3=852Hz

```

```

semilogy(W_L1*fs/2/pi,abs(H_L1),W_L3*fs/2/pi,abs(H_L3));
hold on;
semilogy(697*[1,1],[1e-3,1],852*[1,1],[1e-3,1]);
xlabel('频率/Hz');ylabel('幅频特性/dB');
legend('697Hz 带通滤波器','852Hz 带通滤波器');
axis([0 4000,1e-3, 1.5]);grid on;

```

### 6.3 dtmf\_det\_G\_cover.m

```

% dtmf_det_G_cover.m
%% 利用拨号音识别手机号码 基于 Goertzel 算法
clear;
clc;
fs=8000; %采样率
t=(0:399)/fs;
fc1=697; fc2=770; fc3=852; fc4=941; %低频组
fr1=1209; fr2=1336; fr3=1477; %高频组
%%
num0 = (sin(fc4*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 0
num1 = (sin(fc1*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 1
num2 = (sin(fc1*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 2
num3 = (sin(fc1*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 3
num4 = (sin(fc2*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 4
num5 = (sin(fc2*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 5
num6 = (sin(fc2*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 6
num7 = (sin(fc3*2*pi*t)+sin(fr1*2*pi*t))/2; % 数字 7
num8 = (sin(fc3*2*pi*t)+sin(fr2*2*pi*t))/2; % 数字 8
num9 = (sin(fc3*2*pi*t)+sin(fr3*2*pi*t))/2; % 数字 9
numStar = (sin(fc4*2*pi*t)+sin(fr1*2*pi*t))/2; % 符号*
numJin = (sin(fc4*2*pi*t)+sin(fr3*2*pi*t))/2; % 符号#
%
blk=zeros(size(num1)); %拨号间隔时间

%%
%% 手机号 13910733521

CellNum = [num1 blk num3 blk num9 blk num1 blk num0 blk num7 blk...
            num3 blk num3 blk num5 blk num2 blk num1 blk];

% CellNum = 0.2*randn(size(CellNum))+CellNum; %传输噪声+DTMF 拨号音
%%

filename = 'dtmf.wav';
audiowrite(filename,CellNum,fs);

[CellNum,Fs] = audioread(filename);
%%

sound(CellNum);

```



```

elseif index_lo == 4 && index_hi == 6
    keynum = '0';
elseif index_lo == 4 && index_hi == 5
    keynum = '*';
elseif index_lo == 4 && index_hi == 7
    keynum = '#';
end
Cell_Indent(round(i/2))=keynum; %手机号码识别结果
end
disp(Cell_Indent) %手机号码识别结果
%%

fileID = fopen('dtmf_key.txt','w');
fprintf(fileID,'%6s ',Cell_Indent);
fclose(fileID);

```