

**语音信号处理**

**实验报告**

**实验名称： 语音信号的倒谱分析**

**学生姓名： 赵展文 学号： 2024360809**

**授课教师： 马 英 职称： 教 授**

**院 系： 智能科学与工程学院**

**专业班级： 2024级集成电路工程专业**

**一、验证一个时域信号经过同态处理，是否回到时域？**

程序代码

%% 清除缓存

clear;

close all;

%% 读取音频文件

[file, path] = uigetfile('\*.wav', '选择WAV文件');

filename = fullfile(path, file);

[y, fs] = audioread(filename);

y = y(:, 1); % 单通道

%% 设置帧参数

frame\_length = 1024;

frame\_shift = 256;

hanning\_win = hanning(frame\_length);

%% 分帧处理

fn = fix((length(y) - frame\_length) / frame\_shift) + 1; % 计算帧数

frames = zeros(frame\_length, fn); % 预分配存储空间

for i = 1:fn

start\_idx = (i - 1) \* frame\_shift + 1;

end\_idx = start\_idx + frame\_length - 1;

frames(:, i) = y(start\_idx:end\_idx);

end

%% 加窗处理

cepstrum = zeros(frame\_length, fn); % 倒谱矩阵

for i = 1:fn

frames(:, i) = frames(:, i) .\* hanning\_win;

spectrum = fft(frames(:, i));

log\_spectrum = log(abs(spectrum) + eps);

c = ifft(log\_spectrum);

cepstrum(:, i) = real(c);

end

%% 选择

frame\_idx = 50;

if frame\_idx > fn

frame\_idx = fn;

end

frame\_win = frames(:, frame\_idx);

%% 同态处理与还原

fft\_frame = ifft(log(fft(frame\_win) + eps)); %% esp防止log(0)

ifft\_frame = ifft(exp(fft(fft\_frame)));

%% X轴为样本点索引

x = 1:frame\_length;

%% 绘图

figure;

subplot(2,1,1);

plot(x, frame\_win, 'b-');

title('原始信号');

xlabel('样本点'); ylabel('幅度');

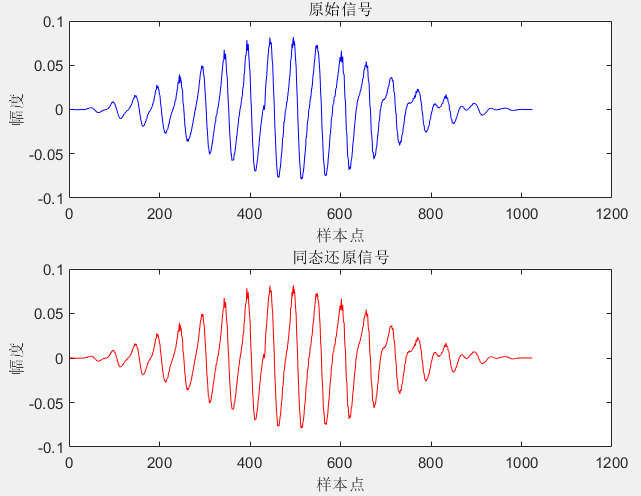
subplot(2,1,2);

plot(x, ifft\_frame, 'r-');

title('同态还原信号');

xlabel('样本点'); ylabel('幅度');

波形图



**二、语音信号倒谱分析，估计基音周期和共振峰频率**

程序代码

%% 清除缓存

clear;

close all;

%% 读取音频文件

[file, path] = uigetfile('\*.wav', '选择WAV文件');

filename = fullfile(path, file);

[y, fs] = audioread(filename);

y = y(:, 1); % 单通道音频

%% 设置帧参数

frame\_length = 1024; % 帧长

frame\_shift = 256; % 帧移

hanning\_win = hanning(frame\_length); % Hanning窗

%% 分帧处理

fn = fix((length(y) - frame\_length) / frame\_shift) + 1; % 计算帧数

frames = zeros(frame\_length, fn); % 预分配存储空间

for i = 1:fn

start\_idx = (i - 1) \* frame\_shift + 1;

end\_idx = start\_idx + frame\_length - 1;

frames(:, i) = y(start\_idx:end\_idx);

end

%% 计算倒谱分析

cepstrum = zeros(frame\_length, fn); % 倒谱矩阵

for i = 1:fn

frame = frames(:, i); % 获取当前帧

frame = frame .\* hanning\_win; % 加窗

% 对信号进行 FFT

spectrum = fft(frame);

% 计算对数功率谱

log\_spectrum = log(abs(spectrum) + eps);

% 对数功率谱进行逆傅里叶变换

c = ifft(log\_spectrum);

cepstrum(:, i) = real(c);

end

%% 绘制倒谱

figure;

subplot(2,1,1);

plot((0:length(y)-1) / fs, y);

xlabel('时间 (秒)');

ylabel('幅度');

title('原始语音信号');

subplot(2,1,2);

plot(c);

xlabel('帧数');ylabel('倒谱系数索引');

title('倒谱分析');

grid on;

%% 基音周期估计

f0\_estimates = zeros(1, fn); % 基音周期 F0 从倒谱的低阶系数中估计

for i = 1:fn

c = cepstrum(:, i);

% 估计基音周期为倒谱系数的最大周期（峰值位置）

[~, max\_lag] = max(c(2:frame\_length)); % 排除第一个系数

f0\_estimates(i) = fs / max\_lag; % 基音频率 F0

end

% 绘制基音周期（F0）

figure;

subplot(3,1,1);

plot((0:length(y)-1) / fs, y);

xlabel('时间 (秒)');

ylabel('幅度');

title('原始语音信号');

subplot(3,1,2);

plot((1:fn) \* frame\_shift / fs, f0\_estimates);

xlabel('时间 (秒)');ylabel('基音频率 (Hz)');

title('基音频率随时间变化');

%% 共振峰频率估计

formant\_estimates = zeros(3, fn); % 假设有3个共振峰

for i = 1:fn

c = cepstrum(:, i);

% 取倒谱的高阶系数

high\_order\_cepstrum = c(10:end); % 去掉低阶倒谱系数（基音周期部分）

% 找到局部最大值作为共振峰频率的估计

[peaks, locs] = findpeaks(high\_order\_cepstrum);

% 提取前3个共振峰频率

formant\_estimates(:, i) = locs(1:3) \* fs / frame\_length;

end

% 绘制共振峰频率

subplot(3,1,3);

plot((1:fn) \* frame\_shift / fs, formant\_estimates');

xlabel('时间 (秒)');ylabel('频率 (Hz)');

title('共振峰频率随时间变化');

legend('F1', 'F2', 'F3');

波形图

