

**语音信号处理**

**实验报告**

**实验名称： 语音信号的频域分析**

**学生姓名： 赵展文 学号： 2024360809**

**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**

**一、语音信号的短时频谱分析**

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); % 取单通道音频

%% 设定帧参数

frame\_length = 256; % 帧长

frame\_shift = 64; % 帧移

%% 分帧处理

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

frame\_time = (((1:fn)-1)\*frame\_shift + frame\_length/2)/fs; % 每帧中心对应的时间（秒）

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

for i = 1:fn

start\_idx = (i - 1) \* frame\_shift + 1; % 计算起始索引

end\_idx = start\_idx + frame\_length - 1; % 计算结束索引

frames(:, i) = y(start\_idx:end\_idx); % 逐帧提取

end

%% 计算 Hanning 窗

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

for i = 1:fn

frames(:, i) = frames(:, i) .\* hanning\_win; % 每帧加窗

end

%% 计算短时傅里叶变换

fft\_frames = fft(frames); % 对每一帧进行 FFT

frame\_length2 = floor(frame\_length/2) + 1;

S = fft\_frames(1:frame\_length2, :); % 取每帧 FFT 的前半部分

freq = (0:frame\_length2-1)\*fs/frame\_length; % 构造频率轴（Hz）

magS = abs(S); % 计算幅度谱

magS\_db = 20\*log10(magS + eps); % 转换为 dB 单位

%% 绘制单帧幅度谱

frame\_idx = 105;

if frame\_idx > fn

frame\_idx = fn;

end

mag\_spec = magS\_db(:, frame\_idx);

figure;

plot(freq, mag\_spec, 'b-');

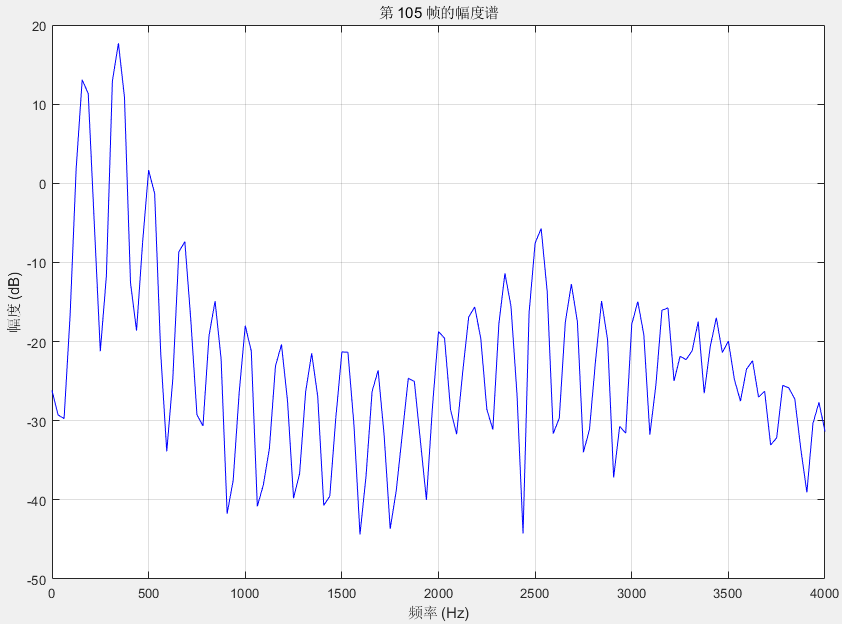
xlabel('频率 (Hz)');

ylabel('幅度 (dB)');

title(sprintf('第 %d 帧的幅度谱', frame\_idx));

grid on;

波形图



**二、语音信号的短时功率谱分析**

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); % 取单通道音频

%% 设定帧参数

frame\_length = 256; % 帧长

frame\_shift = 64; % 帧移

%% 分帧处理

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

frame\_time = (((1:fn)-1)\*frame\_shift + frame\_length/2)/fs; % 每帧中心对应的时间（秒）

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

for i = 1:fn

start\_idx = (i - 1) \* frame\_shift + 1; % 计算起始索引

end\_idx = start\_idx + frame\_length - 1; % 计算结束索引

frames(:, i) = y(start\_idx:end\_idx); % 逐帧提取

end

%% 计算 Hanning 窗

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

for i = 1:fn

frames(:, i) = frames(:, i) .\* hanning\_win; % 每帧加窗

end

%% 计算短时傅里叶变换

fft\_frames = fft(frames); % 对每一帧进行 FFT

frame\_length2 = floor(frame\_length/2) + 1;

S = fft\_frames(1:frame\_length2, :); % 取每帧 FFT 的前半部分

freq = (0:frame\_length2-1)\*fs/frame\_length; % 构造频率轴（Hz）

P = abs(S).^2; % 计算功率谱

P\_db = 10\*log10(P + eps); % 对功率谱转换为dB

frame\_idx = 105;

if frame\_idx > fn

frame\_idx = fn;

end

%% 绘制单帧功率谱

power\_spec = P\_db(:, frame\_idx);

figure;

plot(freq, power\_spec, 'b-');

xlabel('频率 (Hz)');

ylabel('功率 (dB)');

title(sprintf('第 %d 帧的功率谱', frame\_idx));

grid on;

波形图

