

**语音信号处理**

**实验报告**

**实验名称： 语音增强算法分析**

**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**

**一、用任意一种方法实现语音增强**

程序代码

clc; clear; close all;

%% === 读取语音 ===

[x, fs] = audioread("voice/test.wav");

x = x(:, 1);

x = x - mean(x);

x = x / max(abs(x));

%% === 加噪处理 ===

SNR = 10; % 设置信噪比（单位 dB）

signal = Gnoisegen(x, SNR); % 添加白噪声

snr1 = SNR\_singlech(x, signal); % 初始 SNR

%% === 分帧 ===

wlen = 512; inc = 128;

win = hamming(wlen);

x\_framed = buffer(signal, wlen, wlen - inc, 'nodelay');

[~, fn] = size(x\_framed);

%% === VAD检测（基于能量+过零率）===

T1 = 0.01; % 能量门限

[voiceseg, vosl, SF, Ef] = pitch\_vad1(x\_framed, fn, T1);

global\_noise = estimateNoise(x\_framed, SF == 0); % 用无话帧估计噪声

%% === 初始化 ===

X\_klt\_raw = zeros(size(x\_framed)); % KLT增强后语音（未滤波）

X\_klt\_wiener = zeros(size(x\_framed)); % Wiener滤波后语音

%% === 主处理循环 ===

for i = 1:fn

frame = x\_framed(:, i) .\* win;

C = robustCovEst(frame);

[V, D] = eig(C);

klt\_coeff = V' \* frame;

% --- KLT增强 ---

X\_klt\_raw(:, i) = V \* klt\_coeff;

% --- Wiener滤波 ---

signal\_energy = diag(D);

noise\_proj = V' \* global\_noise \* V;

noise\_energy = diag(V' \* global\_noise \* V);

gain = signal\_energy ./ (signal\_energy + noise\_energy + eps);

gain = max(gain, 0.18); % 控制最小增益

klt\_coeff\_filt = gain .\* klt\_coeff;

X\_klt\_wiener(:, i) = V \* klt\_coeff\_filt;

end

%% === 重叠相加恢复语音 ===

output\_klt = overlapAdd(X\_klt\_raw, wlen, inc);

output\_final = overlapAdd(X\_klt\_wiener, wlen, inc);

output\_final = output\_final / max(abs(output\_final));

%% === 可视化 ===

t = (0:length(x)-1) / fs;

figure;

subplot(4,1,1); plot(t, x); title("原始语音信号"); ylabel("幅度"); grid on;

subplot(4,1,2); plot(t, signal); title(sprintf("加噪语音信号 (SNR = %.2f dB)", snr1)); ylabel("幅度"); grid on;

subplot(4,1,3); plot(t, output\_klt(1:length(x))); title("KLT增强后语音"); ylabel("幅度"); grid on;

subplot(4,1,4); plot(t, output\_final(1:length(x))); title("Wiener滤波后语音"); ylabel("幅度"); xlabel("时间/s"); grid on;

%% === 播放增强语音 ===

sound(output\_final, fs);

波形图

