

# 语音信号采集与滤波处理系统设计与实现

## 任务要求

1. 利用MATLAB 设计实现语音信号采集处理软件，通过MIC 实时录制并分析语音信号，包括实时显示信号波形、时域统计特征值、计算信号的频谱等；
2. 采集并录制一段自己的语音信号，选取合适的滤波器性能指标，采用双线性变换设计不同类型（包括巴特沃斯、切比雪夫I 型、切比雪夫II 型、椭圆）滤波器，比较并画出滤波器的频率响应；然后用自己设计的滤波器对采集的信号进行滤波，画出滤波后信号的时域波形和频谱，并对滤波前后的信号进行对比，分析信号的变化；回放语音信号；
3. 基于MATLAB GUI 设计语音信号采集与滤波处理系统界面，实现上述语音采集、滤波处理功能。要求该系统可以选择滤波器的类型，输入滤波器的参数，显示滤波器的频率响应，选择信号，显示滤波前后的频谱，播放功能等。

## 实践原理

- 采样定理
- 时域信号的FFT分析
- 双线性变换
- 滤波器设计

## 实现代码

为了将原始的模拟语音信号转变为数字信号，必须进行采样和量化，进而得到时间和幅度上均为离散的数字语音信号。原始语音信号使用 Windows自带录音机录制，时长为 5s，录音功能代码如下：

```
fs = 44100;           % 采样频率
duration = 5;         % 时间长度(秒)
n = duration*fs;      % 采样点数
t = (1:n)/fs;

%创建一个录音文件: fs =44100, 16-bit, 单通道

recObject = audiorecorder(fs, 16, 1);
fprintf('3秒后开始录音:\n');pause(3);
fprintf('录音中...\n');
recordblocking(recObject, duration);
stop(recObject);
fprintf('录制结束\n');
fprintf('按任意键播放录制音频: \n');pause
play(recObject);      % 播放录制的音频
y = getaudiodata(recObject);

ymax = max(abs(y));   % 归一化
y = y/ymax;

audiowrite('voice1.wav', y, fs); % 写入音频, 进行保存
```

```
figure(1);
plot(t, y);
xlabel('时间/s');
ylabel('幅度');
title('(a)fs = 44100');
```

## 波形分析

```
[x, fs] = audioread('D:\MatlabFile\voice.wav');

L = length(x);
X=fft(x,L);
A = fftshift(X);
A=abs(A);

magX = abs(X);
angX = angle(X);

ws = 2* pi* fs;
w = (-ws/2 + (0:L-1) * ws/L)/(2 * pi);
figure(1)
subplot(221);
plot(t, x);
title('原始信号波形');
xlabel('时间/s');
ylabel('幅度');
subplot(222);
plot(w, abs(A));
title('原始信号频谱');
xlabel('频率/Hz');
ylabel('幅度');
subplot(223);
plot(magX);
title('原始信号幅值');
subplot(224);
plot(angX);
title('原始信号相位');
```

## 加入正弦噪声

```
Au = 0.1;
n = (Au*sin(2*pi*5000*(1:size(x))/fs))'; %加入正弦噪声
% 合成后的语音信号（含噪声）
y = x + n;
% sound(y, fs ,16);
figure(2)
subplot(2,2,1)
plot(t, x) %做原始语音信号的时域图形
title('原语音信号时域波形')
```

```
xlabel('时间/s');
ylabel('幅度');
subplot(2,2,2)
plot(t, y) %做原始语音信号的时域图形
title('加正弦噪声后语音信号时域波形')
xlabel('时间/s');
ylabel('幅度');
y1=fft(y,L);
A1 = fftshift(y1);
A1 = abs(A1);
subplot(2,2,3)
plot(w, abs(A))
title('原始语音信号频谱');
xlabel('频率/Hz');
ylabel('幅度');
subplot(2,2,4)
plot(w, abs(A1))
title('加噪语音信号频谱');
xlabel('频率/Hz');
ylabel('幅度');
```