

## 数字信号处理实验之浮点程序定点化实验相关内容讲解与说明

### 本 C 程序模拟对语音信号做实时按帧滤波

1. 模拟信号去均值、归一化，语音数据规范到正负 1 之间的小数（模拟信号）  
//去均值、归一化，是重要的预处理操作
2. 语音信号采样频率 8kHz、16-bit 分辨率、单声道（数字化存储：数字信号）  
**CoolEdit 录音软件录制单声道、8KHz 采样率、16-bit 分辨率语音数据**  
Tip——resolution，分辨率；不等于精度，精度是精密度和准确度的综合评价  
精度：precision; accuracy; degree of accuracy
3. 指定 16-bit 存储数据：分辨率与动态范围的协调  
//Q 数据格式，Q15：15 位小数、1 位符号位、0 位整数位；Q0：整数
4. 语音样点在存储器里的存储方式、读取方式（按需读取）  
//存储 16-bit 的小数？Q15，二进制存储  
//16-bit 存储空间存储的正负 1 之间的小数数字语音样点!!!
5. 快速处理：按整数处理（Q0）  
//每次读取适当的位宽，8-bit？16-bit？32-bit？  
//能用整数就不用小数  
//把小数数据视为整数进行处理，有没有问题？要做什么相应处理？  
//尽量用短整数表示一个完整数据；能用 8-bit 就不用 16-bit，能用 16-bit 就不用 32-bit  
//能转换成整数运算的数据都做转换，包括语音样点、滤波器系数（程序中已提供）  
//把 16-bit 存储空间存储的正负 1 之间的小数数字语音样点读取成 16-bit 整数!!!  
//对于程序中已经有的小数，如果要用数学进行转换，如何转换？转换计算式？
6. C 语言主程序，要弄明白读取数据的正确做法，正确使用读取、存储函数

### 观测、判断所读取的数据是正确的，如何做？

C 语言里该如何读？get\_w、put\_w 行不行？有没有更好的方式？到底该读成什么数？

Tip——如何判断你 C 语言程序里读取的数据是正确的？与 Matlab 读数交叉验证。

Matlab 函数 wavread 的输入输出参数，[y,fs,nbit] = wavread("\*.wav");

Tip——其特点是把存储的 16-bit 数据读成 double 类型，且  $|y| \leq 1$ ；

把 16-bit 二进制数看成什么数据？Matlab 很强大啊！最经济读取方式是哪种？

Matlab 高版本里 wavread 函数被 audioread 函数所替换。

yC=用 C 语言从 wav 文件里读取的 16 位整数数据[数组];

// 前 22 个数据是 wav 文件的头格式信息。

y→y16=int16(y\*2^15);

//把 Matlab 读取的 double 数据变成 16 位整数

yC[22]= y16(1)?

yC[23]= y16(2)?

yC[24]= y16(3)?

// 判断两种方式读取的两种不同方式获取的若干个语音数据是否相等？

// **这是本实验的第一个关键问题，要解决好！**

### 文件基本格式（二进制和文本格式）

C 程序中要求是读取二进制的文件，wav 文件是二进制文件；wav 文件有头格式，但这些头格式信息也可以当作一般的数据去读取，并不影响程序实现。

文件名后缀可以更改，例如 wav 文件的后缀 wav 更改为 dat，并不会改变文件的二进制文件性质。

7. 浮点子程序，弄清楚舍去保留法的操作过程、定点化，需要一定的特殊辅助操作

**对于舍去保留法，要弄清楚是要舍去哪部分？要保留哪部分？为什么？**

//注意滤波器系数，也要转换成整数，为了快速处理。

//子程序入口参数，滤波器系数，要修改成 16-bit 整数类型。

//程序中的滤波器数据，原来是纯小数，也需要相应更改。

8. 滤波器，绘制出滤波器频率响应，理解滤波器类型、截止频率

//可以用 Matlab 做，可以采用小数。

9. 适当取语音信号帧，做频谱分析

//提取某几帧有效语音帧并绘制其滤波前和滤波后的频谱图

10. 在原始语音信号中添加低频正弦噪声（低于滤波器截止频率 800Hz），再进行滤波

11. 在原始语音信号中添加高频正弦噪声（高于滤波器截止频率 800Hz），再进行滤波  
生成含噪语音信号，要求是在干净语音信号中添加一个低频正弦噪声和一个高频正弦噪声；再对含噪语音信号做滤波实验，要求能够滤除高频正弦噪声而保留低频正弦噪声；提取某几帧含噪语音帧并绘制其滤波前和滤波后的频谱图。

说明：

<1> 生成含噪语音信号，可以用 matlab 来实现，就是干净语音加噪声信号；注意调整二者的幅度，并使合成后的数据幅度要在正负 1 之间；将生成的含噪语音（数据）保存到文件，供 C 语言程序读取。

**<2>正弦信号的抽样的注意事项——正确生成含噪语音数据。**因为实验中用到了把正弦信号作为噪声，并对语音信号加正弦噪声后再做滤波的实验，所以看看关于正弦信号的更多内容，大家可以再看书，理解对正弦信号抽样的要求。对模拟正弦信号采样时考虑采用 8KHz，即采样语音信号的采样率；截短离散时间正弦序列时考虑点数满足整周期点数的倍数。

在 Matlab 环境中编程实现比较方便。

`[y, fs, bits] = wavread('***.wav');`

Sin\_Noise 噪声序列生成方法：例如，400Hz 模拟正弦，采样率为 8KHz，要正确计算出数字角频率  $\omega_0$  是多少，然后得到离散时间正弦序列  $\sin(\omega_0 n)$ ，注意截取 Sin\_Noise 序列长度的要求。

`y_sin = y + k*Sin_Noise;` %其中 k 用于控制信噪比。

再对 y\_sin 做归一化。要先得到 `Max_number = max(abs(y_sin))`，可以考虑再添加一个微小的 delta 量，再做归一化。

然后，可以将归一化的 y\_sin 保存为 wav 格式的文件（使用 wavwrite 函数），或直接保存为一般的二进制文件，供 C 代码做滤波处理。注意 wavread 和 wavwrite 函数的使用方法。

<3> 绘制系统的频率响应、（含噪）语音信号（整体性、不分帧）在滤波前和滤波后的频谱分析，可以在 matlab 中实现，用上同学们掌握的 matlab 函数来实现。

对比两种情况：长度等于离散时间正弦序列整周期点数整倍数和非整倍数情况下的频谱。

12. CoolEdit 软件绘制信号的语谱图