

# 实验一 短时时域分析

## 【实验目的】

- 1、了解语音信号分帧与加窗的重要性和必要性。
- 2、掌握常用的窗函数和加窗分帧处理的原理。
- 3、能编程实现分帧函数，并恢复。
- 4、了解语音信号短时时域分析的原理。
- 5、掌握短时时域分析的一些参数计算方法。
- 6、根据原理能编程实现短时时域分析的参数计算。

## 【实验要求】

- 1、编程要求：编写一段 MATLAB 程序。
- 2、实现功能：实现语音的短时时域分析。

## 【实验原理】

语音信号的时域分析就是分析和提取语音信号的时域参数。语音信号本身就是时域信号，因而时域分析是最早使用，也是应用最广泛的一种分析方法，这种方法直接利用语音信号的时域波形。时域分析通常用于最基本的参数分析及应用，如语音的分割、预处理、分类等。语音信号的时域参数有短时能量、短时过零率、短时自相关函数和短时平均幅度差函数等。这些最基本的短时参数在各种语音信号数字处理技术中都有重要的应用。

### 1、语音分帧

贯穿于语音分析全过程的是“短时分析技术”。因为，语音信号从整体来看其特性及表征其本质特征参数均是随时间而变化的，所以它是一个非平稳态过程，不能用处理平稳信号的数字信号处理技术对其进行分析处理。但是，由于不同的语音是由人的口腔肌肉运动构成声道某种形状而产生的响应，而这种口腔肌肉运动相对于语音频率来说是非常缓慢的，所以从另一方面看，虽然语音信号具有时变特性，但是在一个短时间范围内（一般认为在 10ms~30ms 的短时间内），其特性基本保持不变即相对稳定，因而可以将其看作是一个准稳态过程，即语音信号具有短时平稳性。所以任何语音信号的分析 and 处理必须建立在“短时”的基础上，即进行“短时分析”，将语音信号分为一段一段来分析其特征参数，其中每一段称为一“帧”，帧长一般即取为 10~30ms。这样，对于整体的语音信号来讲，分析出的是由每一帧特征参数组成的特征参数时间序列。

分帧示意图如图 1-1 所示。一般每秒的帧数约为 33~100 帧，视实际情况而定。分帧虽然可以采用连续分段的方法，但一般要采用如图 2-1 所示的交叠分段的方法，这是为了使帧

与帧之间平滑过渡，保持其连续性。前一帧和后一帧的交叠部分称为帧移。帧移与帧长的比值一般取为  $0 \sim 1/2$ 。

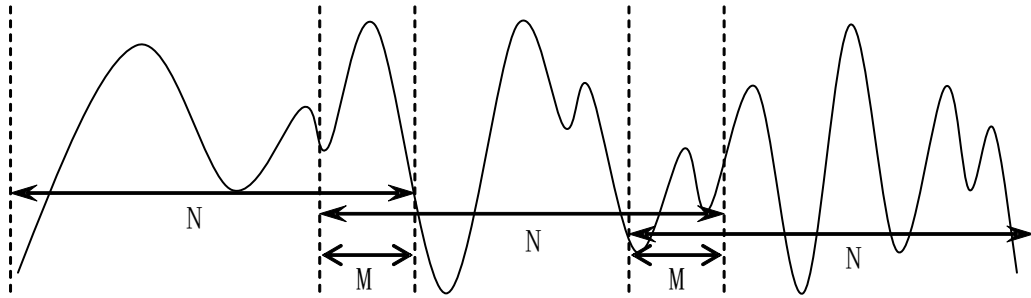


图 1-1 分帧示意图 ( $N$  为帧长,  $M$  为帧移)

分帧是用可移动的有限长度窗口进行加权的方法来实现的, 这就是用一定的窗函数  $w(n)$  来乘  $s(n)$ , 从而形成加窗语音信号  $s_w = s(n) * w(n)$ 。窗函数  $w(n)$  的选择 (形状和长度), 对于短时分析参数的特性影响很大。为此应选择合适的窗口, 使其短时参数更好地反映语音信号的特性变化。选择的依据有两类:

- 1) 窗口的形状: 一个好的窗函数的标准是: 在时域因为是语音波形乘以窗函数, 所以要减小时间窗两端的坡度, 使窗口边缘两端不引起急剧变化而平滑过渡到零, 这样可以使截取出的语音波形缓慢降为零, 减小语音帧的截断效应; 在频域要有较宽的 3dB 带宽以及较小的边带最大值。
- 2) 窗口的长度: 如果长度很大, 则它等效于很窄的低通滤波器, 语音信号通过时, 反映波形细节的高频部分被阻碍, 短时能量随时间变化很小, 不能真实的反映语音信号的幅度变化; 反之, 长度太小时, 滤波器的通带变宽, 短时能量随时间有急剧的变化, 不能得到平滑的能量函数。通常认为在一个语音帧内应包含 1~7 个基音周期。然而不同人的基音周期变化很大, 从女性和儿童的 2ms 到老年男子的 14ms (即基音频率的变化范围为 500Hz~70Hz), 所以  $N$  的选择比较困难。通常在 10kHz 取样频率下,  $N$  折中选择为 100~200 点为宜 (即 10~20ms 持续时间)。常见的窗函数如前一章所列。

## 2、短时能量

设第  $n$  帧语音信号  $x_n(m)$  的短时能量用  $E_n$  表示, 则其计算公式如下:

$$E_n = \sum_{m=0}^{N-1} x_n^2(m) \quad (1-1)$$

$E_n$  是一个度量语音信号幅度值变化的函数, 但它有一个缺陷, 即它对高电平非常敏感 (因为它计算时用的是信号的平方)。

## 【实验步骤】

1、根据语音分帧的思想，编写分帧函数。函数定义如下：

函数格式：

`frameout=enframe(x,win,inc)`

输入参数：`x` 是语音信号；`win` 是帧长或窗函数，若为窗函数，帧长便取窗函数长；`inc` 是帧移。

输出参数：`frameout` 是分帧后的数组，长度为帧长和帧数的乘积。

根据分帧后的语音，绘制连续四帧语音信号（不用窗函数），效果如图 1-2 所示。

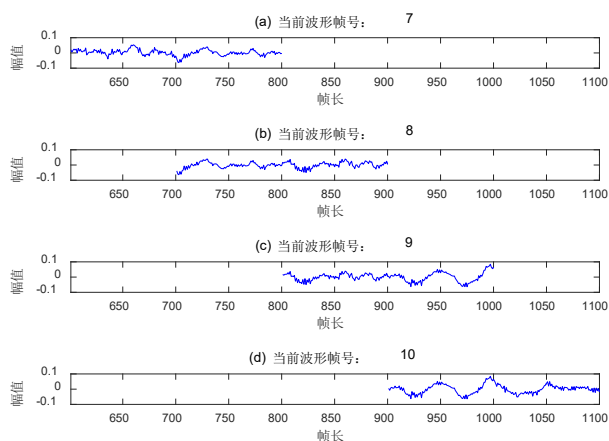


图 1-2 从第 7 帧开始连续四帧语音信号

2、编程实现矩形窗、汉明窗和汉宁窗，效果如图 1-3 所示。

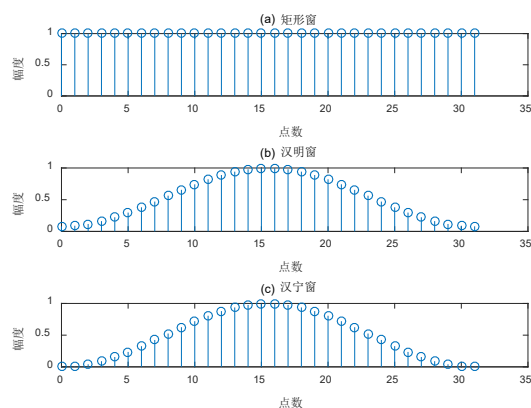


图 1-3 窗函数时域波形

a)矩形窗      b)汉明窗      c)汉宁窗

3、为了显示方便，编程实现 `FrameTimeC` 函数，函数功能为计算分帧后每帧语音中点处对应的时间。函数定义如下：

函数格式：`frametime=FrameTimeC(frameNum, framelen, inc, fs)`

输入参数：`frameNum` 是帧的个数；`framelen` 是帧长；`inc` 是帧移；`fs` 是采样频率。

输出参数：`frametime` 是分帧后每帧对应的时间。

4、编程实现短时能量、短时平均幅度和短时过零率，显示例图如图 1-4 所示。

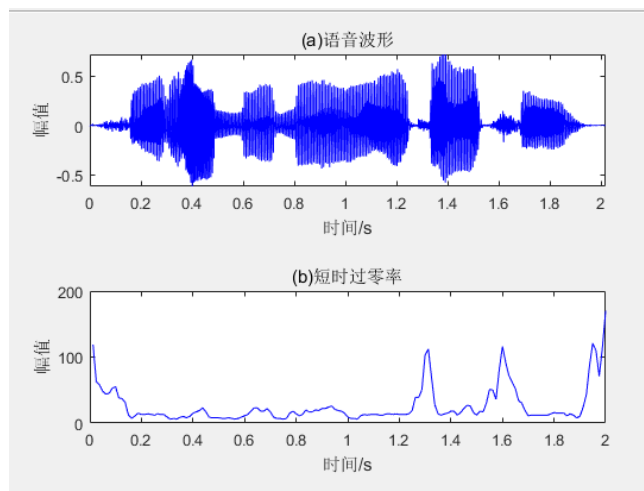


图 1-4 短时过零率的显示例图

每个参数的函数定义格式为：funcvalue=funcname(x) 其中 x 为语音信号，funcvalue 为 [1, 帧数] 的向量。

3) 编程实现短时自相关和短时平均幅度差，显示例图如图 1-5 所示。每个参数的函数定义格式为：funcvalue=funcname(x) 其中 x 为语音信号，funcvalue 为 [帧长, 帧数] 的矩阵。这里显示的语音信号，是语音帧拼接而成，即没有去掉交叠项。

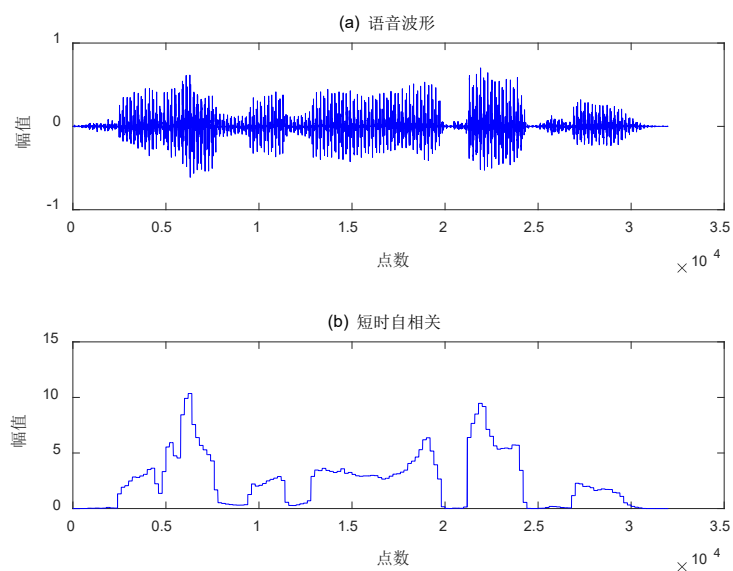


图 1-5 短时自相关的显示例图

### 【思考题】

- 1、编程实现将分帧加窗后的语音信号恢复成原始分帧前的信号。
- 2、编程比较不同的窗函数对短时时域参数估计的影响。