

实验一 语音信号采集实验

【实验目的】

- 1、了解 MATLAB 采集语音信号的原理及常用命令。
- 2、熟练掌握基于 MATLAB 的语音文件的创建、读写等基本操作。
- 3、学会使用 plot 命令来显示语音信号波形，并掌握基本的标注方法。

【实验要求】

- 1、编程要求：编写一段 MATLAB 程序。
- 2、实现功能：实现语音信号采集。

【实验原理】

1.语音信号的采集

语音信号是一种模拟信号，首先须经过采样将其转换为数字信号，实质是把连续信号变为脉冲或数字序列。我们可以用录音软件先录一段 wav 格式的音频。然后用 matlab 的 audioread 函数采集，记住采样频率和采样点。然后用 sound 函数来使用。

2.采样定理

在进行模拟与数字信号的转换过程中，当采样大于最高频率的 2 倍时，则采样之后的数字信号完整的保留了原始信号中的信息，一般实际应用中保证采样频率为信号最高频率的 5~10 倍。

3.采样频率

采样频率是指计算机每秒钟采样多少个声音样本，是描述声音文件的音质、音频、衡量声卡、声音文件的质量标准。采样频率越高，即采样的时间间隔越短，则在单位时间内计算机得到的声音样本数据越多，对声音波形的表示也越准确。

4.采样位数

采样位数即采样值或取样值,用来衡量声音波动变化的参数,是指声卡在采集和播放声音文件时所使用数字声音信号的二进制位数。采样频率是指录音设备在一秒钟内对声音信号的采样次数,采样频率越高声音的还原就越真实越自然。采样位数和采样率对于音频接口来说是最为重要的两个指标。无论采样频率如何,理论上来说采样的位数决定了音频数据最大的力度范围。采样位数越多则捕捉到的信号越精确。

【实验步骤】

- 1) 编写 MATLAB 程序实现录制语音信号“你好，欢迎”，并保存为 C1_1_y_1.wav 文件，要求采样频率为 16000Hz，采样精度 16bit。

2) 使用 `wavread` 函数读取 `C1_1_y_1.wav` 文件, 并使用 `plot` 函数显示出来。要求: 横轴和纵轴带有标注。横轴的单位为秒 (s), 纵轴显示归一化后的数值。图 1-1 为参考图例。

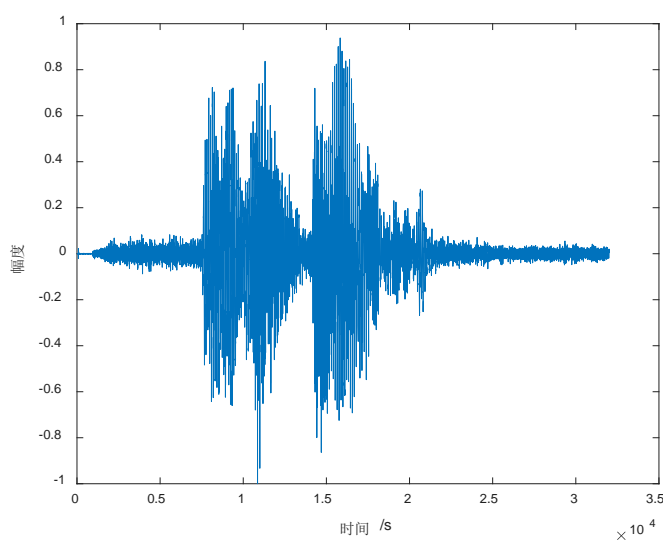


图 1-1 ‘你好, 欢迎’ 时域图

3) 使用 `wavplay` 函数播放录制的语音信号, 并改变播放的采样频率为原始采样频率的倍数, 体验效果。

【思考题】

1. 分析并解释实验要求 3) 的现象原理。
2. 自行录制一段语音, 并存储为 `wav` 文件。要求: 存储为 `wav` 文件时, 分别以采样频率、2 倍采样频率和 1/2 采样频率为三个 `wav` 文件, 并将 `plot` 函数结合 `subplot` 函数在一幅图上显示 3 个波形。横轴和纵轴带有标注。横轴的单位为秒 (s), 纵轴显示的为归一化后的数值。

实验二 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的必要性。
- 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 2-1 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 $0 \sim 1/2$ 。

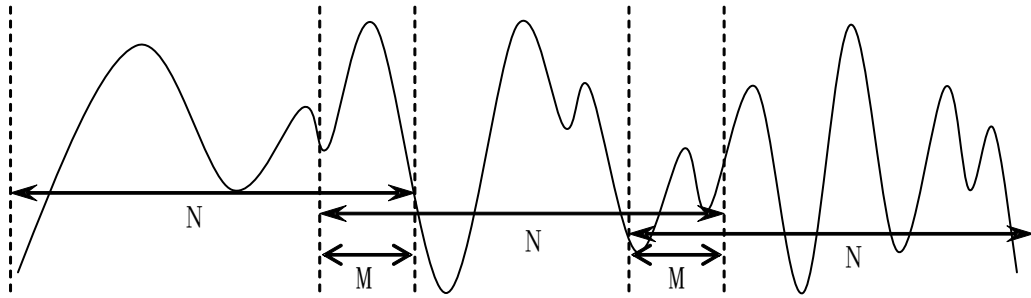


图 2-1 分帧示意图 (N 为帧长, M 为帧移)

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 $1 \sim 7$ 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：

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

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 2-2 所示。

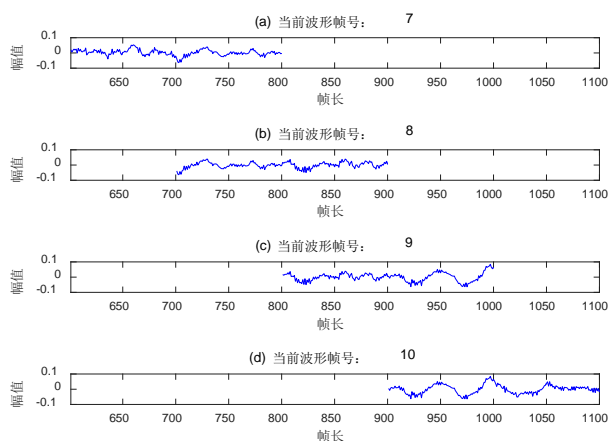


图 2-2 从第 7 帧开始连续四帧语音信号

2、编程实现矩形窗、汉明窗和汉宁窗，效果如图 2-3 所示。

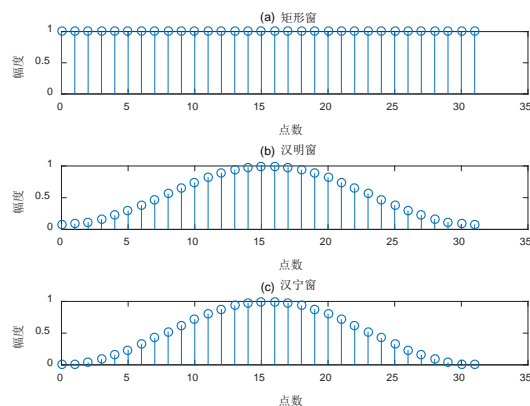


图 2-3 窗函数时域波形

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 2-4 所示。

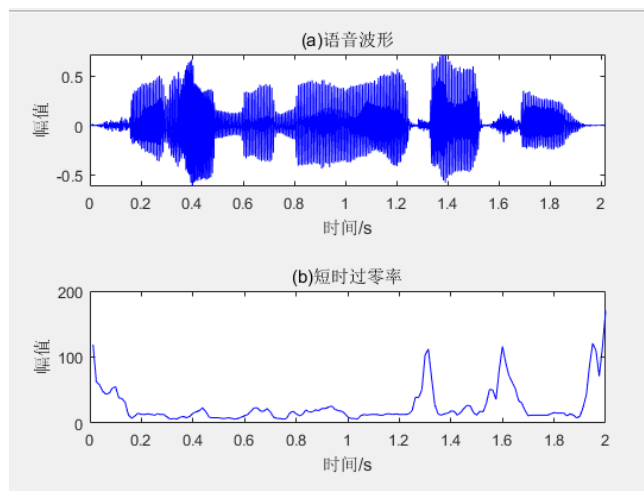


图 2-4 短时过零率的显示例图

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 2-5 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。

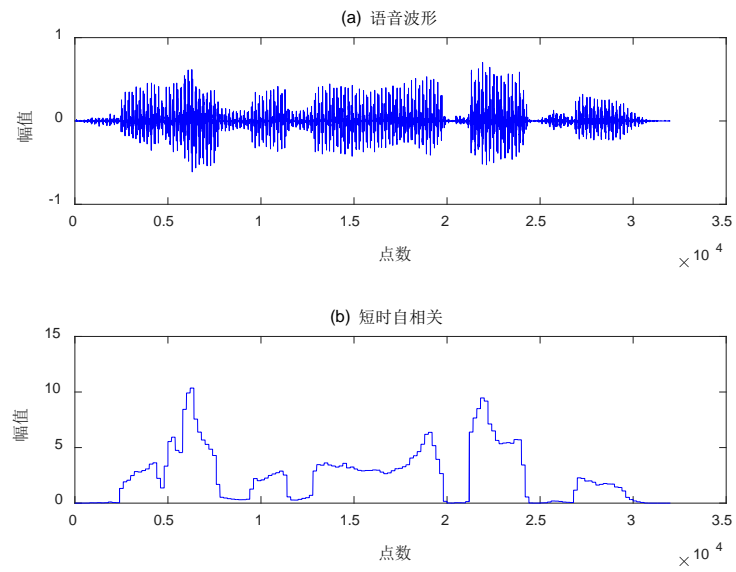


图 2-5 短时自相关的显示例图

【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。

实验三 线性预测分析

【实验目的】

- 1、了解线性预测分析在语音信号处理中的重要性和必要性。
- 2、掌握线性预测分析的基本思想。
- 3、掌握 MATLAB 进行线性预测分析的流程。

【实验要求】

- 1、编程要求：编写一段 MATLAB 程序。
- 2、实现功能：实现语音的线性预测分析（LPC）。

【实验原理】

- 1、语音信号线性预测分析

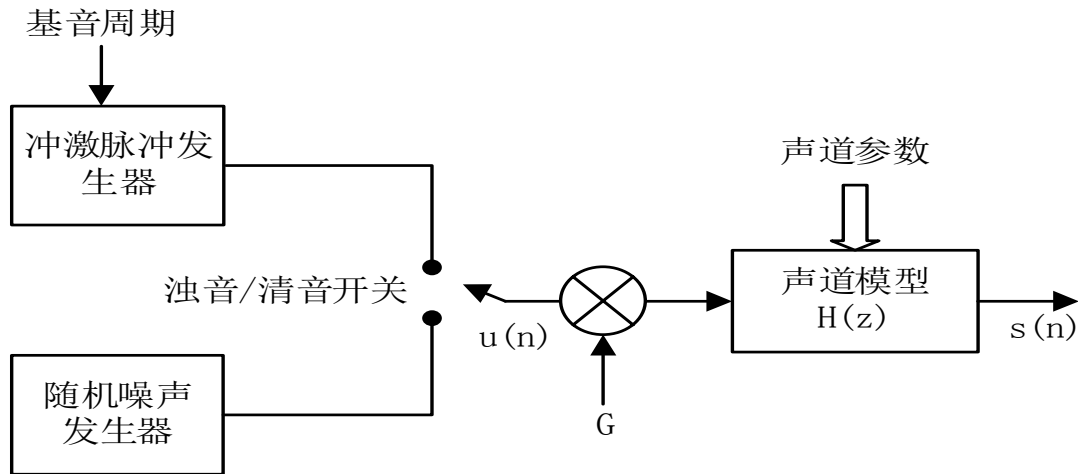


图 3-1 简化的语音产生模型

图 3-1 是简化的语音产生模型，将辐射、声道以及声门激励的全部效应简化为一个时变的数字滤波器来等效，其传递函数为

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (3-1)$$

这种表现形式称为 p 阶线性预测模型，这是一个全极点模型。

此时， $s(n)$ 和 $u(n)$ 间的关系可以用差分方程

$$s(n) = \sum_{i=1}^p a_i s(n-i) + Gu(n) \quad (3-2)$$

表示，称系统

$$\hat{s}(n) = \sum_{i=1}^p a_i s(n-i) \quad (3-3)$$

为线性预测器。 $\hat{s}(n)$ 是 $s(n)$ 的估计值，它由过去 p 个值线性组合得到的，即由 $s(n)$ 过去的值来预测或估计当前值 $s(n)$ 。式中 $a_i (i = 1, 2, \dots, p)$ 是线性预测系数。线性预测系数可

以通过在某个准则下使预测误差 $e(n)$ 达到最小值的方法来决定，预测误差的表示形式如下：

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{i=1}^p a_i s(n-i) \quad (3-4)$$

预测的二次方误差为

$$E = \sum_n e^2(n) = \sum_n [s(n) - \hat{s}(n)]^2 = \sum_n [s(n) - \sum_{i=1}^p a_i s(n-i)]^2 \quad (3-5)$$

为使 E 最小，求 E 对 a_i 的偏导为 0，即

$$\frac{\partial E}{\partial a_j} = 0 \quad (1 \leq j \leq p) \quad (3-6)$$

则有，

$$\frac{\partial E}{\partial a_j} = 2 \sum_n s(n)s(n-j) - 2 \sum_{i=1}^p a_i \sum_n s(n-i)s(n-j) = 0 \quad (1 \leq j \leq p) \quad (3-7)$$

定义 $\phi(j, i) = \sum_n s(n-i)s(n-j)$ ，则式 (3-7) 可简化为

$$\phi(j, 0) - \sum_{i=1}^p a_i \phi(j, i) = 0 \quad (1 \leq j \leq p) \quad (3-8)$$

联立式 (3-5)、式 (3-7) 和式 (3-8)，可得最小均方误差表示为

$$E = \phi(0, 0) - \sum_{i=1}^p a_i \phi(0, i) \quad (3-9)$$

因此，最小误差有一个固定分量 $\phi(0, 0)$ 和一个依赖于预测系数的分量 $\sum_{i=1}^p a_i \phi(0, i)$ 构成。

为求解最佳预测器系数，必须首先求出 $\phi(j, i) (i, j \in [1, p])$ ，然后可按照式 (3-8) 进行求解。

很显然， $\phi(j, i)$ 的计算及方程组的求解都是十分复杂的。

2、线性预测分析的自相关解法

为了有效地进行线性预测分析，求得线性预测系数有必要用一种高效的方法来求解线性方程组。虽然可以用各种各样的方法来解包含 p 个未知数的 p 个线性方程，但是系数矩阵的特殊性质使得解方程的效率比普通解法的效率要高得多。自相关法是经典解法之一，其原理是在整个时间范围内使误差最小，即设 $s(n)$ 在 $0 \leq n \leq N-1$ 以外等于 0，等同于假设 $s(n)$ 经过有限长度的窗（如矩形窗、海宁窗或汉明窗）的处理。

通常， $s(n)$ 的加窗自相关函数定义为

$$r(j) = \sum_{n=0}^{N-1} s(n)s(n-j) \quad 1 \leq j \leq p \quad (3-10)$$

同式 (3-8) 比较可知， $\phi(j, i)$ 等效为 $r(j-i)$ 。但是由于 $r(j)$ 为偶函数，因此 $\phi(j, i)$ 可表示为

$$\phi(j, i) = r(|j-i|) \quad (3-11)$$

此时式 (3-8) 可表示为

$$\sum_{i=1}^p a_i r(|j-i|) = r(j) \quad 1 \leq j \leq p \quad (3-12)$$

则最小均方误差改写为

$$E = r(0) - \sum_{i=1}^p a_i r(i) \quad (3-13)$$

展开式 (3-11), 可得方程组为

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ r(2) & r(1) & r(0) & \cdots & r(p-3) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(p) \end{bmatrix} \quad (3-14)$$

式 (3-14) 左边为相关函数的矩阵, 以对角线为对称, 其主对角线以及和主对角线平行的任何一条斜线上所有的元素相等。这种矩阵称为托普利兹 (Toeplitz) 矩阵, 而这种方程称为 Yule-Walker 方程。对于式 (3-14) 的矩阵方程无需像求解一般矩阵方程那样进行大量的计算, 利用托普利兹矩阵的性质可以得到求解这种方程的一种高效方法。

这种矩阵方程组可以采用递归方法求解, 其基本思想是递归解法分布进行。在递推算法中, 最常用的是莱文逊-杜宾 (Levinson-Durbin) 算法 (如图 3-2 所示)。

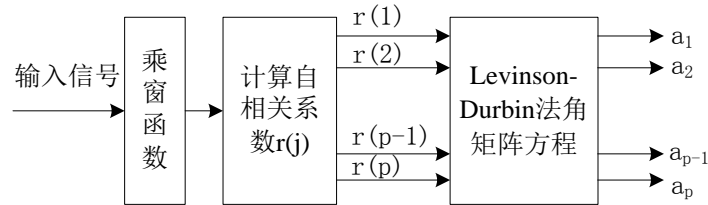


图 3-2 自相关解法

算法的过程和步骤为:

① 当 $i=0$ 时, $E_0 = r(0), a_0 = 1$ (3-15)

② 对于第 i 次递归 ($i = 1, 2, \dots, p$):

i
$$k_i = \frac{1}{E_{i-1}} \left[r(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r(j-i) \right]$$
 (3-16)

ii
$$a_i^{(i)} = k_i$$
 (3-17)

iii 对于 $j=1$ 到 $i-1$
$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}$$
 (3-18)

iv
$$E_i = (1 - k_i^2) E_{i-1}$$
 (3-19)

③ 增益 G 为

$$G = \sqrt{E_p} \quad (3-20)$$

通过对式 (3-16) ~ 式 (3-18) 进行递推求解, 可获得最终解为

$$a_i = a_j^{(p)} \quad 1 \leq j \leq p \quad (3-21)$$

由式 (3-19) 可得

$$E_p = r(0) \prod_{i=1}^p (1 - k_i^2) \quad (3-22)$$

由式 (3-22) 可知, 最小均方误差 E_p 一定要大于 0, 且随着预测器阶数的增加而减小。因此每一步算出的预测误差总是小于前一步的预测误差。这就表明, 虽然预测器的精度会随着阶数的增加而提高, 但误差永远不会消除。由式 (3-22) 还可知, 参数 k_i 一定满足

$$|k_i| < 1, \quad 1 \leq i \leq p \quad (3-23)$$

由递归算法可知, 每一步计算都与 k_i 有关, 说明这个系数具有特殊的意义, 通常称之为反射系数或偏相关系数。可以证明, 它就是多项式 $A(z)$ 的根在单位圆内的充分必要条件, 因此它可以保证系统 $H(k)$ 的稳定性。

3、线性预测的其它参数

用线性预测分析法求得的是一个全极点模型的传递函数。在语音产生模型中, 这一全极点模型与声道滤波器的假设相符合, 而形式上是一自回归滤波器。用全极点模型所表征的声道滤波器, 除预测系数 $\{a_i\}$ 外, 还有其他不同形式的滤波器参数。这些参数一般可由线性预测系数推导得到, 但各有不同的物理意义和特性。在对语音信号做进一步处理时, 为了达到不同的应用目的时, 往往按照这些特性来选择某种合适的参数来描述语音信号。

1) 预测误差及其自相关函数

由式 (3-4) 可知, 预测误差为

$$e(n) = s(n) - \sum_{i=1}^p a_i s(n-i) \quad (3-24)$$

而预测误差的自相关函数为

$$R_e(m) = \sum_{n=0}^{N-1-m} e(n)e(n+m) \quad (3-25)$$

2) 反射系数和声道面积

反射系数 $\{k_i\}$ 在低速率语音编码、语音合成、语音识别和说话人识别等许多领域都是非常重要的特征参数。由式 (3-18) 可得:

$$\begin{cases} a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \\ a_{i-j}^{(i)} = a_{i-j}^{(i-1)} - k_i a_j^{(i-1)} \end{cases} \quad j = 1, \dots, i-1 \quad (3-26)$$

进一步推导，可得：

$$a_j^{(i-1)} = (a_j^{(i)} + a_j^{(i)} a_{i-j}^{(i)}) / (1 - k_i^2) \quad j = 1, \dots, i-1 \quad (3-27)$$

由线性预测系数 $\{a_i\}$ 可递推出反射系数 $\{k_i\}$ ，即

$$\begin{cases} a_j^{(p)} = a_j & j = 1, 2, \dots, p \\ k_i = a_i^{(i)} \\ a_j^{(i-1)} = (a_j^{(i)} + a_j^{(i)} a_{i-j}^{(i)}) / (1 - k_i^2) & j = 1, \dots, i-1 \end{cases} \quad (3-28)$$

反射系数的取值范围为 $[-1, 1]$ ，这是保证相应的系统函数稳定的充分必要条件。从声学理论可知，声道可以被模拟成一系列截面积不等的无损声道的级联。反射系数 $\{k_i\}$ 反映了声波在各管道边界处的反射量，有：

$$k_i = \frac{A_{i+1} - A_i}{A_{i+1} + A_i} \quad (3-29)$$

式中， A_i 是第 i 节声管的面积函数。式 (7-29) 经变换后，可得声管模型各节的面积比为：

$$\frac{A_i}{A_{i+1}} = \frac{(1 - k_i)}{(1 + k_i)} \quad (3-30)$$

3) 线性预测的频谱

由式 (3-1) 可知，一帧语音信号 $x(n)$ 模型可化为一个 p 阶的线性预测模型。当 $z = e^{j\omega}$ 时，能得到线性预测系数的频谱（令 $G=1$ ）：

$$H(e^{j\omega}) = \frac{1}{1 - \sum_{n=1}^p a_n z^{-j\omega n}} \quad (3-31)$$

根据语音信号的数字模型，在不考虑激励和辐射时， $H(e^{j\omega})$ 即 $X(e^{j\omega})$ 的频谱的包络谱。线性预测系数的频谱勾画出了 FFT 频谱的包络，反映了声道的共振峰的结构。

【实验步骤】

1、根据莱文逊-杜宾自相关法求线性预测系数的原理，编写 MATLAB 函数，并与 MATLAB 自带的 LPC 函数进行比较，测试效果如图 3-3 所示。

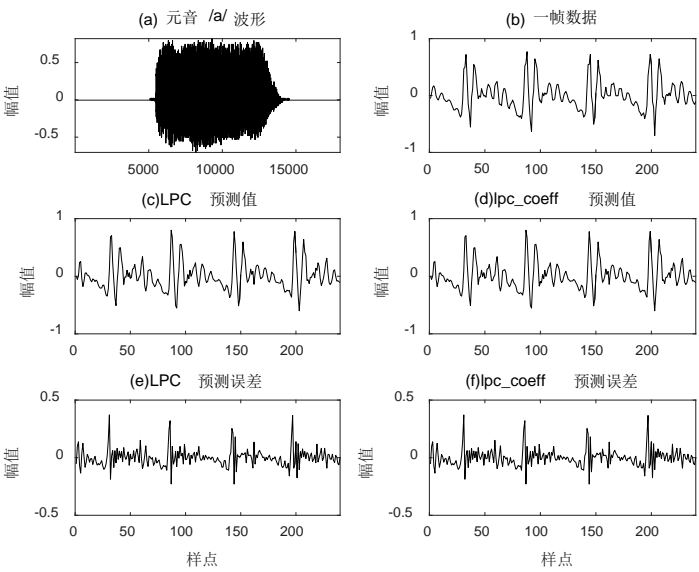


图 3-3 线性预测系数对比

函数定义如下：

名称：lpc_coeff

功能：用莱文逊-杜宾自相关法计算线性预测系数。

调用格式：

[ar,G]=lpc_coeff(s,p)

说明：输入参数 s 是一帧数据； P 是线性预测阶数。输出参数 ar 是按公式 $E_i = (1 - k_i^2)E_{i-1}$ 计算得到的预测系数 $\{a_i\}$ ($i=1, 2, \dots, p$)，共得 p 个预测系数； G 是按公式 $G = \sqrt{E_p}$ 计算得到的增益系数。

2、编写求取 LPC 预测系数的复频谱函数，并与 FFT 频谱进行对比，测试效果如图 3-4 所示。

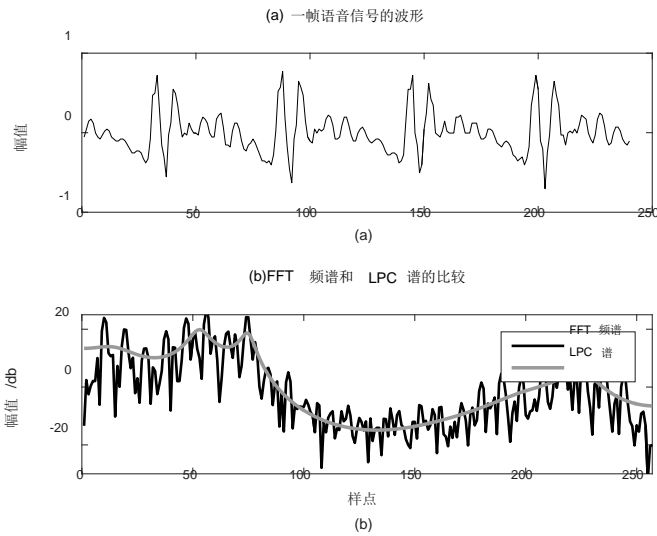


图 3-4 LPC 预测系数的复频谱与 FFT 频谱

LPC 预测系数的复频谱函数的定义如下：

名称：lpcff

功能：计算线性预测系数的复频谱。

调用格式：

$ff=lpcff(ar,np)$

说明：ar 是线性预测系数，np 是 FFT 阶数；输出 ff 是线性预测系数的复倒谱。

3、编程实现由预测系数求取线性预测倒谱系数，并比较与 rceps 函数的区别，测试效果如图 3-5 所示。

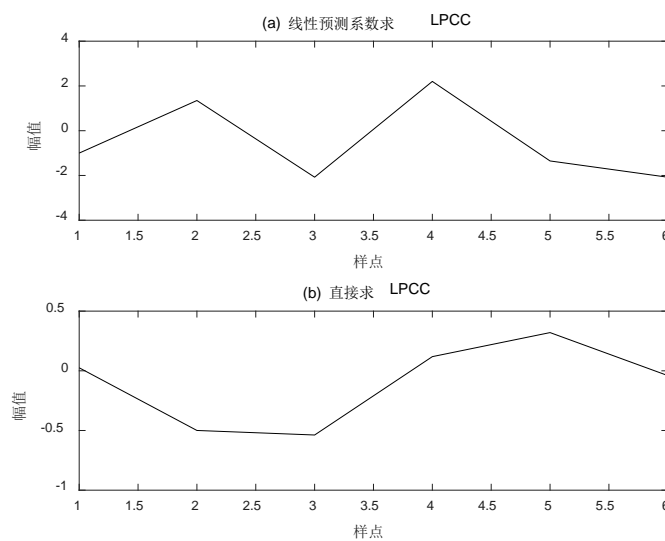


图 3-5 LPCC 比较

名称：lpc_lpccm

功能：由预测系数 ai 求 LPCC。

调用格式：

$lpcc=lpc_lpccm(ar,n_lpc,n_lpcc)$

说明：输入参数 ar 是线性预测系数；n_lpc 是预测系数的长度；n_lpcc 是 LPC 倒谱的长度；输出参数 lpcc 是线性预测倒谱系数。

【思考题】

编程实现预测系数与反射系数间的转换和反射系数与声管面积间的转换。函数定义如下。

(1)由预测系数求反射系数

名称：lpcar2rf

功能：已知预测系数计算出反射系数

调用格式：

$rf=lpcar2rf(ar)$

说明：ar 是预测系数；rf 是反射系数。

(2)由反射系数求出预测系数

名称: lpcrf2ar

功能: 已知反射系数计算出预测系数

调用格式:

$[ar, arp, aru, g] = lpcrf2ar(rf)$

说明: rf 是反射系数; ar 是预测系数; arp 是压力传递函数; aru 是体积速度的传递函数; g 是增益。

(3) 由反射系数求声管面积比

名称: lpcrf2ao

功能: 已知反射系数计算出归正化的声管面积。

调用格式:

$ao = lpcrf2ao(rf)$

说明: rf 是反射系数; ao 是声管面积比。

(4) 由声管面积求反射系数

名称: lpcao2rf

功能: 已知归正化的声管面积比计算出反射系数

调用格式:

$rf = lpcao2rf(ao)$

说明: ao 是声管面积比; rf 是反射系数。

实验四 LPC 编解码实验

【实验目的】

- 1、了解语音信号 LPC 编码的原理
- 2、掌握 LPC 编解码的步骤流程
- 3、根据原理能编程实现 LPC 编码和解码的计算

【实验要求】

- 1、编程要求：编写一段 MATLAB 程序
- 2、实现功能：实现语音信号的 LPC 编码和解码

【实验原理】

线性预测编码（linear predictive coding, LPC）是运用于音频信号处理与语音处理的压缩编码方式，根据线性预测模型的信息表示数字语音信号谱包络。它是最有效的语音分析技术之一，也是低位速下编码方法高质量语音最有用的方法之一，能够提供非常精确的语音参数预测。线性预测编码通过估计共振峰剔除它们在语音信号中的作用，估计保留的蜂鸣音强度与频率来分析语音信号；同时，使用蜂鸣参数与残余信号生成源信号，使用共振峰生成表示声道的滤波器，源信号经过滤波器的处理来逆向合成语音信号。由于语音信号随着时间变化，这个过程是在一段段的语音信号帧上进行处理，通常每秒 30 到 50 帧就能对可理解的信号进行很好的压缩。

线性预测编码通常用于语音的重新合成，它是电话公司使用的声音压缩格式，如 GSM 标准就在使用 LPC 编码格式。它还用作安全无线通信中的格式，在安全的无线通信中，声音必须进行数字化、加密然后通过狭窄的语音信道传输。

线性预测分析的基本思想是：由于语音样点之间存在相关性，所以可以用过去的样点值来预测现在或将来的样点值，即一个语音抽样可以用过去若干个语音抽样或它们的线性组合来逼近。通过使实现语音抽样与线性预测抽样之间的误差在某个准则（通常为最小均方误差准则）下达到最小值来决定一组预测系数。这一组预测系数就反映了语音信号的特性，可以作为语音信号的特征参数用于语音合成和语音识别等。

下面对 LPC 的基本原理和计算过程作一个介绍。

1、线性预测基本原理

线性预测分析对语音的产生过程有一个基本的假设，即认为语音是由一个激励信号通过一个滤波器（响应函数）得到。通过对声道模型的研究，可以认为系统的传递函数符合全极点数字滤波器的形式。系统的传递函数如下：

$$H(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (4-1)$$

其中， p 是极点个数即模型结束， G 是幅值因子， a_k 是模型系数，由 p 和 a_k 决定了声道特性，描述了说话人的特征。对于一个线性预测系统，采样点的输出 $s(n)$ 可以用前面 p 个样本的线性组合来表示，定义系统输出的估计值为：

$$\tilde{s}(n) = \sum_{i=1}^p a_i s(n-i) \quad (4-2)$$

其中系数 a_i 为预测系数， p 为预测阶数。预测误差表示如下：

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{i=1}^p a_i s(n-i) \quad (4-3)$$

定义短时预测均方误差为：

$$E_n = \sum_n e^2(n) = \sum_n [s(n) - \sum_{i=1}^p a_i s(n-i)]^2 \quad (4-4)$$

显然，均方误差越接近于 0，预测的准确度在均方误差最小的意义上为最佳。因此，应满足 E_n 对于各系数 a_i 的偏微分为 0，这样计算可得到线性预测的标准方程为：

$$\sum_n s(n)s(n-j) = \sum_{i=1}^p a_i s(n-i)s(n-j), 1 \leq j \leq p \quad (4-5)$$

可定义：

$$\phi(j, i) = \sum_n s(n-j)s(n-i) \quad (4-6)$$

则上式可简写为：

$$\sum_{i=1}^p a_i \phi(j, i) = \phi(j, 0) \quad (4-7)$$

求解含有 p 个未知数的方程组可得到各预测系数，利用上式，可得最小均方误差为：

$$E_n = \sum_n s^2(n) - \sum_{i=1}^p a_i \sum_n s(n)s(n-i) \quad (4-8)$$

因此，最小误差由一个固定分量和一个依赖于预测系数的分量组成，求解最佳预测系数，需要首先计算 $\phi(i, j)$ ，然后可按上式求出 a_i 。

2、线性预测系数的计算

一般来说，可以采用莱文逊-杜宾（Levinson-Durbin）递归算法来求解线性预测系数，假设 $s(n)$ 在 $[0, N-1]$ 外等于 0，即 $s(n)$ 经过有限长度的窗处理。

$s(n)$ 的自相关函数为:

$$r(j) = \sum_{n=-\infty}^{\infty} s(n)s(n-j) = \sum_{n=0}^{N-1} s(n)s(n-j) \quad (4-9)$$

由上可知, $\phi(j,i)$ 即为 $r(j-i)$ 。 $r(j-i)$ 只与 j 和 i 的相对大小有关, 与 i,j 的取值无关, 因此:

$$\phi(j,i) = r(|j-i|) \quad (4-10)$$

这是一个含 Toeplitz 矩阵的 Yule-Walker 方程。利用 Toeplitz 矩阵的性质可得到求解 Yule-Walker 方程的高效解法—杜宾算法。其迭代过程如下:

(1) 计算自相关系数 $r_n(j)$;

(2) $E^{(0)} = r_n(0)$;

(3) $i = 1$;

(4) 开始按公式 $k_i = \frac{r_n(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} r_n(i-j)}{E^{(i-1)}}$ 进行递推计算;

(5) 使 $i = i+1, i > p$, 则算法结束退出, 否则返回(4), 按上式进行递推。

递推的最终解为:

$$a_j = a_j^{(p)}, 1 \leq j \leq p \quad (4-11)$$

$$E^{(p)} = r_n(0) \prod_{i=1}^p (1 - k_i^2) \quad (4-12)$$

通过 LPC 分析, 由若干帧语音可得到若干组 LPC 参数, 每组参数形成一个描绘该帧语音特征的矢量, 即 LPC 矢量。

LPC 参数是模拟人的发声器官的, 是一种基于语音合成的参数模型, 每段声管对应一个 LPC 模型的极点。一般情况下, 极点个数在 12-16 之间, 就可以足够清晰地描述语音信号的特征了。选择 $p=12$ 可以对绝大多数的语音信号声道模型取得足够近似的逼近。 p 值选择过大虽可以改善逼近效果, 但会增大计算量, 且可能增添一些不必要的细节, 如在用声道模型谱进行共振峰分析时效果变差等。

【实验步骤】

1、为了显示方便, 分别编写编码、解码函数程序 encode1.m、decode1.m。encode1.m 对输入的原模拟信号进行线性预测编码, decode1.m 对生成的 LPC 码进行解码恢复出原信号。具体函数定义如下。

函数格式: [ipitch, irms, irc] = encode1(pitcha, rms, kk)

输入参数: pitcha 是基音频率, rms 表示均方误差, kk 表示预测系数。

输出参数: ipitch 为已编码的基音频率, irms 为量化的均方误差, irc 为量化的预测系数。

函数格式: [voice, pitcha, rms, kk]=decodel(ipitch, irms, irc, rcn)

输入参数: ipitch 为已编码的基音频率, irms 为量化的均方误差, irc 为量化的预测系数, rcn 为系数阶数。

输出参数: voice 为半帧音频信号, pitcha 为解码得到的基音频率, rms 为得到的均方误差, kk 为预测系数。

2、编程实现对信号的编码和解码过程, 波形如图 4-1 所示。

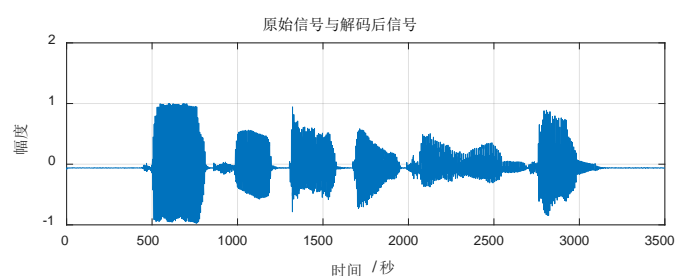


图 4-1 LPC 编解码对比图

【思考题】

编程比较不同的线性预测阶数对 LPC 编解码后语音恢复的影响。

实验五 基于线性预测系数的语音合成

【实验目的】

1、掌握基于线性预测系数与基音参数语音合成的原理

【实验要求】

1、编程要求：编写一段 MATLAB 程序

2、实现功能：实现基于线性预测系数和基音参数的语音合成

【实验原理】

1、预测系数和基音参数语音合成模型

相较于前一实验线性预测模型也可设计成一种源滤波器模型,即由白噪声序列和周期性激励脉冲序列构成的激励源信号,经过选通、放大并通过时变数字滤波器(由语音参数控制的声道模型),获得合成语音信号。语音合成器的示意图如图 5-1 所示。

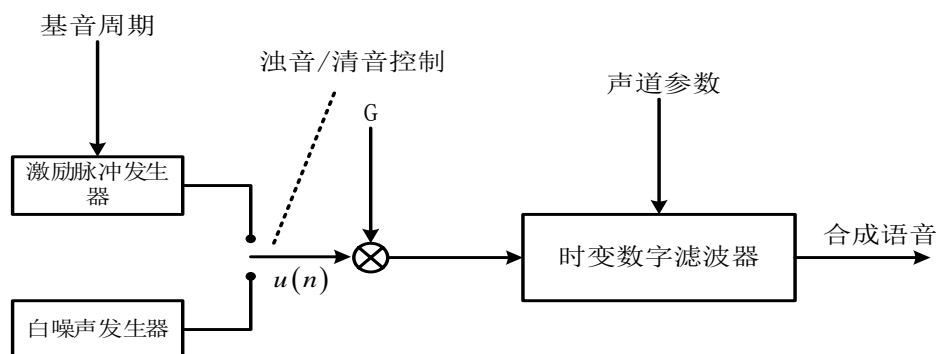


图 5-1 线性预测合成语音模型示意图

图 5-1 所示的线性预测合成语音模型可直接用预测器系数 a_i 构成的递归合成滤波器,其结构如图 5-2 所示。用这种方法定时地改变激励参数 $u(n)$ 和预测系数 a_i ,就能合成出语音。这种结构简单而直观,为了合成一个语音样本,需要进行 p 次乘法和 p 次加法。它合成的语音信号序列为:

$$\tilde{x}(n) = \sum_{i=1}^p a_i \tilde{x}(n-i) + Gu(n) \quad (5-1)$$

式中, a_i 为预测系数; G 为模型增益; $u(n)$ 为激励源信号(白噪声或周期性激励脉冲序列); p 为预测阶数。

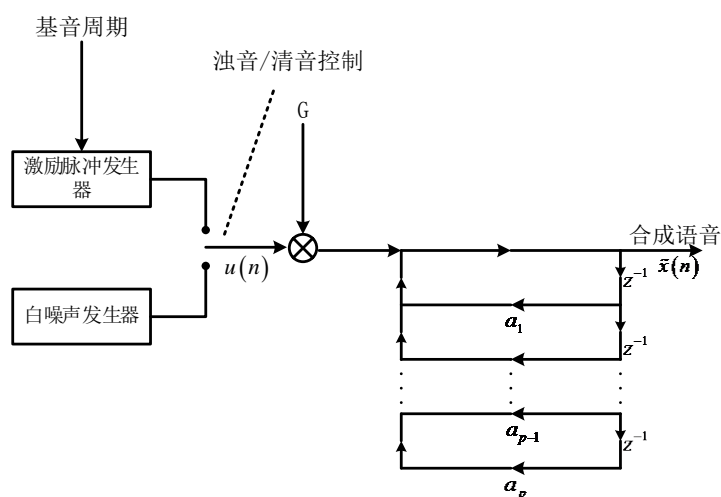


图 5-2 线性预测递归型合成滤波器的语音合成示意图

2、激励脉冲的产生

激励脉冲的产生主要难点在于一帧一帧的连续产生激励脉冲，即要求本帧和上一帧之间激励脉冲序列要连续，同时这一帧和上一帧之间的帧移是 inc 。这就要求本帧的第一个脉冲与上一帧的帧移区内最后一个脉冲之间的间隔要等于本帧的基音周期。

【实验步骤】

1、基于线性预测系数和预测误差的语音合成原理，结合以前所学的内容，基于测试语音，编程实现语音合成。效果如图 5-3 所示。

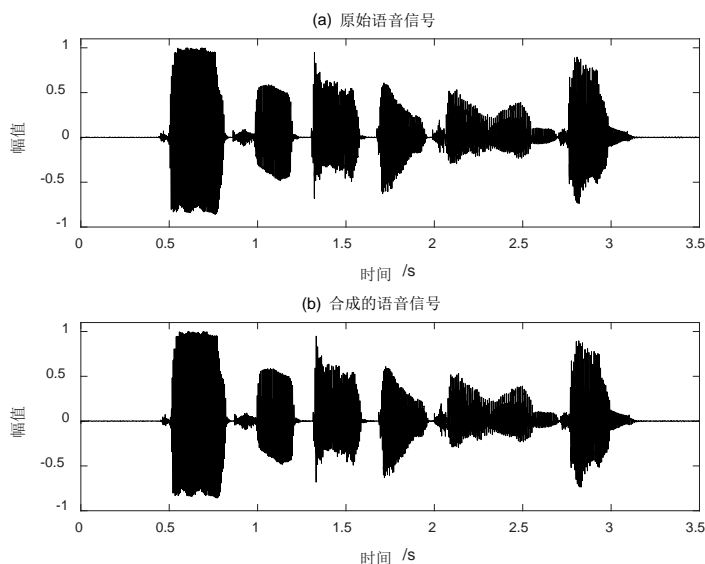


图 5-3 基于线性预测系数和预测误差的语音合成效果图

2、基于线性预测系数和基音参数的语音合成原理，结合以前所学的内容，调用基音检测函数 `pitch_Ceps`、滤波处理函数 `pitfilterml` 等，编程实现语音合成。效果如图 5-4 所示。

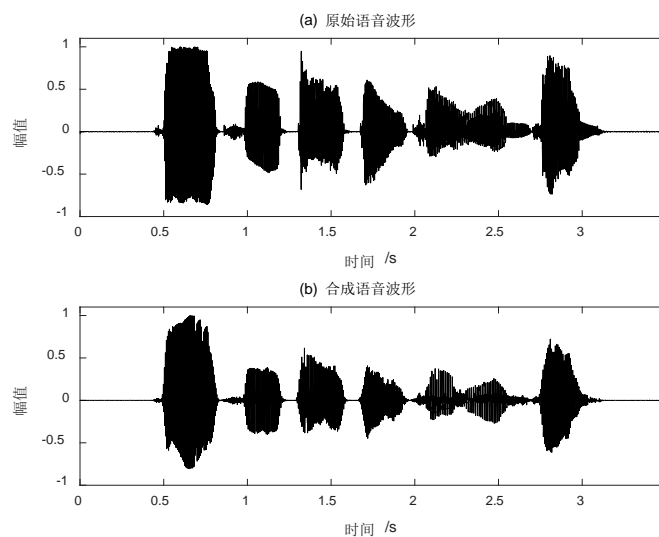


图 5-4 基音线性预测系数和基音参数的语音合成效果图

【思考题】

调用不同的基于检测函数，并比较语音合成的效果。