浙江大学

硕士学位论文

汉语普通话韵律合成的研究

姓名：张后旗

申请学位级别：硕士

专业：电路与系统

指导教师：张礼和

2001.1.1

摘要

本文首先论述了语音信号生成的准稳态模型，详细阐述了基于短时傅里叶变换进行韵律参数修改的基本步骤及其对模型参数时变进程产生的影响。

在此基础上，文中分别从时域和频域研究了时间尺度修改和基音尺度修改的理论依据，并采取相应的方法在计算机中加以实现，针对不同的结果，从时域波形和频域语谱两个角度探讨不同的方法对合成信号质量的影响。

最后，引进国际上八十年代末出现的时域基音同步叠加算法，结合汉语普通话的韵律特点，研究一种实现高自然度，高清晰度的汉语普通话韵律合成的方法。根据合成结果，分别对合成语音的质量及该算法的韵律参数控制能力加以评价，以示其有效性、实用性。

关键词： 语音信号处理；时间尺度修改；基音尺度修改；韵律修改；基音同步叠加。

Research On Prosodic Synthesis Of Chinese Speech

Abstract

In this paper, we firstly reviewed the quasi-stationary model of speech production, expounded the fundamental steps of prosodic modification methods based on Short Time Fourier Transform and its effect on speech model parameters.

Secondly, at time-domain and frequency-domain, we studied the theoretic basis of time-scale and pitch-scale modification respectively, various methods were used to realize all these prosodic parameters modification. According to results, we discussed the effect on quality of synthetic speech produced by various prosodic modification method from time-domain signal waveform and frequency-domain spectrogram respectively.

Finally, we introduced TD-PSOLA algorithm. At the basis of Chinese speech prosodic feature, a high-quality Chinese speech synthesis method using TD-PSOLA algorithm was put forward and realized. The synthetic speech prosodic parameters were analyzed and compared with objective prosodic parameters, the waveform and spectrogram between original speech and synthetic speech were compared too, according to these comparison, the ability of this method to control prosodic parameters and its effect on synthetic speech quality was evaluated.

Key words: speech signal processing; time-scale modification; pitch-scale modification; prosodic modification; pitch synchronous overlap add .

第一章 绪论

§1.1 课题研究的意义

在信息科学和计算机科学迅速发展的今天，言语工程技术受到前所未有的重视，它和数字信号处理、计算机、人工智能等技术学科以及语言学、语音学、生理学、心理学等基础学科都有密切的联系，因此受到广泛的关注。

语音合成技术是言语工程技术的一个重要组成部分，它不仅在人机通讯中充当重要角色，而且对语音的产生和感知模型也有十分重要的意义。言语作为人类进化的最重要标志，是人类社会千万年沿用下来的最常用的通信手段，自然也是人机通信最理想的方式。让计算机也能像人一样会说话和听懂人说的话是人们长期追求的目标。语音合成和语音识别一样是备受青睐的新技术，它可以应用于信息发布系统、语音应答系统、电子邮件中的语音服务以及残疾人语音辅助等方面。科学技术发展到今天，语音合成主要通过计算机来实现，它的功能是将存储在计算机中以文字表达的信息转换成言语的形式输出，因此现代的语音合成系统实质上是一个文语转换系统（Text To Speech）。

语音合成技术经历了一个逐步发展的过程，从参数合成到拼接合成，再到将两者结合[3]。近二十年来，国内外先后已有不少的商用文语转换系统进入市场，如八十年代的 Speech Plus Prose-2000(1982)、DECTalk(1983)、INFOVOX（1983）、Conversant System(1987)等系统，但由于合成语音的自然度和可懂度不高而难以销售。八十年代中期，F.Charpentier 等提出基音同步叠加技术(Pitch-Synchronous-Overlap-Add，PSOLA），既能保持发音的主要音段特征，又能在拼接时灵活调节其音高和音长等韵律特征，给波形拼接技术带来了新生。基于 PSOLA 技术，已开发了法语、德语、日语、意大利语和英语等多种语言的文语转换系统，这些系统的可懂度和自然度都相当高，已逐步接近应用目标

八十年代初，我国开始汉语文语转换系统的研究，虽然起步较晚，但发展迅速，中科院声学所、中国社会科学院语言所、清华大学等都先后研制出了汉语文语转换系统，但都因输出的语音质量问题而不能达到应用要求[1，2，3]。对于一个 TTS 系统，其功能模块可分为文本分析、韵律建模和语音合成三大模块，其中语音合成是 TTS 最基本、最重要的模块，为此，采用好的语音合成方法是研制高质量的 TTS 系统的关键所在。作者以此为研究目标，希望能运用好的语音合成方法，提高汉语普通话语音合成质量，为促进汉语 TTS 系统走向应用做些有益的工作。

§1.2 本文的研究目标及所做的工作

本文的研究目标是提出一种好的方法实现汉语普通话的语音韵律合成，从语音信号生成数字信号模型入手，分别讨论时间尺度的韵律参数修改，基频尺度的韵律参数修改，最后提出实现高质汉语普通话韵律合成的方法，同时尽量减少存储量和运算开销，促使这种合成方法具有实际应用价值。

本文所做的具体工作有：①根据从数字信号处理角度建立的语音信号生成模型，阐明从时域和频域进行时间尺度修改的理论根据并加以实现，将时域和频域修改的结果与原始信号进行比较，给出不同方法各自的优缺点。②阐明从时域和频域进行基频尺度修改的理论根据并加以实现。③引进国际上八十年代末出现的时域 PSOLA 方法，以现有的汉语普通话音系特点为基础，研究运用 TD-PSOLA 实现汉语普通话韵律合成的具体方法并加以实现。对实际广播录音的韵律参数进行分析，其结果作为韵律合成的目标韵律参数，将单独录下的孤立音节运用韵律合成技术拼接成句。通过对合成语句和样句的语调特征进行比较，评估该算法的韵律参数控制能力。通过比较音库中孤立音节与合成成句后对应的音节进行比较，给出合成后　语音的质量（音节清晰度和可懂度）评估。

§1.3 论文的结构

本文第二章介绍了语音生成的模型以及基于短时傅里叶变换（STFT)方法进行韵律修改的一般方法和过程。第三章分别阐述了 STFT 方法（频域）和 WSOLA 方法（时域）实现时间尺度修改的原理及具体实现的结果。第四章分别阐述了 STFT 方法（频域）和 TD-PSOLA 方法（时域）实现基频尺度修改的原理及具体实现的结果。第五章首先介绍了汉语普通话的音系及韵律特点，然后根据汉语普通话的音系及韵律特点，提出运用 TD-PSOLA 实现汉语普通话韵律合成的具体方法，并根据合成结果对该算法的控制韵律参数能力和合成语句质量进行评估。第六章对论文工作进行总结。

第二章 语音信号生成模型及基于 STFT 的韵律修改

本章将要从数字信号处理角度给出语音信号的生成模型，并以此模型为基础，阐述基于短时傅里叶变换基础进行时间尺度和基频尺度修改的一般方法和理论依据。

§2.1 语音生成模型

2.1.1 准稳态模型

根据目前广为接受的语音生成模型[4,8,9,10],采样的语音波形被看成是一个激励信号经一个时变线性滤波器后的输出结果。其激励信号要么是瞬时频率各谐波窄带信号的和（浊音），要么是一个准稳态随机序列，其功率谱平直（清音）。

我们主要研究浊音，用时变的滤波器模拟下列两因素共同的影响[7,8,9,10,11]:

(1)声门以上部分的传输特性（包括唇辐射）

(2)声门波脉冲形状

这个系统的输入输出特性以它时变的单位样值响应$g_n(m)$来表示。$g_n(m)$定义为系统在时刻 n 的单位样值响应，还可以用以下等价的方法来定义，即$g_n(m)$关于 m 的傅里叶变换方法:

$$\sum_{m=-\infty}^{\infty} g_n(m)\exp(-j\omega m) = G(n,\omega)\exp(j\psi(n,\omega)) \qquad (2.1)$$

$G(n,\omega)$表示时变的系统传输函数幅度，$\psi(n,\omega)$表示时变的系统传输函数相位。

$g_n(m)$的非稳定性对应于发音器官的物理运动，通常较时变的语音波形缓慢，在$g_n(m)$所记忆的时长范围内，可以认为它是不变的，即$g_n(m)$是一个准稳态系统。对于浊音来说，激励信号波形$e(n)$表示为一系列谐次相关的复指数和，这些复指数都是单位幅度，零初始相位，以及有一个缓变的基频函数$n \to 2\pi / p(n)$,其中$p(n)$是当时的基音周期，函数$n \to 2\pi / p(n)$被称为基音轮廓

(pitch contour)。

$$e(n) = \sum_{k=0}^{p(n)-1} \exp[j(\phi_k(n))] \qquad (2.2)$$

其中$\phi_k(n)$为激励的 k 次谐波的相位，它定义为时变的谐波角频率$\omega_k(n)$的积分，即

$$\phi_k(n) = \sum_{m=0}^{n} \omega_k(m) = \sum_{m=0}^{n} \frac{2\pi k}{p(m)}$$

$$\omega_k(n) = \frac{2\pi k}{p(n)} \qquad (2.3)$$

注意到激励信号$e(n)$中基频各次谐波的幅度都被假定为常数，故由$G(n,\omega)$独自构成语音信号谱的幅度。同理，由于激励信号的基频各次谐波都有着一个零初始相位，故而仅由系统相位$\psi(n,\omega)$构成语音信号的相位。

由于 $p(n)$ 在时刻附近几乎是常数，

所以激励相位$m \to \phi_k(m)$在 n 附近表示为：

$$\phi_k(m) \approx \phi_k(n) + \omega_k(n)(m-n) \qquad |m-n|很小 \qquad (2.4)$$

根据标准时变滤波器方程,浊音信号 $x(n)$ 模型用激励$g_n(m)$输出来表示。我们有

$$x(n) = \sum_{m=-\infty}^{+\infty} g_n(m)e(n-m) \qquad (2.5)$$

假定基音周期 $P(n)$ 在$g_n(m)$所记忆的时长范围内是不变的，即 $P(n) = P(n-1) = \cdots P(n-m)$,激励信号用当时的各次谐波表达式代替，我们有

$$x(n) = \sum_{K=0}^{P(n)-1} G(n,\omega_k(n))\exp[j(\phi_k(n)+\psi(n,\omega_k(n)))] = \sum_{k=0}^{p(n)-1} A_k(n)\exp[j\theta_k(n)]$$

(2.6)

其中信号 k 次谐波幅度$A_k(n)$即是系统幅度函数在谐波频率$\omega_k(n)$处的取值，信号 k 次谐波的相位$\theta_k(n)$即是激励相位$\phi_k(n)$与系统相位两者的和。

$$A_k(n) = G(n,\omega_k(n))$$

$$\theta_k(n)=\phi_k(n)+\psi(n,\omega_k(n))=\phi_k(n)+\psi_k(n) \qquad (2.7)$$

$\theta_k(n)$ 常被称为信号 k 次谐波的瞬时相位。由于系统相位 $\psi(n,\omega_k(n))$ 是时间 n 的缓变函数，故而在 n 附近，我们可以认为 $\psi(n,\omega_k(n))$ 保持不变，由(2.4)式，有

$$\theta_k(m)=\theta_k(n)+\omega_k(n)(m-n)\ ,\qquad |m-n|很小 \qquad (2.8)$$

2.1.2 浊音的短时傅里叶分析[11]

一个浊音信号 $x(n)$ 的短时傅里叶变换可以很容易地表示成它的各次谐波表达式形式

$$X(t_a(u),\omega)=\sum_{m=-\infty}^{+\infty}h_u(m)x(t_a(u)+m)\exp(-j\omega m) \qquad (2.9)$$

将式 2.6 代入式 2.9,并将 $X(t_a(u),\omega)$ 在频率轴上分 N 点采样

$$X(t_a(u),\Omega_l)=\sum_{m=-\infty}^{\infty}h_u(m)[\sum_{k=0}^{p(t_a(u)+m)-1}A_k(t_a(u)+m)\exp(j\theta_k(t_a(u)+m))]\exp(-j\omega_l m)$$

$0\leqslant l\leqslant N$ (2.10)

假定分析窗 $h_u(m)$ 的时长足够短，使得基音周期 $m\to p(t_a(u)+m)$ 和信号 $x(n)$ 的谐波幅度 $m\to A_k(t_a(u)+m)$ 在 $h_u(m)$ 的时长范围内是一个常数，并假定式 2.8 成立，于是有 $\theta_k(t_a(u)+m)=\theta_k(t_a(u))+m\omega_k(t_a(u))$ ，

代入式 2.10 我们得到

$$X(t_a(u),\Omega_l)=\sum_{k=0}^{p(t_a(u))-1}H_u(\Omega_l-\omega_k(t_a(u)))A_k(t_a(u))\exp[j\theta_k(t_a(u))] \qquad (2.11)$$

其中 $H_u(\omega)$ 是分析窗 $h_u(m)$ 的 STFT。

由式（2.11）可见 ， $x(n)$ 的 STFT 可以表示为将 $p(t_a(u))$ 个 $H_u(\omega)$ 图象相加，每个图象频移 $\omega_k(t_a(u))$ 并被 $A_k(t_a(u))\exp[j\theta_k(t_a(u))]$ 加权。我们假定分析窗是以 0 为对称的，结果，$H_u(\omega)$ 是一个实值函数。令 ω_h 是分析 $H_u(m)$ 的截止频率，如果我们选 ω_h 小于基音谐波频率间隔的一半（被称为窄带分析条件），被移位和加权的 $H_u(\omega)$ 图形将不会有重叠，于是 $X(t_a(u),\Omega_l)$ 简化为

$$X(t_a(u),\Omega_l)=\begin{cases}A_k(t_a(u))\exp[j\theta_k(t_a(u))]H_u(\Omega_l-\omega_k(t_a(u))), & |\Omega_l-\omega_k(t_a(u))|\le\omega_h\\ 0 & \text{其他}\end{cases}$$

(2.12)

$x(n)$ 的 STFT 幅度为

$$M(t_a(u),\Omega_l)\underline{\underline{def}}M_l(u)=\begin{cases}A_k(t_a(u))H_u(\Omega_l-\omega_k(t_a(u))) & , \quad |\Omega_l-\omega_k(t_a(u))|\le\omega_h\\ 0 & \text{其它}\end{cases}$$

(2.13)

$M(t_a(u),\Omega_l)$ 是 $t_a(u)$ 的缓变函数，因为 $A_k(t_a(u))$ 和 $P(t_a(u))$ 都是 $t_a(u)$ 的缓变时间函数。

同样地， $x(n)$ 的相位为

$$\phi(t_a(u),\Omega_l)\underline{\underline{def}}\phi_l(u)=\arctan\left(\frac{\Im(X(t_a(u),\Omega_l))}{\Re(X(t_a(u),\Omega_l))}\right)=\theta_k(t_a(u))\bmod(2\pi)$$

(2.14)

即时相位 $\phi_l(u)$ 带有落入第 l 个频带的单个谐波的即时频率 $\omega_k(t_a(u))$ 的信息。

令 $t_a(u)$ 和 $t_a(u-1)$ 足够近,使得式(2.8)成立，可以通过计算即时相位 $\phi_l(u)$ 的一阶后向差分 $\Delta\phi_l(u)$ 来确定即时频率 $\omega_k(t_a(u))$。

$$\Delta\phi_l(u)\underline{\underline{def}}\phi(t_a(u),\Omega_l)-\phi(t_a(u-1),\Omega_l)$$

$$=(t_a(u)-t_a(u-1))\omega_k(t_a(u))+2n\pi \qquad (2.15)$$

其中 n 是未知的，但可以通过在时域“展开”相位的方法进行估计。

用 $R(u)=t_a(u)-t_a(u-1)$ 表示相继两分析时刻之间的样点数，假定 k 次谐波落入第 l 频带，我们有

$$|(\omega_k(t_a(u))-\Omega_l)R(u)|<\omega_h R(u) \qquad (2.16)$$

其中 ω_h 是分析窗的带宽，再假定 R(u)的长度满足 $\omega_h R(u)<\pi$ 结合(2.15) (2.16) 两式，我们有

$$|\Delta\phi_l(u)-\Omega_l R(u)-2n\pi|<\pi \qquad (2.17)$$

仅有唯一的整数 n 满足不等式（2.17），可以通过$\Delta\phi_l(u)-\Omega_l R(u)$的值来确定 n，一旦 n 的值被确定，将 n 代入(2.15)式，即时频率$\omega_k(t_a(u))$便可计算出来。

即时频率的计算可以总结为按下几步来完成：

(1)根据相继两帧短时谱$X(t_a(u),\Omega_l)$和$X(t_a(u-1),\Omega_l)$计算相位增量

$$Z(t_a(u),\Omega_l)=\phi(t_a(u),\Omega_l)-\phi(t_a(u-1),\Omega_l)-\Omega_l R(u)$$

(2)通过加减2π的一个整数倍的方法修改$Z(t_a(u),\Omega_l)$，使得

$$\left|\overline{Z}(t_a(u),\Omega_l)\right|=\left|Z(t_a(u),\Omega_l)\pm 2n\pi\right|<\pi$$

(3)按下式估计落入第 l 频带的单次谐波的即时频率

$$\lambda(t_a(u),\Omega_l)=\Omega_l+\frac{\overline{Z}(t_a(u),\Omega_l)}{t_a(u)-t_a(u-1)} \qquad (2.18)$$

§2.2 时间/基频尺度的确定[13,14,15,16]

2.2.1 时间尺度修改(Time-Scale Modification)

时间尺度修改指在不影响语音信号的频谱包络的情况下改变其发音的速度。即基音周期轮廓(pitch-contour)和共振峰结构的时变进程在时间尺度上将被重新调整，而其它方面保持不变。

2.2.1.1 时间尺度弯曲函数(Time-Scale warping function)

定义一个任意时间尺度修改，即指定原始语音信号与修改后得到的语音信号两者之间的时间映射关系，这种映射关系 $n \rightarrow n' = D(n)$ 被称为时间尺度弯曲函数。其中 n 指的是原始信号的时间序列，n' 指的是修改后得到的信号的时间序列，通常用一个定积分来定义 D

$$n \rightarrow n' = D(n) = \int_0^n \alpha(m)dm \qquad (2.19)$$

其中$\alpha(n)$是一个时变的时间修改比例，当$\alpha(n)=\alpha$是一个常数时，时间尺度弯曲函数$n \to n'=D(n)=\alpha n$是线性的。当$\alpha>1$时，相应于时间尺度拉伸，发音速度减慢，反之，当$\alpha<1$时，相应于时间尺度压缩，发音速度加快。对于时变的时间尺度修改比例$\alpha(n)$情况，弯曲函数$n \to n'=D(n)$是非线性的。

2.2.1.2 理想的时间尺度修改

时间尺度弯曲函数指定在原始信号中时刻 n 发生的“事情”将会在时间尺度修改了的信号$n'=D(n)$时刻发生，按照语音生成模型，语音参数应按下列方式被转变。

$$n' \to p'(n')=p(D^{-1}(n'))$$

$$n' \to A'_k(n')=G'_k(n')=G(D^{-1}(n'),\omega_k(D^{-1}(n')))$$

$$n' \to \theta'_k(n')=\phi'_k(n')+\psi(D^{-1}(n'),\frac{2\pi k}{p(D^{-1}(n'))})$$

$$n' \to \phi'_k(n')=\sum_{m=0}^{n'}\frac{2\pi k}{p(D^{-1}(m))} \tag{2.20}$$

这组方程表示：

（1）修改后信号的基音轮廓$n' \to p'(n')$被时间尺度弯曲函数 D 重新进行了时间尺度调整。

（2）系统函数（幅度$G'(n',\omega)$和相位$\psi'(n',\omega)$）是原始系统函数在时间尺度上重新调整的结果。

（3）修改后信号 k 次谐波在n'时刻的瞬时频率等于原始信号 k 次谐波频率在$n=D^{-1}(n')$时刻的瞬时频率。这一点根据上述方程很容易证明。

2.2.2 基音尺度修改(Pitch-Scale Modification)

基音尺度修改指的是改变一个语音段的基本频率而不改变它的谱包络（确切的说，共振峰的位置及带宽）以及谱包络的时变进程。

2.2.2.1 基音尺度弯曲函数（Pitch Scale Warping Function）

定义一个任意的基频尺度变换，即指定一个时变的基频修改系数$n \to \beta(n)>0$，它将影响基音轮廓，修改后的基音轮廓$n \to p'(n)$为

$$n \rightarrow p'(n) = \frac{p(n)}{\beta(n)} \tag{2.21}$$

当 $\beta(n) > 1$ 时，当地的基音频率增加到原始基音频率的 $\beta(n)$ 倍（当地的基音周期以 $\frac{1}{\beta(n)}$ 相乘），同理当 $\beta(n) < 1$ 时，当地的基音频率减小到原始基音频率的 $\beta(n)$ （当地的基音周期作响应的增加）。通常情况下， $\beta(n)$ 是一个时间缓变函数。

2.2.2.2 理想的基音尺度修改

参照语音生成模型，对于理想的基音尺度修改，语音参数必须按以下方式进行修改。

$$n' \rightarrow p'(n') = \frac{p(n')}{\beta(n')}$$

$$\mathrm{n}' \rightarrow \mathrm{A}'_{\mathrm{k}}(\mathrm{n}') = \mathrm{G}'_{\mathrm{k}}(\mathrm{n}') = \mathrm{G}(\mathrm{n}', \omega_{\mathrm{k}}(\mathrm{n}')\beta(\mathrm{n}'))$$

$$\mathrm{n}' \rightarrow \theta'_{\mathrm{k}}(\mathrm{n}') = \phi'_{\mathrm{k}}(\mathrm{n}') + \psi(\mathrm{n}', \omega_{\mathrm{k}}(\mathrm{n}')\beta(\mathrm{n}'))$$

$$\mathrm{n}' \rightarrow \phi'_{\mathrm{k}}(\mathrm{n}') = \sum_{\mathrm{m}=0}^{\mathrm{n}'} \omega_{\mathrm{k}}(\mathrm{m})\beta(\mathrm{m}) \tag{2.22}$$

这些方程可以解释如下：

（1） 基音轮廓被时变系数 $\beta(n)$ 重新标度。

（2） 修改后的信号的谐波幅度通过在新的谐波频率处采样原始系统函数获得，因此，它保留了原始信号的共振峰结构。

（3） 修改后的信号的激励瞬时相位 $\phi'_{\mathrm{k}}(\mathrm{n}')$ 的一阶后向差分等于修改后的信号的 k 次谐波频率，即基音频率确实改变了。

与时间尺度修改不同的是，基音尺度修改需要在那些不一定是原始信号谐波频率的频率处估计系统幅度 $G(n', \beta(n')\omega_k(n'))$ 和相位 $\psi(n', \beta(n')\omega_k(n'))$。为此，许多基音尺度修改算法需要明确地将语音信号分解成

①时变的谱包络

②平直的声源谱

由于时变的系统函数 $\mathrm{G}(\mathrm{n}, \omega)$ 不能从输入信号的波形中确切识别出来，为

此，还需要作附加的假设。通常有这样的几种方法估计谱包络，全极点 LPC 模型方法，直接模型方法和声源滤波器分解方法。

§2.3 基于 STFT 进行韵律修改

对于语音信号，我们可以选择由一系列生成参数来生成语音的合成模式，这种合成模式称为参数合成。实现时间/基音尺度调整大体上的途径是，首先分析原始语音信号以获得这些生成参数，然后对这些生成参数实施我们所需要的修改，最终合成相应的信号。在选择这样的一种分析/合成模式过程中，我们必须要妥善处理语音质量和计算开销方面的矛盾，而用这种参数方法，我们很难在这两方面取得一个很好的折衷。通常，一个模型的优缺点主要在于它是否以一种压缩而简化的方式代表语音。一个好的参数模型对语音编码、语音识别以及语音合成有迷人的前景。然而对于韵律修改而言，将原始语音中丰富的声学细节简化将迅速导致一个可感觉到的畸变。为此，通常韵律修改采用非参数方法。由于声音具有诸如声调和音色这些随时间变化而变化的频域特征，非参数方法利用一个时—频表达式，其中，在一个给定的时刻，可感知的声音特性应沿频率轴理想地表达出来。

我们通常将语音看作是一种有着缓变的频域特征的信号（即准稳态信号），为此，我们可以采用将短时分析和傅里叶变换相结合的方法得到一个所谓的短时的傅里叶变换(STFT)作为所需要的时—频表达式。STFT 作为语音分析的一个基本工具，被用于语音合成和修改已许多年了，基于 FFT 算法和迭加合成方法的理论易于理解，且可以获得高效的实施[17,18,19,30] 。其基本思想是：用一个窗函数 w(n)将分析限制在分析时刻周围的一小段之内,在每一分析段之内可以认为 x(n)具有稳定的特性。这样，标准的稳态信号分析工具（如傅里叶变换）便可以作用于每一个分析段。这种方法用于时变系统的分析显然其缺陷是分析的准确性要受加窗和非稳定因素的影响，但在实际的处理过程中，我们用连续或混

叠的信号段进行短时分析以减少这些影响，且易于实时处理。

2.3.1 分析

短时傅里叶变换可以看作是语音信号的时-频表达式，它将信号的一部分作傅里叶变换，然后再移到信号的另一部分依次重复这样的操作。这样，信号便被表示成相应于不同的分析窗位置处短时信号的离散傅里叶系数，相继的分析窗位置$t_a(u)$被称作分析时刻，大多数情况下，STFT 分析以一个恒定的速度来进行，即$t_a(u)=uR$。对于时间/基音尺度修改来说，一个非恒定的分析速度有时会更为方便（基于基音同步分析基础用于时间尺度修改的 WSOLA 算法和用于基频尺度修改的 PSOLA 算法中主要的观点）。

以$x(n)$表示语音信号，$h_u(n)$表示分析窗，并假定$h_u(n)$：①以时刻 0 为中心。②具有确定的时长T_u和对称性，③是一个低通滤波器的单位冲击响应。则短时分析信号$x(t_a(u),n)$和短时分析谱$X(t_a(u),\omega)$分别为：

$$x(t_a(u),n)\underline{\underline{\text{def}}}\,h_u(n)x(t_a(u)+n)$$

$$X(t_a(u),\omega)\underline{\underline{\text{def}}}\sum_{n=-\infty}^{+\infty}h_u(n)x(t_a(u)+n)\exp(-j\omega n) \qquad (2.23)$$

在许多 STFT 分析场合，分析窗是一个固定的窗函数，即$h_u(n)=h(n)$。

2.3.2 修改

对 STFT 要作的修改反映着我们要对原始信号要作的修改。修改阶段由以下两步构成：

（1）修改短时分析谱$X(t_a(u),\omega)$，用以产生一系列短时合成谱$Y(t_s(u),\omega)$。

（2）使这些短时合成谱$Y(t_s(u),\omega)$以一系列新的时刻同步，这些时刻被称为合成时刻，以$t_s(u)$表示。

合成时刻$t_s(u)$序列由分析时刻$t_a(u)$按所需的基音/时间尺度修改来确定，合成时刻的个数也不一定等于分析时刻的个数。对于非恒定的基音/时间尺度修改系数来说，不论分析速度是否恒定，合成时刻通常将是不规则分布的。

2.3.3 合成[12]

2.3.3.1 LSEE-MSTFT 估计

最后一步将短时合成信号序列以合成时刻同步结合起来，以便获得我们所需要的“修改了的”信号。其主要困难是，为了获得我们所需要的韵律修改，修改 $X(t_a(u),\omega)$ 的结果 $Y(t_s(u),\omega)$ 将不再是一个有效的 STFT 序列串，即不存在某个信号，其 STFT 序列串是 $Y(t_s(u),\omega)$。而 $Y(t_s(u),\omega)$ 仍含有最能刻画我们所需要的“修改了的”信号的信息。为此，我们要人为地构造一个信号 $y(n)$，使得 $y(n)$ 的短时谱 $\hat{Y}(t_s(u,\omega))$ 序列串与合成短时谱 $Y(t_s(u),\omega)$ 序列串误差平方和最小，即

$$\hat{Y}(t_s(u),\omega)=\sum_{m=-\infty}^{+\infty} f_u(m)y(t_s(u)+m)\exp(-j\omega m) \tag{2.24}$$

$$\sum_u \int_{-\pi}^{+\pi}\left|\hat{Y}(t_s(u),\omega)-Y(t_s(u),\omega)\right|^2 d\omega \quad \text{最小} \tag{2.25}$$

根据 Parseval 方程给出上式的闭式解

$$y(n)=\frac{\sum_u y_w(u,n-t_s(u))f_u(n-t_s(u))W_u(n-t_s(u))}{\sum_u f_u^2(n-t_s(u))W_u(n-t_s(u))}$$

$$\text{其中}\quad y_w(u,n)=\frac{1}{2\pi}\int_{-\pi}^{\pi} Y(t_s(u),\omega)\exp(j\omega n)d\omega \quad , \tag{2.26}$$

其中 $f_u(n)$ 称为合成窗，$W_u(n)$ 为加权系数，这种合成方程称为 LS-OLA 方程。

合成算法类似于加权叠加，相继的短时合成信号经适当的加权和时移后重新结合起来，分母起一个时变的归一化系数的作用，用以补偿由于相继两窗间的不同叠加而导致的能量的改变。选择不同的加权系数 $W_u(n)$ 将产生不同的合成方案，最直接的方法是令 $W_u(n)=1$，于是得到

$$y(n)=\frac{\sum_u y_w(u,n-t_s(u))f_u(n-t_s(u))}{\sum_u f_u^2(n-t_s(u))} \tag{2.27}$$

另一种选择是令 $W_u(n)=\begin{cases}\dfrac{1}{f_u(n)} & f_u(n)\neq 0\\ 0 & f_u(n)=0\end{cases}$

于是有

$$y(n)=\frac{\sum_u y_w(u,n-t_s(u))}{\sum_u f_u(n-t_s(u))} \tag{2.28}$$

这种方案使叠加合成更为简化。

2.3.3.2 LSEE-MSTFTM 估计

在许多应用场合，我们需要由 STFT 幅度函数序列串$|Y(t_s(u),\omega)|$构造语音信号。上述的最小误差平方方法可以借用，但要稍作修改。同样地，我们也要寻找一个信号 $y(n)$，从最小幅度误差平方意义上来讲，其短时谱 $\hat{Y}(t_s(u,\omega))$ 序列串与我们所需要的信号的其短时谱 $Y(t_s(u),\omega)$ 最接近，即

$$\sum_u \int_{-\pi}^{\pi} \left(|Y(t_s(u),\omega)|-|\hat{Y}(t_s(u),\omega)|\right)^2 d\omega \quad \text{最小} \tag{2.29}$$

运用迭代方法找出 $y(n)$，先任意假定一个 $y^1(n)$ 为 $y(n)$,然后计算 $y^1(n)$ 的 STFT，将其幅度用我们所需要的幅度$|Y(t_s(u),\omega)|$代替，根据修改了的 $y^1(n)$ 的 STFT，运用上述的 LS-OLA 算法，得到一个估计信号，这样的循环不断重复，第 i+1 次的估计信号 $y^{i+1}(n)$ 通过将前一次的估计信号 $y^i(n)$ 的 STFT 幅度用 $|Y(t_s(u),\omega)|$ 替代后再运用 LS-OLA 合成获得，所有的迭代过程可以概括如下：

（1）幅度约束： $\hat{Y}^i(t_s(u),\omega)=|Y(t_s(u),\omega)|\frac{Y^i(t_s(u),\omega)}{|Y^i(t_s(u),\omega)|}$

（2）最小幅度误差平方估计：

$$y^{i+1}(n)=\frac{\sum_u f_u(n-t_s(u))\frac{1}{2\pi}\int_{-\pi}^{\pi}\hat{Y}^i(t_s(u),\omega)\exp(j\omega n)d\omega}{\sum_u f_u^2(n-t_s(u))} \tag{2.30}$$

第三章 时间尺度修改

本章将要分别从时域和频域阐述时间尺度修改的方法并对实际的语音信号进行修改。根据实验结果对合成信号质量加以评估，对这两种方法进行比较，指出他们各自的优缺点并对影响合成信号质量的因素加以探讨。

§3.1 基于 STFT 的时间尺度修改[19]

本节，我们的讨论建立在第二章语音生成的准稳态模型基础上，时间尺度修改靠时间弯曲函数 $t \to D(t)$ 指定且在窄带分析条件下，即式(2.12)、(2.13)、(2.14)成立。

3.1.1 时间尺度修改了的信号的 STFT

令语音信号是浊音，我们采用恒定的分析速度，与每个分析时刻 $t_a(u) = uR$ 相对应着一个合成时刻 $t_s(u) = D(uR)$

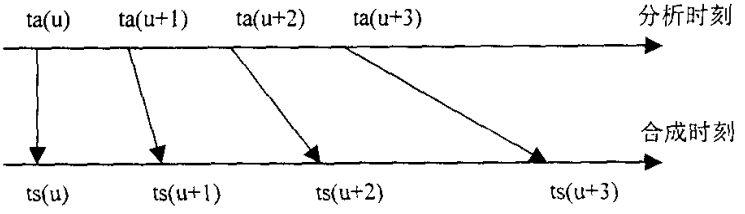


图 3-1

由上图可见，恒定速度的分析时刻往往对应着一个“弹性的”合成时刻。

时间尺度修改了的信号的 STFT $Y(t_s(u), \Omega_l)$ 可以用下式表达

$$Y(t_s(u),\Omega_l) = M[X(D^{-1}(t_s(u)),\Omega_l)]\exp[j\hat{\phi}(t_s(u),\Omega_l)]$$

$$\hat{\phi}(m,\Omega_l) = \hat{\phi}(t_s(u-1),\Omega_l) + \sum_{i=1}^{m-t_s(u-1)} \lambda((u-1)R + \frac{iR}{N(u)},\Omega_l)$$

其中 $t_s(u-1) \le m \le t_s(u)$， $N(u) = t_s(u) - t_s(u-1)$

(3.1)

$N(u)$表示相继两合成时刻之间的样点数，$\lambda(n,\Omega_l)$表示时刻 n 第 l 频带的瞬时频率，其值可由式(2.18)估计。

很容易证明：

①在时间尺度修改后的信号中，$t_s(u)$时刻第 l 频带内的正弦曲线瞬时幅度等于原始信号在$t_a(u)$时刻的第 l 频带内的正弦曲线瞬时幅度。

②在时间尺度修改后的信号中，$t_s(u)$时刻的基音谐波的瞬时频率为：

$$\hat{\lambda}(t_s(u),\Omega_l)=\frac{\hat{\phi}(t_s(u),\Omega_l)-\hat{\phi}(t_s(u-1),\Omega_l)}{N(u)}=\lambda(uR,\Omega_l) \qquad (3.2)$$

即等于原始语音信号在$t_a(u)=uR$时刻的基音谐波的瞬时频率。

令分析时刻靠得足够近，使得在$(u-1)R$和uR之间原始语音信号的基音谐波的瞬时频率保持不变，则式(3.1)简化为：

$$\hat{\phi}(t_s(u),\Omega_l)=\hat{\phi}(t_s(u-1),\Omega_l)+N(u)\lambda((u-1)R,\Omega_l) \qquad (3.3)$$

上述方程可以解释如下：时间尺度修改后的信号的 STFT $Y(t_s(u),\Omega_l)$是通过将原始语音信号的STFT幅度和瞬时频率在时间进程上作修改而不管其瞬时相位，这样便保证了时间尺度修改后的信号含有与原始信号相同的基音谐波频率，只是它们的时间进程按时间弯曲函数$t\rightarrow D(t)$作了重新调整。

最后，将时间尺度调整了的信号的 STFT $Y(t_s(u),\Omega_l)$运用合成方程式(2.27)叠加便得到时间尺度调整了的信号。

尽管上述修改系统基于准周期和缓变参数的假设，但它也可以用于清音信号。用同样的修改系统作用于浊音和清音增加了算法的鲁棒性，因为不需要确定浊音/清音

3.1.2 算法实现：

源程序见附录部分,图 3-2 是实现上述时间尺度修改的算法流程图。其中短时分析信号谱是在分析时刻取一帧短时信号作 FFT 而得到的，第一帧短时合成相位谱取第一帧短时分析相位谱，合成时刻由相应的分析时刻乘以时间尺度修改系数得到。

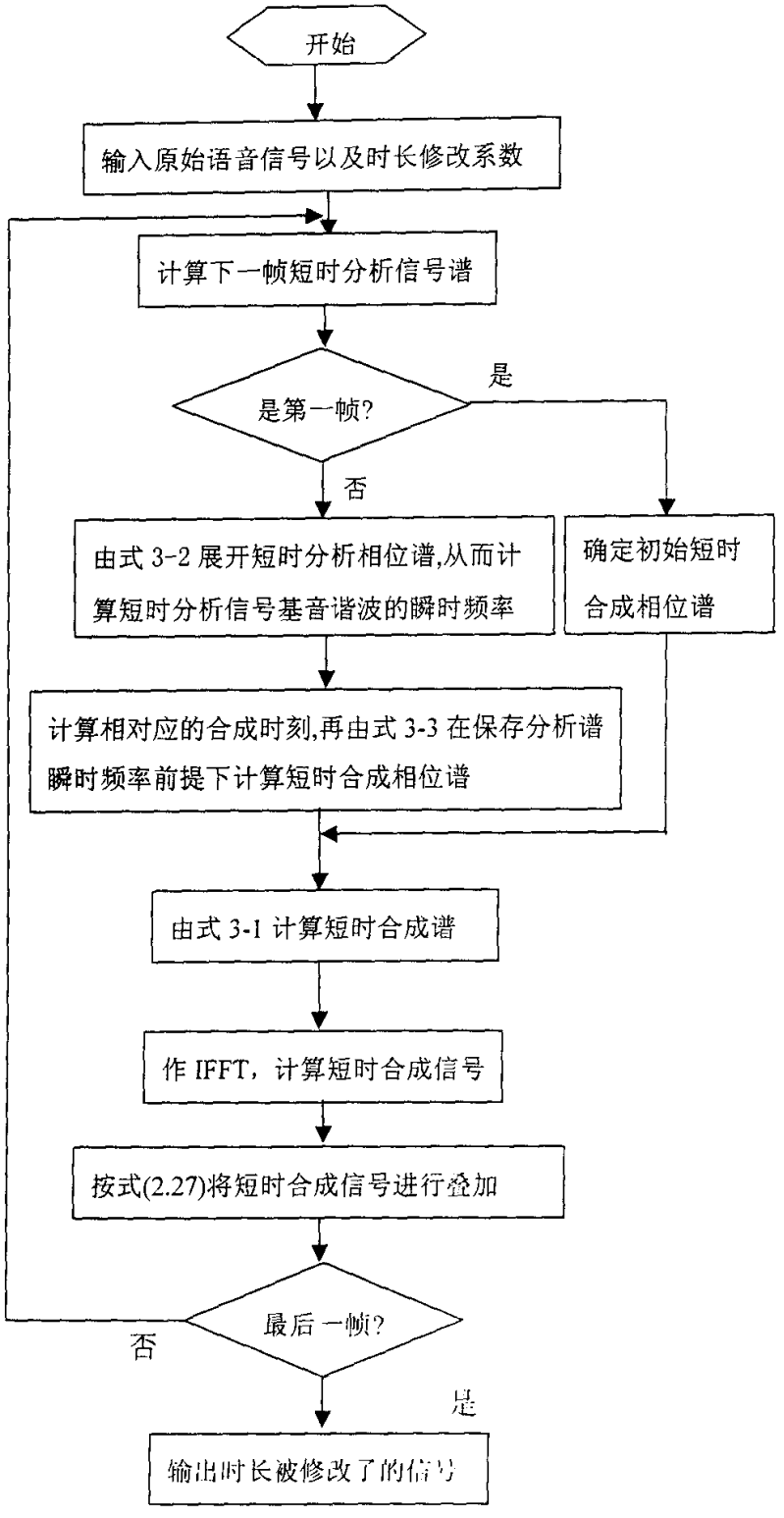
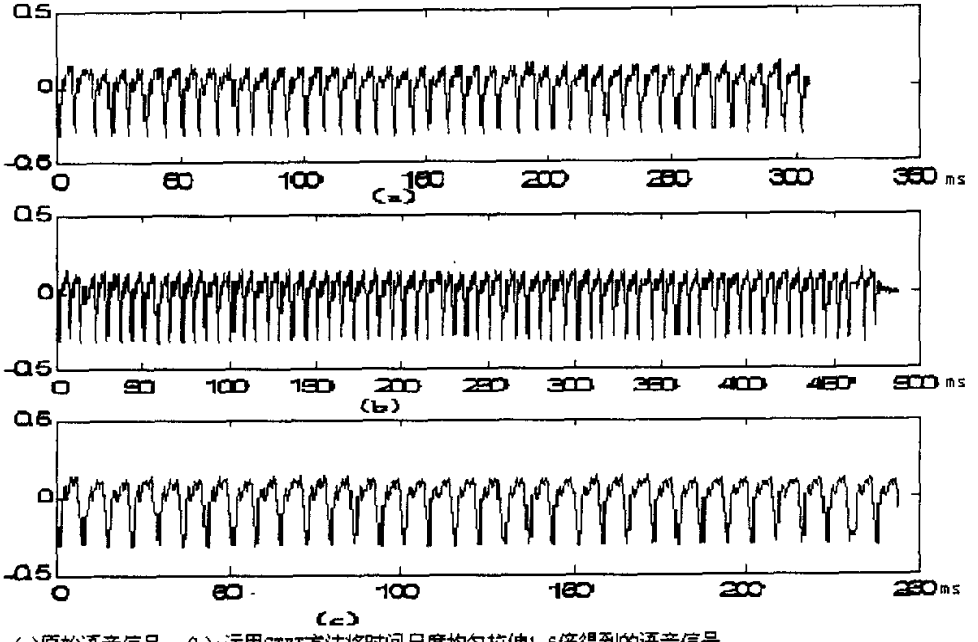

开始
输入原始语音信号以及时长修改系数
计算下一帧短时分析信号谱
是第一帧?
是
否
由式 3-2 展开短时分析相位谱,从而计算短时分析信号基音谐波的瞬时频率
确定初始短时合成相位谱
计算相对应的合成时刻,再由式 3-3 在保存分析谱瞬时频率前提下计算短时合成相位谱
由式 3-1 计算短时合成谱
作 IFFT，计算短时合成信号
按式(2.27)将短时合成信号进行叠加
最后一帧?
否
是
输出时长被修改了的信号

图 3-2

3.1.3 运算结果

3.1.3.1 时域比较

图 3-3 是将元音/e/经 STFT 方法分别将时间尺度均匀减小 0.8 倍和均匀拉长 1.6 倍后得到的信号时域波形。原始语音采样率为 11025Hz,以间隔 5ms 的均匀速度分析，每帧 30ms，1024 点 FFT 谱分辨率。

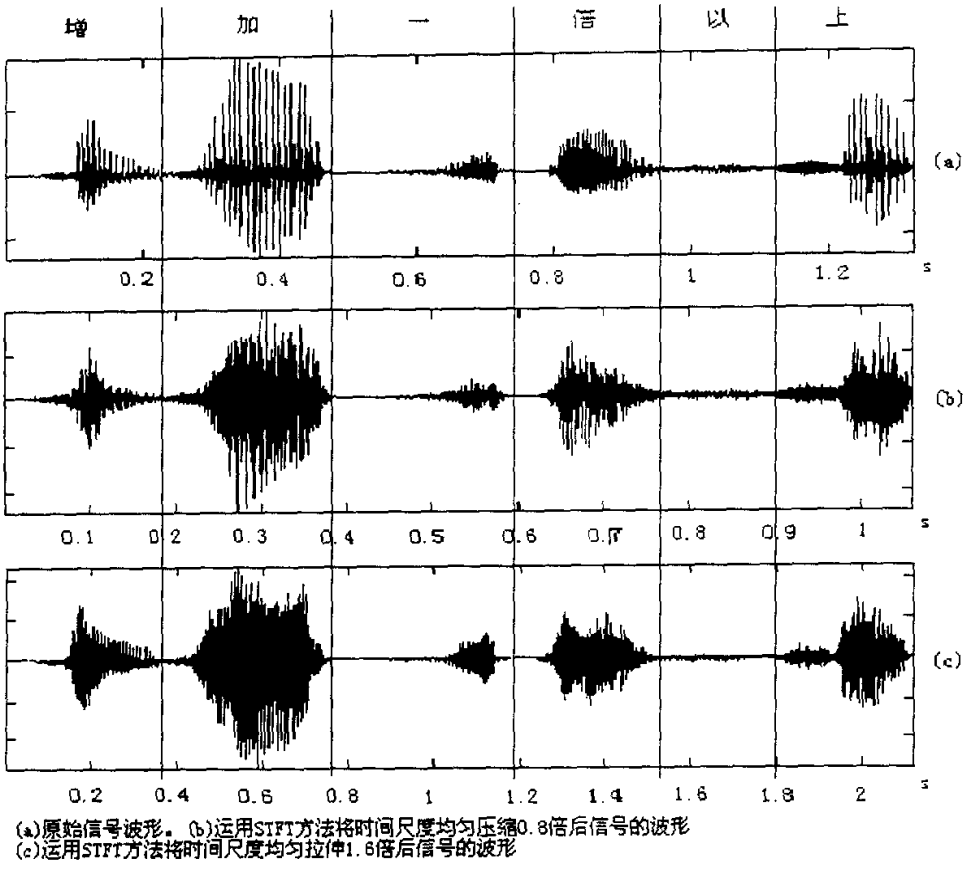


(a)原始语音信号。(b)：运用STFT方法将时间尺度均匀拉伸1.6倍得到的语音信号。
(c)运用STFT方法将时间尺度均匀压缩0.8倍得到的语音信号。

图 3-3

图 3-4 是将 STFT 方法作用于完整语句得到的时间尺度修改了的信号的时域波形。原始语音信号采样率 16000Hz,以间隔 5ms 均匀时刻分析，每帧时长 25ms，1024 点 FFT 谱分辨率。由图 3-3 和图 3-4 所示时域波形可见，对于单元音而言，由于其周期性结构与理想的语音生成模型非常接近，修改后的信号很好地保存了原始信号的准周期性结构,只是时间进程发生了改变而已。而对于实际的语句来说，由于其周期性变化较快，在窄带分析的时长范围内（大于 4 倍基音周期），其周期性结构与我们假设的“周期性不变”有较大的差异，当我们用理想的语音生成模型来进行“相位展开”运算时，必然导致运算结果与实际有相当大的出入，主要表现在时域波形的准周期性结构不太好（“相位展开”运算目的就是使短时合成

信号按周期性同步)。

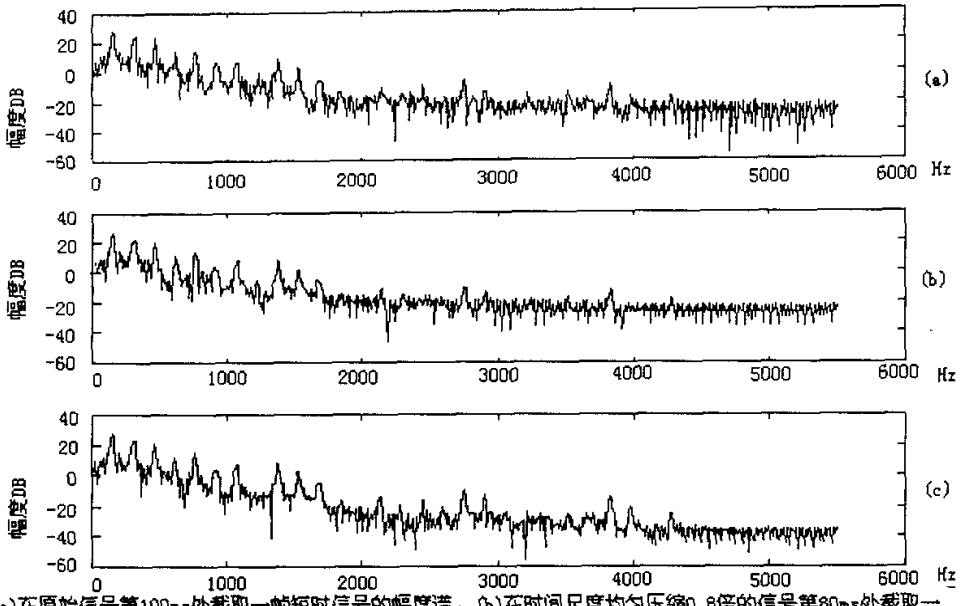


(a)原始信号波形。(b)运用STFT方法将时间尺度均匀压缩0.8倍后信号的波形
(c)运用STFT方法将时间尺度均匀拉伸1.6倍后信号的波形

图 3-4

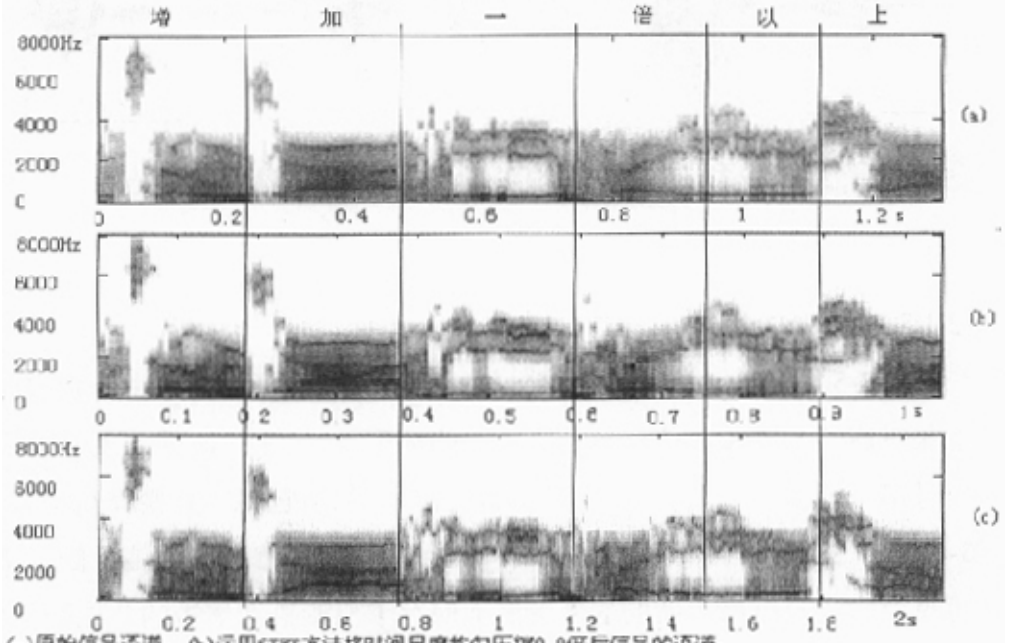
3.1.3.2 频域比较

图 3-5 是在单元音/e/原始信号和时间尺度修改了的信号中分别取时间上相对应的一帧短时信号而计算出来的幅度谱。可见，时间尺度修改后的信号很好地保存了原始语音信号的基音周期和谱包络（共振峰结构）。图 3-6 是将 STFT 方法作用于完整语句得到时间尺度修改了的信号的语谱，可见，时间尺度修改后的信号很好地保存了原始语音信号的共振峰结构，这样便能保证时间尺度修改后的信号保持较高的可懂度。



(a)在原始信号第100ms处截取一帧短时信号的幅度谱。(b)在时间尺度均匀压缩0.8倍的信号第80ms处截取一帧短时信号的幅度谱。(c)在时间尺度均匀拉伸1.6倍的信号第160ms处截取一帧短时信号的幅度谱。

图 3-5



(a)原始信号语谱。(b)运用STFT方法将时间尺度均匀压缩0.8倍后信号的语谱。
(c)运用STFT方法将时间尺度均匀拉伸1.6倍后信号的语谱。

图 3-6

3.1.3.3 听辨结果

对于单元音，经 STFT 方法进行时间尺度修改得到的信号质量非常高，基本上难以区别哪个是原始的，哪个是合成的。对于实际的语句，修改后的信号质量也是比较高，具有很高的可懂度和较好的自然度，但正如我们从其时域波形所见到的，由于其准周期性结构不太好，这样，听起来有轻微的嘶哑现象，不过在实际韵律修改中，时长变化范围一般不会超过 0.5-2.0 倍,这种现象不太严重。

3.1.3.4 主要缺点

基于短时 STFT 方法进行时间尺度修改最大的缺点就是运算开销太大，每一帧的分析和合成都要进行一次 FFT 和逆 FFT，还要涉及到“相位展开”的运算。其次就是在“相位展开”运算过程中，由于实际的语句的短时相位谱与我们假定的“周期性结构”这一理想情况存在着较大的差异（在我们窄带分析的时间范围内（大于四个基音周期），信号的周期性结构变化较大），为此，按理想条件下计算出来短时合成相位谱便于实际的相位谱差别较大，使得合成出来的信号时域波形准周期性结构不太好，相当于声带运动“不规则成分”加重，导致合成的语音听起来有些嘶哑噪声现象。

§3.2 直接在时域进行时间尺度修改[14]

基于 STFT 的时间尺度修改涉及到“相位展开”，运算极其复杂，而在时域直接进行时间尺度修改则显得相对简单，其实质是基于 STFT 的时间尺度修改的一个变化了的方式。

3.2.1 OLA 时间尺度修

根据前文基于 STFT 的时间尺度修改，我们有

$$Y(t_s(u),\Omega_l) = M[X(D^{-1}(t_s(u)),\Omega_l)]\exp[j\hat{\phi}(t_s(u),\Omega_l)]$$

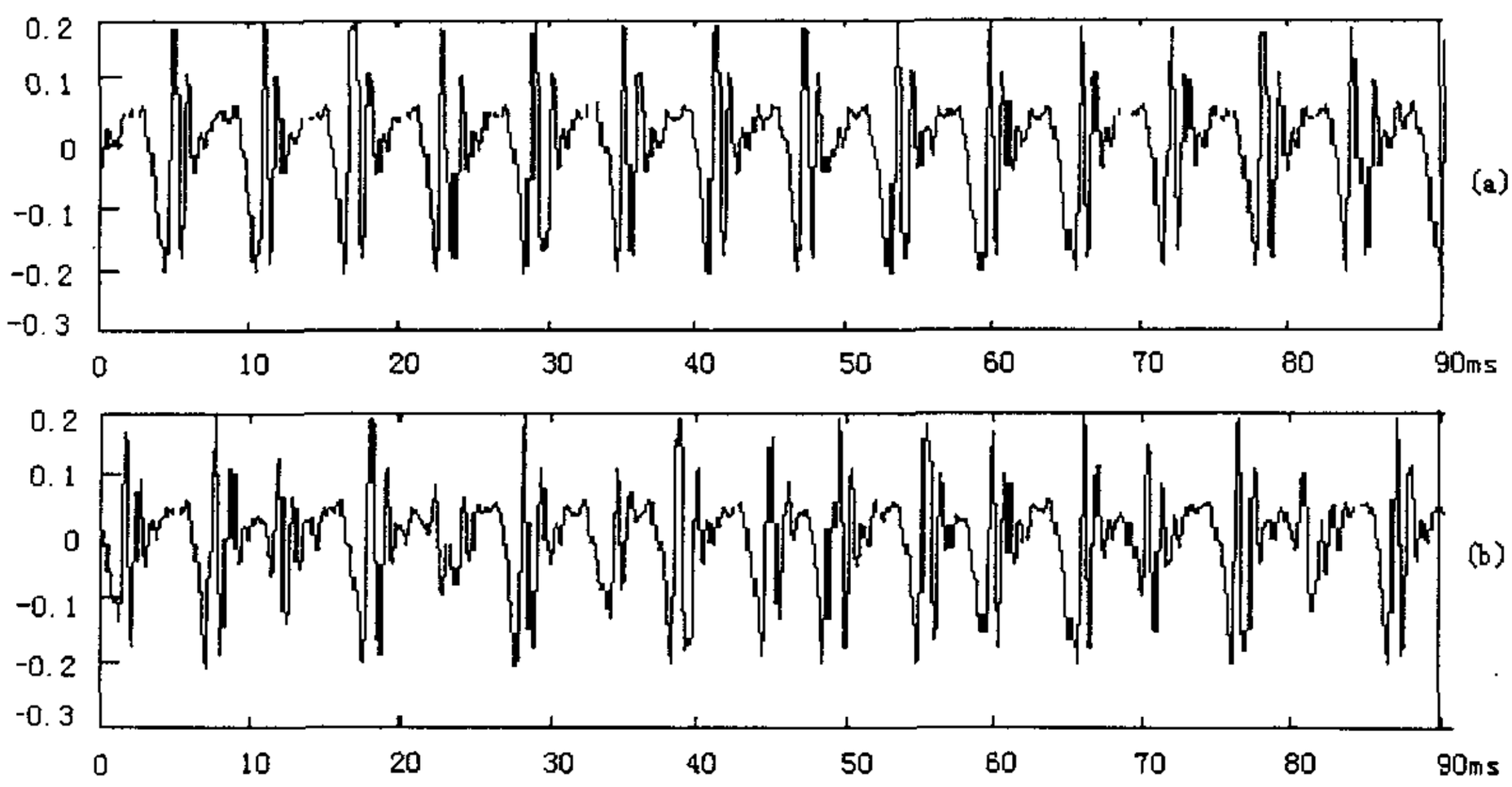
$$\hat{\phi}(t_s(u),\Omega_l) = \hat{\phi}(t_s(u-1),\Omega_l) + N(u)\lambda((u-1)R,\Omega_l)$$

这样才能保证时间尺度修改后的信号 $y(n)$ 具有原始信号的基音谐波频率（即保证具有原来的准周期结构）。如果我们不考虑在 STFT 修改过程中相位的变化直接取 $Y(t_s(u),\Omega_l) = X(D^{-1}(t_s(u)),\Omega_l)$ 则

$y(t_s(u),n) = x(D^{-1}(t_s(u)),n)$ 运用 LS-OLA 合成方程，得到

$$y(n) = \frac{\sum_u f_u(n - t_s(u))x(D^{-1}(t_s(u)),n)}{\sum_u f_u^2(n - t_s(u))}$$

由此方程合成结果如图 3-7 所示：



(a)从原始信号中第100ms处截取一段语音信号
(b)从经过OLA直接叠加而得到的时间尺度均匀拉伸1.6倍的信号中第160ms处截取一段语音信号

图 3-7

由图 3-7 可见，合成的结果非常糟糕，它破坏了原始信号的准周期结构。

对于二元表达式 $x(t_a(u),m)=h_u(n)x(n+t_a(u))$，其中两个时间尺度并非独立的，这样，关于信号 $x(n)$ 时间结构的重要信息将存在于 $t_a(u)$ 和 ω 中。例如，假定一个周期信号 $x(n)=x(n+P)$，如果窗长足够长，每一帧信号 $x(t_a(u),n)$ 含有几个原始周期。同时，这种周期性结构也存在于不同帧之间，即 $x(t_a(u)+p,n)=x(t_a(u),n)$。当任意将帧象 $Y(t_s(u),\Omega_l)=X(D^{-1}(t_s(u)),\Omega_l)$ 那样重新放置时，我们将破坏短时帧内时间结构和短时帧间时间结构的关系，从而导致图 3-7 所示的原始信号的准周期结构被破坏。

复数 STFT $X(t_a(u),\Omega_l)$ 的相位成份 $\Gamma(t_a(u),\Omega_l)$ 带有分析窗内信号的时间结构信息，幅度成份 $|X(t_a(u),\Omega_l)|$ 带有与 ω 有关的可感知的语音特性，其幅度的谐次结构反映了基音频率(pitch)的信息，其谱包络反映了共振峰的信息。由于没有相位信息，故一个幅度谱既没有精确确定 $x(t_a(u),n)$ 时间结构的信息，也没有信号 $x(n)$ 相对于分析窗 $h_u(n)$ 位置的信息。如果我们选择合适长度的窗（几个周期），Griffin 和 Lim 的研究表明[11]，通过 $|Y(t_s(u),\Omega_l)|=|X(D^{-1}(t_s(u)),\Omega_l)|$，运用 LSEE-MSTFT 估计 $y(n)$，经过一定量的迭代（≥100 次）后，得到高质量的时间尺度修改的信号。

3.2.2 高效的时间尺度修改算法

3.2.2.1 同步叠加算法（SOLA）

根据前文的 LS-MSTFTM 迭代可知，迭代过程也就是一个不断使 $|\hat{Y}^i(t_s(u),\Omega_l)|$ 接近于 $|(Y(t_s(u),\Omega_l)|$ 的过程，为此，一个好的初始相位估计 $\Gamma_0(t_a(u),\omega)$（相当于选择一个好的 $y^1(n)$）是非常重要的。Roucos 和 Wilgus 通过实验研究了 LSEE-MSTFTM 进行时间尺度修改。结果表明[17]，用一个好的方法构造的 $y^1(n)$ 本身已具有很高的质量，进一步的迭代将不再需要或事实上并不能提高语音质量，这种算法称为 SOLA（Synchronized Overlap Add）算法。

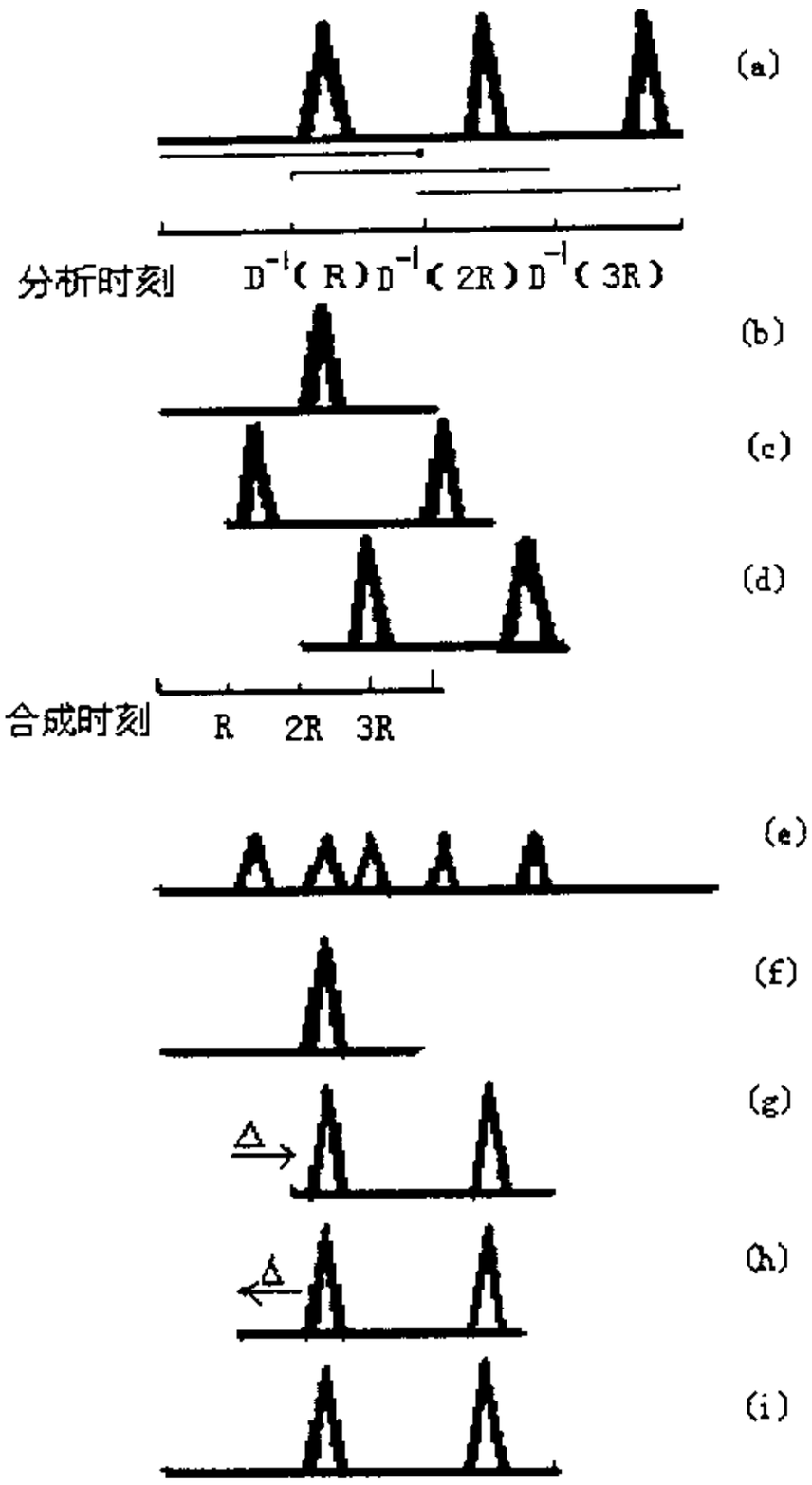


图 3-8

我们令分析阶段和合成阶段使用相同的窗 w(n)，合成时刻 $t_s(u) = uR$ 均匀分布在频域，我们初始估计：

$$Y^0(uR, \Omega_l) = X(D^{-1}(uR), \Omega_l)$$

$$\hat{Y}^0(uR, \Omega_l) = | Y(uR, \Omega_l) | \frac{Y^0(uR, \Omega_l)}{| Y^0(uR, \Omega_l) |} = Y^0(uR, \Omega_l)$$

则
$$y^1(n) = \frac{\sum_u w^2(n - uR)x(n - uR + D^{-1}(uR))}{\sum_u w^2(n - uR)} \qquad (3.4)$$

$y^1(n)$ 被认为是用 LSEE-MSTFTM 迭代运算进行时间尺度修改的一个初始估计信号，它相当于选择一个初始相位 $\Gamma^0(uR, \Omega_l)$ 与每一帧原始短时信号 $x(D^{-1}(uR), n)$ 实际相位相等的初始信号。这种选择对每一帧而言肯定是最好

的，但正如我们前文所讨论的，将位于原始位置$D^{-1}(uR)$处的短时帧重新安放到所需要的合成位置 uR 破坏了帧与帧之间原来的相位关系。如图（3-8）所示[14],从图(a)中取出三帧短时分析信号，按合成时刻重新放置，得到三帧短时合成信号(b)、(c)、(e)，按 LSEE-MSTFT 合成，得到合成信号$y^1(n)$，见图(e)。

根据基于 STFT 的时间尺度修改，我们知道，要使时间尺度修改后的信号 y(n)保存原始信号的基音各次谐波频率（即保存原始信号的准周期结构），必须要满足

$$\hat{\phi}(t_s(u),\Omega_l)-\hat{\phi}(t_s(u-1),\Omega_l)=\lambda(t_a(u-1),\Omega_l)(t_s(u)-t_s(u-1)) \tag{3.5}$$

而现在，对于$y^1(n)$而言，$Y^0(t_s(u),\Omega_l)=X(t_a(u),\Omega_l)$

于是 $\hat{\phi}(t_s(u),\Omega_l)=\phi(t_a(u),\Omega_l)$

$$\hat{\phi}(t_s(u),\Omega_l)-\hat{\phi}(t_s(u-1),\Omega_l)=\phi(t_a(u),\Omega_l)-\phi(t_a(u-1),\Omega_l)$$
$$=\lambda(t_a(u-1),\Omega_l)(t_a(u)-t_a(u-1))+2n\pi$$

正因为$\lambda(t_a(u-1),\Omega_l)(t_s(u)-t_s(u-1))\neq\lambda(t_a(u-1),\Omega_l)(t_a(u)-t_a(u-1))+2n\pi$

才导致原始语音信号的准周期结构被破坏。为此，在合成时为了保存原始信号的准周期结构(只是在时间进程上发生改变)，我们可以在保持$t_s(u-1)$不变的情况下，适当给$t_s(u)$一个偏移Δ，即$t_s(u)=D(t_a(u))+\Delta_u$，使得

$$\lambda(t_a(u-1),\Omega_l)(t_s(u)-t_s(u-1))=\lambda(t_a(u-1),\Omega_l)(t_a(u)-t_a(u-1))+2n\pi$$

从而保证时间尺度修改后的信号保存原始信号准周期结构，如图 3-8(f)、(g)、(h)、(i)所示。如果Δ_u确定，则合成信号便不需经迭代而得到：

$$y(n)=\frac{\sum_u w^2(n-uR-\Delta_u)x(n-uR+D^{-1}(uR)-\Delta_u)}{\sum_u w^2(n-uR-\Delta_u)} \tag{3.6}$$

Δ_u的确定可以在时域进行，因为Δ_u的含义就是要让 y(n)具有原始信号的准周期性结构，故Δ_u的值应使当前短时合成信号与输出信号已合成的部分互相关系数最大。

可见，在 SOLA 算法中，$Y(uR+\Delta_u,\Omega_l)=X(D^{-1}(uR),\Omega_l)$，我们需要的时间弯曲函数并没有精确地实现，为了获得一个同步的短时合成信号，一

个基音周期以内的偏差是允许的。

3.2.2.2 波形相似叠加算法（WSOLA）[27]

WSOLA 是 SOLA 算法的一种变化了的形式。为了进一步降低运算成本，我们希望式(3.6)中分母为一常数，但在上面讨论的 SOLA 算法中，由于 Δ_u 不能预先确定，故无法直接使分母为常数，WSOLA 间接地实现了这一点。

同于前文 SOLA 实现时间尺度修改，为了使(3.5)式成立，我们也可以在保持 $t_s(u)$、$t_s(u-1)$ 和 $t_a(u-1)$ 不变的情况下，使 $t_a(u) = D^{-1}(t_s(u)) + \Delta_u$，这样 $Y(uR, \Omega_1) = X(D^{-1}(uR) + \Delta_u, \Omega_1)$。

如果我们取合适的窗，恒定的合成速度，相邻两帧重叠一半，则式(3.6)分母恒等于常数 1，此时，式(3.6)简化为

$$y(n) = \sum_u w^2(n - uR)x(n + D^{-1}(uR) + \Delta_u - uR) \qquad (3.7)$$

同样，Δ_u 值的确定应基于某种标准，使当前合成短时信号最大程度地与输出信号已合成的部分连续，从而使 y(n)具有原始信号的准周期性结构，WSOLA 算法正是基于波形相似标准来确定 Δ_u。

§3.3 基于 WSOLA 进行时间尺度修改的实现

3.3.1 WSOLA 计算方法

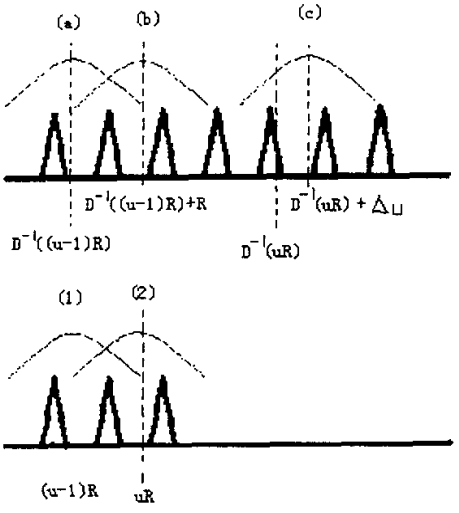


图 3-9

为了说明其原理，我们以一系列脉冲串代表原始周期性信号，合成时刻 $t_s(u)=uR$ 均匀分布。与合成时刻 $t_s(u)$ 对应的分析时刻为 $t_a(u)=D^{-1}(uR)$。

如图 3-9 所示，令短时合成帧(1)是由短时分析帧（a）而来的，短时合成帧(2)如果由由短时分析帧(b)构成，则能完全保证原始信号的连续性。但能否用(b)作(2)要看(b)的位置是否落在下一个分析时刻 $D^{-1}(uR)$ 的最大容许偏差范围内（$[D^{-1}(uR)-\Delta_{max}, D^{-1}(uR)+\Delta_{max}]$）。如果在这个范围内，则使用(b)作(2)，否则，便用(b)作模板，在 $D^{-1}(uR)$ 的最大容许偏差范围内找出与(b)波形最相似的一帧用作(2)，依此类推。

3.3.2 算法流程

WSOLA 方法进行时间尺度修改的源程序见附录部分，其算法流程如图 3-10 所示。在分析时刻偏差允许的范围内寻找与模板最相似的短时分析信号时，搜索起点为 $t_a(u)-\Delta_{max}$，终点为 $t_a(u)+\Delta_{max}$，步长为 1~10（步长越小，效果越好，但运算量相应的增大），Δ_{max} 为最大基音周期的 1/2 左右（不必非常严格）。

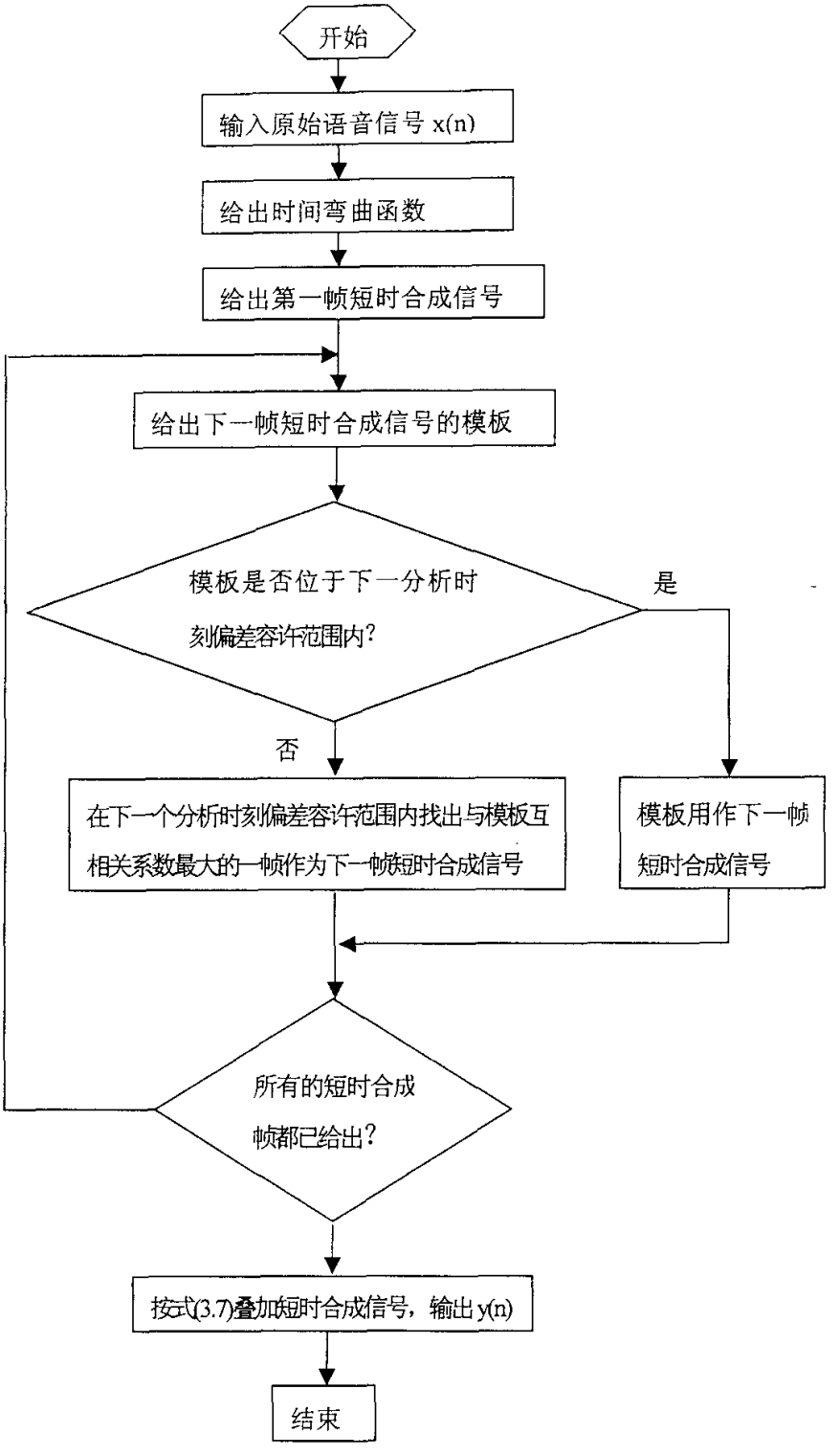

开始
输入原始语音信号 x(n)
给出时间弯曲函数
给出第一帧短时合成信号
给出下一帧短时合成信号的模板
模板是否位于下一分析时刻偏差容许范围内?
是
否
在下一个分析时刻偏差容许范围内找出与模板互相关系数最大的一帧作为下一帧短时合成信号
模板用作下一帧短时合成信号
所有的短时合成帧都已给出?
按式(3.7)叠加短时合成信号，输出 y(n)
结束

图 3-10

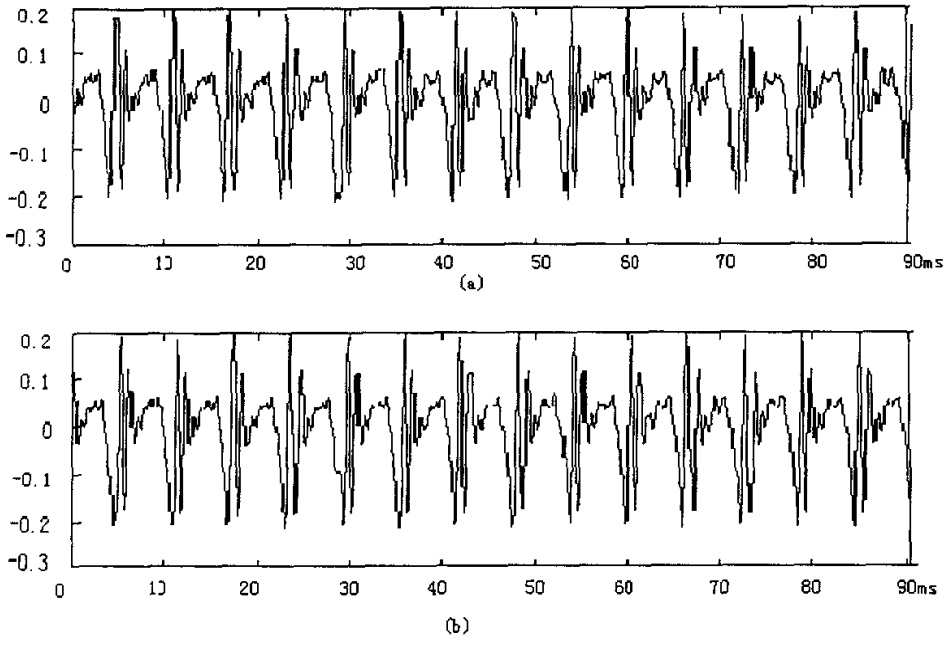
3.3.3 WSOLA 运算结果

3.3.3.1 时域比较

我们先用元音/a/为例，采样率 11.025KHz，23.2ms 的 Hanning 窗，50%交叠，R=11.6ms，$t_s(u) = uR$ 的均匀合成时刻分布，运用 WSOLA 将时长扩大 1.6 倍，结果时域波形如图 3-11 所示。

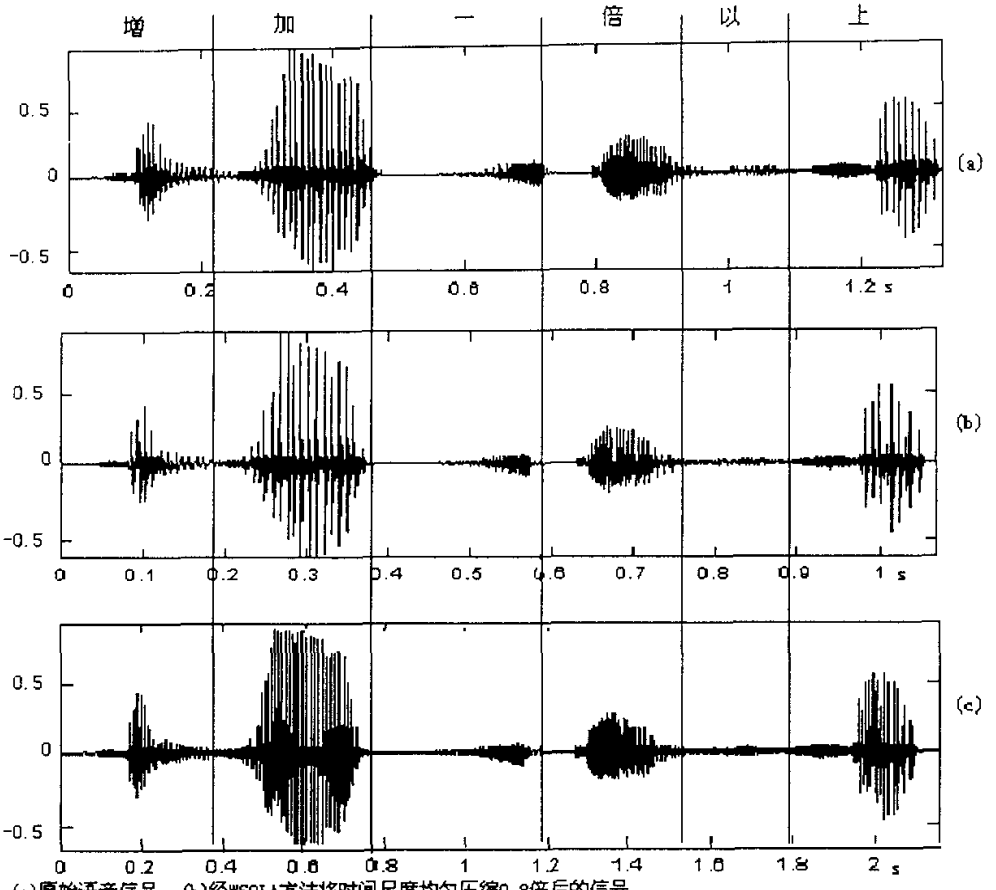
图 3-12 是用 WSOLA 方法对完整的语句进行时间尺度修改得到的信号波形。原始信号采样率 16000Hz, 25ms 的 Hanning 窗，50%交叠，间隔 12.5 等速率合成。

由 3-11、3-12 可见，不论是对单元音还是对实际完整的普通话语句，WSOLA 方法都能极好地保存了原始信号的准周期性结构，合成出来的信号与原始信号波形极为相似。



(a)从原始语音信号第100ms处截取一段语音信号。
(b)从经WSOLA方法将时间尺度均匀拉伸1.6倍得到的信号中第160ms处截取一段语音信号。

图 3-11



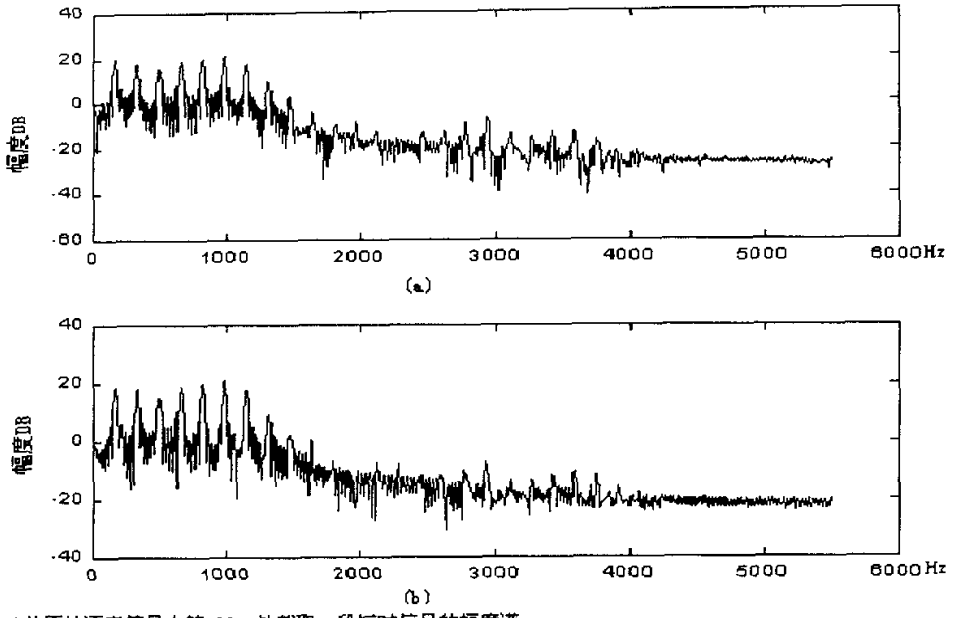
(a)原始语音信号。(b)经WSOLA方法将时间尺度均匀压缩0.8倍后的信号。
(c)经WSOLA方法将时间尺度均匀拉伸1.6倍后的信号。

图 3-12

3.3.3.2 频域比较

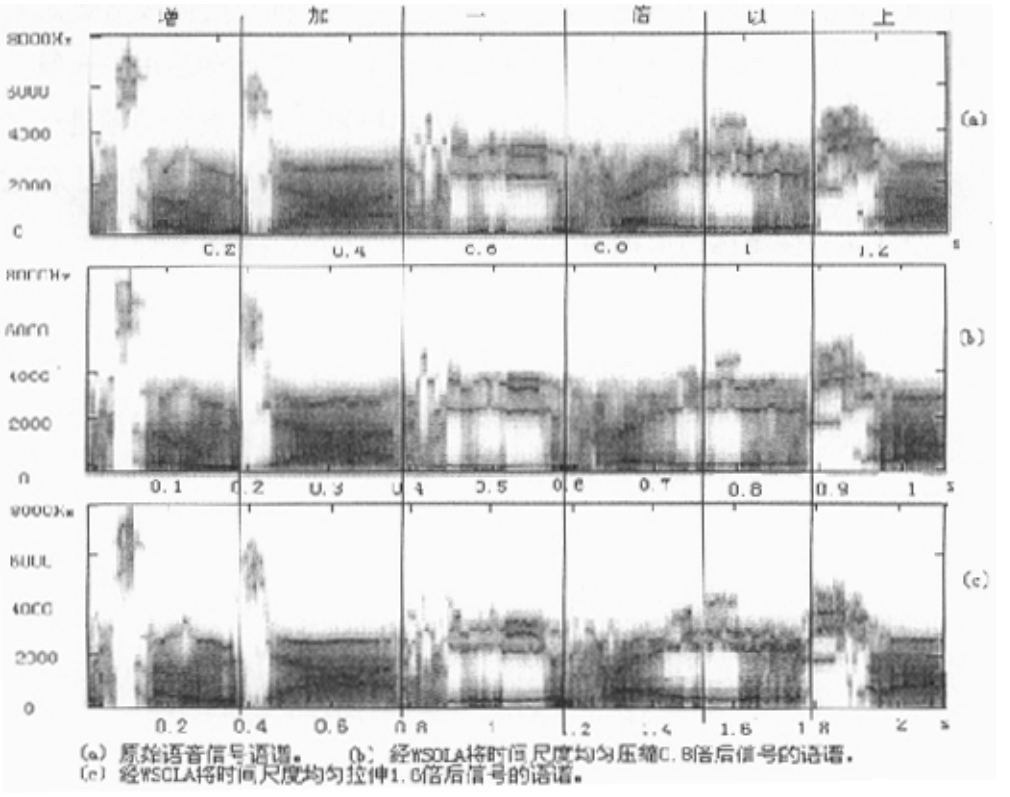
我们分别给出元音/a/时间尺度修改前后时间上相对应两短时信号的幅度谱，由图 3-13 可见，WSOLA 较好地保存了原始信号的基音周期大小和共振峰结构。

再将 WSOLA 作用于完整语句，其语谱如图 3-14 所示。可见，修改后的信号较好地保存了原始信号的共振峰结构（只不过在时间进程上改变而已），这样便能保证合成的信号具有较高的自然度和可懂度。



(a)从原始语音信号中第100ms处截取一段短时信号的幅度谱。
(b)从经WSOLA将时间尺度均匀拉伸1.6倍的信号中第160ms处截取一段短时信号的幅度谱。

图 3-13



(a) 原始语音信号语谱。 (b) 经WSOLA将时间尺度均匀压缩0.8倍后信号的语谱。
(c) 经WSOLA将时间尺度均匀拉伸1.6倍后信号的语谱。

图 3-14

3.3.3.3 听辨结果

运用 WSOLA 方法进行时间尺度修改得到的语音信号有着相当高的质量，在一定范围内（时长修改倍数 0.5-2.0）有着相当高的自然度和可懂度。

3.3.3.4 优点和缺点

运用 WSOLA 方法进行时间尺度修改是直接在时域进行的，为此，每帧短时信号的处理不需要进行 FFT、逆 FFT 以及“相位展开”运算，因而在一定程度上减小了运算开销。同时，由于跳过“相位展开”运算带来的误差，在实际的普通话语句时长修改中，合成出来的信号在保存原始信号的准周期性结构方面比基于 STFT 方法合成出来的信号要好得多。在听觉上没有“嘶哑”现象。

为了保持相邻帧的同步，我们采用自相关方法在一个基音周期以上的范围内搜索与“模板”最相似的一帧作为下一帧短时合成信号。由于是逐点搜索，这样，总的运算开销还是较大。再就是由于 WSOLA 方法实质是在时域对原始信号进行“剪切”与“复制”，当时间尺度修改倍数太大（2.0 倍以上）时，必然出现“回声”较大的现象。好在实际运用中一般不会超过这个范围，这种现象不太明显。有人提出将待“复制”的帧进行“时间反转”，这样，即保持了原始帧的频谱，又能消除“回声”现象，但这必然以牺牲周期性结构的质量为代价。

第四章 基频尺度修改

韵律指语音中所蕴含的时长、基频、强度等超音质特征。在文语转换系统（TTS）中，为了保证合成语音具有良好的可懂度和自然度，韵律合成是极其重要的一个环节。前文我们讨论了时间尺度的修改，而实际的韵律修改还包括基频尺度、音强等方面的修改。本章我们将讨论基频尺度修改。

§4.1 基于 STFT 的基频尺度修改

在第二章我们已经谈到理想的基频尺度修改(Pitch Scale Modification)应满足式(2.22)。在实际的算法中，完成基频尺度修改的方法有多种，但其本质都是直接或间接基于 STFT 基础之上进行的[20,21,22]。

4.1.1 基本步骤

基于直接 STFT 方法进行基频尺度修改原理简单易懂，总体上分以下几步来完成：

①从每一个分析时刻 $t_a(u)$ 附近的一帧信号中抽取谱包络，用以表示声门及声门以上部分的声道传输函数，这个谱包络（共振峰结构）在基频尺度修改过程中应不受影响。

②运用这个谱包络，计算其声源谱，用以近似表示分析时刻 $t_a(u)$ 附近激励信号 $e(n)$ 的频谱。这个激励信号带有与基频相关的信息，在基频尺度修改过程中，这一部分将被修改。

③将修改了的短时声源谱与没有修改的短时谱包络重新结合，得到新的短时合成谱，最后利用叠加方程合成出我们所需要的信号。

④通常基频尺度修改会产生一个相应的时长修改，为此，最后还必须补偿这种时长的修改，用以恢复原始信号的时长。

4.1.2 实现框图

我们可以采用图 4-1 的流程实现基于 STFT 方法的基频尺度修改。

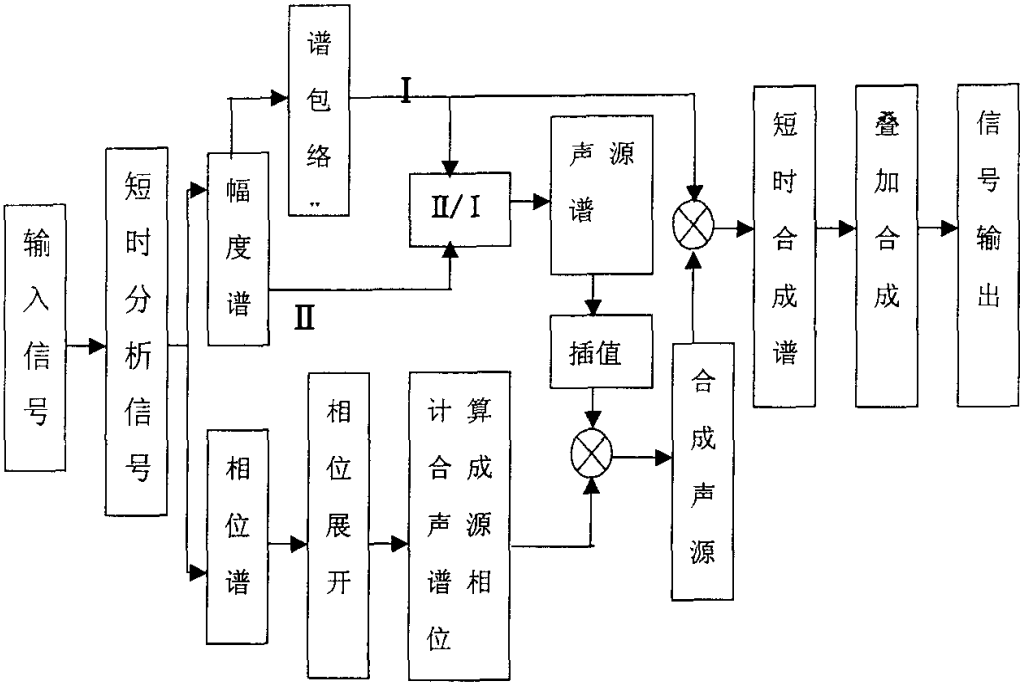


图 4-1

我们的分析依然建立在窄带基础之上（窗长≥4 倍基音周期），谱包络的估计可以有多种方法，如 SEE 声码器（Spectral Envelop Estimation）[20]中采用先在对数幅度谱中进行峰值搜索，再对结果线性插值方法，还有就是利用 LPC 系数的全极点声道传输模型估计出声道传输函数的幅频响应（或者利用 LPC 系数的全极点声道传输模型对短时语音信号进行逆向滤波，得到声源激励信号，用语音信号短时谱除以声源谱也可得到短时谱包络）。[20,21,22]

利用语音信号短时谱除以短时谱包络得到短时声源谱。令基频修改系数为β，对短时声源谱的线性插值将使其谱线伸展或压缩，结果声源谱将占据$[-\pi\beta,\pi\beta]$。当$\beta>1$，即提高基音频率时，我们只需将超出$[-\pi,\pi]$的那些高频谱线丢弃即可。当$\beta<1$，即降低基音频率时，我们要补偿高频部分，最简单的方法就是从低频区复制一部分谱线到高频区。

短时声源谱的插值运算，必然导致短时信号的时长发生相应的改变，

即时长变化系数等于基频变化系数的倒数。当我们再合成这些基频尺度修改了的短时信号时，如果合成时刻按上述时长变化系数的时间弯曲函数算得，叠加时将不会破坏这些短时信号间的相位关系，即合成信号将保持准周期性结构。但我们现在不想改变原始信号的时间尺度，即合成时刻等于分析时刻，按这一系列合成时刻进行叠加，必然要破坏短时信号间的相位关系，从而破坏信号的准周期性结构。为了使这些短时信号同步，应进行“相位展开”运算，先计算出短时合成声源谱的瞬时频率，再根据瞬时频率和合成时刻计算短时合成声源谱的相位。这样，我们便得到以合成时刻同步调的短时合成声源谱。

将没有改变的短时谱包络与基频尺度修改了的短时合成声源谱相乘，便得到短时合成谱，计算其逆 DFT 得到短时合成信号，再按式(2.27)叠加合成，得到基频尺度修改了的语音输出信号。

同样，我们这里讨论的短时频域分析依然基于窄带基础之上。需要说明的是在利用 FFT 计算短时分析信号频谱时，若$\beta<1$，则 FFT 点数 N 应大于或等于T_u/β，其中β 是基频修改系数，T_u 是短时分析信号的点数。在合成时，为了保存合成信号的自然性，合成窗一般与分析窗种类相同，我们这里选用 hanning 窗，但窗长为T_u/β。

4.1.3 算法流程

基于 STFT 实现基频尺度修改的源程序见附录部分，其算法流程图如图 4-2 所示：

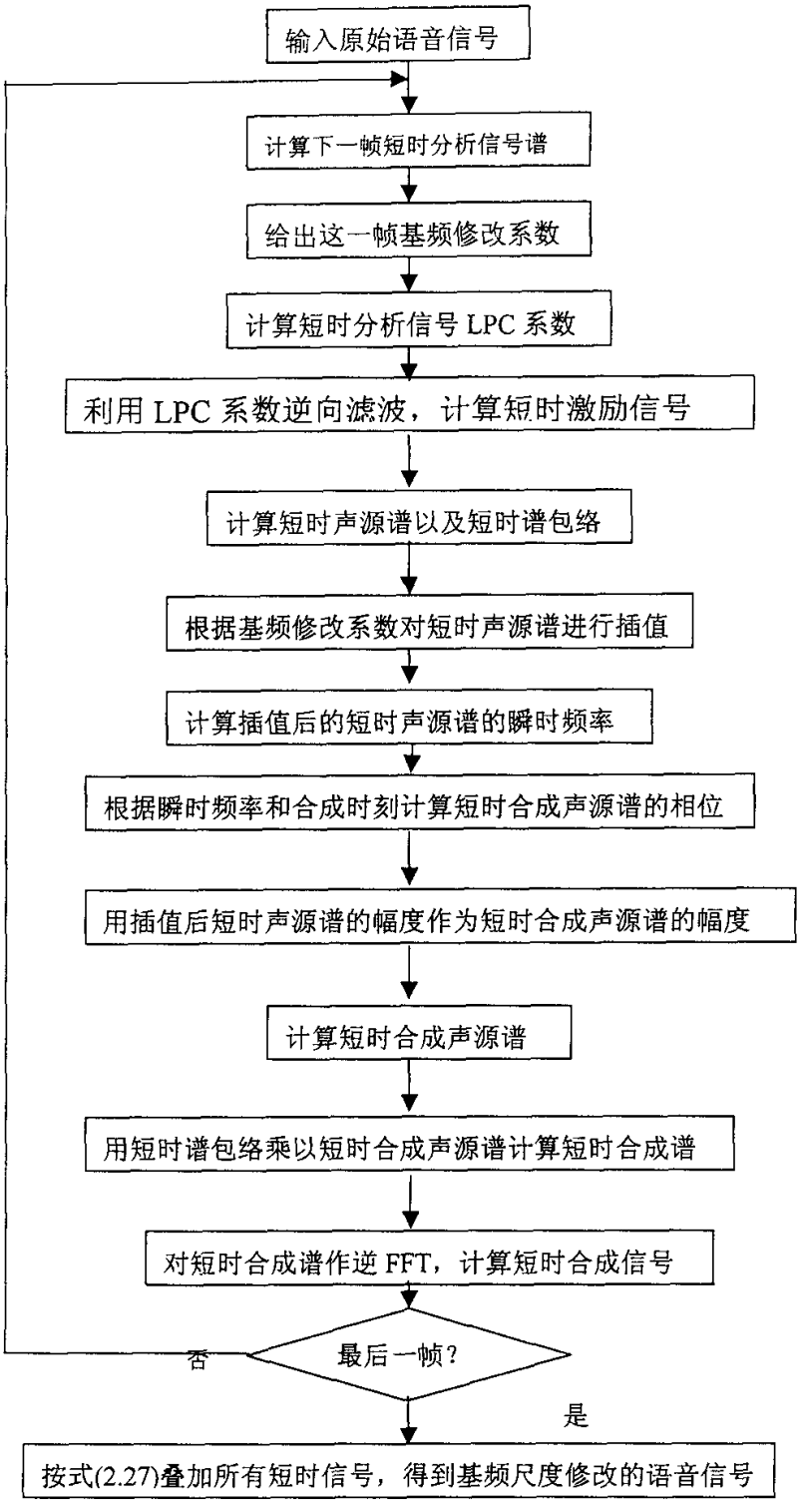
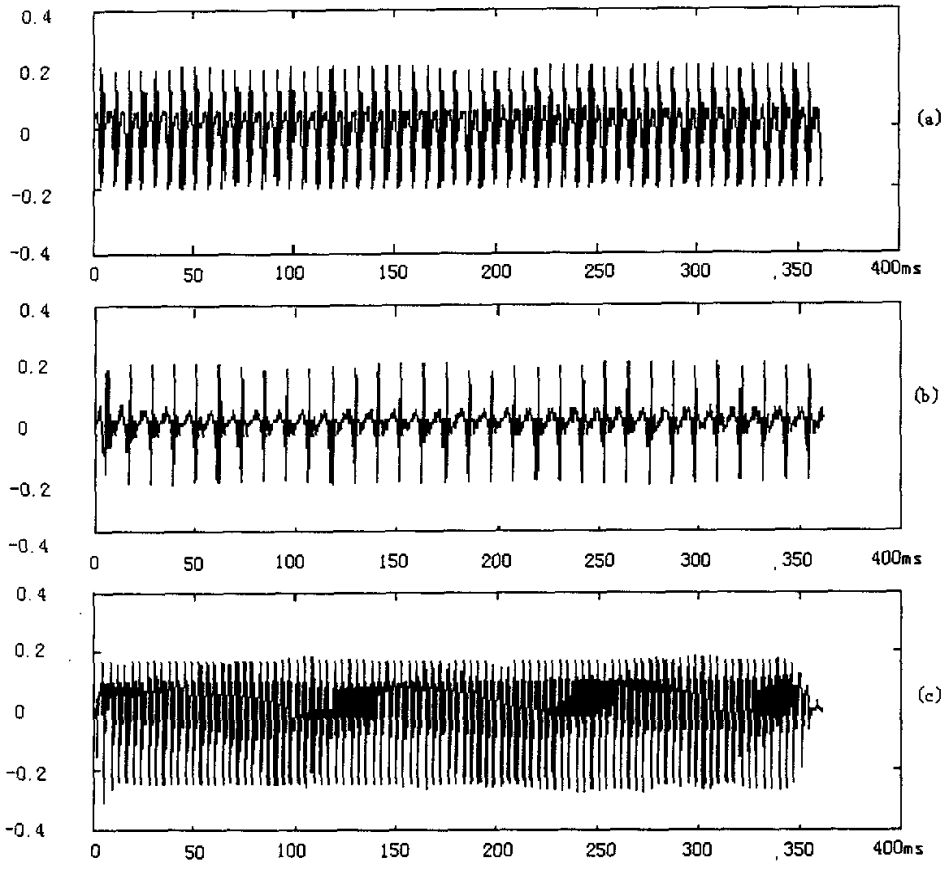

输入原始语音信号
计算下一帧短时分析信号谱
给出这一帧基频修改系数
计算短时分析信号 LPC 系数
利用 LPC 系数逆向滤波，计算短时激励信号
计算短时声源谱以及短时谱包络
根据基频修改系数对短时声源谱进行插值
计算插值后的短时声源谱的瞬时频率
根据瞬时频率和合成时刻计算短时合成声源谱的相位
用插值后短时声源谱的幅度作为短时合成声源谱的幅度
计算短时合成声源谱
用短时谱包络乘以短时合成声源谱计算短时合成谱
对短时合成谱作逆 FFT，计算短时合成信号
最后一帧?
否
是
按式(2.27)叠加所有短时信号，得到基频尺度修改的语音信号

图 4-2

4.1.4 计算结果

4.1.4.1 时域波形比较

我们以元音/a/为原始语音信号，11025Hz 采样率，每帧 30ms,间隔 5ms 均匀时刻分析，分别将原始信号的基音频率尺度均匀缩小 0.6 倍及扩大 1.8 倍，得到的输出信号波形如图 4-3 所示。从时域波形图可见，输出信号的基音周期完全按要求发生改变，且很好地保持着周期性结构，时间尺度没有发生改变。

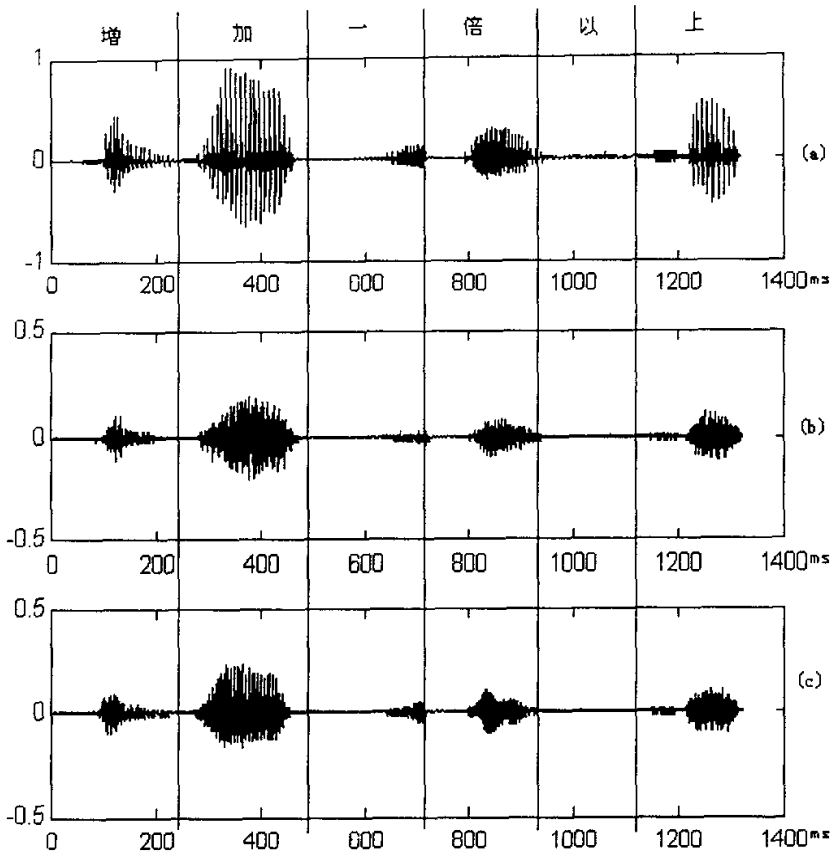


(a) 原始语音信号。 (b) 基频尺度均匀缩小0.6倍后的语音信号。
(c) 基频尺度均匀扩大1.8倍后的语音信号。

图 4-3

图 4-4 是用 STFT 方法将实际的语句的基频尺度作修改，从时域的波

形可见，修改后的信号准周期性结构不太好。



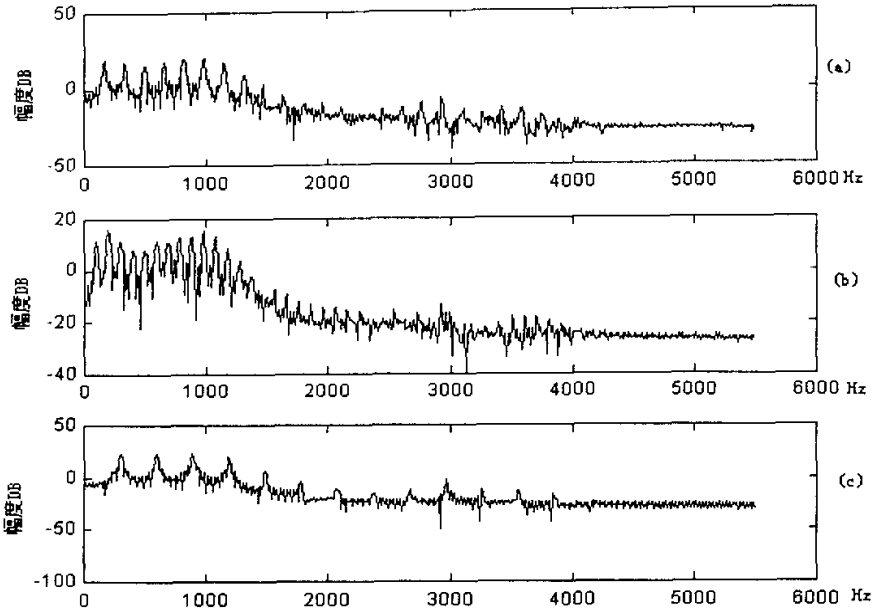
(a) 原始语音信号。(b) 经STFT将基频均匀缩小0.8倍后的信号。
(c) 经STFT将基频均匀扩大1.6倍后的信号。

图 4-4

4.1.4.2 频域比较

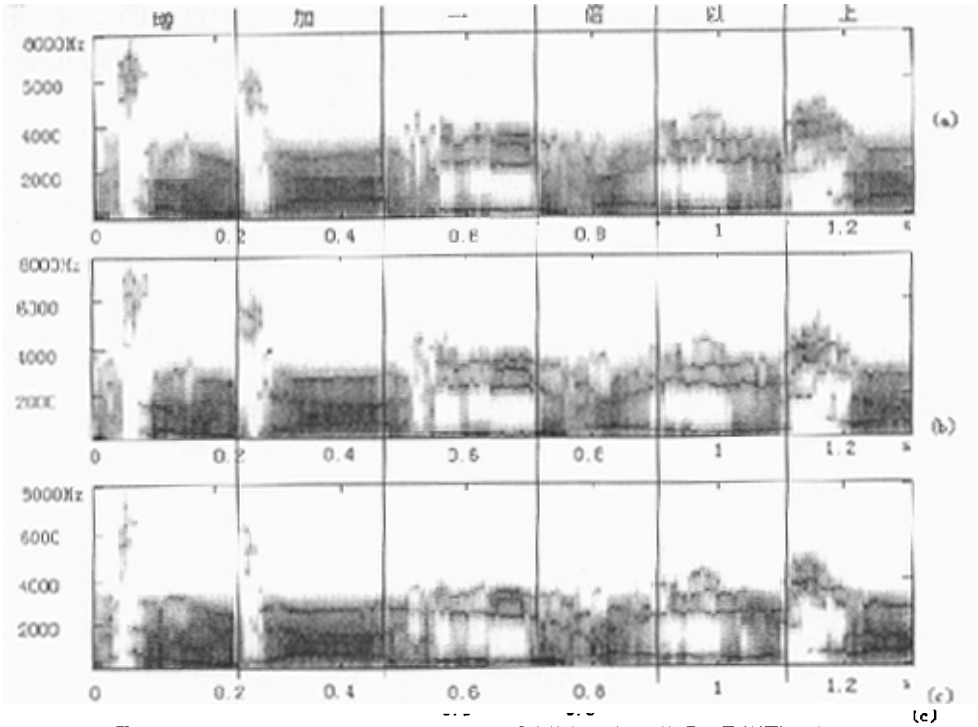
图 4-5 是分别在原始信号和基频尺度修改后的信号中同一位置截取一段语音信号的幅度谱。由此可见，修改后的信号的幅度谱中，代表基频大小的“峰值”间隔发生了相应的改变，而代表着共振峰结构的谱包络却保持不变。

图 4-6 是用 STFT 方法将实际语句的基频尺度作出修改后信号的语谱图。由图可见，尽管图 4-4 中修改后信号的准周期性结构不太好，但其语谱依然与原始信号的语谱很相似，从而保证了修改后的信号保持着较高的自然度和可懂度。



(a) 从原始语音信号中第100ms处截取一段信号的幅度谱。 (b) 从基频尺度均匀缩小0.6倍的信号中第100ms处截取一段信号的幅度谱。 (c) 从基频尺度均匀缩小扩大1.8倍的信号中第100ms处截取一段信号的幅度谱。

图 4-5



(a) 原始语音信号语谱图。 (b) 经STFT将基频尺度均匀缩小0.8倍后信号的语谱图。
(c) 经STFT将基频尺度均扩大1.6倍后信号的语谱图。

图 4-6

4.1.4.3 听辨结果

基于 STFT 进行基频尺度修改得到的语音信号具有相当高的自然度和可懂度，对于单元音来说，其周期性结构特征非常接近理想的语音生成模型，在“相位展开”运算过程中，实际情况与理论情况非常接近，合成出来的信号具有相当好的周期性结构，听起来很难辨别出是合成出来的。但对于实际语句来说，由于其周期性结构变化很大，在窄带分析时长范围内（大于 4 倍基音周期），基于“周期性结构不变”的假设与实际情况差异比较大，因此合成出来的语句波形的周期性结构不好，导致其出现轻微“嘶哑”现象。由于“相位展开”运算是为了使合成短时信号保持同步的，对信号的谱包络没有影响，因此，合成信号的语谱依然与原始信号语谱极为相似，从而使得合成信号保持较高的自然度和可懂度。

4.1.4.4 优点和缺点

与 STFT 方法进行时间尺度修改的情形相似，STFT 方法进行基频尺度修改主要优点就是合成质量较高，但因每一帧都要进行一次 FFT、逆 FFT 以及“相位展开”运算，计算开销太大。再就是“相位展开”运算作用于实际语句时因误差较大导致合成信号的准周期性结构不太好。

§4.2 基音同步叠加方法

前文我们讨论了基于 STFT 方法在频域进行时间尺度和基频尺度修改，这种方法容易理解，但实现起来要进行“相位展开”运算非常复杂，每一帧都要进行一次 FFT 和逆 FFT 运算，导致运算开销太大。本节将要介绍一种新的韵律修改方法——基音同步叠加（Pitch Synchronous Overlap Add）法，即 PSOLA 方法[23,24,25]，它最早是由 F. Charpentier 等在 80 年代末提出来的，这种方法的核心思想是直接对存储于音库中的语音运用 PSOLA 算法进行拼接，从而整合成完整的语音。

4.2.1　PSOLA 方法实现韵律修改的基本步骤

PSOLA 方法实质是 STFT 方法进行韵律修改的一个变化了的形式，它也是遵照 4.1.1 中所述的基于 STFT 方法进行韵律修改的基本步骤，不过它是以一种含蓄的方式进行的，即它将 STFT 方法中声源滤波器分解和声源谱修改这两步以一步来完成。PSOLA 方法实现韵律修改基本上按以下三个步骤进行。

4.2.1.1 分析

同于 STFT 方法，在分析阶段，将语音信号波形分解成一系列短时分析信号 $x(t_a(u),n)$，表示如下：

$$x(t_a(u),n) = h_u(n)x(n - t_a(u)) \tag{4.1}$$

在 PSOLA 方法中，分析时刻 $t_a(u)$ 的设置，在浊音部分是与基音周期同步的，而在清音部分则以恒定速度进行。通常这些基音标记（浊音部分）设置在一个基音周期的开始时刻。粗略的说，这一时刻响应于声门关闭时刻。对于纯净的语音信号，我们可以较为可靠地通过一定的办法估计出这些时刻。

分析窗 $h_u(n)$ 通常选对称的 hanning 窗（当然也可以是其它窗，如 hamming 窗或 Bartlett 窗）。窗长与当地基音周期 $P(s)$ 成正比,即 $T = \mu P(s)$。比例系数 μ 选为 2（在时域 PSOLA 中）到 4（在频域 PSOLA 中）之间。

4.2.1.2 修改

短时分析信号序列将被转变为一系列修改了的短时合成信号，这些短时合成信号序列与一套新的合成信号基音标记同步。这样的一个转变将涉及到三个方面的基本操作：短时信号序列数量的更改，短时信号序列之间的延迟的更改，每一个短时信号的波形可能要发生的更改。合成基音标记t_s的数量取决于基频尺度修改系数β和时间尺度修改系数α，相继两帧短时信号之间的延迟$t_s(u)-t_s(u-1)$等于当地的合成基音周期，用某种算法解出由合成基音标记t_s到分析基音标记t_a之间的映射$t_s \to t_a$，从而指定哪一帧短时分析信号$x(t_a(s))$将被用来产生任何一帧给定的短时合成信号$x(t_s(u))$。

按由短时分析信号产生短时合成信号的方式不同，PSOLA 算法可以分为时域 PSOLA（TD-PSOLA）、频域 PSOLA（FD-PSOLA）以及线性预测 PSOLA（LP-PSOLA）等方式。在时域 PSOLA 中，短时合成信号是由相应的短时分析信号直接拷贝而来，即$x(t_s(u),n)=x(t_a(s),n)$。在频域 PSOLA 中，短时合成信号是将相应的短时分析信号经过一个频域变换后得到。

4.2.1.3 合成

有好几种叠加合成方程可以用来合成最终的合成信号，例如，我们可以利用 LSEE-MSTFT 方案进行估计，这样可以运用式(2.27)或(2.28)叠加得到合成信号输出。

$$y(n)=\frac{\sum_u y_w(u,n-t_s(u))f_u(n-t_s(u))}{\sum_u f_u^2(n-t_s(u))} \tag{2.27}$$

$$y(n)=\frac{\sum_u y_w(u,n-t_s(u))}{\sum_u f_u(n-t_s(u))} \tag{2.28}$$

(2.27)、(2.28)两式中的分母起到一个时变的归一化系数的作用，用以补偿由于相继两帧间的不同程度的叠加而造成的能量的修改。在窄带条件下，这个系数几乎是常数，在宽带条件下，也可以使它成为常数，尤其是当合成窗长度选为合成信号当地基音周期两倍的时候。在这种情况下，合成方程可以简化为下列最简形式：

$$y(n) = \sum_{u} y_w(u, n - t_s(u)) \tag{4.2}$$

其中 $y_w(u, n - t_s(u))$ 是短时合成信号 。

4.2.2 分析基音标记（Analysis Pitch Mark）的确定

在用 PSOLA 进行韵律修改过程中，首先要做的工作就是要确定原始信号的基音标记。在这一方面 N. J. Miller 提出 Data Reduction 方法[26]，C. A. Mcgonegal 等提出 SAPD 算法[27]，C.Ma.等提出 Frobenius norm 方法[28]，根据输入信号的波形较为准确的确定这些基音标记，但无论怎样，利用机器识别这些基音标记总是存在偏差的，而这些偏差将直接影响到最终合成信号的质量。在实际非参数合成的文语转换系统中，我们的原始语音信号是以某种语音单位预先存放在语音库中的。这样，我们便可以预先人工确定出这些基音标记，包括清音部分的“标记”，将它们一起存放到“分析时刻”库中。在运用 PSOLA 进行韵律修改的时候，可以直接查找相应的“分析时刻”库，从中取出分析时刻序列。这种方法既能保证准确程度，又减少了实时运算的开销。

4.2.3 合成基音标记（Synthesis Pitch Mark）的确定[15,16]

4.2.3.1 基频尺度修改

令 $t_a(s)$ 是分析基音标记， $p(t)$ 是分析基音轮廓（Analysis Pitch Contour），$t_s(u)$ 是合成基音标记， p'(t) 是合成基音轮廓（Synthesis Pitch Contour)， $\beta(t)$ 是基频修改系数。

我们有 $p(t_a(s)) = t_a(s+1) - t_a(s)$

$$p(t) = p(t_a(s)), \qquad t_a(s) \le t \le t_a(s+1)$$

$$p'(t_s(u)) = t_s(u+1) - t_s(u)$$

在没有时间尺度修改的情况下，同一时刻，修改后信号的基音周期是 p'(t) 原始信号基音周期 $p(t)$ 的 $\frac{1}{\beta(t)}$ 倍，于是有

$$p'(t_s(u)) = \frac{p(t_s(u))}{\beta(t_s(u))}$$

$$t_s(u+1) - t_s(u) = \frac{1}{t_s(u+1) - t_s(u)} \int_{t_s(u)}^{t_s(u+1)} \frac{p(t)}{\beta(t)} dt \qquad (4.3)$$

其中　$\beta(t) = \beta(t_a(s)) = \beta_s$ ，　　$t_a(s) \le t \le t_a(s+1)$

4.2.3.2 时间尺度修改

令 $\alpha(t)$ 是给定的时间尺度修改系数，我们可以推导出时间弯曲函数 (Time Warping Function):

$D(t_a(1)) = 0$

$D(t) = D(t_a(s)) + \alpha(t - t_a(s))$ ，　　$t_a(s) \le t \le t_a(s+1)$

合成基音轮廓：　　$t \to p'(t) = p(D^{-1}(t))$

我们所需要的合成基音标记 $t_s(u)$：　　$t_s(u+1) - t_s(u) = p'(t_s(u))$

定义一个虚拟的分析时刻 $t'_s(u)$ ： $t_s(u) = D(t'_s(u))$

假定 $t_s(u)$ 和 $t'_s(u)$ 已知，我们可以通过式 4.4 来确定 $t_s(u+1)$ 和 $t'_s(u+1)$，使得 $p'(t_s(u)) = p(t'_s(u))$

$$t_s(u+1) - t_s(u) = \frac{1}{t'_s(u+1) - t'_s(u)} \int_{t'_s(u)}^{t'_s(u+1)} p(t)dt \qquad (4.4)$$

4.2.3.3 时间尺度和基频尺度同时修改

当时间尺度和基频尺度都需要作出修改时，我们可以设想先进行基频尺度修改，得到一个基音轮廓为 $\frac{p(t)}{\beta(t)}$ 的信号，然后将此信号再进行时间尺度修改，按式(4.4)，于是有

$$t_s(u+1) - t_s(u) = \frac{1}{t'_s(u+1) - t'_s(u)} \int_{t'_s(u)}^{t'_s(u+1)} \frac{p(t)}{\beta(t)} dt \qquad (4.5)$$

实际上，当两者都需要作出修改时，我们只需按式(4.5)逐次递推出各个 t_s 即可。

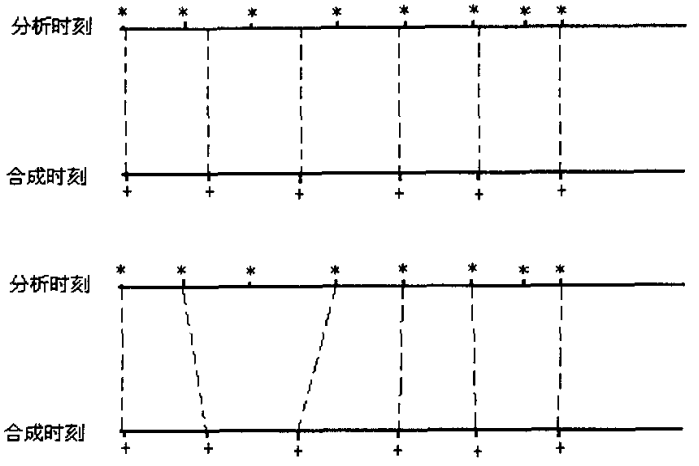
§4.3 直接在时域进行基频尺度修改

根据前文介绍的 PSOLA 方法，我们可以利用时域 PSOLA(TD-PSOLA)方法直接在时域进行基频尺度修改。

4.3.1 TD-PSOLA 进行基频尺度修改

PSOLA 方法是韵律修改的一种高效的算法，它不需通过复杂的“相位展开”方法使短时合成信号与合成时刻同步，而 TD-PSOLA 又是 PSOLA 方法中最有效的一种，因为它在修改阶段只需在短时分析信号序列中选择合适的一帧直接作为短时合成信号，无需进入频域修改，从而省去了频域 PSOLA 方法中每帧必须的 FFT 和逆 FFT 运算。

根据预先标记好的分析基音标记（分析时刻）和基频修改系数 $\beta(t)$，按式(4.3)递推算出合成基音标记（合成时刻）。由于没有进行时间尺度修改，所以“虚拟分析时刻”等于合成时刻，我们可以以最邻近原则选取短时分析时刻，如图 4-7 所示。



上图：虚线代表合成时刻到“虚拟分析时刻”之间的映射
下图：根据“最邻近原则”选择与“虚拟分析时刻”最靠近的分析时刻，由这样的分析时刻为基准获得短时合分析信号的作为相应的短时合成信号

图 4-7

为了利用式(4.2)这个最简合成方程，取 hanning 窗，每帧合成窗长选为当地合成基音周期的 2 倍。

4.3.2 算法流程

我们可以按下列程序流程图实现 TD-PSOLA 方法进行基频尺度修改，具体源程序见附录部分。

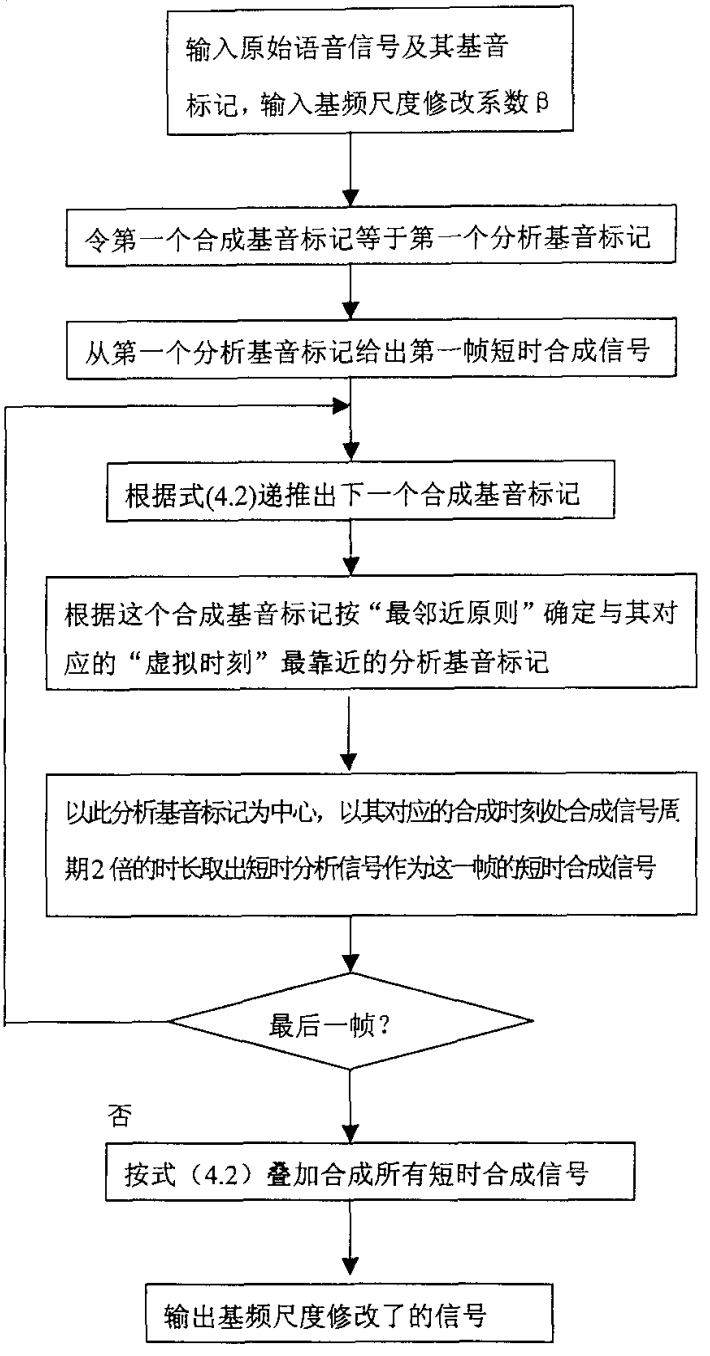


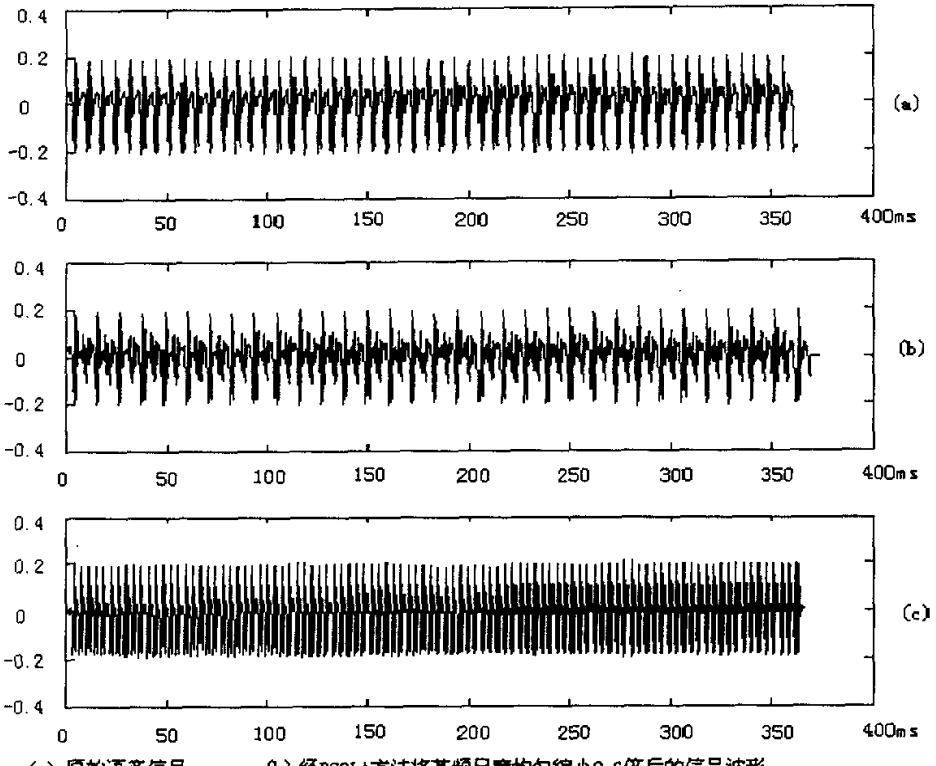
图 4-8

4.3.3 运算结果

4.3.3.1 时域比较

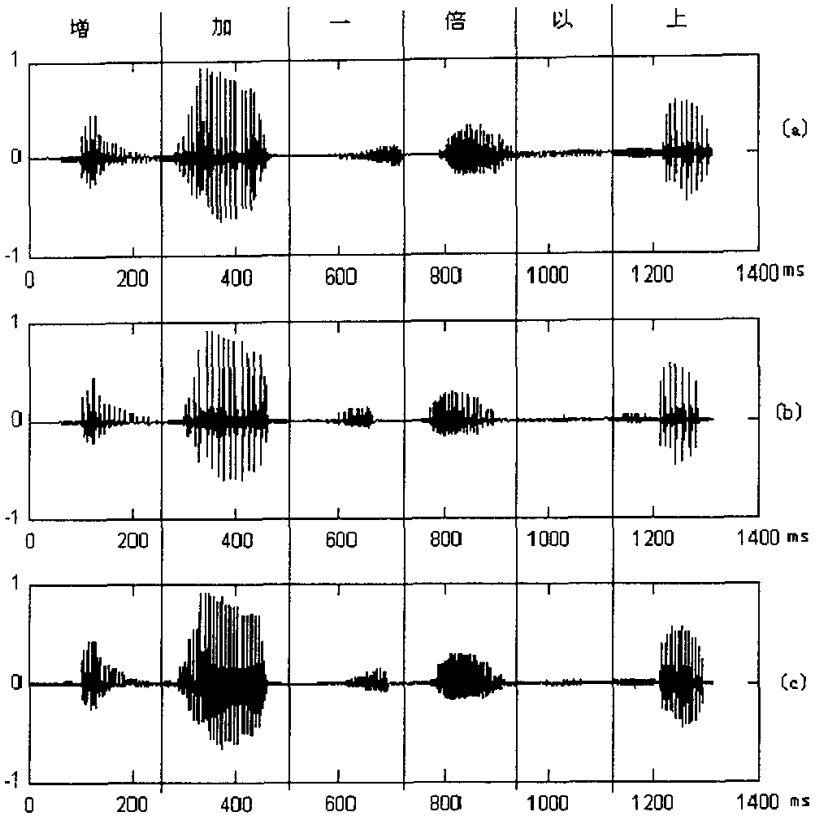
我们首先以单元音/a/为例，原始信号的基音标记预先存放好，每一帧短时合成信号的长度取当地合成基音周期的 2 倍，这样可以使得按式(4.2)叠加方程进行合成，相对于式(2.27)来说，节省了运算开销。

图 4-10 是用 TD-PSOLA 方法作用于一个完整的句子而得到的基频尺度修改后的信号，具体方法是先将原始句子中的每一个字单独切分开来，然后人工为它们找出基音标记，存放在库中，再用 TD-PSOLA 方法对每一个字进行均匀基频尺度修改，最后将修改后的每一个字连结成句。



(a) 原始语音信号。 (b) 经PSOLA方法将基频尺度均匀缩小0.6倍后的信号波形
(c) 经PSOLA方法将基频尺度均匀扩大1.8倍后的信号波形

图 4-9



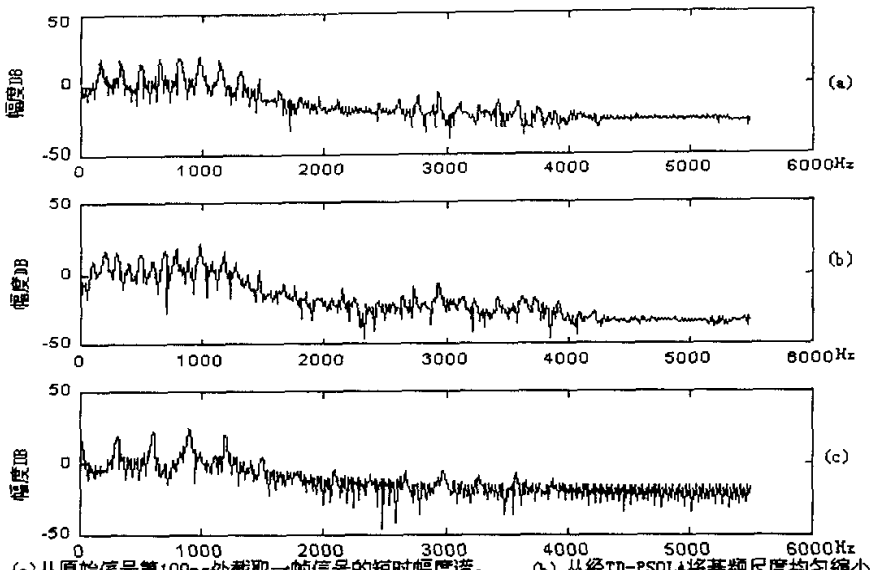
(a) 原始语音信号。 (b) 经TD-PSOLA方法将基频均匀缩小0.8倍后得到的信号。
(c) 经TD-PSOLA方法将基频均匀扩大1.6倍后得到的信号。

图 4-10

比较图 4-10 和图 4-4 可见，TD-PSOLA 方法较 STFT 方法进行基频尺度修改时，修改后的信号能更好得保持准周期性结构。

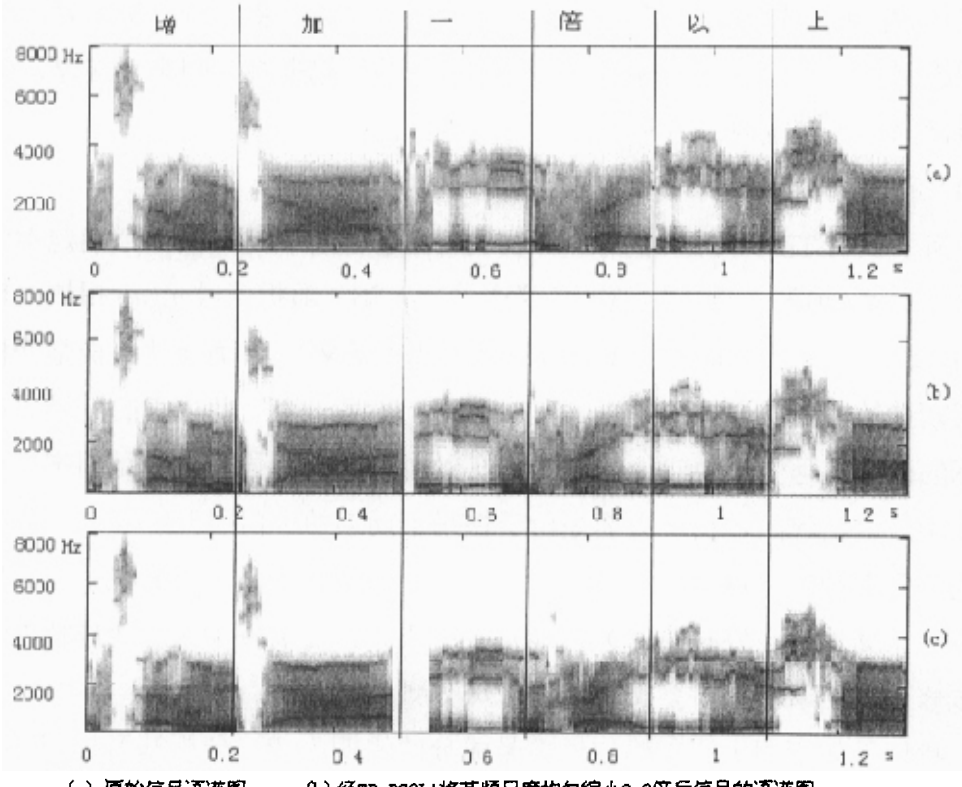
4.3.3.2 频域比较

图 4-11 是从单元音/a/及经 TD-PSOLA 实现基频尺度修改后的信号中同一时刻任取一帧信号作出的幅度谱。类似图 4-5，我们在同样的时刻（第 100ms 处）取同样长度的短时信号。由图 4-11 可见，修改后的信号依然较好地保存了原始信号的谱包络（共振峰结构），而谱线的峰值间距（基音频率）发生了相应的改变。



(a)从原始信号第100ms处截取一帧信号的短时幅度谱。 (b) 从经TD-PSOLA将基频尺度均匀缩小0.6倍的信号中第100ms处截取一帧信号的短时幅度谱。 (c) 从经TD-PSOLA将基频尺度均匀扩大1.6倍的信号中第100ms处截取一帧信号的短时幅度谱。

图 4-11



(a) 原始信号语谱图。 (b) 经TD-PSOLA将基频尺度均匀缩小0.8倍后信号的语谱图。
(c) 经TD-PSOLA将基频尺度均匀扩大1.6倍后信号的语谱图。

图 4-12

比较图 4-11 和图 4-5 可见，用 TD-PSOLA 方法基频尺度修改时，修改后信号的频谱包络不象原始信号频谱那样光滑，而 STFT 方法得到的结果确要好得多。出现这种现象的原因可以从理论上给予解释，Moulines 等在这方面作出过深入的研究[23,25]，本文对此现象不作过多的解释。

图 4-12 是用 TD-PSOLA 方法作用于一个完整的句子而得到的基频尺度修改后信号的语谱图。由图可见，修改后信号的语谱与原始信号的语谱极为相似，这便保证了修改后的信号依然保持很高的可懂度和自然度。就语谱图而言，我们几乎看不出与 STFT 方法得到的信号的语谱有什么差别。

4.3.3.3 听辨结果

基于 TD-PSOLA 方法合成出来的语音具有很高的质量，不论是单元音还是一个完整的语句，在一定的基频修改范围内（基频修改系数在 0.5 到 2 之间），修改后的信号都具有很好的自然度和可懂度。类似于 WSOLA 方法在进行时间尺度修改，由于 TD-PSOLA 合成的实质是将短时分析信号进行复制或拷贝，这样在复制时便会导致合成信号听起来有“回声”现象，但在通常的修改范围内，这种现象不很明显。

4.3.4 TD-PSOLA 方法实现基频修改的解释

为了说明 TD-PSOLA 的原理，我们将语音浊音假定为由一个确定性的周期信号 $d_a(m)$ 和一个零均值的广义平稳（WSS）随机信号 $n_a(m)$ 相加而构成的（这一假定虽带有理想成分但却很有现实意义）。其确定性成份是严格的周期性的，而随机成分用以解释一个周期与另一个周期之间的差异（在实际的语音信号波形中我们能明显的观察到），这些差异一方面是因声带运动的不规则引起的，另一方面是由于在每个周期的声门展开阶段来自肺部的湍流而引起的。这个随机成分在某些音（如摩擦浊音）中或在一定的发音条件（如浊音在发轻声时）可能是主要成分。为了减少人工合成特性，基频修改时应该不要过多影响其随机成分频谱。

我们作如下假定：①确定性信号成分的周期为 P，分析基音标记设置为与基频同步，即 $t_a(n)=nP$ ； ②基音周期修改系数 β 是常数且等于时间尺度修改系数，这样，合成基音标记与分析基音标记便是一一对应关系，

$t_s(n)=n\beta P$； ③在合成时用简化的叠加方程，并且所有的分析窗$h_n(m)$都等于原型窗$h(m)$（该窗的选取应是满足归一化条件$\sum h(t_s(n)-m)=1,\forall m$）。

根据以上假设，合成信号的确定成分为

$$d_s(m)=\sum_n h(t_s(n)-m)d_a(m-t_s(n))=\sum_n x_0(m-n\beta P) \qquad (4.6)$$

其中$x_0(m)=h(-m)d_a(m)$

再根据离散泊松方程，可以证明，合成信号的确定性成份的傅里叶级数为：

$$d_s(m)=\frac{1}{\beta P}\sum_{k=0}^{\beta P-1}X_0(\Omega_k)\exp(j\Omega_k m), \qquad (4.7)$$

其中 $\Omega_k=\frac{2\pi k}{\beta P}$，$X_0(\omega)$是$x_0(m)$的DFT。

由此可见，合成信号的确定成分的谱包络等于原始信号确定成分的谱包络，不同的是谱线的位置出现在$\Omega_k=\frac{2\pi k}{\beta P}$处，而原始信号确定成分的谱线应该在$\omega_k=\frac{2\pi k}{P}$处。

TD-PSOLA 对随机成分的影响在此不详细讨论，Moulines 和 Charpentier 的研究结果是：合成后信号的随机成分不再是白噪声，其自相关是原始随机成分自相关$\sigma^2\delta(\tau)$以周期$(1-\beta)P$重复加权叠加，加权系数是分析窗函数的自相关。

第五章 汉语普通话的韵律合成

§5.1 汉语普通话的声学特征

语音的声学特征包括两方面内容，即音段的特征和超音段的特征。本章将着重讨论与非参数合成密切相关的超音段特征。

5.1.1 汉语普通话的音系特点

汉语普通话的音系特点是界限分明和音节带有声调音位。形成汉语的文字是一字节为单位的文字，一个汉字代表语言里的一个音节。受书面语的影响，说话时，往往一个音节之间稍有停顿和力度减弱，其次绝大多数音节的开头有辅音声母，辅音处在以元音为主的韵母前面，两者形成不同性质的声波交替出现，在听觉上也容易形成音节与音节的界限。另外，对标准音的规范要求，也是促使人们在发音时音节字音的相对独立性和完整性[31]。

汉语的音节分为声母、韵母、声调三部分，声母是处于音节开头的辅音，若音节开头没有辅音则称为零声母。声母中除/m/、/n/、/l/、/r/和零声母以外都是清音，加上零声母，汉语共有 2 2 个声母。见表 5-1

表 5-1

发音方式	浊 / 清度	是否送气	成阻部位					
			唇	齿	齿龈	卷舌	腭	软腭
爆破音	清	不送气	b		d			g
		送气	p		t			k
鼻音	清	送气	m		n			
摩擦音	清	送气	f	s		sh	x	h
	浊	送气				r		
边音	浊	送气			l			
塞擦音	清	不送气		z		zh	j	
		送气		c		ch	q	

音节中声母后面的部分叫做韵母，韵母又可以分为韵头、韵腹、韵尾三部分。其中韵腹是每个音节必须有的。韵母的韵头和韵腹都是元音，韵尾除元音以外还包括两个鼻音/-n/和/-ng/。汉语普通话共有 38 个韵母，按韵头的不同分为①开口呼；②齐齿呼；③合口呼；④撮口呼等四类。具体见表 5-2。

表 5-2

发音方式	单元音韵母	复合元音韵母	复鼻尾音韵母
开口	a o e ê er	ai ei ao ou	an en ang eng
齐齿	i	ia ie iao iou(iu)	ian in iang ing
合口	u	ua uo uai uei(ui)	uan uen(un) uang ueng ong
撮口	ü	üe	üan ün iong

音节具有声调是汉语的一个重要特征。这里的声调具体指汉语中区分意义的声调调型(pitch contour),即阴、阳、上、去四种声调(tone)，如“实验(shi2yan4)”和“誓言(shi4yan2)”两词的声韵结合完全相同，但由于音节声调不同，听者能准确区分它们，从而理解它们所代表的意思。

按声、韵、调的汉语音系结构，可能的音节组合形式有 22×38×4=3344 种，但实际上汉语有严格的组合规则，如/j/、/q/、/x/后面只能是/i/、/ü/开头的韵母。又如当韵母部分是以音位/i/开头的时候，可以出现在声母位置上的辅音只有/b/、/p/、/m/、/d/、/t/、/n/、/l/、/j/、/q/、/x/等十个。而当韵母是以元音/u/开头时，除了/j/、/q/、/x/以外只有两个辅音/n/和/l/可以出现在声母位置上。另外可以组成音节的声韵结构也不是每一种情况都有 4 种声调的音节，因此，实际使用的汉语音节在 1270 个左右。

5.1.2 声调的描述

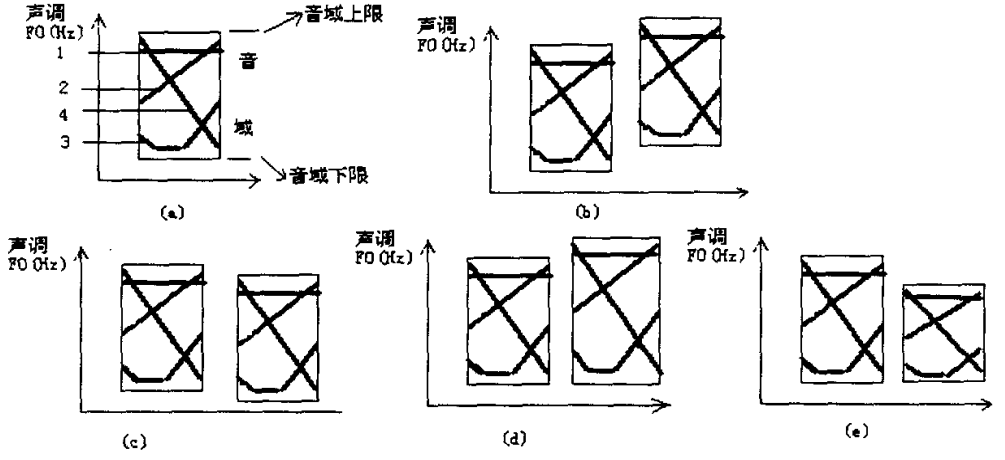
声调加载音节的浊音段，而声母并不携带声调信息[41,42,43]，汉语声调主要体现在韵母段的音高上。虽然如此，绝对的音高值对声调的感知是没有意义的。声调必须用音节间相互比较而显示的相对音高或音节内部音高的升降变化来描述。

汉语普通话共有四个声调，分别是阴平，阳平，上声和去声。沿用音程原理将可以声调的高低分为五级，分别用 1、2、3、4、5 来表示，其中“1”表示最低调值，“5”表示最高调值，这种标调方法被人们称为“五度制标调方法”[9,40]。 五度调制不与任何绝对的基频相对应，他们只有相对音高的高低差别。用五度制表示，普通话的四声分别是 55，35，214 和 51。对汉语的一个词内或一句话内的每个音节来说，都可以用五度调制来描述其声调，但实际上不同位置上的 1 和 5 所代表的绝对音高值并不相同。为了能使每个

位置上的声调的五度制描述都能在绝对音高的坐标系中反映出来，就需要引进音域及音域的上下限的概念。赵元任把音域定义为“不同声调之间以及声调升降极限内的音高范围”[40]。沈炯对音域的概念作了进一步详细的描述[34]：为了将声调和语调分解成相对的音高体系，有必要使用声调音域这个概念，它是语流中特定位置上声调音高特征分布的总音域。在声调音域内，“高”和“低”是相对比而存在的声调特征，“升”和“降”是曲拱本身相对变化的声调特征。高低和升降幅度的相对音高，由声调音域决定。声调音域在语流中不断改变它的高低宽窄。语调就是以句子为单位的音域系列。

在连续语流中，音域的变化，对应着音域上限或下限在绝对音高坐标系中的移动。音域的加宽和压缩导致音域内部曲拱发生量变，单曲拱所反映的声调特征并不改变。一旦确定了音域上下限在绝对音高坐标系中的对应音高值，音域内部各声调的绝对音高也就相应地确定下来。图 5-1 为音高曲线与低音线位置、音域大小和声调的关系示意图。

图 5-1



(a)图：音域及其上、下限和声调在绝对音高坐标系中的关系；(b)图：提高音域；
(c)图：降低音域；(d)图：加宽音域；(e)图：收缩音域。

5.1.3 声调的静态特性和动态特性[32,33]

近十年来，人们通过对声调的各种研究，发现声调在孤立音节中和连续语流两种不同语音环境下有不同的声学表现。在孤立的语音环境中，声调的

调型是相对稳定的，但在连续语流中，相邻音节声调相互影响，同一声调在不同语音环境中会发生不同的变化，因此，对声调的研究有必要分成静态和动态两种情况分别进行。

前文所述的五度制描述四声的调值分别是 55、35、214 和 51，这实际上是静态声调的描述。当音节处于不同的环境中，其调值可能发生不同的变化。声调的动态特性可分为两种，一种是连续变调，这时声调由于受相邻声调的影响，失去原有的声调特征，而转化为其它声调(即声调类型发生变化)。对这种调性的变化目前已达到共识，其主要变调规则如下：①上声变调，上声在非上声之前变半上；上上相连，前上变阳平。②去声变调，去声在去声之前变半去。③“一”，“七”，“八”，“不”的变调，它们在去声之前变阳平，在其它声调之前读本调。④轻声变调，一个字读成轻声后将失去本调，它的音高完全由前音节决定。

声调的另一种动态特性是声调的声调类型不变，但由于受相邻音节声调的影响、语调的影响及重音的影响而发生音域的变化，有时也伴随声调中曲拱的变化。汉语中有二字组、三字组和四字组等主要的几种声调模式。二字组的声调特点是各种声调两两搭配，前字除上声外，其它的组合都成为两个单字调顺式连接的趋势，而后字的调型比本调稍低。三字组的声调特点是是：①除上声外，三个字调相连时是有一条起伏平顺的调势；②中间字是伴随着首字调尾的高低和末字调头的高低来决定自己的调型；③两个上声相连时，前上低（半上）则后上高（变阳平），前上高则后略低（变过渡调或句尾调）；④所有的变调都是顺势相连而不是陡起陡落的，但如前一调型降至最低，则后调另行高起。四字组的声调可以用单字调，双字调和三字调的变调规则来进行变化，具体如何变化与四字组的结构有关。

5.1.4 汉语的重音

重音是一个重要的韵律特征，因此，研究其声学表现对汉语合成有重要的意义。人们往往容易认为重音主要是增加强度，而实际上汉语的重音首先是扩大音域和持续时间，其次才是增加强度[41]。其主要特点有：①基频升高是强调重音的重要声学表现。基频升高的方式与声调的音高特征和曲拱特

征密切相关。具体地说，阴平的基频整个的上升，阳平主要是高音点上升，有时伴随着低音点下降，去声主要是高音点上升，上声读成半上时低音点下降，读成全上时调尾上升。②强调重音的时长普遍加长。就重读阳平和去声来说，音高升高，时长加长；阴平重读时时长与音高的变化存在着互补的关系，基频较高时，时长增加的幅度相对减少，反之，时长增加的幅度加大；对于上声重读时，基频变化幅度不大，时长有明显的加长。③音节强调重读时对其强度没有明显影响。

5.1.5 汉语的语调

象汉语这样的声调语言的基音变化，不能看成各种声调的简单连接，在不同声调之间即声调升降极限以内的音高范围，也是一个变量，随着发音的力量和发声的力量而改变。这种音高范围（即音域）的变化，在连续语流中因轻重、快慢以及其他种种原因会发生上移、下移、加宽或收缩等变化[40]。音节的声调和句子的语调就好像“小波浪跨在大波浪上面，实际结果是这两种波浪的代数和”[41]。

声调和语调可以进一步被分解为两个相对音高体系，声调音域在语流中不断改变它的高低宽窄，语调就是以句子为单位的声调音域系列。经过这种切分，语调是由一连串声调音域组织起来的音高调节形式，声调是在声调音域中滑动的曲拱。语调对声调音域有调节作用，声调音域的改变表现在声调曲拱发生的量变上。语调对音域的上限和下限分别起调节作用，上限的调节变化和语义的加强相关，下限的调节变化和节奏结构的完整性相关[34]。

总之，调域的上下限用来限制声调的变化范围，调域在语流中的不同位置上的上升、下降或展宽、压窄用来体现语调。

5.1.6 影响音节时长的因素

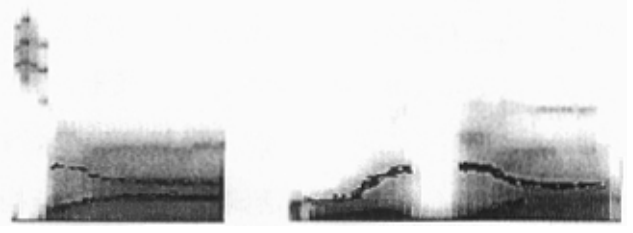
对普通话时长特征研究可以在音位层、音节层、音节组合层和语句层等不同层次上进行。目前在音位层和音节层上的研究结果比较多。对单发音节来说，通常阴平 436ms，阳平 455ms，上声 483ms，去声 425ms[39]。在连续语流中，声母的发音方法直接影响它的时长，不同单韵母的时长也不同，单韵母的时长对声母的时长还有补偿关系，说话速度、声调对时长都有影响，

此外，音节与音节组合成词时，音节的位置、多音节词的组合结构都将对音节的时长产生影响[36]。

5.1.7 音节间的协同发音

协同发音（co-articulation）是指“两个以上的发音特征同时出现”的语音现象。汉语普通话的音节内部往往存在着这种现象，其语谱表现为不同音位之间出现共振峰的平滑过渡，例如图 5-2 中“家”的语谱，我们明显的可以看出音位/i/的共振峰向/a/的过渡。

图 5-2



“家”的语谱 “国家”的语谱

协同发音效应不仅存在于音节内部的各音位之间，也存在于相邻音节之间，例如图 5-2 中“国家”的语谱，音节“国”尾部的第二共振峰向“家”中前部音位/i/的第二共振峰过渡。许毅分析了前后音节各种组合关系中的共振峰过渡特性，他的研究成果得到唐涤飞的证实。音节间共振峰过渡在语音感知中远没有基频的过渡重要，一般人对音节间的共振峰的过渡是不敏感的[38]。

§5.2 TD-PSOLA 方法进行汉语普通话韵律合成

在汉语普通话文语转换系统中，韵律合成是一个极其重要的环节，它是保证合成语音具有良好自然度和可懂度的关键。根据本章前一节的讨论我们知道，汉语普通话是一种具有声调和语调的语言，因此在实际的韵律合成过程中，我们既要进行时间尺度修改，又要进行基频尺度修改。很显然，这种修改尺度并不是均匀的。通常可以先进行时间尺度修改，然后再进行基频尺度修改。由于在处理时我们是分帧进行，如果分别进行时间、基频尺度修改，必然导致运算开销太大，很难保证韵律合成的实时性要求。为此，本节采用 TD-PSOLA 方法同时进行时间、基频尺度修改。通常，时间用均匀尺度修改对自然度影响不太大，故而本节为简单起见，将时间尺度修改采用均匀方法。

5.2.1. TD-PSOLA 方法进行汉语普通话韵律合成模块结构

对于一个实际的汉语文语转换系统，我们可以采用图 5-3 流程图来实现。在本章，我们将重点讨论 TD-PSOLA 方法的韵律合成模块，程序的源代码见附录部分，图 5-4 是这一模块的程序流程图。

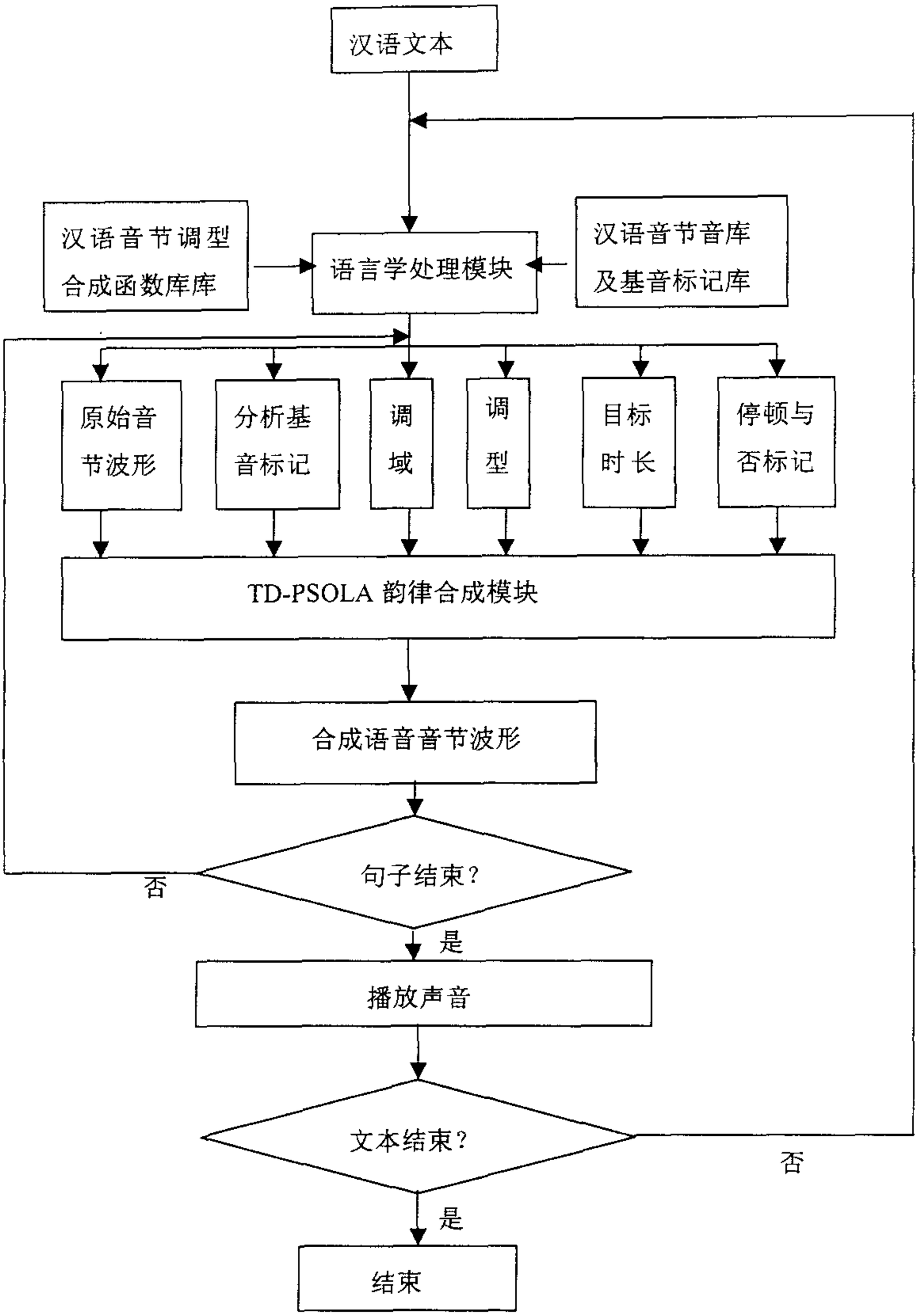

汉语文本
汉语音节调型
合成函数库库
语言学处理模块
汉语音节音库
及基音标记库
原始音
节波形
分析基
音标记
调
域
调
型
目标
时长
停顿与
否标记
TD-PSOLA 韵律合成模块
合成语音音节波形
句子结束？
否
是
播放声音
文本结束？
否
是
结束

图 5-3

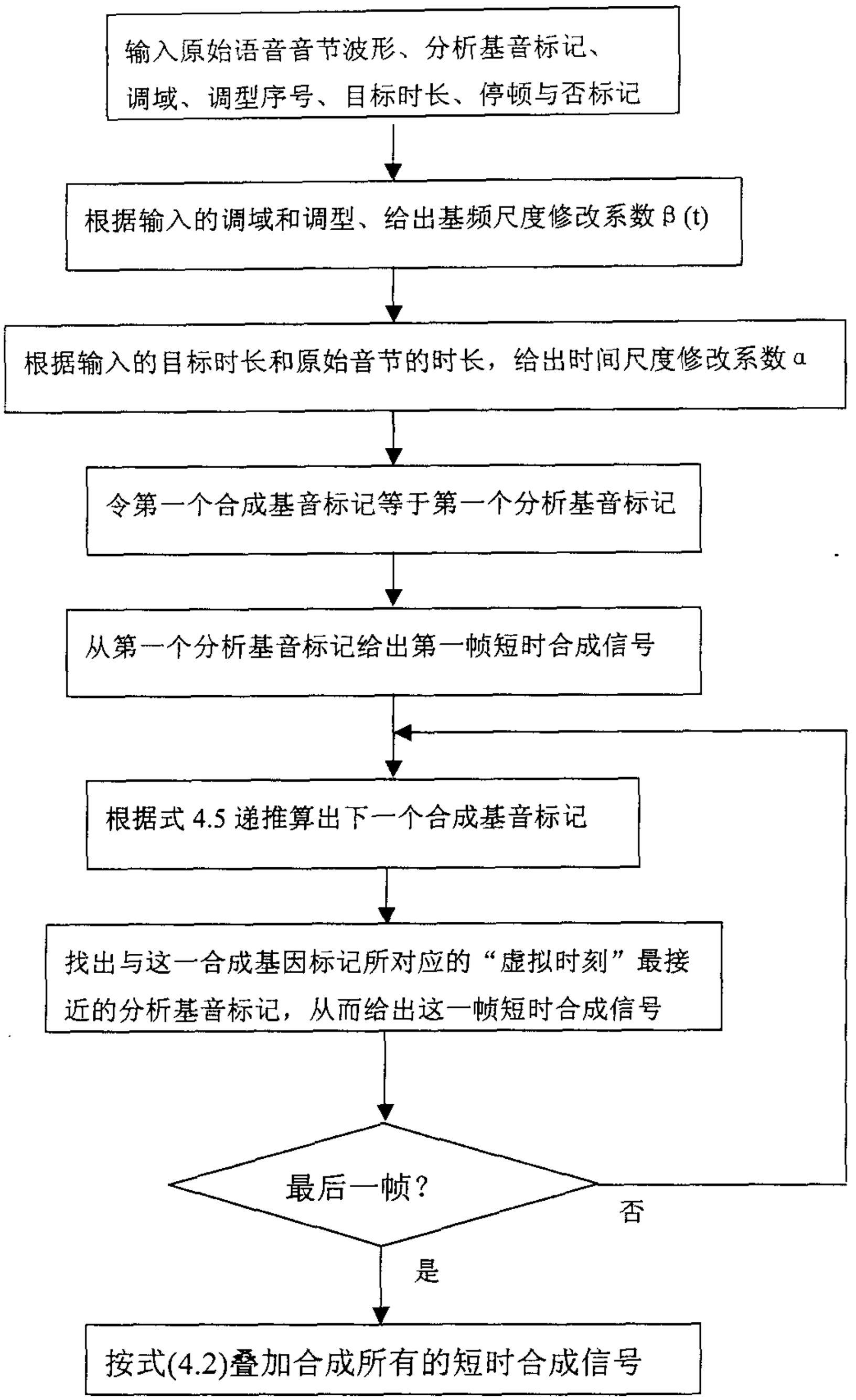

输入原始语音音节波形、分析基音标记、
调域、调型序号、目标时长、停顿与否标记
根据输入的调域和调型、给出基频尺度修改系数 β (t)
根据输入的目标时长和原始音节的时长，给出时间尺度修改系数 α
令第一个合成基音标记等于第一个分析基音标记
从第一个分析基音标记给出第一帧短时合成信号
根据式 4.5 递推算出下一个合成基音标记
找出与这一合成基因标记所对应的“虚拟时刻”最接近的分析基音标记，从而给出这一帧短时合成信号
最后一帧？
否
是
按式(4.2)叠加合成所有的短时合成信号

图 5-4

我们可以将汉语普通话所有的音节预先录好音存入音库中，并将一些儿化音（如“字儿”、“小鸡儿”、“今儿”、“丝儿”）当作单音节录入库中（20-30个）。然后人工对这些音节（或儿化音节）进行基音标记，对于他们声母中的清音部分，按均匀间隔也对他们进行“标记”。这样，每个音节的所有这些“标记”便构成“分析时刻”序列，将所有的“分析时刻”序列存入基音标记库。

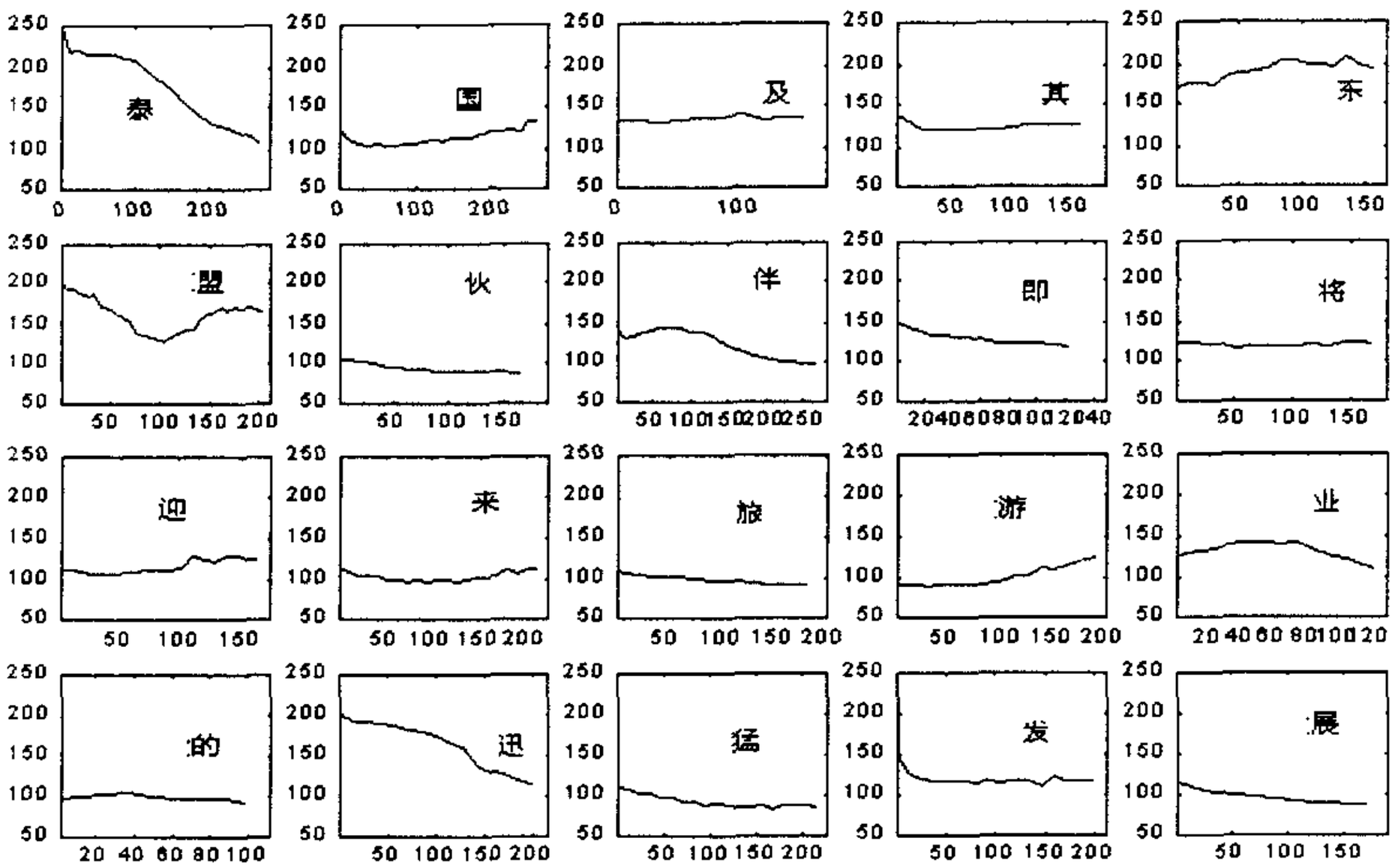
本章前面我们谈到，实际的汉语具有声调和语调两个方面的不同特性，这样，实际的汉语语句中每个音节的调型、调域和时长便与这个音节单独发音时的调型、调域和时长往往有很大不同，同一句中不同音节的调域往往也是不同的，即五度标记法中的 1、2、3、4、5 不仅对于不同人来说代表的基音频率不同，就是同一个人说的同一句话中，标记不同音节调型的这五个“音阶”也是不同的。因此，语言学处理模块应对输入的文本进行语义分析，然后根据汉语的句子发音的词调规则和语调规则给出每个音节的调域、调型曲线以及时长等参数，韵律合成模块根据语言学处理模块给出的这些韵律参数对原始音库中的音节进行韵律合成（韵律修改），最后将组成句子的所有音节修改后的波形进行拼接。

本文讨论的重点是韵律合成模块，在这一模块中，我们通过用几个子模块实现韵律合成。datainput 模块分别从音库中读入语音波形数据及从基音标记库中读入这个音节的分析时刻序列，pitchmodi 模块根据目的调型和原始分析时刻$t_a(s)$给出基频修改系数$\beta(t)$和分析基音轮廓$P(t)$，该模块将输入的音节分浊、清两部分分别处理，对于清音段，基频是没有意义的，置$\beta(t)\equiv 1$；对浊音段，先根据输入的目标调型给出目标基音轮廓，再根据目标基音轮廓和该音节原始基音轮廓确定该段基频修改系数$\beta(t)$。Timemodi 模块根据目的时长和音节原始时长给出时长修改系数。Synpmark 模块根据基频修改系数$\beta(t)$、分析基音轮廓$P(t)$、时间修改系数 α 和原始分析时刻$t_a(s)$，根据(4.5)给出合成时刻$t_s(u)$（合成基音标记）和虚拟分析时刻$t_v(u)$(意义见图 4-7)。Synthesis 模块根据第 u 个合成时刻$t_s(u)$对应的虚拟分析时刻$t_v(u)$，找出与

它最近的分析时刻$t_a(s)$，以$t_a(s)$为中心，在原始信号中取$2P'(t_s(u))$长度的短时分析信号作为第 u 帧短时合成信号（$P'(t_s(u))$是第 u 个合成时刻$t_s(u)$处的合成信号周期）。最后按式(4.2)叠加合成所有的短时合成信号，得到该音节的韵律合成信号。

5.2.2　韵律合成参数的表示

我们分析了一句广播发音“泰国及其东盟伙伴，即将迎来旅游业的迅猛发展”的主要韵律参数，其“基音轮廓（contour）图”如下：



样句的基音轮廓图，横坐标是时间，单位ms。纵坐标是基音频率 ，单位是Hz

图 5-5

根据分析得到的基音轮廓,我们将这一句每一个音节的调型曲线和调域以及时长记录于表 5-3:

表 5-3

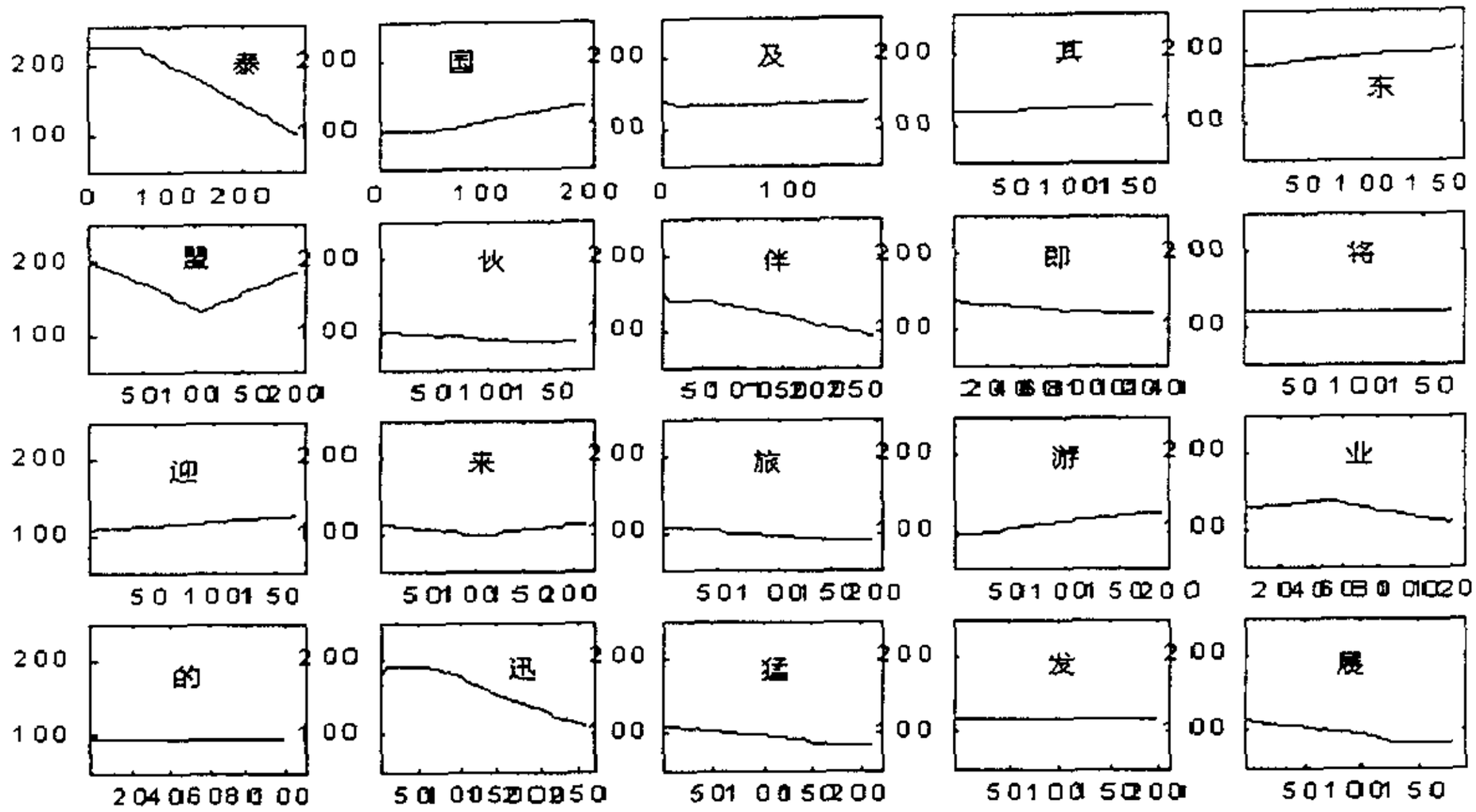
音节	调型	调域	时长	音节	调型	调域	时长
泰	51	105Hz-220Hz	284ms	迎	35	85Hz-125Hz	186ms
国	335	71Hz-135Hz	202ms	来	535	84Hz-112Hz	222ms
及	35	124Hz-136Hz	173ms	旅	211	92Hz-152Hz	202ms
其	35	115Hz-127Hz	183ms	游	35	55Hz-125Hz	201ms
东	35	150Hz-200Hz	167ms	业	351	110Hz-140Hz	130ms
盟	514	130Hz-200Hz	210ms	的	55	95Hz-95Hz	111ms
伙	211	88Hz-152Hz	191ms	迅	51	110Hz-190Hz	271ms
伴	51	95Hz-145Hz	282ms	猛	211	85Hz-185Hz	224ms
即	211	120Hz-180Hz	149ms	发	55	118Hz-118Hz	212ms
将	55	120Hz-120Hz	181ms	展	211	85Hz-205Hz	190ms

我们采用五度标记法来表示调型，其中音阶 1 表示调域中最低基音频率，音阶 5 表示调域中最高基音频率，余者在最低基频与最高基频之间均匀分布，这样，TD-PSOLA 根据音阶组合生成合成信号的基音轮廓。

在合成之前，我们单独录下这二十个音节，人工记录下它们的基音标记，给出分析时刻，存放到库中，然后运用 TD-PSOLA 韵律合成模块，按表 5-1 的韵律参数对音库中的单音节进行韵律修改，最后拼接成句。

5.2.3 韵律参数合成效果

我们对上述语句按上述参数的合成结果进行了韵律参数的提取，其基音轮廓如图 5-6 所示：比较图 5-6 和图 5-5 可见，合成后的基音轮廓与样句的基音轮廓图曲线非常相似。表 5-4 是输入的韵律参数（样句韵律参数）和实际合成获得的参数比较结果，通过比较可见，该算法能够相当精确的实现韵律参数修改。



韵律合成后语句的基音轮廓图，横坐标是时间，单位是ms，纵坐标是基音频率，单位是Hz

图 5-6

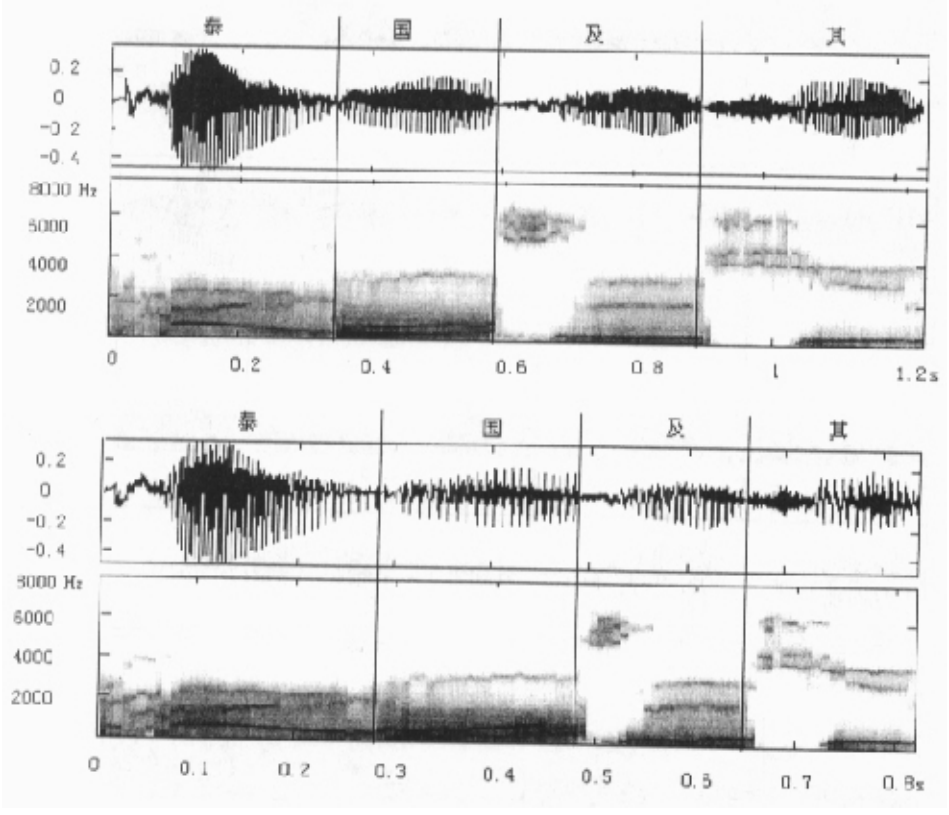
表 5-4

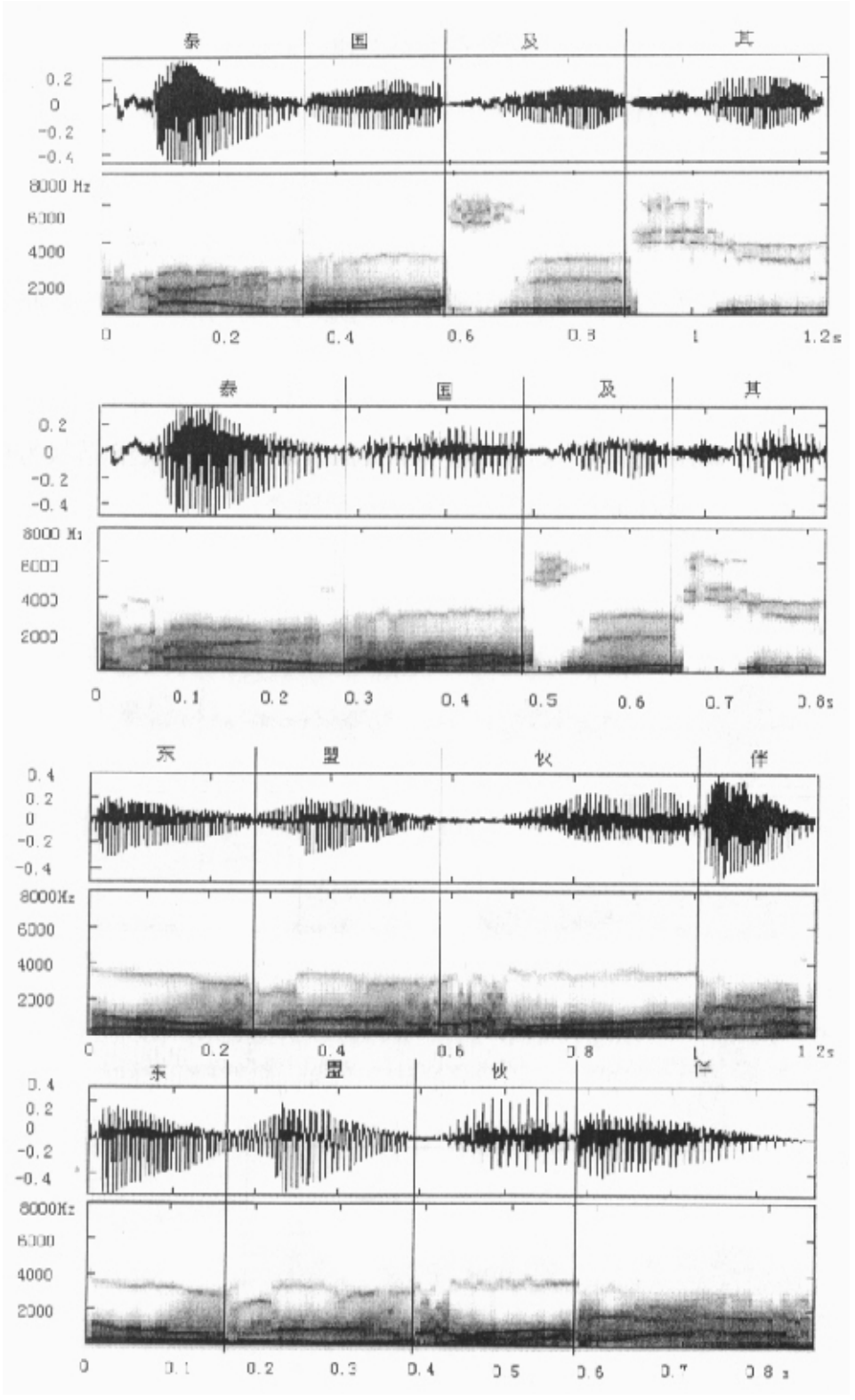
音节	输入时长	输入调域	输出合成信号时长	输出合成信号调域
泰	284ms	105Hz-220Hz	284ms	104Hz-222Hz
国	202ms	71Hz-135Hz	202ms	70Hz-136Hz
及	173ms	124Hz-136Hz	173ms	125Hz-139Hz
其	183ms	115Hz-127Hz	183ms	117Hz-127Hz
东	167ms	150Hz-200Hz	167ms	154Hz-202Hz
盟	210ms	130Hz-200Hz	210ms	133Hz-198Hz
伙	191ms	88Hz-152Hz	191ms	88Hz-148Hz
伴	282ms	95Hz-145Hz	282ms	95Hz-140Hz
即	149ms	120Hz-180Hz	149ms	121Hz-173Hz
将	181ms	120Hz-120Hz	181ms	120Hz-120Hz
迎	186ms	85Hz-125Hz	186ms	88Hz-126Hz
来	222ms	84Hz-112Hz	222ms	86Hz-112Hz
旅	202ms	92Hz-152Hz	202ms	92Hz-152Hz
游	201ms	55Hz-125Hz	201ms	63Hz-125Hz
业	130ms	110Hz-140Hz	130ms	110Hz-139Hz
的	111ms	95Hz-95Hz	111ms	95Hz-95Hz
迅	271ms	110Hz-190Hz	271ms	110Hz-190Hz
猛	224ms	85Hz-185Hz	224ms	86Hz-178Hz
发	212ms	118Hz-118Hz	212ms	119Hz-119Hz
展	190ms	85Hz-205Hz	190ms	86Hz-194Hz

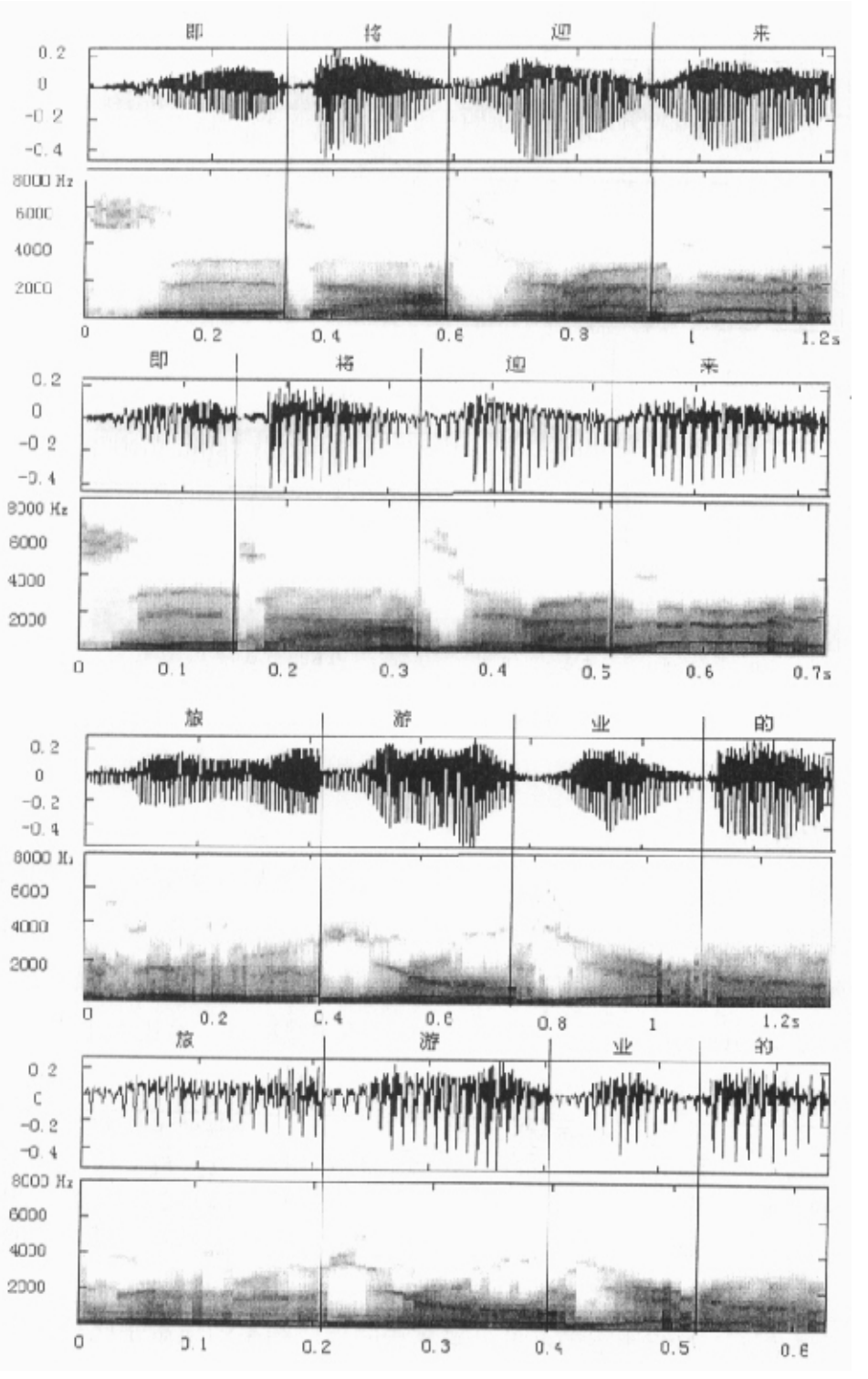
§5.3 汉语普通话韵律合成质量

5.3.1 合成信号时域和频域特性

我们分别将用于合成语句的音库中各个单音节的波形和语谱与合成语句的波形和语谱进行比较。图 5-7 中上面的部分是各个原始单音节的波形和语谱，下面的部分是合成语句的波形和语谱。比较韵律修改前后的时域波形可见，时间尺度已作了相当大幅度的改变，基频修改的幅度其实也较大，合成信号依然保存着良好的准周期性。比较韵律修改前后的语谱可见，尽管信号已作了相当大幅度的基频和时间尺度修改，但语谱依然与原始音节极为相似，这便保证了合成后的信号依然保持很高的清晰度。




泰
国
及
其
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.2
0.4
0.6
0.8
1
1.2s
泰
国
及
其
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8s
东
盟
伙
伴
0.4
0.2
0
-0.2
-0.4
8000Hz
6000
4000
2000
0
0.2
0.4
0.6
0.8
1
1.2s
东
盟
伙
伴
0.4
0.2
0
-0.2
-0.4
8000Hz
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8 s


即
将
迎
来
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.2
0.4
0.6
0.8
1
1.2s
即
将
迎
来
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7s
旅
游
业
的
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.2
0.4
0.6
0.8
1
1.2s
旅
游
业
的
0.2
0
-0.2
-0.4
8000 Hz
6000
4000
2000
0
0.1
0.2
0.3
0.4
0.5
0.6

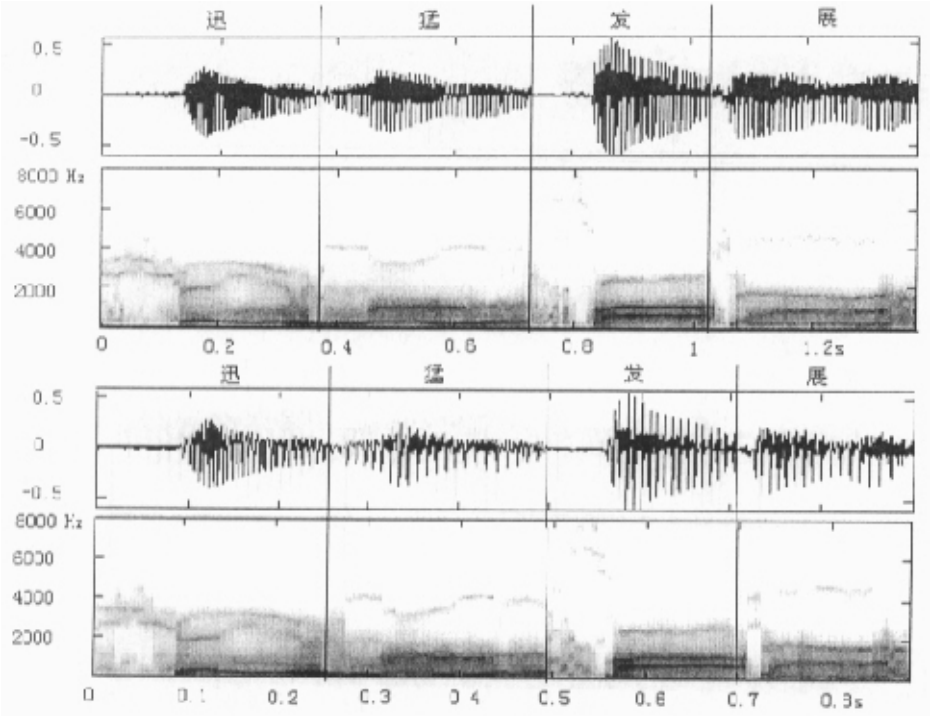


图 5-7

5.3.2 听辨结果

我们让 20 名大学生首先在未知内容对这句合成出来的句子的情况下进行听辨实验，听完三遍后写出句子内容。然后对语调的自然度进行评估，最后让他们听通过自然录音得到的样句（其韵律参数作为我们合成语句时的输入参数），指出合成语句的语调与样句语调的相似程度，结果如下：

表 5-5

听写正确率	合成语句自然度					语调的相似程度			
	很自然	比较自然	可以忍受	难以忍受	极不自然	很相似	比较相似	不太相似	极不相似
98%	20%	65%	10%	5%	0%	35%	55%	10%	0%

通过表 5-5 的统计结果可见，本算法的合成结果的质量是比较好的。

5.3.3 存在的问题

TD-PSOLA 虽然能较好的进行韵律合成，尤其是其运算开销是目前各种

非参数合成方法中最小的，但这种方法最大的缺陷是对语音的共振峰参数的影响还不能控制。因此，虽然对于一个音节内部的协同发音（在语谱中表现为音节内部不同音素之间的共振峰的平滑过渡），这种算法能将它很好的保存下来（只能是保存，并非合成出来的），但是，对于语句中音节之间的协同发音现象（在语谱中表现为音节之间的共振峰的平滑过渡），TD-PSOLA 却无法修改原始音库中孤立音节的共振峰，使不同音节之间出现共振峰的平滑过渡，从而模拟自然发音语句中的这种协同发音现象。根据前面我们讨论的唐涤飞的对音节间共振峰听觉感知特性的研究结果我们知道，虽然 TD-PSOLA 不能合成音节间的共振峰过渡，但由于一般人对此感知不太敏感，因此合成效果听起来还是比较自然清晰的。但是要想进一步提高合成的自然度和清晰度，还得考虑协同发音效应。

第六章 总结

本文以语音的非参数韵律合成为研究目标，讨论了在时域和频域进行韵律合成的理论依据和实际合成的方法以及合成的效果，最后着重讨论了怎样用 TD-PSOLA 方法实现高质量的汉语普通话的韵律合成。

通过研究基于 STFT 基础之上的非参数韵律合成可见，直接 STFT 方法除了可以修改韵律参数外，还具有修改共振峰结构的能力，但是运算开销太大，尤其在“相位展开”时理想情况与实际情况差距较大，从而导致合成信号波形周期性结构差。TD-PSOLA 算法在实现韵律参数修改方面具有运算开销少，韵律参数合成精度高，合成语音质量高等优点，在汉语普通话合成方面，比以往各种合成方法（如各种共振峰合成器、波形编码合成器）无论在运算开销还是在语句的自然韵律合成上都有着好得多的效果。就所需的存储开销而言，也仅需 20M-30M 的存储空间即可容纳整个汉语普通话的音库以及基音标记库，无需额外硬件，在个人 PC 机上完全可以达到实时性的要求。

对于实际的文语转换系统，语言处理模块对要合成的句子给出好的韵律参数是非常很重要的，但如何能根据给定的韵律参数加以精确合成，并且达到高清晰度高自然度便是最终实现高质量的文语转换系统的关键。目前世界上已研究出多种语言的基于 PSOLA 技术的 TTS 系统，国内一些科研单位对汉语文语转换系统也进行了大量的研究。通常 TD-PSOLA 在进行韵律修改时，不特别考虑清音，而是将目标调型参数加到整个基音单元上，这样基频这一韵律参数便不能精确实现，再就是时间和基频尺度分别进行修改，也必然导致合成韵律参数误差较大。本文根据汉语普通话的音系特点，为了实现高精确度的韵律参数合成，采用动态的合成调型的方法。即根据输入的调型序号、调域和时长，动态的合成出目标基音轮廓曲。这样就能考虑汉语音节中声母中清音无调这一特性，对同一个音节，实现在清音段，相邻合成基音标记间隔等于分析基音标记间隔（恒等于一常数，没有基音周期意义），而目标基音轮廓曲线全部加在音节中的浊音段部分，并

且实现时间和基频尺度同时修改，从而不仅节约了运算开销，更主要的是减少了因计算合成基音标记不准确而产生的合成韵律参数的误差。

虽然本文的方法能够合成出高质量的汉语普通话语句，但是它存在的缺陷也不容忽视，如何合成出实际自然发音语言中的协同发音问题，如何消除背景回声，如何对语流中的强度变化提出一个合理的轮廓从而加以合成，如何用变化的时间尺度修改系数更为精确的修改音节内部各音位的时长等，尚需要做大量的工作。

参考文献

[1]中国科技大学人机语音通信国家重点实验室，唐浩，专题报道“语音合成技术应用实例”，《产品与技术》，2000.3.20
[2]清华大学计算机智能技术与系统国家重点实验室，吴志勇，蔡莲红，专题报道，“语音合成技术的原理”，《产品与技术》，2000.3.20
[3] 中国科技大学人机语音通信国家重点实验室，王仁华，专题报道“让计算机开口讲话——语音合成技术及国内外发展现状”，《产品与技术》，2000.3.20
[4]T.W.Parson 著,王成义等译,《语音处理》,国防工业出版社，1990 .
[5]张贤达，《现代信号处理》,清华大学出版社，1995.
[6]Kay, S. M 著，黄建国等译，《现代谱估计》，科学出版社，1994.
[7]A. V. Oppenheim and R. W. Schafer “Digital Signal Processing”. Englewood Cliffs,NJ:Prentice-Hall,1975.
[8]L. R. Rabiner and R. W. Schafer , “Digital Processing of Speech Signals”. Englewood Cliffs,NJ:Prentice-Hall,1978.
[9]杨行俊，迟惠生，《语言信号数字处理》,电子工业出版社, 1995.
[10]M.R.Portnoff, “Time-frequency representation of digital signals and systems based on short-time Fourier analysis”, IEEE Trans. On ASSP, vol28, pp55-69,1980
[11].M.R.Portnoff “Short-time Fourier analysis of sampled speech” ,IEEE Trans. On ASSP, vol29, no3, pp364-367, 1981.
[12].D.Griffin and J.Lim “ Signal estimation from modified short_time Fourier transform” ,IEEE Trans.On ASSP vol32,no.2, pp236-243,1984.
[13]M.R.Portnoff “Time-scale Modification Based on Short-Time Fourier Analysis”,IEEE Trans. On ASSP vol29,no.3,pp374-390,1981.
[14]J.B.Allen “Short-time spectral analysis of sampled speech” ,IEEE Trans. On ASSP,vol25,pp235-238,1977.
[15]E.Moulines and J.Laroche “None-parametric techniques for pitch-scale and time-scale modification of speech”, Speech Communication ,vol.16, no.2,pp175-207,1995
[16]H.Valbret,E.Moulines and J.P.Tubach, “Voice transformation using PSOLA techniques”, Speech Communication ,vol.11, nos.2-3,pp175-187,1992
[17]S.Roucos and A.Wilgus “High quality time-scale modification of speech”, in Proc.ICASSP,pp493-496,1985.
[18]S.Hamid Nawab ,Thomos F. Quatieri and Jae S.Lim “Signal Resconstruction From Short-Time Fourier Transform Magnitude”, IEEE Trans.On ASSP vol31,no.4, pp786-794,1983.
[19]W.Verhelst and M.Roelands, “An Overlap-add technique based on waveform similarity (wsola) for high-quality time-scale modification of speech ”,in Proc.ICASSPpp554-557,1993.
[20]DOUGLAS B.PAUL “The Spectral Envelop Estimation Vocoder ”, IEEE Trans.On ASSP vol29, no.4,1981.
[21]STEPHANIE SENEFF “System to Independently Modify Excitation and/or Spectrum of Speech Waveform Without Explicit Pitch Extraction”, IEEE Trans.On ASSP vol30, pp566-578,1982.

speech wave," J.Acoust.Soc.Amer,vol50,pp637-655,1971
[23]E.Moulines and F.Charpentier "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones ",Speech Comm.,vol.9, no.5, pp.453-467,1990.
[24]F.Charpentier and M.Stella "Diphone synthesis using an overlap-add technique for speech waveform concatenation ", in Proc. Int. Conf. On Acoust. Speech and Sig. Proc. pp.2015-2018, 1986.
[25]C. Hamon, E. Moulines and F. Charpentier , "A diphone synthesis system based on time-domain modifications of speech" ,in Proc. Int. Conf. Acoust.,speech,Sig. Proc. ,Glasgow, pp.238-241,1989.
[26]N. J. Miller , "Pitch Detection By Data Reduction",IEEE trans. On ASSP vol23, pp.72-74,1975.
[27]C. A. Mcgonegal, L. R .Rabiner and A. E. Rosenberg, "A Semiautomatic Pitch Detector", IEEE trans. On ASSP vol23, pp.570-574,1975.
[28]C. Ma, Y. Kamp, and L.Willems, " A frobenius norm approach to glottal closure detection from the speech signal ", IEEE Trans. On Speech and Audio,vol.2, no.2, pp.258-265,1994.
[29]K. Lukaszewickz and M. Karjalainen , "Microphonemic method of speech synthesis" .Proc. Int. Conf. Acoust. Speech, Signal Proc.,Dallas,p-p.1426-1429, 1987.
[30]R.E.CROCHIERE, "A Weighted Overlap-Add Method of Short –Time Fourier Analysis/Synthesis ", IEEE Trans。 On Acoustic, Speech, and Signal Proc. vol28,no.1,pp99-102,1980
[31]周同春,《汉语语音学》,北京师范大学出版社,1989.
[32]吴宗济,"普通话语句中的声调变化",中国语文,1982 年第 6 期,P439-449.
[33]吴宗济,"普通话三字组变调规律",中国语言学报,1985 年第 2 期,P70-92.
[34]沈炯,"北京话的声调和语调",北京语音实验录,北京大学出版社,P72-130,1985.
[35]沈炯等,"汉语语势重音的音理(简要报告)",语文研究,1994 年第 3 期,P10-15.
[36]冯隆,"北京话语流中声韵调的时长",北京语音实验录,北京大学出版社,P11-195.
[37]初敏,吕士楠和陆亚民,"利用基音同步叠加技术合成汉语的研究",第三届全国人及语音通讯学术会议论文集,P.394-397,1994.
[38]唐涤飞,吕士楠,周同春,王仁华,"汉语合成协同发音规则初探",第六届全国语音图像通讯信号处理学术会议(SICS'93)论文,P75-78.
[39] 初敏,"高清晰度高自然度汉语文语转换系统的研究",中国科学院声学研究所,1995.
[40]赵元任,"Tone and Intonation in Chinese",历史语言研究所集刊,4 本 2 分,P.121-134,1933.
[41]赵元任,《汉语口语语法》,吕叔湘译,商务印书馆,1979.
[42]林焘和王士元,"声调感知问题",中国语言学报,1984 年第 2 期,P59-69.
[43]林茂灿,"北京话声调分布域的感知实验研究",中国社会科学院语言研究所语音实验室《语音研究报告》(1992-1993),p19-29.
[44]Michiael W.Macon and Mark A. Clememts , "Speech Concatenation and

synthesis using an overlap-add sinusoidal model",in Proc. ICASSP 96,pp.361-364
[45] Michiael W.Macon and Mark A. "An enhanced ABS/OLA sinusoidal model for wavform synthesis in TTS", Eurospeech ,1999,vol.5, pp.2327-2330.
[46]E.Bryan George , "Practical high-quality speech and voice synthesis using fixed frame rate ABS/OLA sinusoidal modeling",in Proc. ICASSP 98 vol. 1,pp.301-304
[47]Ann Syrdal, Yannis Stylianou ,Laurie Garrison ,Alistair Conkie and Juergen Schroeter "TD-PSOLA Versus Harmonic Plus Noise Model In Diphone Based Speech Synthesis",in Proc. ICASSP,1998,pp.273-276.

附 录

MATLAB 语 言 程 序 文 件:

（1） 基于 STFT 在频域进行时间尺度修改的源程序文件

tscale.m 文 件

```
function [x,y]=tscale(wav,coeff)
fid=fopen(wav,'rb');
data=fread(fid,'int16');
fclose(fid);
L=length(data);
for i=1:L-22  %eliminate the head file
   x(i)=data(i+22)/2^15;
end
N=600;
M=1024;
anaw=hanning(N);
synw=hanning(N);
R=N/4;
L0=fix(length(x)*coeff);
for i=1:L0
   y0(i)=0;
   f(i)=0;
end
L1=fix((length(x))/R);
for m=1:L1
   m
   ta(m)=(m-1)*R;%ta is analysis time instants
   ts(m)=round(coeff*ta(m));%ts is synthesis time instants

   for i=1:N
   if ta(m)+i>length(x)
      Sw(i)=0;
   else
      Sw(i)=x(ta(m)+i)*anaw(i);
   end
end
```


75

```
    anaspec=fft(Sw,M);%short time analysis spectrum
for i=1:M/2
    anaspecmag(i)=abs(anaspec(i));
    anaspecpha(i)=angle(anaspec(i));
end
if (m<=1)
   synspecpha=anaspecpha;
else
      for i=1:M/2

z(i)=anaspecpha(i)-preanaspecpha(i)-2*pi*(i-1)/M*(ta(m)-ta(m-1));
          while(z(i)>=pi)
             z(i)=z(i)-2*pi;
          end
          while(z(i)<=-pi)
             z(i)=z(i)+2*pi;
          end
          Langmita(i)=2*pi/M*(i-1)+z(i)/(ta(m)-ta(m-1));
          synspecpha(i)=synspecpha(i)+(ts(m)-ts(m-1))*Langmita(i);
    end

end
preanaspecpha=anaspecpha;

temp0=anaspecmag.*exp(j*synspecpha);
for i=1:M/2
   synspec(i)=temp0(i);
end
synspec(M/2+1)=conj(synspec(M/2));
for i=M/2+2:M
   synspec(i)=conj(synspec(M-i+2));
end

temp=ifft(synspec,M);
synsig=real(temp(1:N)).*synw';
```


76

```
for i=1:N
  if ts(m)+i<=L0
     y0(ts(m)+i)=y0(ts(m)+i)+synsig(i);
     f(ts(m)+i)=f(ts(m)+i)+synw(i)*synw(i);
    end
end
```

```
clear Sw synsig z temp temp0 anaspec anaspecmag anaspecpha synspec
synspecmag
end
for i=1:L0
   y(i)=y0(i)/f(i);
end

   subplot(2,1,1)
   plot(x)
   subplot(2,1,2)
   plot(y)
```

（2）WSOLA 方法在时域实现时间尺度修改的源程序文件

wsola.m 文 件:

```
function [x,y]=wsola(wav,coeff)
fid=fopen(wav,'rb');
data=fread(fid,'int16');
fclose(fid);
L=length(data);
for i=1:L-22  %eliminate the head file
   x(i)=data(i+22)/2^15;
end

N=256;
w=hanning(N);
R=N/2;
L1=fix((length(x)*coeff/R));
averD(1)=R;
```


77

```
for i=1:N
   synseg(1,i)=x(i)*w(i);
   model(i)=x(i+R);
end
ts=2*R;
for m=2:L1
   averD(m)=fix(m*R/coeff);
     if abs(ts-averD(m))<50
   for i=1:N
      synseg(m,i)=model(i)*w(i);
      if ts+i>length(x);
         x(ts+i)=0;
      end
      model(i)=x(ts+i);
   end
    ts=ts+R;
else
   delta=similar(model,averD(m),x);
for i=1:N
   synseg(m,i)=x(averD(m)+delta+i-R)*w(i);
   if averD(m)+delta+i>length(x)
    x(averD(m)+delta+i)=0;
   end
      model(i)=x(averD(m)+delta+i);

end

ts=averD(m)+delta+R;
clear delta
    end
  end
for i=1:fix(length(x)*coeff)
   y1(i)=0;
end
for m=1:L1
```


78

```
    for i=1:N
        if (m-1)*R+i<=length(x)*coeff
            y1((m-1)*R+i)=y1((m-1)*R+i)+synseg(m,i);
        end

    end
end
```

```
y=y1;
subplot(2,1,1)
plot(x)
subplot(2,1,2)
plot(y)
```

similar.m 文 件

```
function shift=similar(template,Ta,signal)
N=256;
R=N/2;
w=hanning(N);
k=-1;
for j=Ta-50:Ta+49
    if i+j-R>0
    for i=1:N
        if i+j-R>length(signal)
            signal(i+j-R)=0;
        end
        cadiseg(i)=signal(j+i-R);
    end
    temp=corrcoef(template,cadiseg);
cor=temp(1,2);
if cor>=k
    k=cor;
    shift=j-Ta;
    end
else
```

```
    shift=0;
    end
end
```

（3） 基于 STFT 在频域进行基频尺度修改的源程序文件：

pscale.m 文件

```
function [x,y]=pscale(wav)
fid=fopen(wav,'rb');
data=fread(fid,'int16');
fclose(fid);
L=length(data);
for i=1:L-22 %eliminate the head file
   x(i)=data(i+22)/2^15;
end
N=360;
M=1024;
anaw=hanning(N);
R=N/6;
for i=1:length(x)
   y0(i)=0;
   f(i)=0.01;
end
ts(1)=0;
L0=fix((length(x))/R);
for m=1:L0
      a(m)=1.6;
   ta(m)=(m-1)*R;%ta is analysis time instants
   if m>1
      ts(m)=fix(ts(m-1)+R/a(m));
   end

for i=1:N
   if ta(m)+i>length(x)
      Sw(i)=0;
   else
```


80

```
            Sw(i)=x(ta(m)+i)*anaw(i);
        end
    end
    A=lpc(Sw,25);%LPC coefficient
    anasoursesig=filter(A,1,Sw);%short time analysis sourse signal
    anasoursespec=fft(anasoursesig,M);%short  time  analysis  sourse
spectrum
    anaspec=fft(Sw,M);%short time analysis spectrum
    specenvelop=abs(anaspec./anasoursespec);
    xi=1:1/a(m):M;
    temp=interp1(anasoursespec,xi,'liner');
    L1=length(temp);
    if a(m)>=1
       for i=1:M/2
          modisoursespec(i)=temp(i);
       end
    else
       for i=1:fix(L1/2)
          modisoursespec(i)=temp(i);
       end
       for i=fix(L1/2)+1:M/2
          modisoursespec(i)=temp(i-fix(L1/2));
       end
    end
    clear temp;
    %modisoursespec(M/2+1)=modisoursespec(M/2);
    %for i=M/2+2:M
       %modisoursespec(i)=conj(modisoursespec(M-i+2));
    %end
    modisoursespecmag=abs(modisoursespec);
    modisoursespecpha=angle(modisoursespec);
    if (m<=1)
       synsoursespecpha=modisoursespecpha;
    else
```

```
        for i=1:M/2

z(i)=modisoursespecpha(i)-premodisoursespecpha(i)-2*pi*(i-1)/M*(ts(m)-
ts(m-1));
            while(z(i)>=pi)
                z(i)=z(i)-2*pi;
            end
            while(z(i)<=-pi)
                z(i)=z(i)+2*pi;
            end
            langmita(i)=2*pi/M*(i-1)+z(i)/(ts(m)-ts(m-1));
            synsoursespecpha(i)=synsoursespecpha(i)+R*langmita(i);
        end
    end
    premodisoursespecpha=modisoursespecpha;
    synsoursespec=modisoursespecmag.*exp(j*synsoursespecpha);
     synw=hanning(round(N/a(m)));
     for i=1:M/2
        synspec(i)=synsoursespec(i)*specenvelop(i);
      end
      synspec(M/2+1)=conj(synspec(M/2));
      for i=M/2+2:M
        synspec(i)=conj(synspec(M-i+2));
      end
          templ=real(ifft(synspec,M));

      synsig=templ(1:round(N/a(m))).*synw';
    for i=1:round(N/a(m))
        if (m-1)*R+i<=length(x)
            y0((m-1)*R+i)=y0((m-1)*R+i)+synsig(i);
            f((m-1)*R+i)=f((m-1)*R+i)+synw(i)*synw(i);
        end
    end
clear synsig synw A anasoursesig anasoursespec modisoursespecpha;
end
```


82

```
for i=1:length(x)
   y(i)=y0(i)/f(i);
end

subplot(2,1,1)
plot(x)
subplot(2,1,2)
plot(y)
zoom xon
```

（4） 运用 TD-PSOLA 在时域进行基频尺度修改的源程序

psola.m 文 件

```
function [y,y1]=psola(wav,anapmark,pitchcoef)
fid=fopen(wav,'rb');
data=fread(fid,'int16');
fclose(fid);
L=length(data);
for i=1:L-22  %eliminate the head file
   y(i)=data(i+22)/2^15;
end
fid=fopen(anapmark,'rb');
ta=fread(fid,'int16');
fclose(fid);
ori_duration=length(y);
%'ts' is synthetic instant
%'tv' is virtual analysis instsnt
for t=1:ta(1)
   pa(t)=ta(2)-ta(1);
end

for s=1:length(ta)-1
       for t=ta(s):ta(s+1)-1
       pa(t)=ta(s+1)-ta(s);
    end
end
L=length(pa);
```

```
for t=L:L+500
    pa(t)=pa(L);
end
ts(1)=ta(1);
u=1;
while(ts(u)<=ori_duration)
  u=u+1;
    shift=1;
    ts(u)=ts(u-1)+shift;
    sum=pa(ts(u-1))/pitchcoef;
    while abs((ts(u)-ts(u-1))-sum/(ts(u)-ts(u-1)))>=1
          shift=shift+1;
        ts(u)=ts(u-1)+shift;
        sum=sum+pa(ts(u))/pitchcoef;
    end
end
y1=synthesis(y,ta,ts);
subplot(2,1,1)
plot(y)
subplot(2,1,2)
plot(y1)
```

synthesis.m 文 件

```
function y1=synthesis(y,ta,ts)
%'y' is original speech signal
%'ta' is analysis instant
% 'ts' is synthetic instant
ps=diff(ts);% 'ps' is synthetic signal period
w=hanning(2*ps(1))';
if ta(1)<=ps(1)
   for i=1:ta(1)+ps(1)
      Sm(i)=y(i)*w(i+ps(1)-ta(1));%the first  short-time analysis frame
   end%Define the first short-time synthetic frame is equal to the first
analysis short time frame
else
```


84

```
    for i=1:2*ps(l)
        Sm(i)=y(ta(l)-ps(l)+i)*w(i);
    end
end
y1=Sm;
L1=length(ts);
for m=2:L1%Sm is the m-th short-time synthetic fame
    n=mapping(ta,ts(m));
    L2=2*ps(m-1);
    clear w
    w=hanning(L2)';
    for i=1:L2
        t=ta(n)-ps(m-1)+i;
        if t<=length(y)&t>=1
            Sm(i)=y(t)*w(i);
        else
            Sm(i)=0;
        end
        end
        % synthesis overlap add
          L3=length(y1);
        for i=1:L2/2
            y1(L3-L2/2+i)=y1(L3-L2/2+i)+Sm(i);
        end
        for i=L2/2+1:L2
            y1(L3-L2/2+i)=Sm(i);
        end
    end
```

mapping.m 文 件

```
function n=mapping(anap,p)
  s=2;
  while((p-anap(s-1))/(anap(s)-anap(s-1))<0|...
        (p-anap(s-1))/(anap(s)-anap(s-1))>1)&s<length(anap)
     s=s+1;
```


85

```
end
if s==length(anap)
   n=s;
else
      au=(p-anap(s-1))/(anap(s)-anap(s-1));
      if (au<=1)&(au>1/2)
      n=s;
      else
       n=s-1;
      end
   end
```

（5）运用TD-PSOLA进行汉语普通话韵律合成的源程序文件

datainput.m 文件

```
function [y,ta,yuan]=datainput(wav,anapmark)
fid=fopen(wav,'rb');
data=fread(fid,'int16');
fclose(fid);
L=length(data);
for i=1:L-22  %eliminate the head file
   y(i)=data(i+22)/2^15;
end
fid=fopen(anapmark,'rb');
temp=fread(fid,'int16');
for i=1:length(temp)-1
   ta(i)=temp(i+1);
end
yuan=temp(1);%yuan is the first pmark of voice phone
fclose(fid);
```

pitchmodi.m 文件

```
function
[pa,pitchcoef]=pitchmodi(yuan,ta,highesttune,lowesttune,type)
%yuan is the first sample point of the voice section
tune1=lowesttune;
tune2=tune1+(highesttune-lowesttune)/4;
```


86

```
tune3=tune1+(highesttune-lowesttune)/2;
tune4=tune1+(highesttune-lowesttune)*3/4;
tune5=highesttune;
L=length(ta);
    T=ta(2)-ta(1);
    for t=1:ta(1)-1
       pa(t)=T;%pa(t) is the analysis pitch cotour
       pitchcoef(t)=1;
    end
    L1=length(ta)-1;
    for k=1:L1
        T=ta(k+1)-ta(k);
        if ta(k)>=yuan
           break
        else
           for t=ta(k):ta(k+1)-1
                   pa(t)=T;
              pitchcoef(t)=1;
              end
           end
          end
          clear L1 T
    if type==55
       L1= length(ta)-1;
      for s=k:L1
         T=ta(s+1)-ta(s);
         PC=tune5/(16000/T);
         for t=ta(s):ta(s+1)-1
            pa(t)=T;
            pitchcoef(t)=PC;
          end
         end
    end
    if type==35
       temp=tune3;
```


87

```
Ll=length(ta)-1;
for s=k:Ll
   temp=temp+(tune5-tune3)/(length(ta)-k);
   T=ta(s+1)-ta(s);
   PC=temp/(16000/T);
        for t=ta(s):ta(s+1)-1
         pa(t)=T;
         pitchcoef(t)=PC;
        end
   end
end
if type==214
   temp=tune2;
   L1=length(ta)-1;
   L2=k+fix(2/3*(length(ta)-k))-1;
    DELTAP=(tune1-tune2)/(2/3*(length(ta)-k));
   for s=k:L2
      temp=temp+DELTAP;
      T=ta(s+1)-ta(s);
      PC=temp/(16000/T);
    for t=ta(s):ta(s+1)-1
      pa(t)=T;
      pitchcoef(t)=PC;
    end
   end
DELTAP=(tune4-tune1)/(1/3*(length(ta)-k));
for s=L2+1:L1
   temp=temp+DELTAP;
   T=ta(s+1)-ta(s);
    PC=temp/(16000/T);
  for t=ta(s):ta(s+1)-1
    pa(t)=T;
    pitchcoef(t)=PC;
 end
end
```


88

```
end
if type==211
   temp=tune2;
   L1=length(ta)-1;
   L2=k+fix(3/4*(length(ta)-k))-1;
    DELTAP=(tune1-tune2)/(3/4*(length(ta)-k));
   for s=k:L2
   temp=temp+DELTAP;
    T=ta(s+1)-ta(s);
    PC=temp/(16000/T);
    for t=ta(s):ta(s+1)-1
          pa(t)=T;
          pitchcoef(t)=PC;
   end
  end
for s=L2+1:L1
    T=ta(s+1)-ta(s);
    PC=temp/(16000/T);
    for t=ta(s):ta(s+1)-1
       pa(t)=T;
       pitchcoef(t)=PC;
    end
   end
end
if type==51
     temp=tune5;
   L1=length(ta)-1;
   L2=k+fix(1/4*(length(ta)-k))-1;
   for s=k: L2
   T=ta(s+1)-ta(s);
   PC=temp/(16000/T);
   for t=ta(s):ta(s+1)-1
    pa(t)=T;
    pitchcoef(t)=PC;
end
```


87

```
end
DELTAP=(tune1-tune5)/(3/4*(length(ta)-k));
for s=L2+1:L1
   temp=temp+DELTAP;
   T=ta(s+1)-ta(s);
   PC=temp/(16000/T);
   for t=ta(s):ta(s+1)-1
        pa(t)=T;
        pitchcoef(t)=PC;
    end
  end
end
if type==514
   temp=tune5;
   L1=length(ta)-1;
   L2=k+fix(1/2*(length(ta)-k))-1;
    DELTAP=(tune1-tune5)/(1/2*(length(ta)-k));
   for s=k:L2
     temp=temp+DELTAP;
     T=ta(s+1)-ta(s);
     PC=temp/(16000/T);
  for t=ta(s):ta(s+1)-1
     pa(t)=T;
     pitchcoef(t)=PC;
  end
 end
DELTAP=(tune4-tune1)/(1/2*(length(ta)-k));
for s=L2+1:L1
   temp=temp+DELTAP;
  T=ta(s+1)-ta(s);
   PC=temp/(16000/T);
  for t=ta(s):ta(s+1)-1
    pa(t)=T;
    pitchcoef(t)=PC;
end
```


70

```
    end
end
if type==351
    temp=tune3;
    L1=length(ta)-1;
    L2=k+fix(1/2*(length(ta)-k))-1;
      DELTAP=(tune5-tune3)/(1/2*(length(ta)-k));
     for s=k:L2
        temp=temp+DELTAP;
        T=ta(s+1)-ta(s);
        PC=temp/(16000/T);
     for t=ta(s):ta(s+1)-1
         pa(t)=T;
         pitchcoef(t)=PC;
     end
    end
DELTAP=(tune1-tune5)/(1/2*(length(ta)-k));
for s=L2+1:L1
        temp=temp+DELTAP;
        T=ta(s+1)-ta(s);
         PC=temp/(16000/T);
        for t=ta(s):ta(s+1)-1
          pa(t)=T;
          pitchcoef(t)=PC;
    end
   end
end
if type==335
    temp=tune3;
L1=length(ta)-1;
L2=k+fix(1/3*(length(ta)-k))-1;
 for s=k:L2
    T=ta(s+1)-ta(s);
    PC=temp/(16000/T);
 for t=ta(s):ta(s+1)-1
```

```
        pa(t)=T;
        pitchcoef(t)=PC;
    end
  end
DELTAP=(tune5-tune3)/(2/3*(length(ta)-k));
for s=L2+1:L1
    temp=temp+DELTAP;
    T=ta(s+1)-ta(s);
     PC=temp/(16000/T);
    for t=ta(s):ta(s+1)-1
      pa(t)=T;
      pitchcoef(t)=PC;
    end
  end
end
if type==535
   temp=tune5;
   L1=length(ta)-1;
   L2=k+fix(1/2*(length(ta)-k))-1;
   DELTAP=(tune3-tune5)/(1/2*(length(ta)-k));
  for s=k:L2
     temp=temp+DELTAP;
     T=ta(s+1)-ta(s);
     PC=temp/(16000/T);
  for t=ta(s):ta(s+1)-1
      pa(t)=T;
      pitchcoef(t)=PC;
  end
end
DELTAP=(tune5-tune3)/(1/2*(length(ta)-k));
for s=L2+1:L1
    temp=temp+DELTAP;
    T=ta(s+1)-ta(s);
     PC=temp/(16000/T);
    for t=ta(s):ta(s+1)-1
```

```
        pa(t)=T;
        pitchcoef(t)=PC;
    end
   end
end
clear L L1 L2 T PC DELTAP temp
```

timemodi.m 文件

```
function  timecoef=timemodi(ori_duration,obj_duration)
timecoef=obj_duration/ori_duration;
```

synpmark.m 文件

```
function [ts,tv]=synpmark(ta,pitchcoef,pa,timecoef,obj_duration)
%'ts' is synthetic instant
%'tv' is virtual analysis instsnt
L=length(pa);
for t=L:L+500
   pa(t)=pa(L);
   pitchcoef(t)=pitchcoef(L);
end
ts(1)=ta(1);
tv(1)=round(ts(1)/timecoef);
u=1;
%while(ts(u)<=obj_duration)
  %u=u+1;
 %  shift=1;
    %ts(u)=ts(u-1)+shift;
    %tv(u-1)=round(ts(u-1)/timecoef);
  %tv(u)=round(ts(u)/timecoef);
    %sum=pa(tv(u-1))/pitchcoef(tv(u-1));
    %while abs((ts(u)-ts(u-1))*(tv(u)-tv(u-1))-sum)>=(tv(u)-tv(u-1))
 %   shift=shift+1;
%    ts(u)=ts(u-1)+shift;
 %temp=round(ts(u)/timecoef);
 %   sum=sum+pa(tv(u))/pitchcoef(tv(u))*(temp-tv(u));
    %tv(u)=temp;
%end
```


93

```
%end
while(ts(u)<=obj_duration)
  u=u+1;
    shift=0;
    ts(u)=ts(u-1);
    tv(u-1)=round(ts(u-1)/timecoef);
  tv(u)=round(ts(u)/timecoef);
  sum=0;
   while abs((ts(u)-ts(u-1))*(tv(u)-tv(u-1))-sum)>=tv(u)-tv(u-1)
       shift=shift+1;
    ts(u)=ts(u-1)+shift;
    temp=ts(u)/timecoef;
       sum=sum+pa(round(tv(u)))/pitchcoef(round(tv(u)))*(temp-tv(u));
       tv(u)=temp;
    end
end
ts=round(ts)
tv=round(tv);
```

synthesis 文件:

```
function y1=synthesis(y, ta, tv, ts, obj_duration)
%'y' is original speech signal
%'ta' is analysis instant
%'tv' is virtual analysis instant
% 'ts' is synthetic instant
ps=diff(ts);% 'ps' is synthetic signal period
w=hanning(2*ps(1))';
if ta(1)<=ps(1)
   for i=1:ta(1)+ps(1)
      Sm(i)=y(i)*w(i+ps(1)-ta(1));%the first  short-time analysis frame
   end%Define the first short-time synthetic frame is equal to the first
analysis short time frame
else
   for i=1:2*ps(1)
      Sm(i)=y(ta(1)-ps(1)+i)*w(i);
   end
```


94

```
end
yl=Sm;
Ll=length(ts);
over=0;%over flag
for m=2:Ll%Sm is the m-th short-time synthetic fame
    n=mapping(ta,tv(m));
   L2=2*ps(m-1);
   clear w
   w=hanning(L2)';
   for i=1:L2
      t=ta(n)-ps(m-1)+i;
      if t<=length(y)&t>=1
         Sm(i)=y(t)*w(i);
      else
         Sm(i)=0;
      end
   end
% synthesis overlap add
%L3=length(yl);
L3=ts(m-1);
   if over==1|L3>obj_duration
      over=1;
      break;
   else

   for i=1:L2/2
      if over==1|L3-L2/2+i>obj_duration
         over=1;
         break;
      else
         if L3-L2/2+i>0
            yl(L3-L2/2+i)=yl(L3-L2/2+i)+Sm(i);
         end
      end
end
```

```
  for i=L2/2+1:L2
     if over==1|L3-L2/2+i>obj_duration
        over=1;
        break;
     else
          y1(L3-L2/2+i)=Sm(i);
   end
end
end
```

mapping.m 文 件

```
function n=mapping(anap,p)
  s=2;
  while((p-anap(s-1))/(anap(s)-anap(s-1))<0|...
        (p-anap(s-1))/(anap(s)-anap(s-1))>1)&s<length(anap)
     s=s+1;
  end
  if s==length(anap)
     n=s;
  else
        au=(p-anap(s-1))/(anap(s)-anap(s-1));
        if (au<=1)&(au>1/2)
        n=s;
        else
        n=s-1;
       end
   end
```

psola 文 件

```
function
[y,y1]=psola(wav,anapmark,obj_duration,highesttune,lowesttune,type)
%[y,y1]=tdpsola(wav,anapmark,duration,intialtune,endtune,type)
%'wav' is the original speech signal which is saved at wav type,
%'pmark' is anlysis pmark file and is saved at binary type.'obj_duration'
%is the modified signal time duration.'highesttune' is the highest tune of
%the five level tunetype and 'lowesttune' is the lowest tune of the five
```

```
%level tune type
% for example
% [y y1]=psola('guo35.wav','guo35pmark.bin',3000,130,80,51);
[y,ta,yuan]=datainput(wav,anapmark);
[pa,pitchcoef]=pitchmodi(yuan,ta,highesttune,lowesttune,type);
ori_duration=length(pa);
%'ori_duration' is the time duration of the original signal
timecoef=timemodi(ori_duration,obj_duration);
[ts,tv]=synpmark(ta,pitchcoef,pa,timecoef,obj_duration);
y1=synthesis(y,ta,tv,ts,obj_duration);
```

致 谢

在本人的两年半研究生学习中，先后得到导师俞振利副教授、张礼和教授的悉心培养和热情指导，导师的严谨治学态度和兢兢业业的工作精神给我留下了深刻的印象。

信电系的许多老师也给予我多方的支持和帮助。

谨以此致以衷心的感谢！