Speech Synthesis Based on Sinusoidal Modeling

A THESIS

Presented to

The Academic Faculty

By

Michael W. Macon

In Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering

Georgia Institute of Technology

October 1996

Speech Synthesis Based on Sinusoidal Modeling

Approved:

Mark A. Clements, Chairman

Mark J. T. Smith

Thomas P. Barnwell

Date approved by Chairman _____

To my parents and mmss

Acknowledgments

Before all, I thank God for blessing me with the talents, opportunities, and influences that led me down this path.

I thank my advisor, Prof. Mark Clements, for guiding me and at the same time letting me roam free to explore over the last 5 years. His support, encouragement, and words of recommendation were invaluable in pointing me in the right direction. I also thank the other members of the reading committee, Prof. Mark J. T. Smith and Prof. Tom Barnwell, for their time and suggestions, as well as Stacy Schultz and Kay Gilstrap for organizing things every step of the way. I thank James Oliverio and E. Bryan George for taking the time to serve as outside members of my committee.

I would also like to thank Dr. George for opening up several key opportunities that shaped my educational experience. The opportunity to extend his earlier work would not have been possible without his many efforts to lobby on my behalf for internship opportunities and financial support from Sanders, Inc. and Texas Instruments. Summer work experiences at these companies and in the Motorola Paging Products Group were an invaluable part of my educational experience, and I thank many, including Alan McCree, Kathy Brown, Steve Bardenhagen, Zaffer Merchant, and especially David Morgan and Vishu Vishwanathan for making those summers worthwhile.

My work related to text-to-speech synthesis would not have been possible without the help of Andrew Breen, Mike Edgington, and the rest of the Speech Synthesis and Analysis Group at British Telecom. I especially thank them for one of the most

fruitful months of my graduate career during my visit in December 1995, and for the use of the LAUREATE II software in my work.

The work related to singing synthesis would not have been possible without the hard work of Leslie Jensen-Link, whose musical talent and software development skills were an indispensible contribution to the goal of making a computer program that could sing "Frère Jacques" as well as her husband Matthew. Thanks to Matthew for (mis)use of his voice. On a related note, thanks go to Fay Salvaras and RKM Studios and Glen Matulo of Orphan Studio for help in generating slick-sounding demos.

I thank several professional contacts, including Drs. Tom Quatieri and Bob McAulay of MIT Lincoln Laboratory for discussions on sinusoidal modeling (and loud accordions); and Dr. John Westerkamp of the University of Dayton for inspiring me to pursue graduate education in digital signal processing. Deepest appreciation goes to Prof. Ron Cole of the Oregon Graduate Institute for his efforts to open the door to the exciting future that lies before me.

On a personal level, I would like to thank many more people than I can list here (but I'll try): my parents, bro' Bob, and extended family, for their love, nurturing, and non-stop support throughout my whole life; my 'other brothers' the fabulous Hannes boys; many Atlanta friends, including Kelly Novak, Jeanne Buck, Gail & Andrew Owens, Mark Jones & Kitty Swain, and *SOUP*; and friends from DSP Groups past and present, including Alex^2 Potamianos, Haluk Aydinoglu, Jeff Schodorf, Jonathan Su, Mat Hans, Steve Kogon, *les frères* Arrowood, Dan Drake, Ram Rao, Greg Smith, Tarald Tronnes, Tom Gardos, and many others.

Finally, last on this page but first in my heart, Amy Zepp-soon-to-be-Macon, M.D., for helping make my dreams come true with her unwavering support, love, and affection.

Contents

List of Tables

List of Figures

Summary

In this research, the application of the *Analysis-by-Synthesis/Overlap-Add* sinusoidal model to synthesis of speech and singing voice is investigated, and a set of basic extensions and improvements of the capabilities of the model are developed. First, the application of the model to concatenation-based text-to-speech (TTS) synthesis is described. Methods for concatenating segments extracted from a corpus of recorded speech are presented, and challenges associated with removing perceptible mismatches in time/frequency structure around the segment boundaries are identified. Methods for smoothing the signal near these boundaries using the sinusoidal model are presented. The implementation of this model within a commercial TTS system serves as a research testbed. Results of a comparison between the new method and the commonly-used *Pitch-Synchronous Overlap Add* (PSOLA) method indicate that the method performs equally as well as the PSOLA method in the cases tested.

Next, through the extension of the text-to-speech synthesis method to the synthesis of singing, it is shown that the flexibility of the sinusoidal model approach enables the incorporation of various musically-interesting effects into the synthesized signal. These effects include vibrato, pitch variation and transition effects, and changes correlated with change in vocal effort. Also in this system, methods of corpus design and unit selection specifically designed for singing synthesis are developed. Despite the fact that a relatively small voice inventory is used, the system is capable of synthesizing a musically-pleasing singing voice that maintains the perceived identity of the vocalist recorded to create the unit inventory.

Finally, several improvements to the sinsusoidal model itself are detailed. The causes of artifacts present in the original ABS/OLA model are found by theoretical and empirical analysis, and methods for eliminating or diminishing these artifacts are presented. Among the innovations is a method for phase randomization based on subframe synthesis of the signal. It is shown through the results of a subjective comparison test that the method improves the quality of unvoiced speech synthesized using the model.

CHAPTER 1

INTRODUCTION

1.1 Text-to-Speech Synthesis

The problem of automatic conversion of textual information to synthetic speech has been a subject of research since the advent of the digital computer. Speech is the primary method of communication among humans, and it is natural to strive to enable humans to interact with computers using a speech-based interface. This technology has also been used to compensate for the loss of speech-related faculties by humans. The applications of text-to-speech (TTS) technology as a means of compensating for handicaps are many and varied, including reading machines for the blind, teaching aids for children with speech disabilities, and other aids for those with vocal impairments [1, 2, 3, 4].

Beyond these specialized applications, TTS also has begun to play a major role in "information access" technologies. The worldwide telephone network provides an infrastructure for access to various information sources, and speech provides a natural interface to such information. Presently in many such *interactive voice response* (IVR) systems, prerecorded messages read by a human being are used to provide information or prompt the user. This strategy becomes impractical, however, in applications where [5]

- the text is unpredictable or dynamic (e.g., up-to-the-minute weather reports or stock price quotes, email reading),

- access to a large database is required (e.g., catalog order information),
- cases where new prompts need to be frequently recorded, but must have a constant voice identity.

Text-to-speech synthesis can provide a significant advantage in these applications and many others.

The earliest attempts to synthesize speech (circa 1930) involved mechanical or electrical resonant filters that were excited to produce synthetic voice sounds or "copy" existing recorded speech [6]. Later, as the source/filter theory of speech production was further advanced by Fant [7] and others, more elaborate models and implementations of the vocal tract transfer function were developed. Along with many others, the work of Klatt [8] was instrumental in advancing the state of the art in such *formant synthesis* systems, in which voiced speech production is represented as the excitation of a set of series or parallel resonances (the vocal tract) by a shaped pulse train (glottal pulses).

More recently, the approach taken by many researchers in the TTS community has been to move away from the use of such speech production or acoustical models. Instead, systems based on the concatenation of subword-sized units of *recorded speech* have emerged. This approach backs away from traditional scientific attempts to *model* subtle aspects of the speech signal, and instead simply *encodes* all unknown information by storing samples of the actual waveform. This approach has resulted in a significant advancement in the quality and "naturalness" of speech produced by state-of-the-art TTS systems.

Until fairly recently, the memory and storage limitations of general-purpose computing hardware have made concatenation-based TTS systems rather impractical, since the quality of results generally improves as the size of the inventory of concatenated segments increases. Modern personal computers and workstations are now quite well-suited to this type of system, and concatenation-based systems have become the method of choice for many commercial efforts [5, 9].

A TTS system based on concatenation poses special research problems. Since there is no acoustic production model to control as in formant synthesis, *prosody modification* must be performed to change the durations and pitch period of the concatenated segments. Due to pitch differences, coarticulation, and other allophonic variations, the speech signals across the point of concatenation may be fairly different. The voice quality and other characteristics of the talker's speech may also vary across the joining boundary of concatenated segments. Various smoothing operations must be incorporated into the synthesis process to make these discontinuities less perceptible. These signal processing techniques lie at the heart of the synthesis algorithms, and are crucial to the synthesis of high-quality, natural-sounding speech.

The predominant technique for prosodic modification within the concatenative TTS context has been *Pitch-Synchronous Overlap Add* (PSOLA) resynthesis of the speech waveform, which involves extracting, copying, and repositioning windowed speech waveforms [10, 11]. Although fairly good results can be achieved with this method in many cases, it is not without shortcomings. These shortcomings can result in synthetic speech that possesses significant objectionable artifacts. The existence of such artifacts implies that a TTS system output is less likely to be intelligible or natural-sounding, thus making information reception more difficult for users.

Since concatenation-based synthesis relies on encoding and prosodic modification of actual speech waveforms, a wide array of speech signal modeling techniques can be applied to the problem. Many of these are found in the speech coding literature. Early concatenation work involved analysis, concatenation, and smoothing of linear prediction (LP) parameters, excited by a pitch pulse train or white noise [12]. As reflected in much of the speech coding literature, this representation is itself quite a coarse approximation, and it produces poor speech quality when certain assumptions fail to hold true [13]. Many variations on this source/filter representation have been developed for coding applications over the past 25 years, including analysis-by-synthesis methods such as multipulse LPC [14], code-excited linear prediction

(CELP) [15], and various others. Each of these takes a step towards a closer approximation of the original signal. However, not all are well suited to high-quality prosodic modification, because the focus in their development has been on finding an "efficient" (i.e., easy to code) model, rather than a "flexible" model that is useful for high-quality prosody modification.

1.2 Sinusoidal Modeling

The desire for an alternative speech model that is both an "efficient" and more general representation than pitch-excited LPC led to the development of a *sinusoidal model* of speech in the mid-1980's. Although similar ideas were published by others around the same time, the main pioneers of this work were R. J. McAulay and T. F. Quatieri. This model has been used in a wide range of applications, including speech modification [16, 17], co-channel speech separation [18], aids for the hearing-impaired [19, 20], speech enhancement [21], audio signal modeling [22], psychoacoustic models [23], and others. As is apparent from this list, the sinusoidal model is a useful and flexible signal model that lends itself well to many applications.

An extension to McAulay and Quatieri's work was proposed by George and Smith [24, 25, 26]. This algorithm, called the *Analysis-by-Synthesis/Overlap-Add* (ABS/OLA) model, was found to have some advantages over McAulay and Quatieri's model in terms of signal quality and synthesis computational complexity. The main application of this model has been as a method for prosodic modification of speech, so this makes it a natural candidate for application in a text-to-speech system.

Although sinusoidal models represent a significant advancement in the area of speech modification, they do fall short in some respects. Since sinusoidal models are best suited to representing *periodic* signals, the representation of unvoiced speech often suffers, especially under modification. This shortcoming can lead to undesirable artifacts in synthesis, again detracting from the quality of the overall system.

1.3 Research Overview

This thesis presents the application of a sinusoidal model analysis/synthesis algorithm to concatenation-based speech synthesis. As mentioned above, the ABS/OLA sinusoidal model is an attractive candidate for this type of application, since it is capable of achieving high-quality speech modification, while offering a relatively low computational complexity in synthesis. Algorithms for concatenating and smoothing subword speech units taken from an inventory of sinusoidally-modeled speech are presented, and the performance of the method is examined in detail. An extension of this method to the synthesis of *singing voice* is presented as well, to futher demonstrate the flexibility of the model.

As mentioned, sinusoidal models are generally capable of high-quality speech modification, but still suffer from certain artifacts, mainly in unvoiced speech. This thesis also presents algorithms for mitigating these deficiencies. Several enhancements of and extensions to the ABS/OLA sinusoidal model will be presented, including methods for improving the results in modification of unvoiced speech.

Through the presentation of this improved sinusoidal model and its application in a TTS system, it will be shown that this model can provide a framework for high-quality speech and voice synthesis, offering advantages over competing methods. The thesis is organized as follows:

Chapter 2 first presents background information on the topic of sinusoidal modeling. This section introduces time-scale and pitch modification methods based on sinusoidal models and presents other relevant issues for following sections. The second half of the chapter presents an overview of current approaches to the text-to-speech conversion process, with the purpose of putting the research presented into its proper perspective within the scope of the various challenges associated with TTS.

Chapter 3 begins with an in-depth theoretical analysis of the time-scale and pitch

modification algorithms in the ABS/OLA sinusoidal model, conducted with the purpose of explaining commonly-encountered artifacts. After this analysis, several extensions to the model are proposed and their implementations discussed and evaluated.

Chapter 4 details the application of the improved ABS/OLA model to a text-to-speech system. The commercially-available LAUREATE II TTS system from British Telecom is used as a basis for the implementation. Algorithms for analysis, concatenation, smoothing, and synthesis of the synthetic speech waveform are presented, as are the results of a comparison with a competing synthesis method.

Chapter 5 presents an extension of the sinusoidal model framework to the synthesis of singing, demonstrating the ability of the model to synthesize and control subtle aspects of the speech signal. It also offers the opportunity to explore the application of the model in an environment less reliant on a linguistically-based front-end analysis of text.

Finally, the thesis concludes with a summary of contributions and directions for future research.

CHAPTER 2

BACKGROUND

2.1 Sinusoidal Modeling

Sinusoidal models attempt to represent the input signal by a small number of sinusoidal components at any given instant in time, while still maintaining "perceptual equality" with the original input signal. A "sparse" representation of this sort is advantageous for speech coding, where it is desirable to transmit only a minimal number of parameters that describe the signal. For the purposes of speech modification, a sparse representation is also desirable, because it leads to more tractable and computationally efficient solutions to time-scale and pitch modification.

2.1.1 The McAulay/Quatieri Model

Analysis/Synthesis

McAulay and Quatieri introduced the initial form of their sinusoidal model in 1984 [27]. In the model, the speech signal $s(t)$ is modeled as the sum of a small number of sinusoids with time-varying amplitudes and frequencies,

$$s(t) = \sum_{l=1}^{L} A_l(t) \cos(\phi_l(t)) \tag{2.1}$$

where $A_l(t)$ represents the amplitude and $\phi_l(t)$ represents the phase of the lth sinusoidal component. The phase $\phi_l(t)$ is found by the integrating $\omega_l(t)$, the time-varying

instantaneous frequency of the lth sinusoid,

$$\phi_l(t) = \int_0^t \omega_l(\tau)d\tau + \phi_l(0). \tag{2.2}$$

It should be noted that phase offsets of the various components *relative to each other* are not constrained in any way.

To find the parameters of the model, $A_l(t)$ and $\phi_l(t)$, the DFT of windowed signal frames is calculated, and the peaks of the spectral magnitude are selected from each frame. Depending on the pitch period of the speaker, anywhere from 20 to 80 peaks are typically necessary to represent a given frame. The amplitudes and instantaneous frequencies of the peaks are then noted, and a "nearest-neighbor" peak matching algorithm is used to relate the frequencies of sinusoids in one frame to those in the next frame. During stationary segments of the utterance (e.g., sustained vowels), these frequencies match each other with minimal variation of amplitude or frequency. However, when characteristics of the utterance change abruptly, as during unvoiced sounds and transitions, the parameters vary considerably from frame to frame.

Based on the results of the peak-matching algorithm, parameter "tracks" are created by linearly interpolating the component amplitudes and frequencies to describe the evolution of one frame into another. "Births" and "deaths" of parameter tracks are also allowed to account for the possibility of a changing number of peaks from one frame to the next. Figures 2.1(a) and 2.1(b) give examples of the frequency tracks obtained by this procedure. Once these tracks are obtained, reconstruction can be accomplished by substituting the amplitude and frequency parameters into Equations (2.1) and (2.2) for each value of the time index t.

The model thus describes the speech waveform solely by the amplitude and frequency values of a relatively small number of sinusoids from each frame. An important point to note is that the parameter tracks used to reconstruct the waveform are described in a *functional* form. This offers a very simple and intuitive framework for *time-scale* and *frequency-scale modification* of the speech waveform, since the time

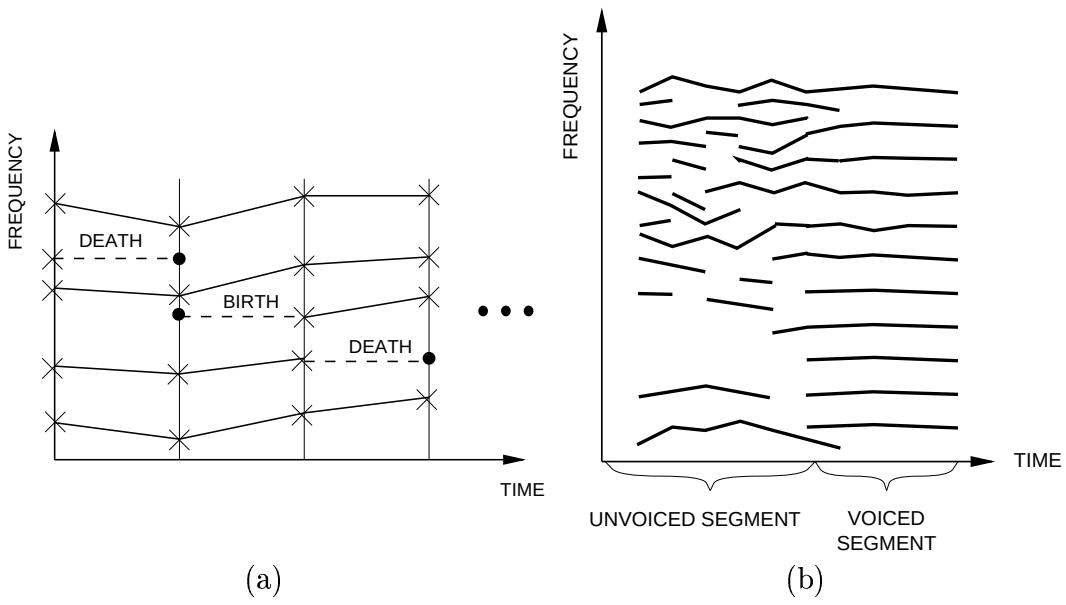

Figure 2.1: Examples of sinusoidal model frequency tracks: (a) Nearest-neighbor matching, (b) Resulting sinusoidal frequency tracks (after [27]).

index t and the frequency tracks $\{\omega_l(t)\}$ in these functions can be altered prior to resynthesis.

In [28], McAulay and Quatieri introduce explicit representation of the phase[1] of each component sinewave into the model, addressing phase coherence issues associated with the original "magnitude-only" model [27]. In this formulation, A_l^k, ω_l^k, and θ_l^k, the amplitude, frequency, and phase, respectively, of the lth sinewave component of the signal, are estimated from the DFT magnitude peaks in the kth frame. As in the first model, the amplitudes and frequencies are submitted to the matching algorithm and linearly interpolated to derive the lth amplitude/frequency track given by $A_l(t)$ and $\omega_l(t)$.

In contrast to the magnitude-only model, however, here the phase function $\phi_l(t)$ is not defined simply as the integral of the instantaneous frequency. Instead, a cubic *phase interpolation function* of the form

$$\phi_l(t) = \zeta + \gamma t + \alpha t^2 + \eta t^3 \tag{2.3}$$

is fit to the set of measured phases $\theta_l^k\}$ at the frame boundaries by constraining the slope of $\phi_l(t)$ (i.e., the frequency) and its values at the beginning and end of the frame to be the measured values and then imposing a smoothness constraint [28].

With the phase functions $\{\phi_l^k(t)\}_{l=1}^{L^k}$ determined for each of the L^k frequency tracks in the kth frame, the signal is resynthesized by the relation

$$\tilde{s}^k(t) = \sum_{l=1}^{L^k} A_l^k(t)\cos(\phi_l^k(t)). \tag{2.4}$$

for each frame. A block diagram of the complete analysis/synthesis system is depicted in Figure 2.2.

In a simple extension to their sinusoidal model, McAulay and Quatieri [16] proposed an approach to time-scale and pitch modification based on the following

[1]Note that there are two notions of *phase* being used here–the relative phase offsets of component sine waves as estimated from the DFT, and the phase $\phi_l(t)$, which is the argument of the $\cos(\cdot)$ function in Equation (2.1).

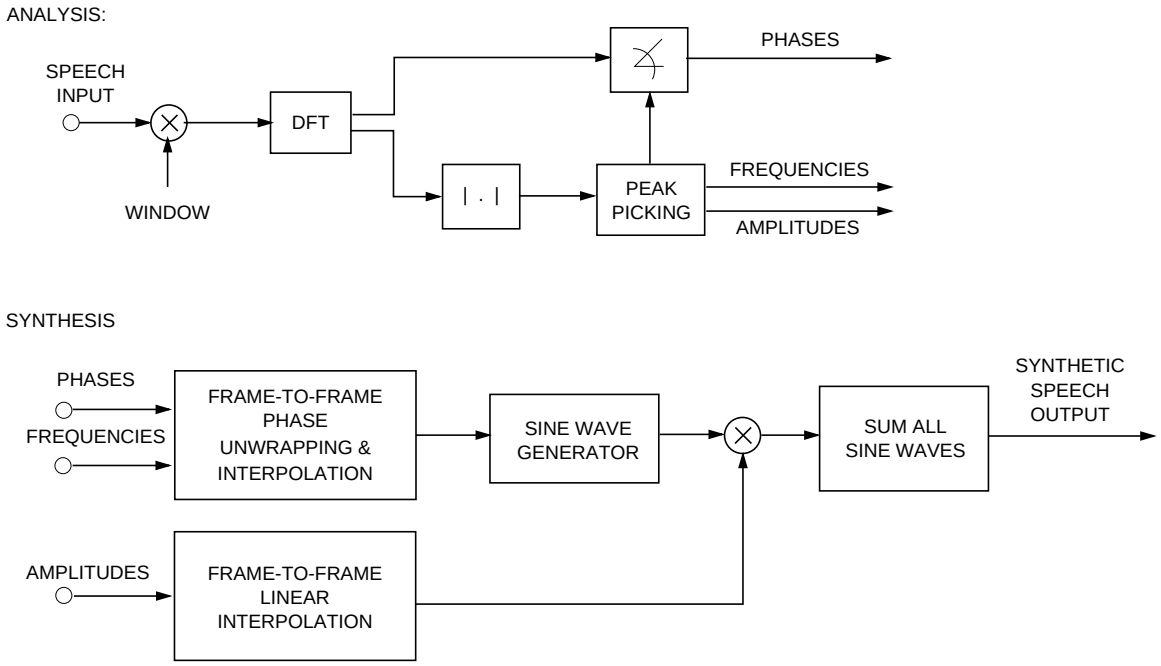

Figure 2.2: Block diagram of the sinusoidal analysis/synthesis system (from [17]).

idea: Since the parameter tracks $A_l(t)$ and $\phi_l(t)$ are described as functions of time, the time-scale evolution of these parameters can simply be altered to achieve time-scale modification. Frequency-scale modification can be achieved by shifting the sinusoidal frequencies prior to resynthesis.

Time-scale modification

In time-scale modification of speech, the goal is to change the speaker's apparent rate of articulation without changing the pitch of the speaker's voice. This can be accomplished by first decoupling the vocal tract and glottal excitation contributions to the speech signal, and then modifying each separately. Specifically, the time evolution of the vocal tract parameters must be scaled (reflecting the modified rate of articulation), while the excitation signal is modified in such a way that the pitch contour and voicing characteristics change at the modified time scale, but the pitch scale remains unaltered.

In [16], McAulay and Quatieri extend the original sinusoidal model to achieve these speech transformations. In this extension, the vocal tract system function magnitude is estimated from the signal, and the system phase is derived by imposing a minimum phase assumption. The residual amplitude and phase after the vocal tract effects have been removed are then assumed to be the amplitude and phase of the excitation components. With these contributions separated, independent modifications of the vocal tract and excitation time scales are facilitated.

Assuming the desired time variable t' is a scaled version of the original time variable t gives $t' = \rho t$, where $\rho > 1$ corresponds to a slower articulation and $\rho < 1$ corresponds to more rapid articulation. With this, the time-scale modified amplitude of the lth sinewave track is given by

$$A'_l(t') = a_l(t'/\rho)M_l(t'/\rho), \tag{2.5}$$

where $a_l(t)$ and $M_l(t)$ represent the original excitation and system amplitudes as a function of time, respectively. $\Omega'_l(t')$, the lth modified excitation phase, is obtained

by integrating the time-scaled instantaneous frequency track and adding this to the phase offset carried over from the previous frame. Specifically,

$$\Omega_l'(t') = \int_0^{t'} \omega_l(\tau'/\rho)d\tau + \phi_l'(0) \tag{2.6}$$

where the time index at the beginning of the current frame is assumed to be $t = 0$, and $\phi_l(0)$ is the phase offset at the beginning of the frame. The lth track system phase $\psi_l(t)$ is modified in a manner similar to the way the amplitude tracks are modified in (2.5), giving the expression for the composite model phase

$$\phi_l'(t') = \Omega_l'(t') + \psi_l'(t'). \tag{2.7}$$

Finally, the time-scaled waveform is synthesized as

$$\hat{s}(t) = \sum_{l=1}^{L} A_l'(t') \cos[\phi_l'(t')]. \tag{2.8}$$

Although this method has the desired effect on the signal's time scale and pitch characteristics, the authors reported a "reverberant artifact" present in the output [17] due to the breakdown in phase coherence of the excitation sinewaves. At no point in the modification procedure are steps taken to preserve the correct phase relationship between the modified components, and this results in a perceptible phase error propagation.

In a refinement of this time-scale modification algorithm, referred to as "shape-invariant" modification [17], this problem is circumvented, resulting in modified speech that has no reverberant artifact and retains the shape of the original waveform. The major improvements in the algorithm come as a result of incorporating the concept of a *pitch pulse onset time* into the excitation model. The pitch pulse onset time is defined to be the first occurrence of a pitch pulse in a frame [29]. An estimator function for this parameter can be derived starting from the following hypothesis: if the glottal excitation pulse train is modeled as a sum of sinusoids, then at the point in time when a pitch pulse occurs, all the sinusoids must constructively interfere.[2] This

[2]Thus the argument of the $\cos(\cdot)$ function in Equation (2.1) must equal 0 or π for each component.

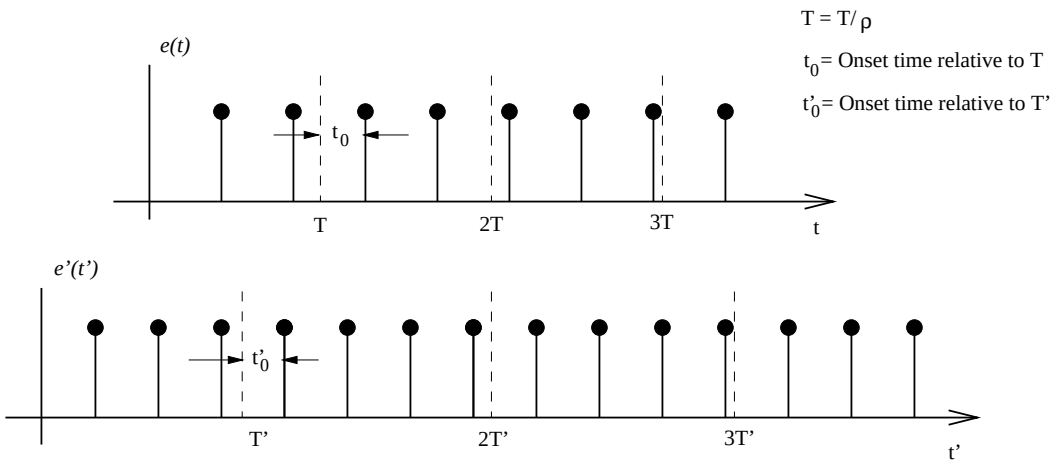

Figure 2.3: Modified pitch pulse onset time: $e'(t')$ represents the modified version of the excitation $e(t)$ (after [30]).

constrains $\{\Omega_l(t)\}$, the set of excitation component phases, to obey a linear function of frequency

$$\Omega_l^k(t) = (t - t_0^k)\omega_l(t), \tag{2.9}$$

where t_0^k is the pitch pulse onset time measured with respect to the kth frame boundary. The onset time in each frame is estimated by finding the maximum of an "onset-time likelihood function" derived in [29].

This linear phase model can be used to find the excitation component phases at the frame boundaries. The system phase is then found by subtracting the excitation phase from the measured composite phase. The amplitude $A_l(t)$ and vocal tract system phase $\psi_l(t)$ can then be time-scale modified as in Equations (2.5) and (2.6) above. However, the modified excitation phase $\Omega_l'(t')$ must be found by calculating a new onset time, t_0', for the modified frame. This new onset time is found by accumulating pitch periods of the previous frame and finding the time at which the first pitch period falls in the current frame, as shown in Figure 2.3. Note that this assumes that accurate pitch estimation has been performed for the analysis frame.

Once the modified system and excitation phases at a particular frame's boundaries have been found, they are summed to form a composite phase estimate. The

phase track for each model component is then found by use of the cubic interpolation procedure, and the speech is reconstructed via Equation (2.8).

Pitch and frequency-scale modification

Frequency-scale modification is the dual to time-scale modification: the goal is to alter the frequency domain characteristics without altering the time scale of the signal. Compression or expansion of the spectral envelope can be accomplished simply by scaling the instantaneous frequency of each component sinewave in the model to the desired value and resynthesizing the signal. A more challenging task, however, is to change the pitch of the signal without changing the spectral envelope or time-scale.

Within the context of the shape-invariant modification system [17], pitch-modification can be achieved by first scaling the estimated pitch period contour $P(t)$ by a desired factor β, (i.e., $P'(t) = P(t)/\beta$, where $\beta > 1$ corresponds to higher pitch and vice-versa). From this, the modified onset time t'_0 for each frame can be computed as described above, and the lth excitation phase can be computed with its frequency scaled as

$$\Omega'_l(t) = (t - t'_0)\beta\omega_l. \tag{2.10}$$

Now, since the spectral envelope due to the vocal tract system response must maintain the same shape as in the original signal, the system magnitude and phase responses for each track must be resampled at the scaled frequencies

$$\begin{aligned} M'_l(t) &= \hat{M}(\beta\omega_l, t) \\ \psi'_l(t) &= \hat{\psi}(\beta\omega_l, t) \end{aligned} \tag{2.11}$$

where $\hat{M}(\omega, t)$ and $\hat{\psi}(\omega, t)$ represent system magnitude and phase responses interpolated from the estimated sinewave parameter tracks at time t. From this point, the phases can be interpolated across frames by the cubic phase function, as in Equation (2.3), and the pitch-scaled signal can be synthesized.

This "shape-invariant modification" is made possible by the explicit control of phase available in the sinusoidal signal model. The assumptions made in formulating the model are general enough to provide natural-sounding modified speech and audio signals, yet flexible enough to facilitate relatively easy modification.

2.1.2 The Analysis-by-Synthesis/Overlap-Add Model

In [24, 25, 26], a sinusoidal model based on different analysis, synthesis, and modification algorithms is proposed by George and Smith. Termed the *Analysis-by-Synthesis/Overlap-Add* (ABS/OLA) sinusoidal model, this algorithm relies on an iterative analysis-by-synthesis parameter estimation algorithm in lieu of the peak-picking of the McAulay/Quatieri model. Resynthesis of the modified signal is accomplished by using a simple inverse FFT and overlap-add procedure.

Analysis and synthesis

In the ABS/OLA model, the input signal $x[n]$ is represented by a sum of overlapped short-time signal frames,

$$x[n] = \sigma[n] \sum_{k=0}^{K-1} w_s[n - kN_s] s_k[n - kN_s] \tag{2.12}$$

where K is the number of synthesis frames, N_s is the synthesis frame length, $w_s[n]$ is a symmetric window function that is nonzero over the interval $[-N_s, N_s]$, and $s_k[n]$ represents the kth frame "synthetic contribution" to the synthesized signal. Each synthetic contribution $s_k[n]$ is represented as the sum of a small number of *constant-frequency* sinusoidal components, given by

$$s_k[n] = \sum_{l=0}^{L-1} A_l^k \cos(\omega_l^k n + \phi_l^k) \tag{2.13}$$

where L is the number of sinusoidal components in the frame, and A_l^k, ω_l^k, and ϕ_l^k are the kth frame sinusoidal amplitudes, frequencies, and phases, respectively. The

slowly-varying gain term $\sigma[n]$ is used to improve the accuracy of the sinusoidal representation during transient signal segments.[3]

An iterative analysis-by-synthesis procedure is performed to find the "optimal" component amplitudes, frequencies, and phases for the signal frame. This analysis-by-synthesis procedure seeks to minimize the approximation error between the original and modeled signals by searching for the sinusoidal component that will minimize a mean-squared error at each iteration of the algorithm. It is shown in [24] that this analysis method results in an increase in model component accuracy over the peak-picking method; i.e., better segmental SNR values are obtained for nominal numbers of sinusoids, and the perceived speech quality is judged to be better than for the peak-picking analysis. However, analysis-by-synthesis also has the drawback of being much more computationally intensive than peak-picking.

The overlap-add synthesis algorithm in the ABS/OLA system is another feature that provides some advantages over competing sinusoidal models. Resynthesis of the signal is accomplished by a simple inverse FFT and overlap-add procedure, while in the McAulay/Quatieri model, a set of arbitrary-frequency sinusoidal oscillator outputs must be computed for each output sample. Computational complexity is concentrated in the analysis routine of the ABS/OLA system, while the synthesis routine is relatively simple. This trait is advantageous in applications where analysis can be performed off-line, but synthesis must be performed quickly, such as text-to-speech waveform synthesis.

Time- and frequency-scale modification

In the McAulay/Quatieri model, the specification of a set of time-dependent parameter tracks provides a conceptually simple method for speech modification. These parameter tracks are not available, however, within the framework of the overlap-add

[3] Another interpretation of this envelope is as a function that slightly modifies the basis functions of the time-frequency representation used to model each frame.

synthesis procedure described above. In [24, 25, 26], a method that provides high-quality speech and music modification within the overlap-add context is presented. The foundation of this modification algorithm rests on a "quasi-harmonic" ordering of model components, providing a framework for frequency-scale and time-scale modification that preserves phase relationships between components (hence waveform shape). A "phasor interpolation" algorithm is incorporated into the modification procedure to resample the excitation spectral envelope during pitch-shifting. This algorithm will be discussed in greater detail in Chapter 3.

2.1.3 Hybrid Sinusoidal/Noise Models

Although sinusoidal signal models have been shown to provide the capability for high quality modification of speech and music signals, they are not without their limits. Objectionable artifacts can be noticed in drastic pitch and time-scale modifications of signals that do not adhere to the implicit assumptions of a sinusoidal model. For example, although noise-like signals are modeled well by a sufficient number of sinusoids that vary in amplitude and frequency [28], when evolution of sinusoidal components is slowed during time-scale modification, their periodicity becomes perceptible and a "tonal" artifact is audible [17]. In performing pitch-lowering using the ABS/OLA algorithm, a "choppy" quality is imparted to noise-like speech segments, resulting from modulations introduced during the pitch modification process. These shortcomings stem from the fact that sinusoidal signal models treat noise-like energy in the same manner as periodic signal components, ignoring underlying differences in the waveforms and physiological production of voiced and unvoiced sounds.

Several "hybrid" models have been proposed to attempt to mitigate this problem by treating voiced speech and noise-like energy differently in the representation. These involve using both sinusoidal components and some form of noise-like energy to represent the signal. Perhaps the most widely known work in this area relates to the Multiband Excitation (MBE) vocoder developed by Griffin and Lim [31]. In

this model, a harmonic set of sinusoids is first fit to the signal spectrum in each analysis frame. Once the least-squares-optimal amplitudes of these harmonics have been found, a spectral-fit comparison is made between the harmonic model spectrum and the original signal spectrum across several frequency bands. If the signal-to-noise ratio in a band lies below a certain threshold, then the band is declared to be "unvoiced." This band-by-band voicing decision is used to determine the method of synthesis—in voiced bands, the sinusoidal amplitudes are used, while in unvoiced bands, the sinusoids are replaced by noise energy. The noise part is synthesized by taking the inverse FFT of a random spectrum set to zero in the voiced bands.

The noise synthesis method in the MBE model is roughly equivalent to exciting a time-varying filter with white noise. Others have more explicitly used this same filtering idea. For a music synthesis application, Serra and Smith [32, 33] propose a system where the sinusoidal part of the signal is estimated (with some hand-editing) and subtracted from the signal spectrum. The residual spectral envelope is then smoothed and multiplied by a random noise spectrum to generate noise with the desired spectral shape.

In a similar operation viewed from another perspective, other authors have proposed representing bandpass noise using sinusoids modulated by lowpass random processes, rather than by filtering a white noise signal [34, 35, 36, 37]. Some of these involve a joint fitting of the sinusoids and so called "narrowband basis functions" to the spectral magnitude of each frame, while others depend on voicing decisions or manual setting of the modulation parameters [38].

A similar method to the Serra and Smith technique (though automatic) involves subtracting sinusoidal components from the signal, and then estimating a complex aperiodic spectral component via iterative reestimation [39, 40]. This process produces an aperiodic component that preserves the time structure of the original stochastic component of the signal, as demonstrated by the authors in tests with synthetic signals. This contrasts with the two filtering methods described above, in

which the time domain characteristics of the noise are controlled only by the time support of the synthesis frame.

Although this method does not lend itself well to a compact, convenient model, it does highlight an important point. Many authors have reported a lack of "perceptual fusion" of the harmonic and stochastic parts of these such models – meaning that the two components can be discriminated by the listener, and they do not sound like they come from the same source. It has been found that to create a natural-sounding synthetic voice, it is important to maintain the time-structure of the noise, which is concentrated around the instants of glottal opening and closing [41] in voiced speech. Similar characteristics have been found in the analysis and synthesis of reed musical instrument sounds, where it is important to maintain the time coherence of noise pulses with the reed motion [42].

Others have attempted to model noise waveforms directly in the time domain. Richard *et al.* use the framework of "formant waveforms" and Poisson random processes to model unvoiced speech in a manner that maintains the time envelope shape [43, 44, 45].

Finally, in a revision of the "noise filtering" approach for time-scale modification, Laroche *et al.* propose subtracting quasi-harmonic sinusoidal components from the speech signal in the time-domain, then fitting an AR model to the residual spectrum. However, they also incorporate a smoothed time-domain energy distribution function that maintains general time-domain amplitude characteristics [46, 47].

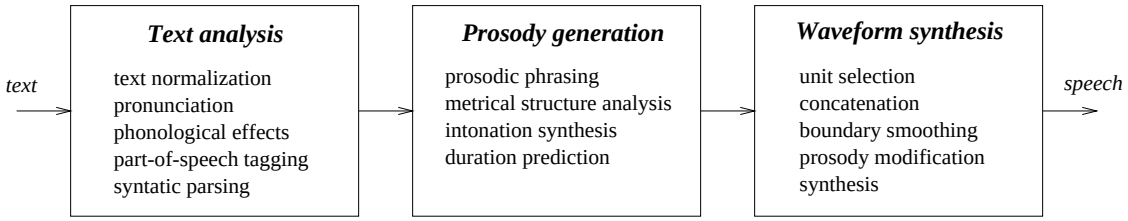

Figure 2.4: Block diagram of a concatenation-based TTS system.

2.2 Text-to-Speech Synthesis

Current approaches to the text-to-speech synthesis problem incorporate ideas from a diverse set of fields, including linguistic theory, perceptual psychology, speech production science, digital signal processing, and structured software design. The general text-to-speech problem can be broken into three parts (shown in Figure 2.4): (1) the automatic conversion of text into an abstract linguistic representation, (2) generation of prosody from this linguistic representation, and (3) synthesis of the speech waveform. The focus of the research proposed here is not on an entire TTS system, but rather on a subsystem within it. This section will present an overview of these three blocks, the first two of which serve as a "front-end" to the techniques developed in this thesis. This introduction will serve to put the research work described in the Chapter 4 into its proper perspective.

2.2.1 Text Analysis

Text segmentation and normalization The most useful TTS system is one that is able to accept input of text in any format and turn this text into speech in a manner consistent with human expectations. The formats and expectations vary considerably with the application. For instance, an automatic email reader should be able to extract information such as the name of the sender from the mail header, and be able to make sense of the typically "free-form" layout, capitalization, and

punctuation style of informal email messages. In any case, *text segmentation* must be performed on the input to delimit words, sentences, and paragraphs in the input text. *Text normalization* must also be performed to convert numbers, dates, symbols, and abbreviations into words that can be spoken.

These tasks are sometimes more difficult than one would expect. For example, the segmentation and normalization of the following text is quite easy for humans, who can incorporate many preconditioned expectations,

```
I gave Dr. Jones $8.00 on Park Dr. He lives on St. James St.
```

but it is much more challenging to develop a computer algorithm that accomplishes this same task, due to the ambiguities involved.

Morphological decomposition and pronunciation Once the text has been segmented and normalized, it is necessary to derive pronunciations for the input words. Grapheme-to-phoneme conversion algorithms [6] have been a topic of much research, but most current systems employ large online lexicons to obtain pronunciations, relying on grapheme-to-phoneme rules only as a backup. The size of an online lexicon can be drastically reduced by first doing a *morphological decomposition* of each input word into "headwords," prefixes, and affixes [48]. This eliminates the need to include all surface forms of a word such as "love" (lover, lovers, loved, loving, ...) in the lexicon. For example, the word "incoming" can be decomposed into the following set of *morphs*:

$$\texttt{incoming} \rightarrow \texttt{in + come + ing}$$

Rules for adding and/or dropping letters like the final "e" must also be included. Irregular forms such as

$$\texttt{saw} \rightarrow \texttt{see + (PAST)}$$

can be included as new headword entries for simplicity.

From this morph decomposition, a pronunciation can be found in the lexicon for known words. This pronunciation can be specified in terms of either a *phonemic* symbol set, which involves the minimal set of contrastive sound units in a language, or a larger set of *phonetic* symbols, which attempt to classify allophonic variations of sounds associated with each phoneme.

Tagging and parsing To generate other phonological information, it is necessary to assign part-of-speech information to each word (*tagging*), and build up the phrase-, clause-, and sentence-level structure of the input text (*parsing*).

Some part of speech information may be obtained directly from entries in the lexicon, but this does not, by any means, remove all ambiguity. Many words are used as multiple parts of speech. A probabilistic approach to the tagging problem, called *stochastic tagging* [49], is commonly used. In this approach, the goal is to maximize the probability that a given word sequence $\mathcal{W}$ has the tag sequence $\mathcal{T}$, $P(\mathcal{T}|\mathcal{W})$. The conditional probabilities of certain tags, given surrounding tags and words, can be estimated from large corpora of hand-marked text. Given these probabilities, the tag sequence that maximizes $P(\mathcal{T}|\mathcal{W})$ can be found using Viterbi decoding.

Once tagging has been performed, other useful information may be extracted from the lexicon entry for each word. For example, lexical (word-level) stress information is needed to disambiguate between the noun and verb forms of words such as "increase."

Parsing of the sentence can be performed by rule-based grammar methods, statistical methods, or hybrids of the two approaches. Again, several types of ambiguities crop up, including the common problem of prepositional phrase attachment, as in the sentence, *"The man saw the boy with a telescope."* This sentence could imply that either the boy or the man had a telescope. A syntactical parse tree of this sentence is shown in Figure 2.5. Although it is desirable to produce a very rich, complete, syntactical parse, this is not always necessary. A partial parse that locates

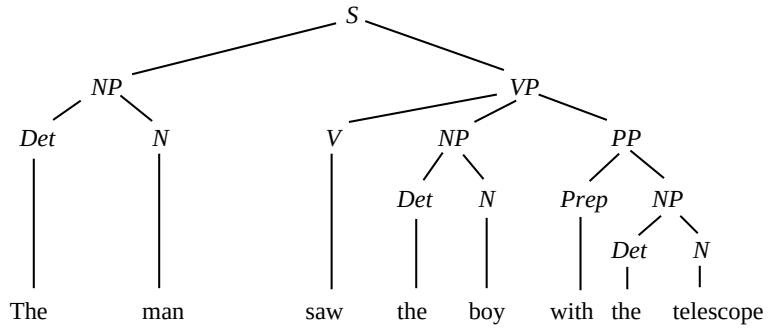

Figure 2.5: Syntactical parse tree for the sentence "The man saw the boy with the telescope." (after [50]). Note that this is not the only possible syntactic parse of this sentence.

phrases and clauses may be adequate to derive the necessary prosodic information in later stages of the TTS conversion.

Continuous speech effects Further modifications of pronuncation occur as a result of syntactic context and speech rate effects. *Function words* often take on *strong* and *weak forms*, depending on context. For example, "the" is pronounced differently when it precedes a word beginning with a consonant than when it precedes a vowel.

Other effects occur when speaking rate is increased, and it is important to model these effects to synthesize speech that does not sound "over-articulated" or otherwise unnatural. Among these processes are the deletion of weak syllables in words and elimination of consonants at word boundaries due to increased speech rate. In British English, another common effect is the insertion of an */r/* sound between a word ending in a vowel and an adjacent word beginning with a vowel.

2.2.2 Prosody Generation

As mentioned earlier, the term *prosody* refers to the "musical" qualities of speech—aspects such as rhythm and intonation that are separate from the sequence of phonemes in an utterance. The perceived "lack of naturalness" observed in even state-of-the-art

speech synthesizers today is primarily due to the failure of current systems to generate human-like rhythmic and intonational information from unrestricted text [51].

It should be noted that prosody generation is considered to be a very difficult research problem. Natural prosody comes from a complex combination of linguistic, pragmatic, and environmental factors, including semantic information (word meaning), dialogue context, emotional state of the speaker, and speaking style. Much of this information is impossible even for humans to infer from a small passage of printed text, unless many assumptions are made. To sidestep this problem, most TTS systems attempt to generate *discourse-neutral* prosody, which makes weak assumptions about semantics. This makes it possible to generate reasonable results from the sentence syntax and structure only.

Prosodic phrasing In natural speech, *prosodic phrases* serve to divide sentences into smaller units and aid in syntactic disambiguation. Breaks between these phrases are marked by phrase-final duration lengthening and pause insertion, as well as intonational features. Rule- and stochastic-based methods for finding the locations and "salience" (strength) of these phrase boundaries exist [52]. These methods use the syntactical structure and any semantic information that may be available to derive a hierarchical structure for the prosodic phrases of the input text passage. This structure is a key element of further steps in the process.

Rhythmic/metrical structure The common underpinning for many modern approaches to prosody generation is the field of *metrical phonology* [53, 54, 55]. This theory aims to quantify *stress* as a set of relative *prominence* comparisons between paired constituents in a sentence. These comparisons are represented in a binary *metrical tree* structure as shown in Figure 2.6. In the bottom tree nodes, weakly-stressed syllables are marked with a "*w*," and stronger syllables are marked with an "*s*." In the level above this, the syllable with stonger relative prominence, called the *metrical head* of the constituents below, is again marked with "*s*."

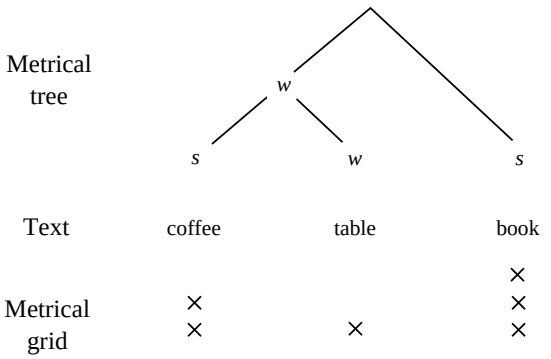

Figure 2.6: An example of a metrical tree and metrical grid for the phrase *"coffee table book,"* (after [56]).

This tree is created from lexical stress information at the bottom-most level, and then by a set of rules above this level. So called "stress shifts" can occur, which change the prominence of certain syllables based on context. For example, the stress of the word "thirteen" changes from a "*w s*" pattern to "*s w*" in the phrase "thirteen men." These effects can be modeled by first transforming the metrical tree to a *metrical grid*, as shown at the bottom of Figure 2.6. From here, rules are applied to the grid to make the stress pattern satisfy rules of *eurhythmy*, the theory that human perception finds certain rhythmic patterns more pleasing than others.

Intonation synthesis An essential part of the expressiveness of human speech is the "melodic" component of an utterance. The fundamental frequency (F_0) contour of a sentence often disambiguates among several possible meanings of a sentence and conveys more subtle information such as stress, emphasis, or the talker's emotional state. Thus, the generation of appropriate fundamental frequency contours is a vital step in the synthesis of natural-sounding speech.

Most intonation algorithms separate the generation of the F_0 contour into 2

distinct phases, one coming from *intonational phonology*, the other from *phonetics*.[4] The phonological component serves as an abstract means of describing intonational events, while the phonetic component converts this sequence of events into a graph of F_0 versus time.

Pierrehumbert [57] has developed a widely-used phonological framework for the description and transcription of English intonation. This framework consists of sets of *pitch accents* associated with prominent syllables, and *boundary tones*, which describe events at the boundaries of prosodic phrases. Each of these tones and accents can be characterized as being either "high" or "low" or a combination of these two. An abstract representation such as this is advantageous, because it separates this phonological component of intonation from more specific segmental details involved in producing the actual F_0 contour.

Based on this intonation theory, an algorithm for converting the above mentioned phonological description into a function of F_0 versus time is used [58]. Target values associated with pitch accents are superimposed on a downward-sloping function. This function models *declination*, the general tendency for a talker's pitch range to narrow and drift downward over the course of a phrase. Interpolation between these targets is then performed to create the pitch contour.

Various other methods for determining the F_0 contour also exist. Among these are models that represent the F_0 contour as the response of a lowpass filter to an input of step functions and impulses corresponding to linguistic events in the utterance [6, 59], and methods employing dynamical control system models that can be trained from a marked speech corpus [60].

Segmental duration prediction The *duration* of segmental units also contributes to the perception of rhythm in speech, an important cue for naturalness. There is con-

[4] *Phonology* is the branch of linguistics that deals with abstract entities such as the "accents" and "tones" described here. In contrast, *phonetics* deals with the manifestation of these abstract entities as audible cues in speech.

siderable debate over just which segmental unit is best suited to measuring duration. The *phone*, or phonetic realization of a particular phoneme, is convenient for various reasons, but the *syllable* or other units [61] are easier to relate to other phonological theories. Again, both rule-based and statistical-training methods exist [62], both of which depend on factors such as phonetic identity and context, speaking rate and style, stress, and prominence.

2.2.3 Waveform Synthesis

The third block of Figure 2.4 is the focus of the research described in this thesis. At the "back-end" of a text-to-speech system is a module that converts the aforementioned linguistic representation to an actual speech waveform. There are currently three basic categories of methods for accomplishing this: *articulation-based synthesis, formant synthesis,* and *concatenation* of recorded speech segments.[5] An overview of these three waveform synthesis methods is given in this section.

Articulation-based methods One reasonable approach to synthesizing a speech waveform is to model the physics of the speech production mechanism. If the simultaneous motions of the diaphragm, glottal folds, vocal tract cavities, and lips could be accurately simulated, then realistic speech would be produced by such a model. Although many researchers have developed speech synthesis algorithms based on simplifications of such a model (see for example [63, 64, 65, 66]), the task of producing natural-sounding speech from articulatory rules is extremely complex. Scientific knowledge of the intricate behavior and properties of the vocal tract remains rudimentary. Many researchers are currently developing more accurate models of speech production [67, 68] that are increasing understanding of the articulatory process.

[5] The rightmost block of Figure 2.4 is specific to the concatenation case.

Formant synthesis One principle that has formed the basis of much digital speech processing work over the past 25 years has been the "source/filter" model of speech production proposed by Fant [7]. In this model, the speech waveform is modeled as the result of passing an excitation source through a slowly-varying resonant tube-like structure (the vocal tract). The "transfer function" of the resonant vocal tract structure is changed by motion of the articulators, and can be described quite well by a set of three to five resonances, or *formants*.

In *formant synthesis*, pioneered by Klatt [8], Holmes [69], and others, synthesis is achieved by applying a set of heuristic rules for controlling the frequencies and amplitudes of these formants and the characteristics of the excitation source. Although isolated phonemic units can be characterized almost solely by their formant frequencies and motions, the formant locations in continuous natural speech are heavily influenced by context. Because of this fact, the rules necessary to control a formant synthesizer are rather complex.

Although careful refinements of formant synthesis have resulted in quite intelligible synthetic speech, the output speech lacks many of the subtle qualities that listeners perceive as "naturalness." With careful manipulation of model parameter trajectories, it is possible to "copy" recordings of speech quite well using a formant synthesizer, but specifying general rules for these trajectories that result in very natural-sounding speech is a difficult problem.

Concatenation The shortcomings of the previously mentioned synthesis methods arise because they each rely on the integrity of a simplified mathematical model of speech production. Many perceptually significant, but ill-understood, properties of the speech waveform are not present in the result. One way to sidestep these shortcomings is to concatenate short pieces of digitally recorded speech to form larger utterances. The synthesis algorithms developed in this research utilize this concatenation strategy. The next section describes in greater detail some issues associated

with the concatenation method.

2.2.4 Concatenation-Based Synthesis

Except in some very limited applications, it is not appropriate to piece together individual words to form new sentences in a text-to-speech system. The set of necessary words would be enormous, and a system of this sort would have no way to pronounce names or new words that were not already present in its archives. Instead, subword-sized units must be concatenated to form words and longer utterances. *Phonemes*, the set of basic units that comprise the sounds of a language, are an attractive set to consider. There are approximately 44 phonemes in English; these could in theory be concatenated to produce any word. However, the manifestation of a given phoneme is not invariant with context; *coarticulation* effects between adjacent phonemes change their acoustic representation significantly. As an alternative to the phoneme as a basic synthesis unit, researchers have proposed using other subword units such as the *diphone*, the acoustic segment beginning at the center of one phoneme and ending at the center of another [70], or the demisyllable, a unit derived from the syllable [71]. Given an inventory of such subword synthesis units, synthesis is performed by choosing the proper recorded subword units and concatenating them to generate speech.

Inventory structure/unit selection

Although fixed sets of diphones and demisyllables can produce reasonably intelligible synthetic speech, better results can be obtained when more general methods of unit selection are employed. Many of these methods use a large corpus of annotated single-speaker speech as the inventory, instead of a hand-edited library of single units. This approach allows for more choice in unit selection, and also eliminates the tedious process of diphone segmentation by hand.

The goal in such systems is to find subword speech units that are taken from a context most closely matched to the context of the unit in the synthesized utterance.

This process will, in theory, choose a unit that most closely represents the coarticulatory effects of the neighboring speech segments. One example of a method for organizing and classifying such a database is "context-oriented clustering" (COC) [72]. In this method, each set of phones in an annotated database is subdivided into clusters by a recursive binary splitting operation. (For example, the phoneme */a/* is separated into units preceded by */b/* and units not preceded by */b/*.) At each step of the procedure, an intracluster spectral variance measure is used to decide which cluster to split. This process is continued until all clusters have a sufficiently small variance or contain a minimum number of units. An extension of this procedure, called "multilayer COC," has also been proposed [73, 74]. In this algorithm, stress and syntactic boundary information in the database is also used in the splitting operation.

Other approaches incorporate not only phonemic and phonological annotation, but also acoustic characteristics of individual units. These methods are also able to handle gracefully situations where an exact phonemic context match is not available. In [75, 76, 77], tree splitting operations are performed to select units based on cepstral distance, rate of change of cepstra, and other measures. In [78, 79, 80, 81], this set of features is expanded to include prosodic characteristics such as F_0, duration, and energy as well. In this work, the problem is set up within the context of dynamic programming, with each unit selected having a "unit distortion" and a "continuity distortion." The selection process then consists of selecting a path through a lattice of available units such that a combination of transition and node cost functions is minimized.

Prosodic modification

Once the synthesis units have been chosen, independent control of energy, fundamental frequency and time-scale evolution is required to create a synthetic utterance with the correct prosodic information. A review of speech modification methods that have been applied to TTS follows.

Time-domain methods Among the simplest prosodic modification methods is *time-domain pitch-synchronous overlap-add* synthesis (TD-PSOLA), originally developed by researchers at the *Centre National d'Etudes des Télécommunications* (CNET) in France [11, 10]. This method involves a windowing of the speech signal $s[n]$ using time-domain windows centered on successive pitch pulses. This produces a sequence of short-time signals

$$s_k[n] = h[n]s[n - kT_o],$$

where $h[n]$ is an analysis window of length μTo. The factor μ is typically 2, meaning the window spans two pitch pulses in the waveform. Since the analysis window is positioned on successive pitch pulses, the overlap between successive windows is proportional to μ. Because of the pitch-synchronous nature of the analysis, accurate *pitch pulse marks* must be found for each pitch period. This usually involves an automatic glottal epoch detection procedure with some hand-correction. TD-PSOLA synthesis is accomplished by positioning the short-time signals $s_k[n]$ along the time axis with some overlap, and then summing.

Time-scale modification using time-domain PSOLA is achieved by deleting or replicating short time signals prior to the overlap-add procedure. This preserves the formant structure of the waveform, while changing the time-scale evolution of the utterance, and works fairly well for stationary voiced speech signals. However, when the time-scale of unvoiced speech is expanded by this method, periodicities are introduced into the waveform by the replication of individual short-time segments, and these periodicities are manifested as a "tonal" artifact.

Pitch modification is accomplished by simply repositioning the short-time segments relative to each other prior to overlap-add, as shown in Figure 2.7. In ideal conditions, this preserves the formant structure, while altering the fundamental frequency of the signal. The window length factor μ is very critical in this process. If μ is large, reverberant artifacts occur, since the pitch periods within the short-time signal cannot be realigned (equivalently, spectral lines appear in the Fourier trans-

form of $s_k[n]$). As μ is made small, the window shape has a greater influence on the individual pitch period waveform shapes, leading to broadening of the formants. Also, the position of the waveform within the analysis window becomes critical. A detailed analysis of these effects is given in [11].

The biggest advantage of time-domain PSOLA, of course, is its simplicity. Synthesis can be implemented with a complexity of approximately seven operations per sample [82], making it well-suited to real-time applications [83, 84, 85].

This description of PSOLA is reminiscent of the time-domain analysis of pitch modification in the ABS/OLA sinusoidal model given in Section 3.1.3. Comparisons can be drawn between PSOLA and phasor interpolation in the ABS/OLA model, and it is clear that some of the artifacts produced by these methods will be similar to each other. However, the fact that no explicit *model* for the signal exists in the PSOLA case implies that it is much more difficult to develop methods for mitigating these artifacts.

Time-domain source/filter models The source/filter model used in standard LPC vocoders has been used as a prosodic modification framework for many years. Early systems used vocal tract parameters taken from natural speech along with a synthetic pulse excitation [86, 12]. This simple representation makes prosodic modifications almost trivial, since the excitation sequence is synthetically generated, but the speech quality is similar to that of a formant synthesizer.

Extensions of this technique instead use modification of the LP residual to achieve greater naturalness in the modified speech. For instance, the time-domain PSOLA technique described above can be applied to the residual itself (LP-PSOLA) [11, 87]. A model that provides for slightly more efficient storage is the use of a "multi-pulse" representation of the residual [88, 89].

These techniques have the advantage of decoupling the formant structure and pitch information, making the algorithm somewhat less sensitive to window size and

alignment than PSOLA. A long window in LP parameter estimation can be used to reduce the window effects on the formants, and a shorter window can be used to avoid reverberation in the pitch modification.

Frequency domain methods Another alternative to PSOLA-based pitch shifting of the LP residual is frequency domain modification. In [90, 87], a technique for using the FFT to modify pitch is presented (called FD-PSOLA). An FFT of the windowed speech segment is computed, from which the formant structure is removed via spectral flattening. Rescaling of the frequency axis is then used to change the position of the residual signal harmonics, and an inverse FFT is used to transform the signal to the time domain after reintroducing the spectral envelope shape. This is equivalent to sampling rate conversion of the residual signal. Drawbacks of this approach include reduction in the signal bandwidth during pitch lowering and migration of noise across frequency bands.

A few authors have recently proposed direct frequency-domain synthesis of the speech waveform using sinusoidal models [91, 92, 93], including the "Multiband Excitation" (MBE) model proposed by Griffin and Lim [31] and the sinusoidal model developed by McAulay and Quatieri. The method presented in Chapter 4 falls into this category.

Unit concatenation issues

An added complication in the TTS application is the fact that slightly dissimilar speech units must be *concatenated* as well as modified.

As shown in Figure 2.8 (within the context of PSOLA), this can result in several types of mismatches at the concatenation point. Linear phase mismatches in the signal cause misalignments of the pitch pulses in voiced speech (panel (a)), which can be perceived as a "garbled" speech quality by the listener. Gross differences in the phase distributions of the signals can result in a variation of waveform shape

across the boundary, but this is less perceptible. When the fundamental frequency of one inventory segment is much higher than its neighbor at the concatenation point, differences in waveform shape are often apparent, as shown in panel (b). These can be attributed to the formant estimate accuracy inherent to the method (TD-PSOLA is poor in this regard, as can be seen in Figure 2.8). Finally, spectral tilt and formant frequencies and bandwidths can differ across the boundary, resulting in a perceived discontinuity of vowel quality (panel (c)). These mismatch artifacts are universal aspects of the concatenation problem, but their manifestation may depend to some degree on the method of prosodic modification being used.

In [94, 95], an algorithm that reduces many of the artifacts of concatenation is proposed as a preprocessing step before application of TD-PSOLA. This algorithm involves an off-line resynthesis of the speech unit inventory using the MBE model. Upon resynthesis, the speech is forced to have a constant pitch value. This implies that (*i*) pitch marking is trivial, and (*ii*) interpolation of spectral shape across the unit join boundaries can be accomplished by simple time-domain interpolation of the units. This algorithm does achieve its desired goal of making concatenation simpler, but produces speech with the well-known "buzzy" artifacts associated with the MBE vocoder.

In Chapter 4, the ABS/OLA sinusoidal model is applied to address the research problems associated with concatenation.

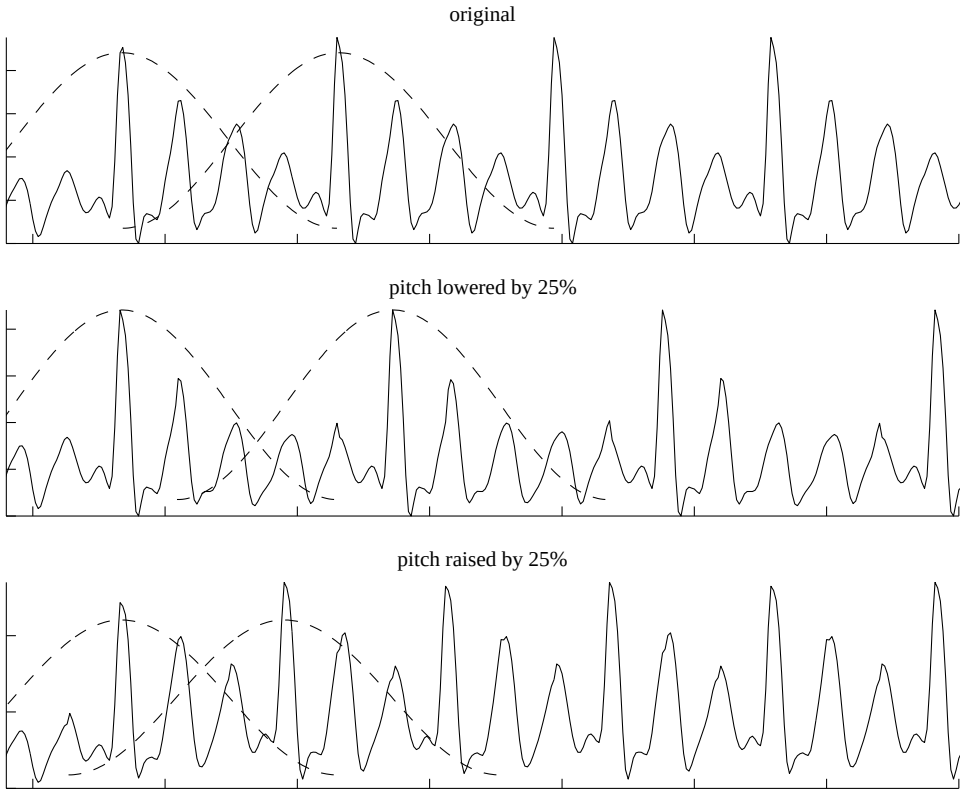

Figure 2.7: Pitch modification via PSOLA: *top:* original voiced speech; *middle:* pitch lowered by 25%; *bottom:* pitch raised by 25%

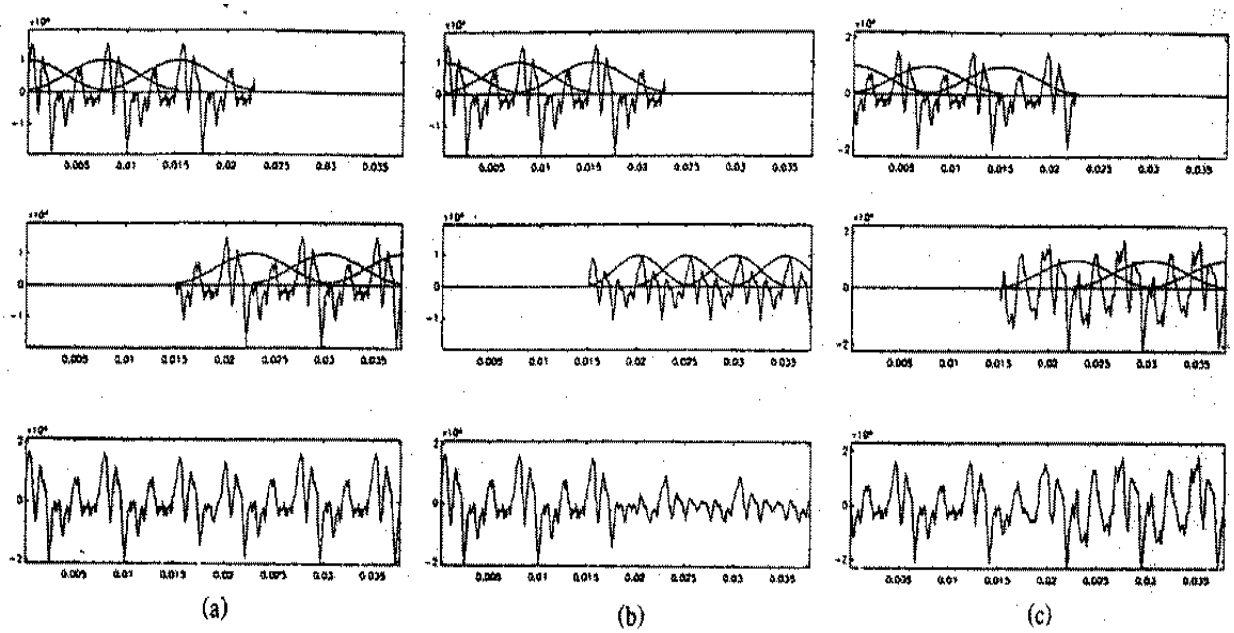

Figure 2.8: Concatenation artifacts: (a) phase mismatch, (b) F_0 mismatch, (c) spectral envelope mismatch

CHAPTER 3

AN IMPROVED SINUSOIDAL MODEL

3.1 Overlap-Add Sinusoidal Speech Modification

As described in the previous chapter, the Analysis-by-Synthesis/Overlap-Add (ABS/OLA) sinusoidal model is capable of high-quality prosodic modification and resynthesis of speech and music. However, the results obtained with this model are not without artifacts that merit investigation.

In this section, the pitch modification and time-scale modification algorithms in the original ABS/OLA work are described and analyzed in detail. This analysis serves to point out the causes of some synthesis artifacts, and it motivates several extensions of the model described later in the section. These improvements focus on (*i*) removal of undesirable modulations caused by pitch modification, (*ii*) mitigation of "tonal noise" artifacts in unvoiced speech modification, and (*iii*) improvement of pitch pulse onset time estimation in the model.

3.1.1 Frequency-Scale and Time-Scale Modification

The overlap-add synthesis of the kth frame of a speech signal $x[n]$ using the ABS/OLA model can be expressed as follows:

$$x[n + kN_s] = \sigma[n + kN_s] \left(w_s[n] s^k[n] + w_s[n - N_s] s^{k+1}[n - N_s] \right) \qquad (3.1)$$

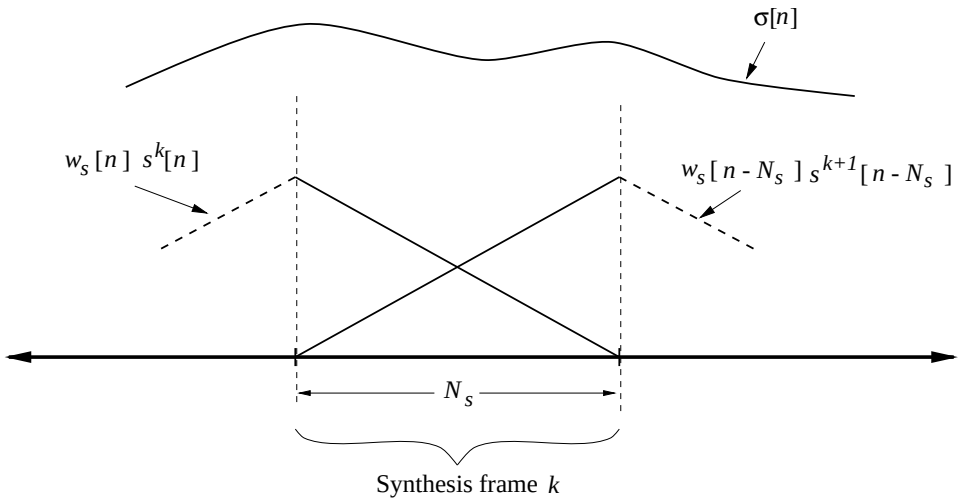

Figure 3.1: Overlap-add synthesis of a single frame using the ABS/OLA model.

where $\sigma[n]$ is the time-varying gain contour mentioned in Section 2.1.2, $w_s[n]$ is the synthesis window, and $s^k[n]$ and $s^{k+1}[n]$ are the "synthetic contributions" generated from analysis parameters of analysis frames k and $k+1$, respectively. This equation is depicted schematically in Figure 3.1. Each synthetic contribution $s^k[n]$ can be written as a sum of quasi-harmonic, constant-frequency sinusoids

$$s^k[n] = \sum_{j=0}^{J[k]} A_j^k \cos\left((j\omega_0^k + \Delta_j^k)n + \phi_j^k\right), \tag{3.2}$$

where ω_0^k is the fundamental frequency estimate for the frame, A_j^k, ϕ_j^k are the jth component amplitudes and phases, respectively, and Δ_j^k is the jth component *differential frequency*.

A straightforward approach for time-scale modification with such a model is simply to scale the length of each synthesis frame by a factor ρ, such that the new length of synthesis frame k is $\rho_k N_s$ ($\rho > 1$ implies *slower* speech). Furthermore, the time-evolution of the gain envelope $\sigma[n]$ must also be scaled to keep it synchronous with the sinusoidal frames. Frequency-scale modification can be performed by simply scaling each component's frequency in Equation (3.2) by a factor β ($\beta > 1$ corre-

sponds to a *higher* frequency). Alternatively, *pitch modification* can be performed by removing spectral shape characteristics due to vocal tract formants, frequency-scaling this "residual" model, and then reintroducing this resonant structure, as described in Sections 2.1.1 and 2.1.2. It is also important to note that modification of the frequency- and time-scale of the sinusoidal components changes the locations of pitch pulses in each frame. Because the overlap-add procedure relies on the coherent overlap of these pulses between adjacent frames, a time shift δ^k must be imparted to realign the frames after modification. Derivation of an expression for this shift is given in Section 4.5, as is a graphical illustration of the problem.

Unfortunately, the modification scheme described above leads to poor results for all but very small modification factors. This is mainly due to the fact that the sinusoidal components are *not*, in general, harmonically-related to each other. Because the frequencies are not multiples of a fundamental, phase offsets between components do not repeat periodically, and the "pulse-like" time-domain structure of a typical voiced speech signal is not maintained, as shown in Figure 3.2(b). This time-varying phase evolution is perceived by the listener as a "reverberant" or "rough" quality.

In [24], a method for overcoming this problem is developed. The solution involves viewing each quasi-harmonic component as the product of a harmonic term and a slowly-varying phase modulation term as follows:

$$\begin{aligned} s^k[n] &= \Re e\left\{\sum_{l=0}^{J[k]} A_l^k e^{j((l\omega_0^k+\Delta_l^k)n+\phi_l^k)}\right\} \\ &= \Re e\left\{\sum_{l=0}^{J[k]} \left(e^{j\Delta_l^k n}\right)\left(A_l^k e^{j(l\omega_0^k n+\phi_l^k)}\right)\right\}. \end{aligned} \tag{3.3}$$

By expanding or contracting the time-scale of this modulating term, the phase offsets at the ends of the modified frame can be made to agree with those at the ends of the original unmodified frame, and the problem of phase coherence breakdown is eliminated. Figure 3.2 shows this effect. Thus the synthesis Equations (3.1) and (3.2)

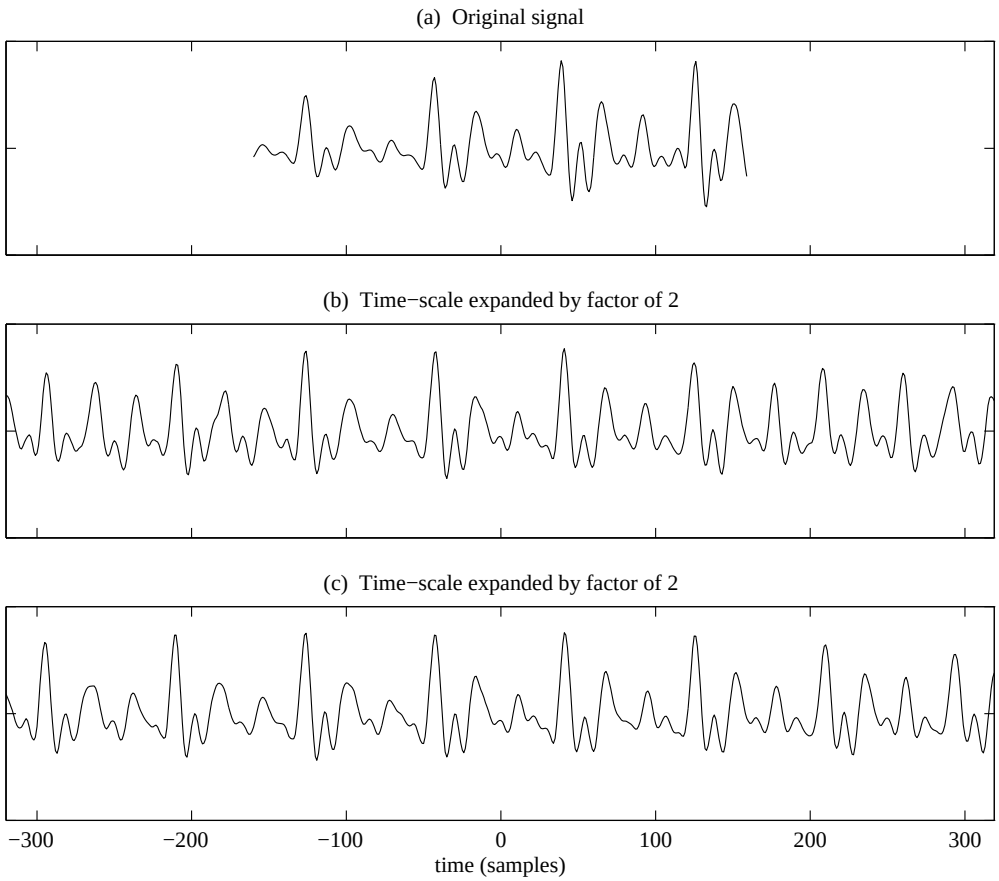

Figure 3.2: Phase coherence breakdown due to differential frequency terms in quasiharmonic model: (a)Original speech; (b) Result after time-scale modification without scaling of differential frequency terms; (c) Result after time-scaling of differential frequency terms.

for a modified signal $\hat{x}[n]$ become

$$\hat{x}[n+N_k] = \sigma\left[\frac{n}{\rho_k} + kN_s\right]\left\{w\left[\frac{n}{\rho_k}\right]s^k[n] + w\left[\frac{n}{\rho_k} - N_s\right]s^{k+1}[n-\rho_k N_s]\right\} \qquad (3.4)$$

where

$$\begin{aligned} s^k[n] &= \sum_{j=0}^{J[k]} A_j^k \cos\left(j\beta_k\omega_0^k(n+\delta^k) + \frac{\Delta_j^k n}{\rho_k} + \phi_j^k\right) \\ s^{k+1}[n] &= \sum_{j=0}^{J[k]} A_j^{k+1} \cos\left(j\beta_{k+1}\omega_0^{k+1}(n+\delta^{k+1}) + \frac{\Delta_j^{k+1} n}{\rho_k} + \phi_j^{k+1}\right) \end{aligned}$$

for $0 \leq n < \rho_k N_s$, where N_k is the beginning of the current synthesis frame ($N_k = N_s \sum_{i=0}^{k-1} \rho_i$).

An expression for the frequency of a single component can be written as

$$\hat{\omega} = j\beta\omega_0 + \Delta_j/\rho. \qquad (3.5)$$

Modifying the differential frequency terms in this manner has the effect of making the quasi-harmonic set of sinusoidal components *more harmonic* in structure when the time scale of the frame is expanded, becoming a harmonic series in the limit. This results in very good quality time-scale modification of voiced speech for a wide range of modification scales.

However, this "harmonization" strategy becomes a detriment to the modified speech quality when applied to time-scale expansion of unvoiced speech. Reducing the magnitude of the differential frequency terms has the effect of making the resynthesized noise more "tonal" in nature, since the components become more nearly harmonically related.

As mentioned by McAulay and Quatieri in [28] and other papers, the key to the ability of the sinusoidal model to represent unvoiced speech lies in the pseudo-random variation of component amplitudes, phases, and frequencies from frame to frame. This important variation is reduced in this case for two reasons: (*i*) The time scale is being *expanded*, meaning that the component frequencies do not change as rapidly as in the unmodified model; and (*ii*) fundamental frequency contours in the

model are usually smoothed over several adjacent frames to improve modified voiced speech quality, causing the nearly harmonic tones to tend to persist even longer. This tonal quality of unvoiced speech after time-scale expansion is a classic problem that has been mentioned by several authors [34, 38, 46, 16, 17, 47]. The presence of this artifact is addressed by a proposed extension to the ABS/OLA model described in Section 3.3.

3.1.2 Excitation Modification

As mentioned above, *pitch* modification, as opposed to *frequency* modification, can be performed by maintaining the speech formant structure while changing the sinusoidal component frequencies. This can be accomplished by simply dividing out the amplitude and subtracting the phases of a spectral envelope system function estimate from the sinusoidal parameters for the frame. What remains is a sinusoidal model for a signal representing the glottal excitation input to the vocal tract.

Simply frequency-scaling the sinusoidal excitation components for the frame, although an intuitive solution, can cause problems. These problems arise because the bandwidth of the modeled signal is changed by frequency scaling—high frequency energy is lost in pitch lowering, for example. Also, noisy regions in the original spectrum can possibly be moved to lie under formant peaks, causing other objectionable artifacts. The solution proposed by George and Smith [24, 25] is instead to *interpolate* and *resample* the excitation spectrum to achieve pitch modification in a process termed "phasor interpolation."

Given the excitation amplitudes b_l and phases θ_l for the frame, interpolation from the sinusoidal line spectrum to a smooth envelope $E(\omega)$ is computed by

$$E(\omega) = \sum_{l=0}^{J} b_l e^{j\theta_l} I(\omega - l\omega_0). \tag{3.6}$$

Since this is a complex-valued interpolation, 2π phase discontinuities between adjacent sinusoids can cause undesirable effects. For this reason, it is necessary to have

an *unwrapped* phase response. A reasonable approach to this problem is simply to remove the linear phase component due to the *pitch pulse onset time* shift described in Section 2.1.1, which causes the phases to become roughly centered around 0 for voiced speech signals, as shown in Figure 3.3. The interpolation function $I(\omega)$ proposed in [24, 25] is

$$I(\omega) = \begin{cases} \cos^2(\pi\omega/2\omega_0), & |\omega| \leq \omega_0 \\ 0, & \text{otherwise} \end{cases}, \tag{3.7}$$

which has the desirable property that $I(0) = 1$ and $I(l\omega_0) = 0$ for $l \neq 0$. After interpolation, the function $E(\omega)$ is then resampled at the new pitch harmonic locations to generate the pitch-shifted set of sinusoidal components. The set of differential frequencies Δ_l is interpolated and resampled by a similar method.

The phasor interpolation method works well in many cases, but still produces certain artifacts in the resynthesized speech. In particular, a very "pulsy" structure is imparted to the excitation when the pitch is lowered. Also, when pitch lowering is applied to unvoiced speech, an annoying "choppy" quality arises, due to time-domain modulations of the noise amplitude. This artifact also becomes audible in pitch lowering of breathy or partially devoiced speech.[1] When pitch raising is applied to unvoiced speech, tonal artifacts similar to those mentioned previously become very apparent.

3.1.3 Time-Domain Interpretation

The next several paragraphs provide an analysis of the phasor interpolation scheme described above, with the goal of finding the cause of the aforementioned artifacts. This analysis motivates a set of improvements to the model, described later in this section.

[1] *Devoicing* is the manifestation of a typically voiced phoneme (e.g., a vowel) as a predominantly unvoiced sound, often occuring in unstressed syllables or function words such as "the" or "a"—a phonological process referred to as *reduction*.

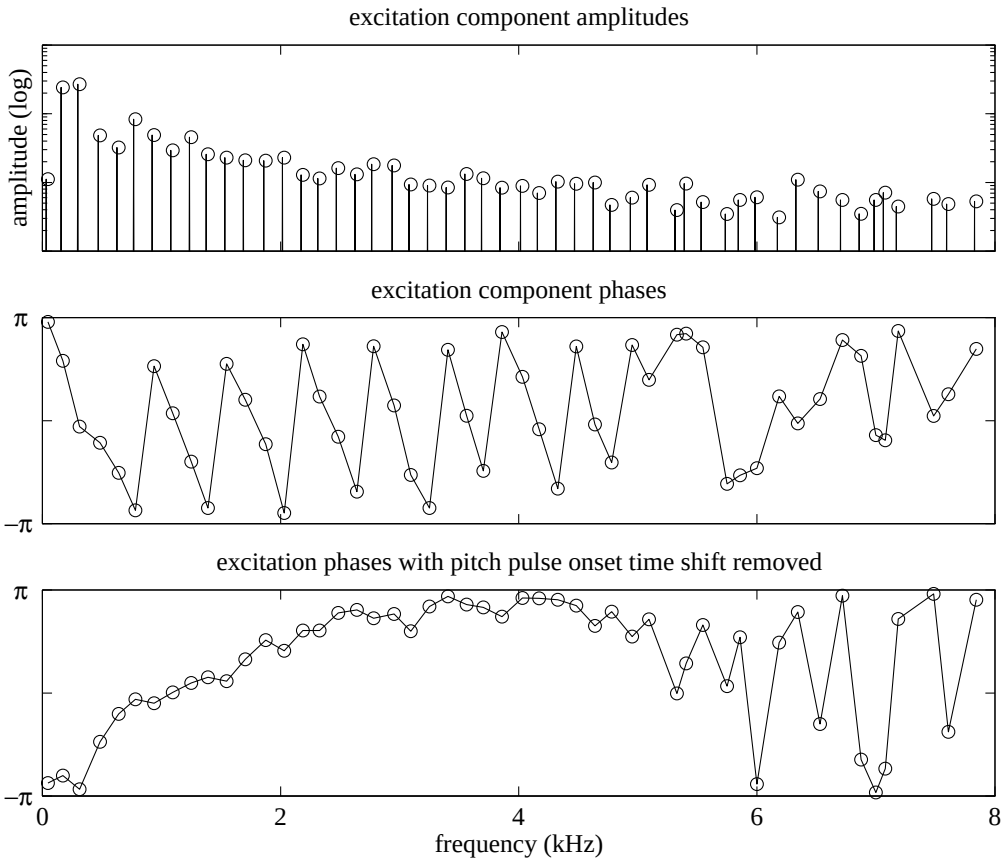

Figure 3.3: Unwrapping phases via removal of linear phase shift: *top:* Sinusoidal component amplitudes; *middle:* component phases; *bottom:* component phases after removal of pitch pulse onset time linear phase shift. (Phases above 5 kHz are considered perceptually insignificant.)

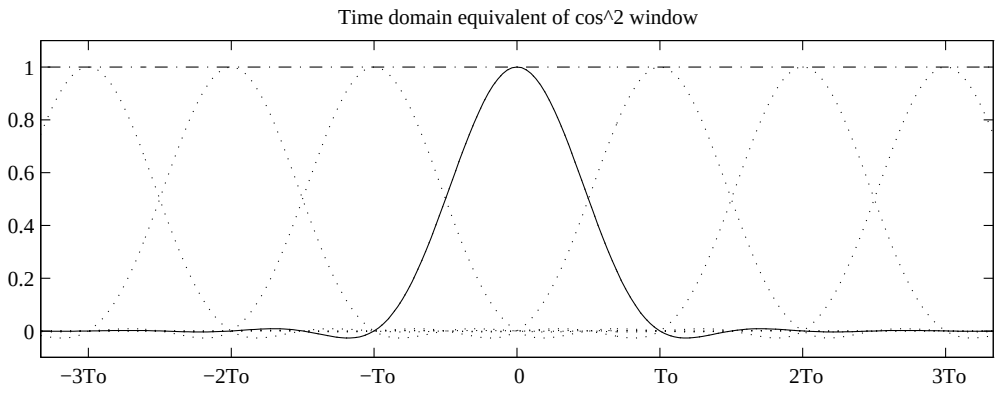

Figure 3.4: Time-domain equivalent of phasor interpolation window. Note that shifts of window sum to unity.

Equation (3.6) above can be rewritten as a convolution

$$E(\omega) = S(\omega) * I(\omega), \tag{3.8}$$

where

$$S(\omega) = \sum_{l=0}^{L} b_l e^{j\theta_l} \delta(\omega - l\omega_0),$$

and we have assumed L harmonically-related sinusoids for simplicity. This convolution can be written in the time domain as the product

$$e[n] = s[n]i[n], \tag{3.9}$$

where $s[n]$ is a (slightly dispersed) pitch pulse train and $i[n]$ is the inverse Fourier transform of the interpolation function $I(\omega)$. For the particular function in Equation (3.7), the following expression for $i[n]$ can be found

$$i[n] = \frac{1}{2\pi}\left[\frac{\sin(\omega_0(n - T_0/2))}{2(n - T_0/2)} + \frac{\sin(\omega_0(n + T_0/2))}{2(n + T_0/2)} + \frac{\sin(\omega_0 n)}{n}\right], \tag{3.10}$$

where $T_0 = 2\pi/\omega_0$. As seen in Figure 3.4, this window function has zero crossings at multiples of T_0, and sums to unity when superimposed at shifts of kT_0. Since the removal of the pitch pulse onset phase term tends to move excitation pulses to the frame center, Equation (3.9) results in the extraction of a single prototype pulse

from $s[n]$ via windowing by $i[n]$. This pulse is completely described by the smoothed spectrum $E(\omega)$.

Resampling $E(\omega)$ at multiples of a new fundamental frequency $2\pi/\hat{T}_0$ can be represented by

$$\hat{S}(\omega) = \sum_{l=-\infty}^{\infty} E(\omega)\delta\left(\omega - l\frac{2\pi}{\hat{T}_0}\right), \tag{3.11}$$

which can be written in the time-domain as

$$\hat{s}[n] = e[n] * p[n], \tag{3.12}$$

where

$$p[n] = \hat{T}_0 \sum_{k=-\infty}^{\infty} \delta[n - \hat{T}_0 k].$$

Thus, this resampling operation results in periodic replication of the prototype pulse at the new fundamental period $\hat{T}_0$. This process is depicted in Figure 3.5 for the case of a voiced speech input. The complex interpolation and resampling operation is followed by reintroduction of the differential frequency terms from Equation (3.3). In keeping with the argument in Section 3.1.1, these differential frequencies can be interpreted as slowly varying phase modulation terms that serve to modify the waveform shape slightly across the duration of the synthesis frame. In Figure 3.5, it can be noted that the waveform shape changes slightly across the frame duration due to these differential frequency terms.

As seen in Figure 3.5, the phasor interpolation is capable of performing a reasonable modification of the pitch of the speech signal. However, as mentioned above, an amplitude modulation occurs during pitch lowering ($\beta < 1$), imparting a choppy sound to unvoiced speech and a pulsy or buzzy sound to voiced speech. The cause of these artifacts can be seen in the time-domain interpretation of phasor interpolation described above. As shown in Figure 3.4, the time-domain versions of the interpolation window $i[n]$ sum to unity when the spectrum is resampled at the original fundamental frequency $\omega_0 = 2\pi/T_0$. However, when a modified fundamental frequency $\hat{\omega}_0 = \beta\omega_0$ is used, the superposition of these windows does *not* sum to 1. The

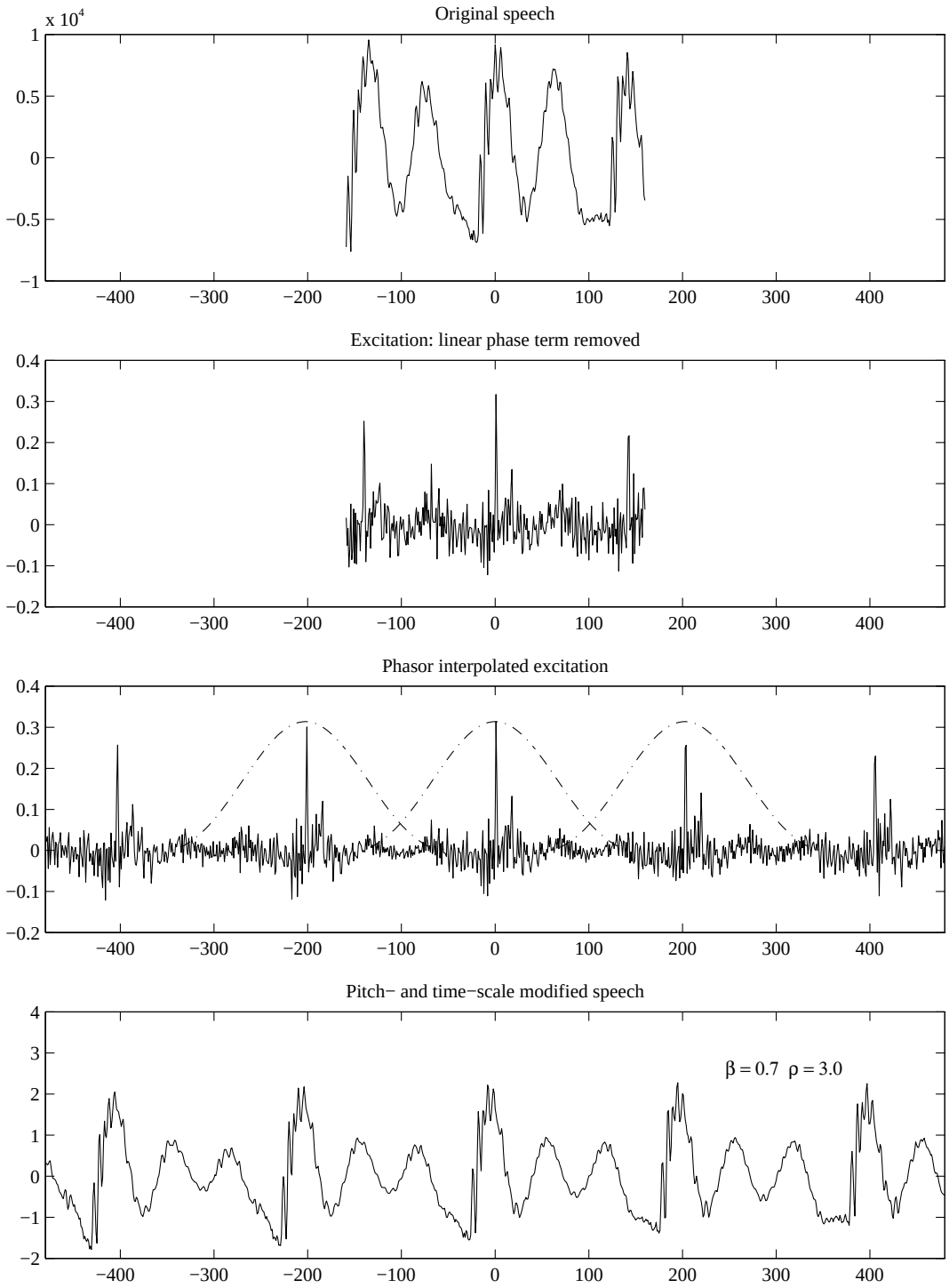

Figure 3.5: Phasor interpolation applied to voiced speech.

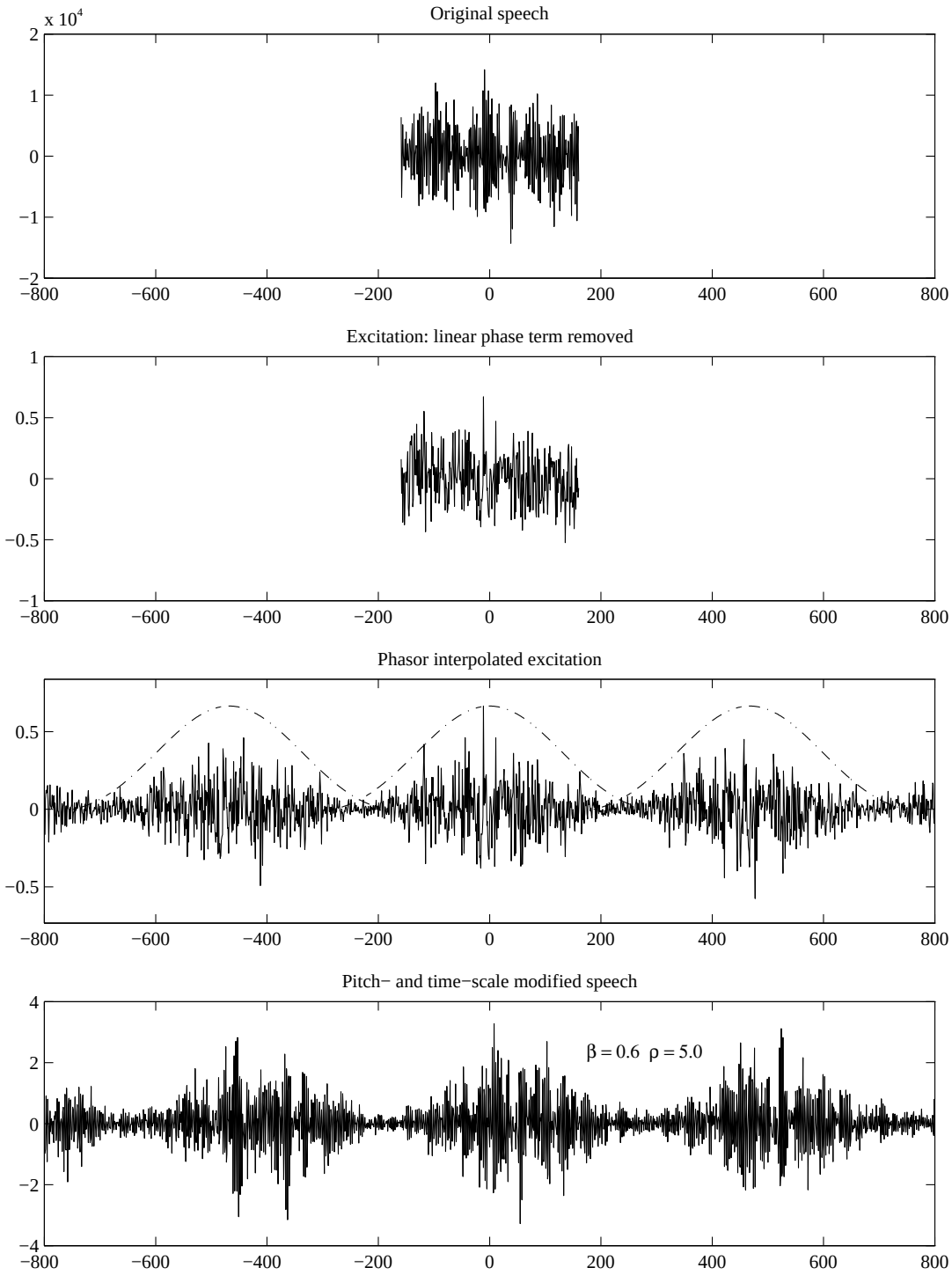

Figure 3.6: Phasor interpolation applied to unvoiced speech.

undesirable effect of this on unvoiced speech is illustrated in Figure 3.6. The time-domain equivalent of phasor interpolation causes a modulation of the pitch-modified unvoiced speech when ($\beta < 1$). This artifact is perceived as a "choppy" sound by the listener. Furthermore, this modulation produces the "buzzy" or "pulsy" structure of pitch-lowered voiced speech, since the amplitude of the residual excitation signal lying between the pulses is made unnaturally small by the window modulation effects.

3.2 Compensation for Modulation Effects

The time-domain analysis of phasor interpolation given above suggests a remedy for this undesired modulation effect. An expression for the modulation can be derived, and compensation for this effect can be made.

The effective amplitude modulation of the output signal in a frame can be written as

$$c[n] = i[n] * p[n], \tag{3.13}$$

where $p[n]$ is the pulse train given by

$$p[n] = T_0 \sum_{k=-\infty}^{\infty} \delta[n - \frac{T_0}{\beta}k].$$

This convolution can be rewritten in the frequency domain as

$$c[n] = \frac{\beta}{2\pi} \int_{-\pi}^{\pi} I(\omega) \sum_{k=-\infty}^{\infty} \delta\left(\omega - k\beta\omega_0\right) e^{j\omega n} d\omega, \tag{3.14}$$

where $\omega_0 = 2\pi/T_0$. Since $I(\omega)$ is defined to be nonzero over $|\omega| < \omega_0$, this simplifies to

$$\begin{aligned} c[n] &= \beta \sum_{k=-K}^{K} I\left(k\beta\omega_0\right) e^{jk\beta\omega_0 n} \\ &= \beta \left[I(0) + 2 \sum_{k=1}^{K} I\left(k\beta\omega_0\right) \cos\left(k\beta\omega_0 n\right) \right], \end{aligned} \tag{3.15}$$

with $K = \lfloor \frac{1}{\beta} \rfloor$. This is depicted schematically in Figure 3.7. As β is reduced, more

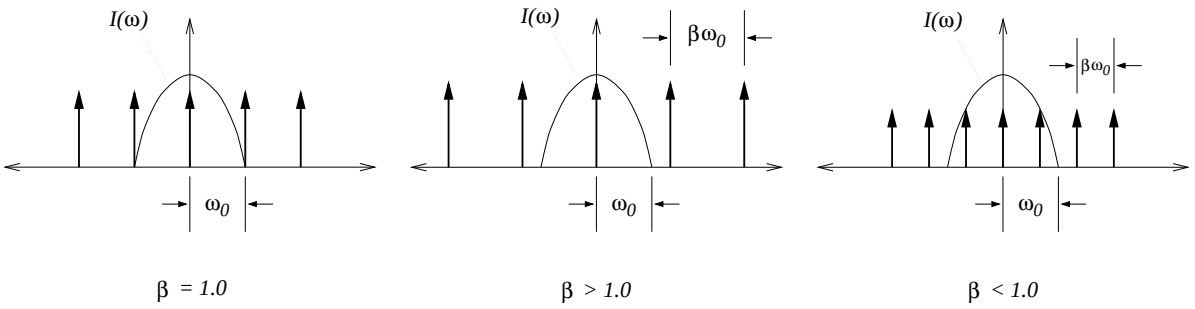

Figure 3.7: Illustration of modulation components introduced by phasor interpolation. Depending on the value of β, one or more components of the pulse train spectrum affect the pitch modified signal.

and more components of the pulse train spectrum are added to the modulation signal $c[n]$. It should also be noted that the case where $\beta \geq 1$ corresponds only to a constant gain being applied to the signal frame.

Given this expression for the modulation envelope $c[n]$, the effects of this modulation can be reversed by simply dividing each sample $x[n]$ in the pitch-modified frame by $c[n]$, i.e.,

$$\hat{x}[n] = \frac{x[n]}{c[n]}. \tag{3.16}$$

As shown in Figure 3.8, this has the desired effect of restoring the time envelope of the modified signal. However, a potential problem arises—since $i[n]$ contains zeros at multiples of T_0, some values of $c[n]$ will approach zero as β approaches 0.5 or lower, causing the method to become unstable.

To sidestep this problem, a limit can be placed on the minimum of $c[n]$. When $\beta \geq 0.5$, Equation (3.15) can be rewritten as

$$c[n] = g_0 + g_1 \cos(\omega_k n), \tag{3.17}$$

where

$$\begin{aligned} g_0 &= \beta\, I(0) \\ g_1 &= 2\beta\, I(\beta\omega_0). \end{aligned}$$

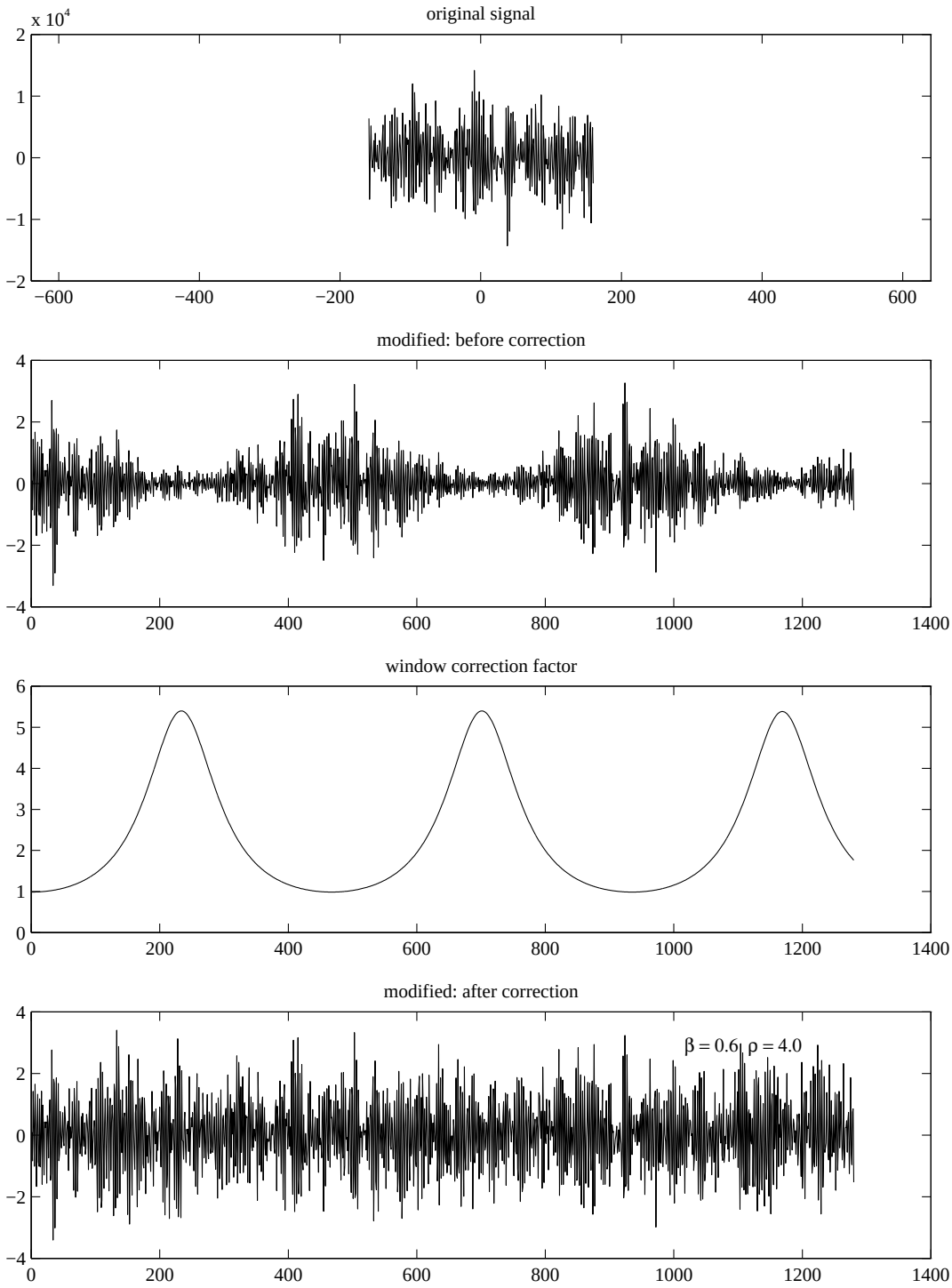

Figure 3.8: Modulation compensation applied to unvoiced speech.

It can be shown that, for the choice of window $I(\omega)$ given in Equation (3.7), the gain $1/c[n]$ will be limited to a value C_{max} by setting

$$\begin{aligned} g_0 &= \beta \frac{1 + 1/C_{max}}{2} \\ g_1 &= \beta \frac{1 - 1/C_{max}}{2} \end{aligned} \tag{3.18}$$

whenever

$$\beta - 2\beta \cos^2(\beta\pi/2) > 1/C_{max}. \tag{3.19}$$

Figure 3.9 shows the effect of this limiting on $c[n]$ and $1/c[n]$ as β is varied from 0.75 to 0.50.

For the case where $\beta < 0.5$, there will be multiple terms in the expression for $c[n]$, and the limiting method above will no longer be applicable. The instability in this case can be simply avoided by introducing a bias factor $1/C_{max}$ into the division in Equation (3.16) and taking the absolute value of $c[n]$:

$$\hat{x}[n] = \frac{x[n]}{|c[n]| + 1/C_{max}}. \tag{3.20}$$

However, it is rare that pitch modifications of $\beta < 0.5$ (lowering by an octave) are attempted in practical applications.

The effect of this compensation algorithm on *voiced* speech should also be considered. In Figure 3.10, an example of this case is presented. It can be seen that division by $c[n]$ during voiced speech tends to make the waveform less "pulsy," since a gain larger than unity is applied between the pitch pulses. This can be interpreted in the frequency domain as a sharpening of the formants that narrows their bandwidth and causes these resonances to "ring" more after excitation by a pitch pulse, as seen in Figures 3.10 and 3.11.

This effect alone is desirable – the final output waveform maintains the shape of the original without the pulsy artifact. However, a slight reverberance is also imparted to the voiced speech by this method. This effect may arise because the excitation signal produced by phasor interpolation exhibits some strange phase behavior between

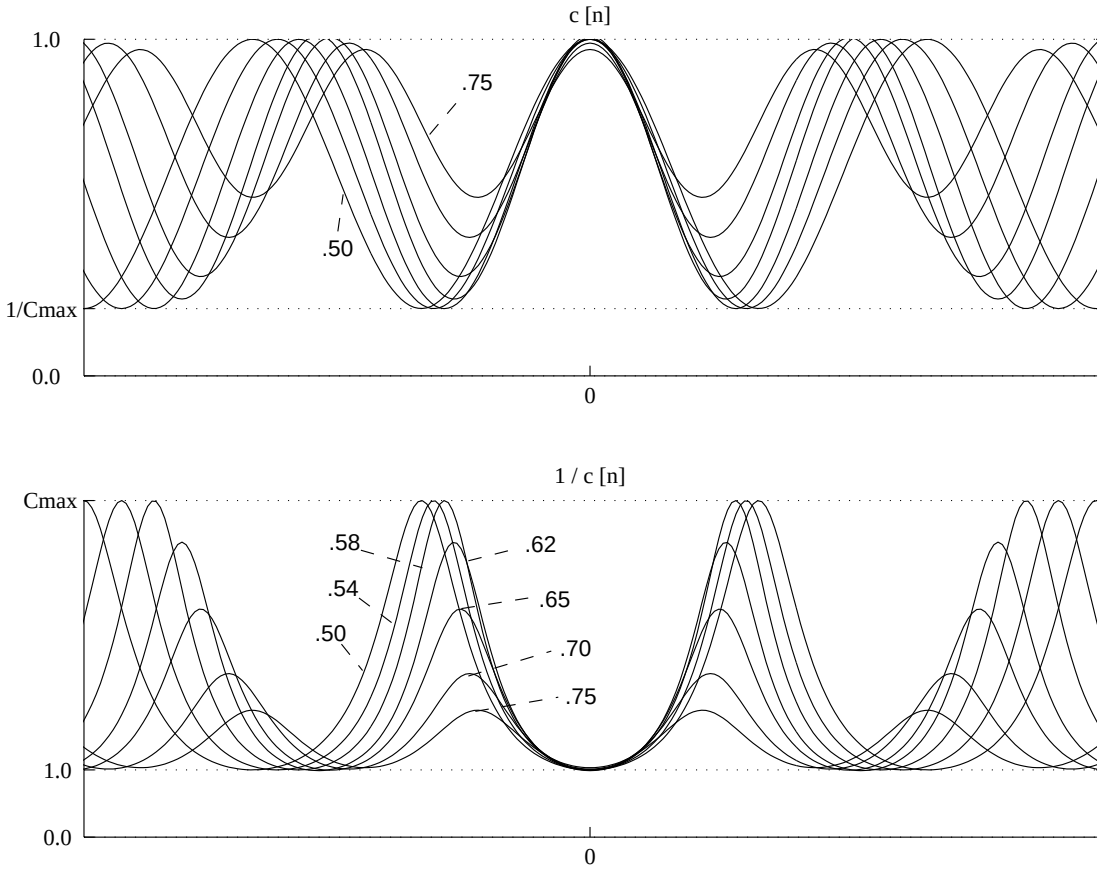

Figure 3.9: Effect of limiting $c[n]$ and $1/c[n]$ in the pitch modification compensation algorithm as β is varied from 0.75 to 0.50.

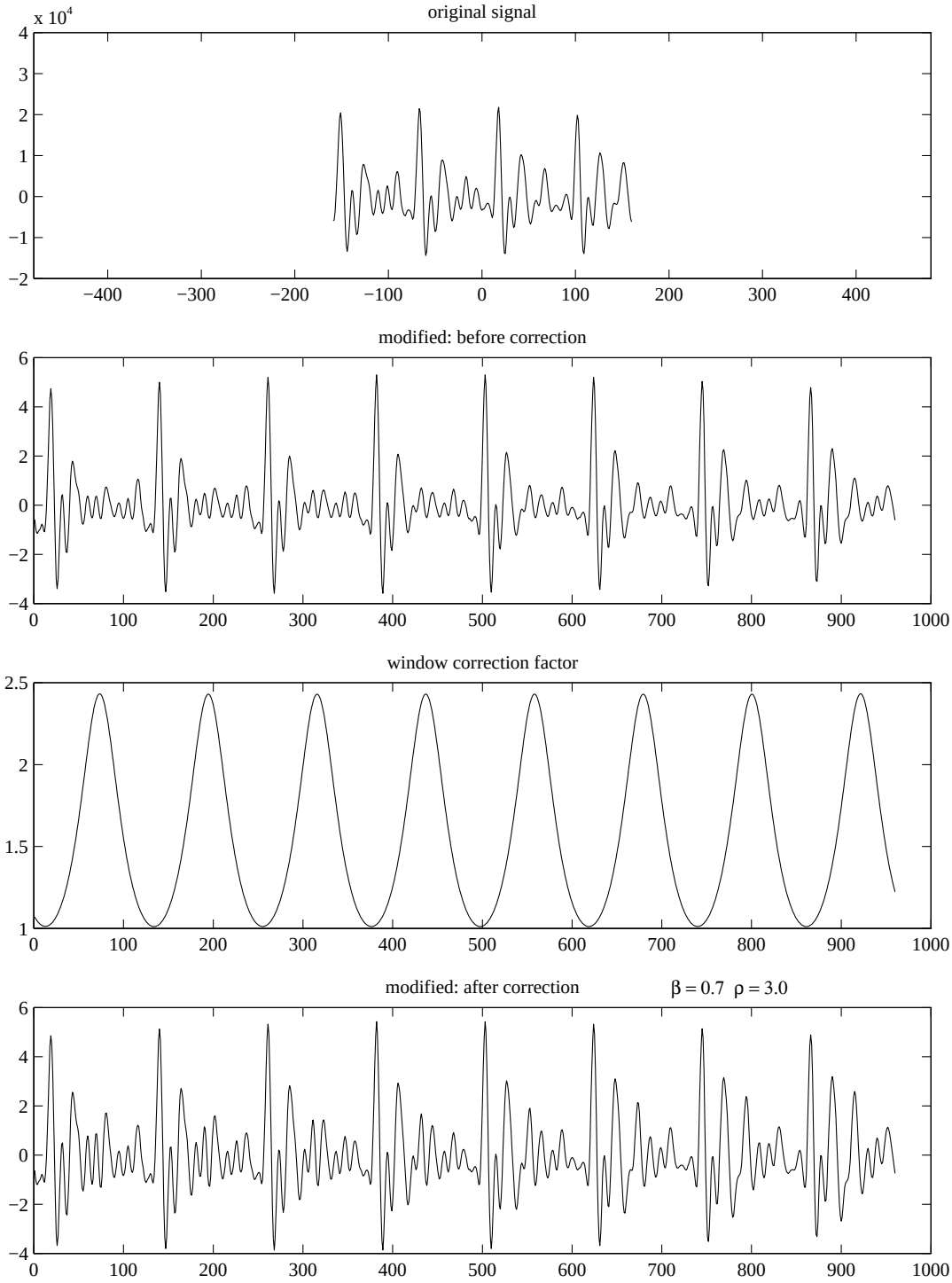

Figure 3.10: Modulation compensation applied to voiced speech.

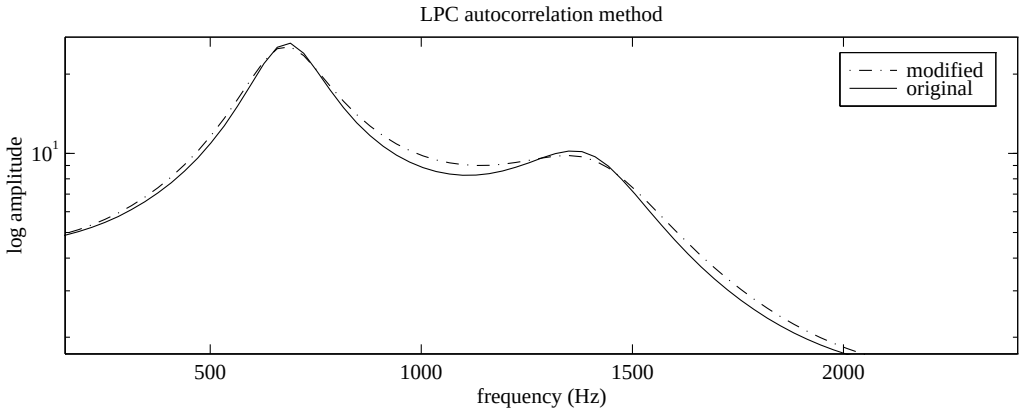

Figure 3.11: Effects of modulation compensation on the first 2 formants of a vowel sound. The compensation algorithm results in a slight reduction of formant bandwidths.

pitch pulses. By close examination of Figure 3.10, it can be seen that the ringing of the formant waveforms becomes less regular just before the onset of a new pulse. From the time domain framework developed above, this artifact can also be viewed as a phase distortion caused by the overlapping pitch period windows.

3.3 Phase Dithering

For convenience, voiced and unvoiced speech are represented in the same manner by the sinusoidal model. A commonly-cited problem in sinusoidal model-based speech modification algorithms is the existence of so-called "tonal" artifacts in unvoiced speech after time-scale expansion or raising of the pitch. One method for circumventing these artifacts is to use an entirely different model for unvoiced portions of the speech signal. As described in Section 2.1.3, researchers have proposed harmonic/stochastic decompositions of the signal for coding [31, 35] or modification [46, 33]. Most of these are based on representing the periodic portion of the signal by a sinusoidal model and then modeling the residual signal as the output of a time-varying

filter excited by white noise. Although decompositions such as these can mitigate some types of artifacts, it has been noted that the output signal often suffers from a lack of "perceptual fusion" of the two signal components [44]. This results in the sinusoidal and noise parts being perceived as two distinct sources by the listener, rather than as a single, unified source.

The algorithm presented in this section (and in [96]) is an extension of the Analysis-by-Synthesis/Overlap-Add (ABS/OLA) sinusoidal model. In this extension, noise-like segments of the signal are represented by sinusoidal components, but the phases of these sinusoids are manipulated to preserve the "perceptual randomness" of the signal after modification – that is, to remove the perception of tonality in the signal. A perceptual motivation for this algorithm is given, as is a frequency-domain interpretation of its resulting effect on the signal. Finally, the results of a subjective comparison test evaluating the effectiveness of the algorithm are also presented.

3.3.1 Phase Randomization Synthesis Algorithm

Perceptual motivation Empirically, it has been found that the ABS/OLA model is capable of faithfully reproducing both voiced and unvoiced sounds when a frame update period of 10 milliseconds or less is used in synthesis. However, when time-scale expansion and/or pitch raising operations are performed, the unvoiced segments take on the above-mentioned "tonal" character.

This role of time-scale expansion in causing this artifact can be explained in terms of current theories of pitch perception. One theory suggests that the brain assigns the perceived pitch of a tone complex based on the intervals between peaks in the fine time structure of the signal at various points on the basilar membrane[2], integrated over a time interval on the order of several milliseconds [97]. Thus, any arbitrary set of sinusoidal components with constant amplitude and frequency (e.g., a time-expanded synthesis frame) will produce regular patterns at various places

[2]The basilar membrane behaves roughly as a filter bank.

across the basilar membrane, and the brain will recognize prominent periodicities in these patterns. When the the sinusoidal components remain stationary for a duration significantly large with respect to the integration time of this human pitch detection mechanism, the resynthesized speech signal begins to take on a tonal character.

It has also been observed that this tonal artifact is exacerbated by pitch-raising modification. In [28], McAulay and Quatieri justify the use of the sinusoidal representation for unvoiced speech by an argument based on the Karhunen-Loève expansion for noise-like signals. They conclude that this representation for unvoiced speech is valid when the sinusoidal components are spaced "closely enough" to each other in frequency that the ensemble power spectral density is relatively smooth as a function of frequency. When the fundamental frequency of the sinusoidal components is raised in a given frame, the components become more widely spaced in frequency, leaving a spectral shape that is less smooth and more peaked as the pitch is raised. Thus, the model for the noise becomes less mathematically representative of the signal characteristics. From a human perception viewpoint, the tone complex representing the noise-like signal becomes more sparse, and spectral lines become more prominent. This effect tends to worsen the perceived tonal noise artifact.

The top and middle panels of Figure 3.14 show the periodogram of an 80 ms segment of unvoiced speech signal (phoneme */s/*) before and after time-scale expansion and pitch raising. Note that the spectrum of the modified signal, which possesses a significant tonal noise artifact, distinctly exhibits the presence of these tonal components.

The above perceptual arguments suggest that the perception of randomness in the modified signal can be maintained by (*i*) disrupting long-term periodicities in the time waveform over the course of the synthesis frame, and (*ii*) maintaining the smoothness of the original signal spectrum. The next section presents a method that is capable of achieving these objectives.

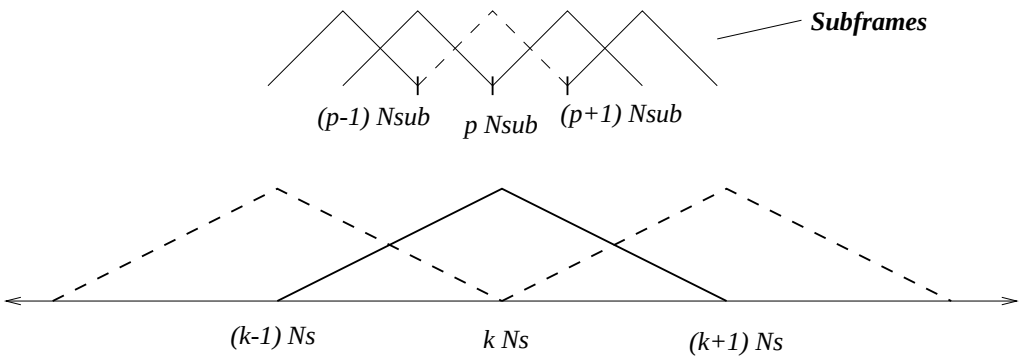

Figure 3.12: Subframe overlap-add synthesis.

Overlap-add phase dithering It has been found experimentally that the above goals can be realized by modulating the phase of the sinusoidal model components in each frame of unvoiced speech. The nominal frequency of each component is kept the same, but the time structure of combined sets of these components along the basilar membrane no longer exhibits the periodicities originally detectable by the listener. One simple way to implement such an idea within the context of an *overlap-add* model is to subdivide each time-scale expanded frame, as shown in Figure 3.12, and randomize (or "dither") the phase offsets between components in each subframe.

Referring to Equation (2.13), each N_s-sample frame can be divided into subframes of length N_{sub}, where $N_{sub} < N_s$. It is possible to resynthesize a signal *identical to the original* frame $s_k[n]$ by

$$s_k[n] = \sum_{m=-\infty}^{\infty} w_s[n - mN_{sub}] \sum_{l=0}^{L-1} A_l \cos(\omega_l n + \phi_{l,m}), \tag{3.21}$$

where $w_s[n]$ is a window function that is nonzero over $[-N_{sub}, N_{sub}]$, and the frame k notation has been suppressed. Equation (3.21) and the original synthesis Equation (2.13) can be made equal by letting $\phi_{l,m} = \phi_l^k$ for all m, where ϕ_l^k is the original phase estimate for the frame in Equation (2.13). (In practice, the limits of the sum on m can be made finite, since $s_k[n]$ is multiplied by a finite time support window.)

Alternatively, the phase offsets between sinusoidal components in each subframe

can be *varied* by adding a random offset $V_l\,\psi_{l,m}$ to each $\phi_{l,m}$ term:

$$\phi_{l,m} = \phi_l^k + V_l\,\psi_{l,m} \tag{3.22}$$

where $\psi_{l,m}$ is a uniform random variable over $[-\pi, \pi]$ and V_l is a weighting factor that takes on values in $[0, 1]$. This suggests the possibility of using a "soft-decision" scaling of V_l that depends on the nature of the input signal over each frame; the random variable $\psi_{l,m}$ can be weighted by the factor $V_l \in [0, 1]$ to produce varying degrees of phase randomization, or equivalently, disruption of the signal periodicities. Thus, when $V_l = 0$ for all l, the synthetic contribution $s_k[n]$ will be resynthesized in its original form, but when $V_l = 1$, the phase offsets will be completely random from subframe to subframe. This scaling can also be varied across frequency (the l index) to introduce frequency-dependent phase randomization.

Although the previous equations have been presented as time-domain summations of cosines, $s[n]$ can be computed much more efficiently using the inverse FFT, as mentioned in Section 2.1.2. This idea can be easily extended to the computation of Equation (3.21) by using a sequence of N_s/N_{sub} IFFT's and an overlap-add procedure analogous to that used in the original model [25].[3]

Frequency-domain interpretation Interpreting the above algorithm in the frequency domain provides several interesting insights into its behavior. Specifically, the effect on each component can be described as a frequency modulation that spreads the effective bandwidth of each component, smoothing the signal spectrum.

Rewriting the subframe overlap-add equation (3.21) in terms of complex signals and substituting Equation (3.22), we obtain

$$s_k[n] = \Re e \left\{ \sum_{m=-\infty}^{\infty} w_s[n - mN_{sub}] \sum_{l=0}^{L-1} A_l e^{j(\omega_l n + \phi_l + V_l \psi_{l,m})} \right\}. \tag{3.23}$$

[3]The use of a *shorter* IFFT for the subframes is desirable from the standpoint of computational complexity, but results in a slight loss of accuracy in Equation (3.21), due to quantization of the frequencies to FFT bin values.

This equation can be rewritten to incorporate a set of functions $b_l[n]$ that are frequency modulated by respective sinusoidal signal components,

$$s_k[n] = \Re e \left\{ \sum_{l=0}^{L-1} b_l[n] A_l e^{j(\omega_l n + \phi_l)} \right\} \tag{3.24}$$

where

$$b_l[n] = \sum_{m=-\infty}^{\infty} w_s[n - mN_{sub}] e^{jV_l \psi_{l,m}}.$$

In this equation, the term $e^{jV_l\psi_{l,m}}$ can be viewed as a complex-valued random process with a sampling rate of $1/N_{sub}$, and $w_s[n]$ as an interpolation filter that performs a conversion to the sampling rate of the speech signal. The transform of $b_l[n]$ can be written as

$$B_l(e^{j\omega}) = W_s(e^{j\omega}) \sum_{m=-\infty}^{\infty} e^{-j(m\omega N_{sub} - V_l \psi_{l,m})}, \tag{3.25}$$

where $W_s(e^{j\omega})$ is the Fourier transform of the subframe synthesis window.

If V_l is set to 0 for all l, then the summation will equal a pulse train whose pulses coincide with the nulls of $W_s(e^{j\omega})$, resulting in $B_l(e^{j\omega}) = \delta(\omega)$; the sinusoidal component is left unmodified. However, if $V_l > 0$, then the elements of the frequency-domain pulse train will *not* coincide with the window transform nulls, but will instead be weighted by nonzero samples of the window transform. This means that, as V_l is increased, $B_l(e^{j\omega})$ will on average assume the shape of the window transform $W_s(e^{j\omega})$, as shown in Figure 3.13. Thus, the bandwidth of $B_l(e^{j\omega})$ will be increased by making the subframe durations shorter (i.e., decreasing N_{sub}), since this will widen the mainlobe of the window transform.

The increase in bandwidth of each sinusoidal component results in a smoothing of the resynthesized signal spectrum. This is demonstrated in Figure 3.14, where the time-scale and pitch modified signal spectrum is shown with and without the phase randomization algorithm applied.

Other work The use of a modulating function such as $b_l[n]$ to preserve randomness in the sinusoidal representation of noise is reminiscent of ideas in [35], where "nar-

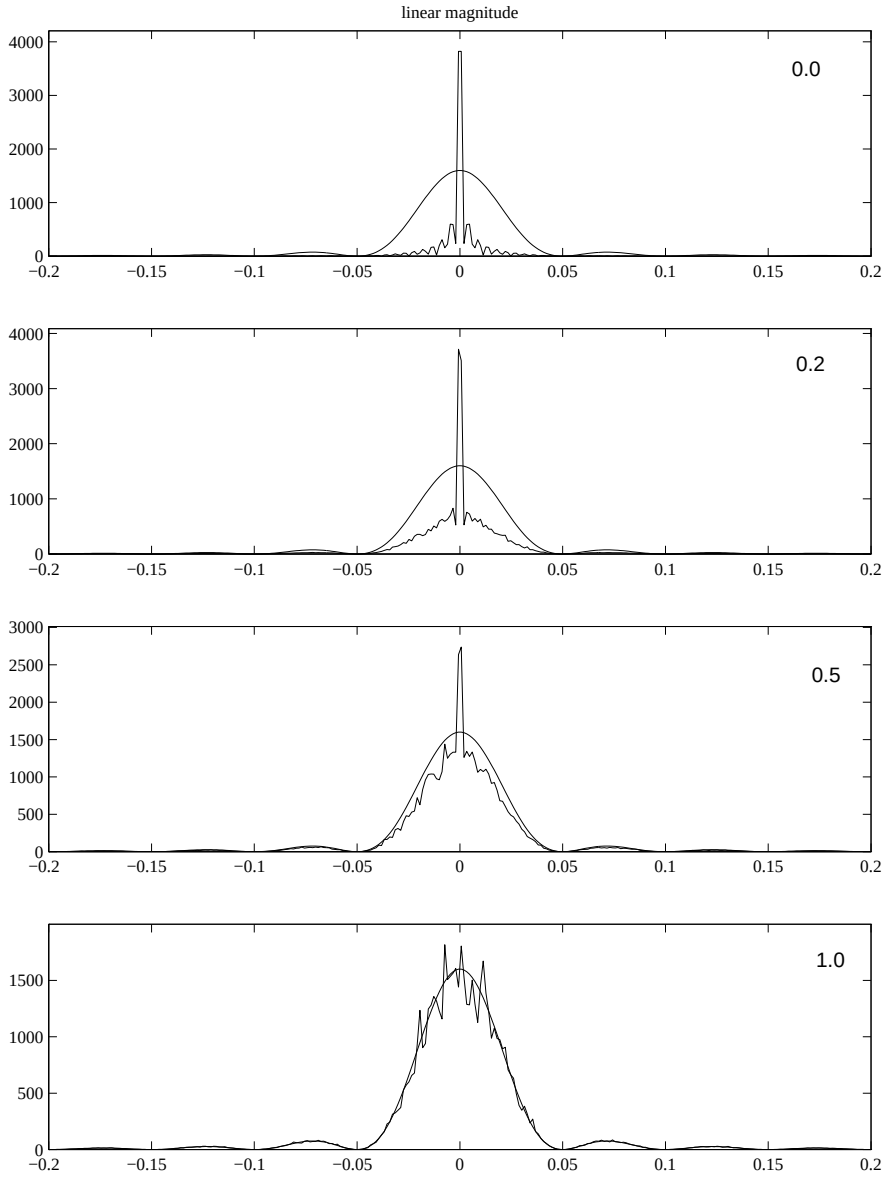

Figure 3.13: Illustration of effect of phase randomization on frequency domain sinusoidal basis functions as V_l is varied in Equation (3.22) (averaged over 30 trials). Value of V_l is given in top right corner of each plot. Note that the bandwidth of the stochastic basis functions approaches the bandwidth of the window transform $W_s(e^{j\omega})$ as V_l approaches 1.0.

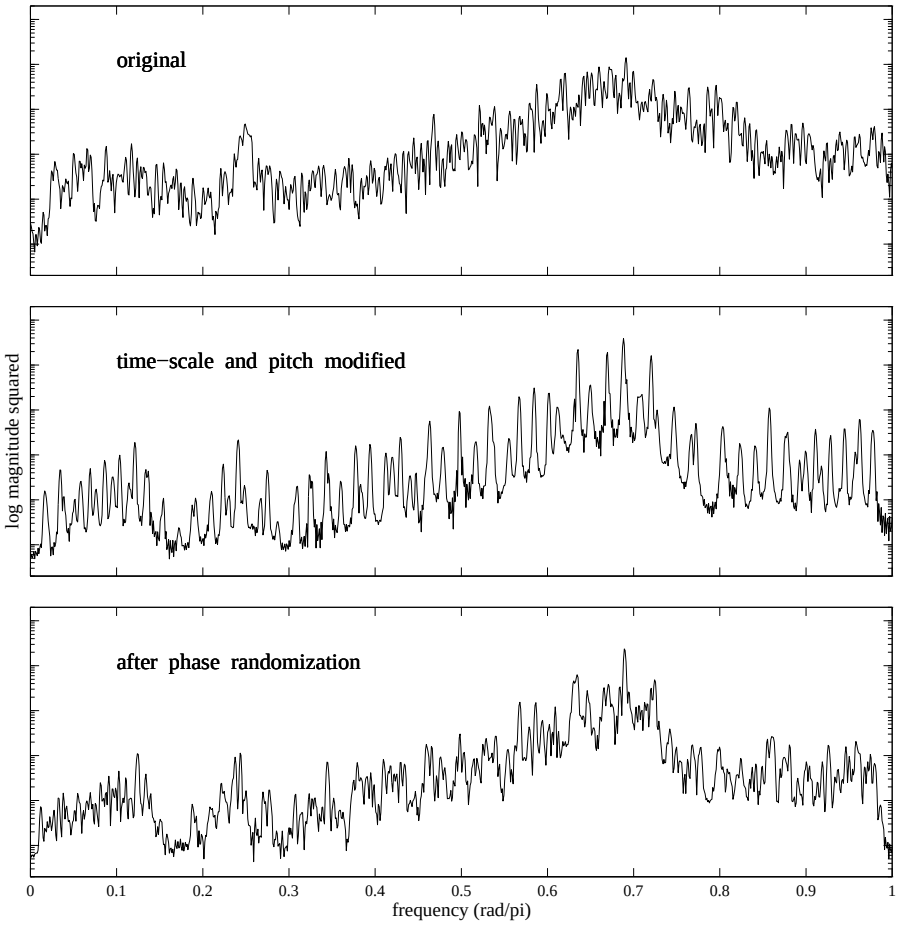

Figure 3.14: Periodogram (50 ms rectangular window) of 80 ms unvoiced speech segment. *(top)* original signal; *(middle)* signal after time-scale expansion by a factor of 4 and upward pitch shift by a factor of 2; *(bottom)* resulting signal after modification using phase randomization algorithm. (frame length before modification = 10 ms, $N_{sub} = 5, \psi_{l,m} \sim U[-\pi, \pi]$)

rowband basis functions" were used to represent unvoiced speech in a speech coding application. In the algorithm proposed here, however, a straightforward extension of the ABS/OLA synthesis procedure provides for a computationally efficient synthesis of these modulated components, avoiding filtering of long, randomly generated sequences. The effects of sinusoidal phase coherence on voiced speech quality have also been studied in the development of speech coding applications using the sinusoidal transform coder [30].

3.3.2 Analysis Algorithm

Voicing measure The incorporation of a voicing decision is necessary to preserve the phase coherence of voiced speech segments. Several approaches to estimating the "degree of voicing" are mentioned in the sinusoidal modeling and speech coding literature. In [98], the signal-to-noise ratio between a set of harmonic components and the original speech spectrum is mapped to the degree of voicing, with the implication that a harmonic model will better fit the spectrum in voiced speech. A similar notion is used in a frequency-dependent voicing decision in [31]. The synthesis method developed in this paper can be coupled with any of these analysis methods to implement frequency-dependent voicing decisions.

In experiments with this algorithm, an analysis process similar to those in [98] and in [31] has been used. This analysis relies on the following assumptions:

- The sinusoidal components in the model will be very nearly harmonically related in voiced portions of the speech signal.
- These harmonically-related sinusoidal components will constitute a large part of the energy in the spectrum of the analyzed frame.

The analysis algorithm takes these assumptions into account by measuring the signal-to-noise ratio

$$S = \frac{\int_a^b |A(\omega)|^2}{\int_a^b |S(\omega) - A(\omega)|^2} \tag{3.26}$$

where $A(\omega)$ represents the sinusoidal component line spectrum with frequencies shifted to the nearest harmonic (i.e., no longer quasiharmonic components), and $S(\omega)$ represents the original signal spectrum. The limits a and b represent the upper and lower edges of frequency bands over which the SNR measure is computed. The SNR values obtained are then mapped to a "degree of voicing" in the interval $[0, 1]$. Histograms of SNR values over hand-marked voiced and unvoiced speech are then used to set thresholds and establish a "soft-decision" voiced/unvoiced measure.

Although this method works well in most cases, it has a critical flaw. It is highly dependent on the stationarity of the signal and on correct estimation of the pitch. If signal characteristics over the analysis frame duration change rapidly, the spectrum will *not* consist of a set of harmonically-spaced discrete spectral lines, even in voiced speech. Neither will this occur when the pitch period changes rapidly over the frame duration. Instead, frames falling into these cases will be classified as "unvoiced," resulting in a "hoarse" speech quality in synthesis.

One example of such a degenerate case is shown in Figure 3.15. The nonstationarity of the signal in the glottal stop of the phrase "... needle and ..." (between samples 1000 and 1500 of the figure) results in a spectrum that does not fit the harmonic structure assumed for voiced speech. Despite this, the signal is clearly *not* unvoiced – classifying the frames in the vicinity of this aperiodic glottal activity as unvoiced results in a distortion of important time-domain waveform characteristics. This type of distortion impairs the intelligibility of the resynthesized speech and causes audible quality degradation.

Ad hoc methods of detecting transient portions of the signal can be derived, and these can be used to "turn-off" the phase dithering algorithm when the signal is nonstationary.

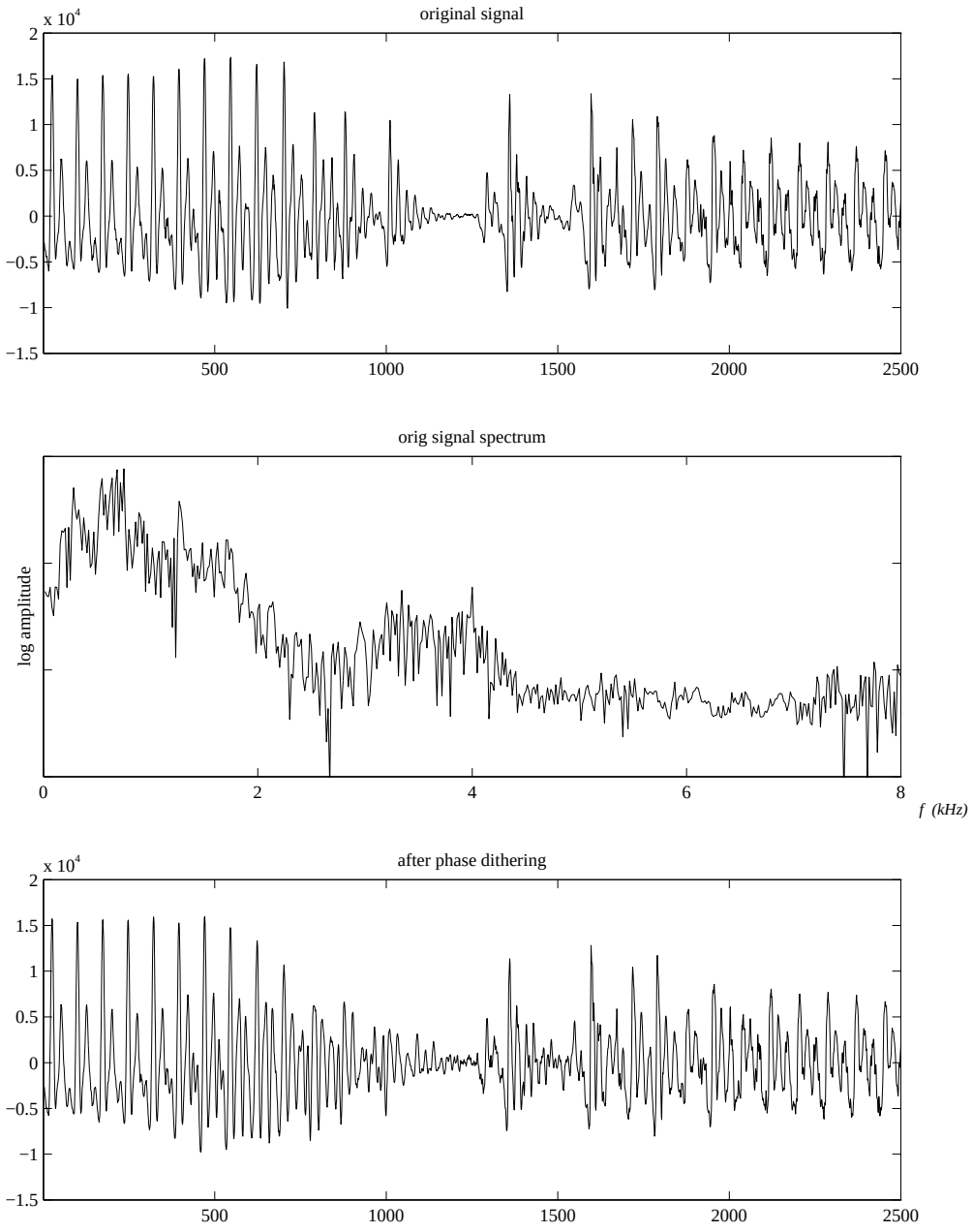

Figure 3.15: Effect of incorrect voicing decision on glottal stop (between samples 1000–1500) in the phrase "... needle and ..." *top:* original signal; *middle:* spectral magnitude of original – note that spectrum does not fit the assumed form of voiced speech; *bottom:* resynthesized signal after phase dithering applied with sinusoidal model – glottal stop waveform is disrupted.

3.3.3 Results

Subjective comparison test To confirm the appropriateness of the phase randomization approach, a subjective comparison test was conducted using 25 volunteer students and employees of the Center for Signal and Image Processing. Of these 25 subjects, two were experienced in subjective speech quality assessments, and 23 were naïve listeners. The subjects were asked to compare 32 pairs of utterances, where each pair consisted of one utterance synthesized with the phase randomization algorithm applied and one synthesized using ABS/OLA without this extension. The order of the sentence pairs and the elements within each pair were selected randomly for each subject. For each trial, the two synthesized utterances were presented via headphones. The subject was then asked to select utterance "A" or "B" according to his or her preference "in terms of overall sound quality."

The speech material used as input to the algorithm consisted of eight short phrases extracted from sentences in the TIMIT database [99], shown in Table 3.1. The material used was selected to represent an equal number of male and female voices and to contain several unvoiced phonemes. The sinusoidal model analysis procedure was run on each of the sentences, and a "hard" voicing decision was made based on a comparison of the quasiharmonic sinusoidal model components to a harmonic spectrum, as described in the previous section. Additional constraints were applied to prevent a decision of "unvoiced" in voicing onsets and other voiced transient signal segments. In a method similar to that employed in STC speech coding [98], this V/UV decision was used to control V_l in Equation (3.22) via a "voicing cutoff frequency" ω_c, such that

$$V_l = \begin{cases} 1 & \text{if } \omega_l > \omega_c \\ 0 & \text{otherwise} \end{cases}, \tag{3.27}$$

where ω_l is the lth sinusoid frequency. In frames declared voiced, ω_c was set to π, while in unvoiced frames ω_c was set to 0. A time-domain smoothing of the sequence of ω_c values between these extremes was performed by passing the sucessive values of ω_c

Table 3.1: Phrases used in subjective comparison test of phase randomization algorithm.

key	utterance text
femA	*...shimmers on the ocean...*
femB	*...cyclical programs...*
femC	*...seamstresses attach zippers...*
femE	*...through Sequoia national forest...*
maleA	*...his scalp was blistered...*
maleB	*...catastrophic economic cutbacks...*
maleC	*...four extra eggs for breakfast...*
maleD	*...many wealthy tycoons splurged...*

through a lowpass filter to provide a gradual transition between voiced and unvoiced speech.

Four test conditions were applied to each of the eight sentences. Time-scale modifications by factors of 2.0, 3.0, and 4.0 (slower speech) were applied with no pitch modification, and time-scale modification by a factor of 3.0 was also applied in combination with a pitch modification by a factor of 1.5 (higher pitch).

Test results and discussion The results of the four test conditions described above are given in Table 3.2. Each value given represents a percentage of responses preferring the phase randomization method over the standard modification method, averaged over the eight utterances and 25 test subjects. Based on this number of trials, the test results show a preference for the phase randomization method that is statistically significant ($p < 0.001$) in all cases.

Although it should be expected that the algorithm would provide greater improvement of speech quality in more drastic modifications, this was not observed in the response percentages for tests B, C, and D. One explanation of this effect is as follows: The subjects were instructed only to compare "overall sound quality" and not any specific aspect of the speech signals. Since most of the subjects participating in the test were not experienced in critical listening tests for speech processing, they

Table 3.2: Results of subjective comparison test of utterances synthesized with and without application of phase randomization method in unvoiced speech.

test	modification factors	% preferring phase rand
A	$\beta = 1.0, \rho = 2.0$	81.0
B	$\beta = 1.0, \rho = 3.0$	79.0
C	$\beta = 1.0, \rho = 4.0$	73.5
D	$\beta = 1.5, \rho = 3.0$	72.5

tended to judge *both* exemplars as more "unnatural" than normal speech for drastic modifications of time scale or pitch. Because of this, the response percentages tended to gravitate towards a result more consistent with guessing rather than definite preference of one or another method. This theory was confirmed by interviews with subjects after the test. It is also interesting to note that the two subjects who had previous critical listening experience chose the phase randomization method in 100% of the tested cases.

3.4 Pitch Pulse Onset Time Estimation

One major source of artifacts arising in sinusoidal model-based speech modification is errors in the *pitch pulse onset time* estimation algorithm, as discussed in Section 2.1.1, the pitch pulse onset time is defined to be the time index of the pitch pulse closest to the frame center. In the ABS/OLA model, this quantity plays an important role in (i) aligning adjacent frames in OLA synthesis, and (ii) removing linear phase terms prior to pitch modification. Objectionable artifacts arise for two reasons:

1. When pitch pulse onset time estimation errors occur such that the *difference* between successive onset time locations is not an integer multiple of T_0, non-coherent overlap between adjacent frames occurs in overlap-add synthesis. Figure 3.16 compares the waveform appearance of correctly aligned frames to the

situation where pitch pulse onset times in adjacent frames are not consistent with each other. The susceptibility of the algorithm to such errors is discussed more thoroughly in Section 4.5 within the context of frame alignment in text-to-speech synthesis.

2. Errors in locating the *absolute locations* of pitch pulses (independent of interframe onset time differences) cause problems in the "phasor interpolation" method of modifying the pitch period [24]. As mentioned in the time-domain analysis of the ABS/OLA pitch modification in Section 3.1.3, the pitch pulse onset time is used to remove linear phase offsets, as indicated in Figure 3.5. These errors result in a garbled speech quality.

   The original onset time algorithm (based on the work of McAulay and Quatieri [17, 30, 100]) attempts to locate these absolute locations and works well for strongly voiced speech. However, in weakly voiced speech segments, the estimator becomes unreliable, leading to the artifacts mentioned.

These observations suggest that a necessary component is an onset time estimator that can robustly find the onset time location within a speech frame *and* can maintain consistency with the pitch period from frame to frame. With these two objectives in mind, a correction algorithm was designed to correct gross errors in the pitch pulse onset time estimates [101]. This results in a more robust, consistent onset time estimate and eliminates misalignment artifacts.

Pitch pulse onset correction algorithm The following algorithm has been devised to more robustly estimate the pitch pulse onset times of each analyzed frame.

1. After sinusoidal model analysis, the following estimator function (from [100]) is evaluated for each set (frame) of sinusoidal parameters to obtain a pitch pulse onset time estimate τ_{est}:

$$\tau_{est} = \arg\left(\max_{\tau}\left\{\sum_{l=0}^{J} b_l^2 \cos(\psi_l + \omega_l \tau)\right\}\right), \tag{3.28}$$

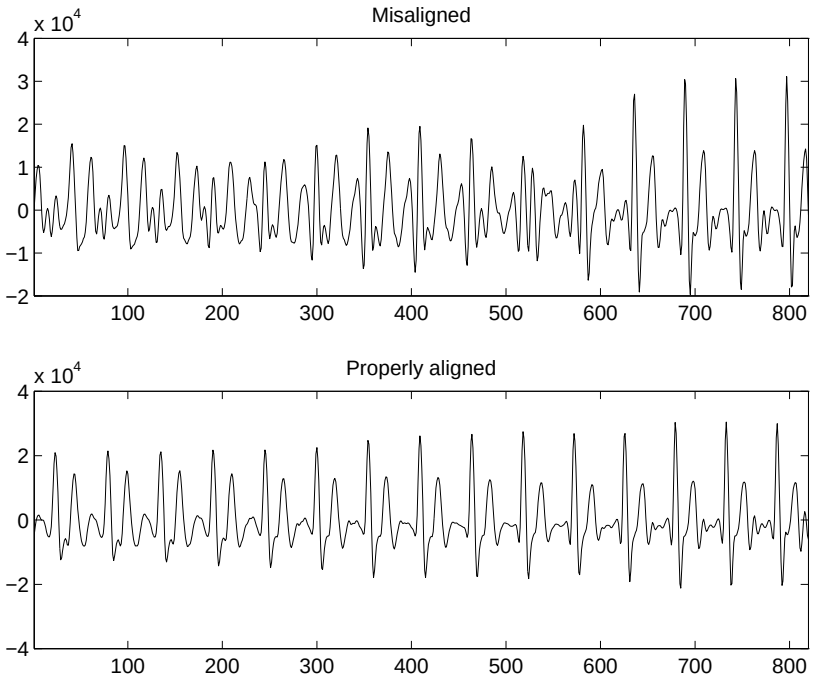

Figure 3.16: Effects of pitch pulse onset time estimation errors: *Upper plot:* Effect of pitch pulse misalignment on resynthesized speech. *Lower plot:* Effect of correct pulse alignment.

where J is the number of sinusoids in the frame, and ω_l is the lth component frequency. The excitation amplitude b_l and phase ψ_l are obtained by dividing out the (complex) vocal tract transfer function $H(\omega_l)$ from the sinusoidal components. In practice, τ is evaluated over a grid of onset time values between $-T_o/2$ and $T_o/2$, where T_o is the pitch period.

2. For each of the estimated onset times $\tau_{est}(k)$ (onset estimate for the kth frame), a *predicted* onset time τ_{pred} for frame $k+1$ is found from the pitch period T_o:

$$\tau_{pred}(k+1) = \tau_{est}(k) + T_o^{avg} \left\lfloor \frac{N_s - \tau_{est}(k)}{T_o^{avg}} + 1 \right\rfloor - Ns, \tag{3.29}$$

where T_o^{avg} is the average of $T_o(k)$ and $T_o(k+1)$ and N_s is the analysis frame length.

3. The interframe onset time differences Δ_{est} and Δ_{pred} are then computed for each frame by

$$\begin{aligned} \Delta_{est}(k+1) &= \mathrm{mod}(\tau_{est}(k+1) - \tau_{est}(k), T_o^{avg}) \\ \Delta_{pred}(k+1) &= \mathrm{mod}(\tau_{pred}(k+1) - \tau_{est}(k), T_o^{avg}), \end{aligned} \tag{3.30}$$

where $\mathrm{mod}(x, n)$ represents x modulo n.

4. For each frame, the agreement of Δ_{est} and Δ_{pred} is checked. If $\Delta_{est} \approx \Delta_{pred}$, then the estimated onset time is consistent with the pitch period, and τ_{est} is a reasonable estimate of the pitch pulse onset time. Frames that satisfy

$$\left| \frac{\Delta_{est} - \Delta_{pred}}{T_o^{avg}} \right| < \nu \tag{3.31}$$

are labeled as "consistent," all others are labeled as "inconsistent."[4]

5. Typically, consistent estimates will occur in groups, generally over strongly voiced segments of the utterance. From the entire set of onset time estimates,

[4] $\nu = 0.1$ in the implementation.

runs of K or more[5] consecutive consistent estimates are marked as "anchors." For each frame k in the set of anchor frames, the final pitch pulse onset time $\hat{\tau}(k)$ is simply given by

$$\hat{\tau}(k) = \tau_{est}(k). \tag{3.32}$$

6. The energy of each of the frames with inconsistent estimates (frames between the anchor sets) is computed from the sinusoidal parameters. In each set of non-anchor frames, the frame with the minimum energy is selected as a "target," as depicted in Figure 3.17.

7. Starting from the right end of each anchor, the onset time for each inconsistent frame is determined by the following recursion:

$$\hat{\tau}(k+1) = \hat{\tau}(k) + T_o^{avg} \left\lfloor \frac{N_s - \hat{\tau}(k)}{T_o^{avg}} + 1 \right\rfloor - N_s, \qquad k = k_0, k_1, ..., k_M \tag{3.33}$$

where $\hat{\tau}(k_0)$ is the onset time value at the right end of a given anchor set, and k_M is the nearest target frame's index.

8. Starting from the left end of each anchor, the onset time for each inconsistent frame is predicted "backward in time" by the following recursion:

$$\hat{\tau}(k-1) = \hat{\tau}(k) + T_{oB}^{avg} \left\lfloor \frac{N_s + \hat{\tau}(k)}{T_{oB}^{avg}} \right\rfloor + N_s, \qquad k = k_0, k_{-1}, ..., k_{-M} \tag{3.34}$$

where $T_{oB}^{avg} = (T_o(k) + T_o(k-1))/2$, $\hat{\tau}(k_0)$ is the onset time value at the left end of a given anchor set, and k_{-M} is the nearest target frame to the left.

This specifies the pitch pulse onset time for all frames in the analyzed signal. Estimates consistent with the pitch period are accepted, while estimates that do not agree with the pitch period are forced to agree by replacing them with values predicted from the pitch estimate.

The pitch pulse onset correction algorithm results in improved quality after time-scale and pitch modification of continuous speech. The algorithm is able to

[5] $K = 3$ in the implementation.

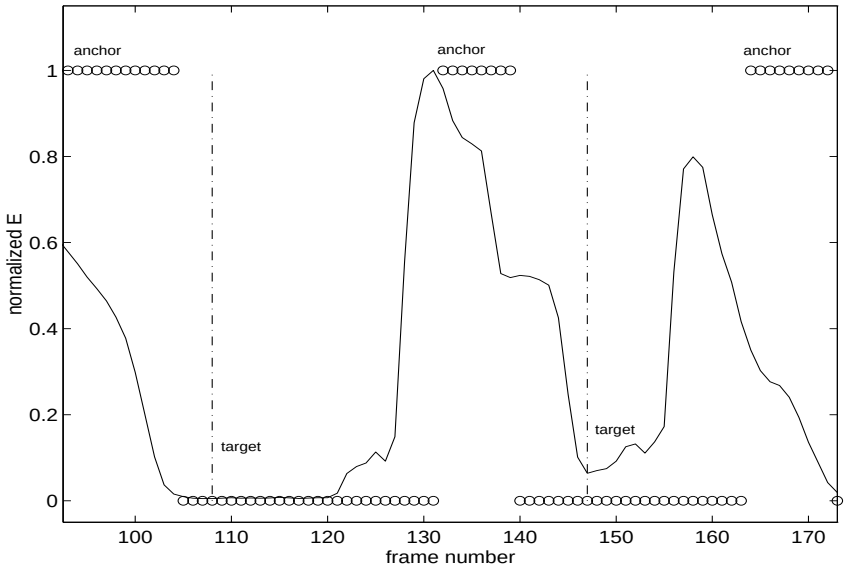

Figure 3.17: "Anchor frames" in pitch pulse onset time correction algorithm. "Consistent" estimates are denoted by circles with a y-value of 1; "inconsistent" estimates by circles with a y-value of 0. The energy contour shown is used to select "target" frames for onset time prediction.

correct spurious errors and inconsistencies in the algorithm results. This is especially true of unstressed or phrase-final speech segments, which tend to exhibit greater irregularities in voicing characteristics and waveform pulse shape [102].

However, it should be noted that long sets of erroneous, yet self-consistent, onset estimates are not corrected by this method. For instance, the glottal excitation patterns of male speakers sometimes exhibit a phenomenon called *diplophonic double-pulsing*, which produces a "dual-pulse" excitation signal [102]. This is caused by a second, shorter opening of the glottis during the pitch cycle. The pitch pulse onset estimator will often choose this second pulse as the onset location for several frames in a row, and this causes artifacts in the output speech. However, the correction algorithm is not able to detect this error, since the secondary pulses are still separated from each other by an interval of length T_0.

Further implications of pitch pulse onset time errors are described in the next chapter, within the context of text-to-speech synthesis.

CHAPTER 4

TEXT-TO-SPEECH SYNTHESIS USING A SINUSOIDAL MODEL

This section presents the application of the improved sinusoidal model described in Chapter 3 within the framework of a concatenation-based text-to-speech system [103, 104]. First, the "front-end" system used to perform the necessary linguistic analysis of the input text is briefly described. An overview of the proposed method is then given, and each part of this algorithm is described in detail. Finally, its performance is compared to that of an existing method.

4.1 Overview of Method

4.1.1 Laureate TTS System from British Telecom

The LAUREATE II system from British Telecom ranks among the best text-to-speech systems in the commercial market in its capability to produce high-quality, natural sounding synthesized speech. It is based on a concatenation method similar to that described in Section 2.2.4, and this makes it well-suited to serve as a testbed for the algorithms described in this Chapter. The BT research group responsible for development of LAUREATE II has generously lent parts of their system to the Center for Signal and Image Processing at Georgia Tech for the purpose of serving as a technology testbed.

Furthermore, this system has been designed with a very modular structure to make it well suited to be both a research-oriented and a commercial software package. As shown in Figure 4.1, the system is broken into components that follow quite closely the module descriptions given in Section 2.2. Each of these components is designed to interact with a "core" application, which stores data in a "linguistic object" that is not tied to the theories embodied by any of the components. Each component can also access data specific to itself, via external data files. For example, the "Unit selection" and "Realization" components use data files that describe the speech data inventory.

The focus of the work described in this proposal is on the "Realization" component of Figure 4.1. This component is responsible for retrieving units from the speech database, concatenating these units, smoothing at the boundaries, and applying the necessary prosodic contours to the synthetic speech.

4.1.2 Sinusoidal Model Synthesis Module

Figure 4.2 shows a block diagram of the sinusoidal model synthesis algorithm, which is a subsystem of the "Realization" component in Figure 4.1. A brief overview of the function of each block is as follows:

Inventory preprocessing The first step in the synthesis process involves preprocessing the entire inventory of speech used by the synthesizer. This consists of applying the ABS/OLA sinusoidal model analysis to each sentence in the speech database, and organizing the resulting model parameters into a form suitable for the rest of the synthesis software.

Unit retrieval In the "Unit selection" module of Figure 4.1, a selection process similar to those described in Section 2.2.4 is performed, and pointers to these units in the inventory are stored in the "linguistic data object" related to the sentence. The synthesis module must at this point extract the needed sinusoidal

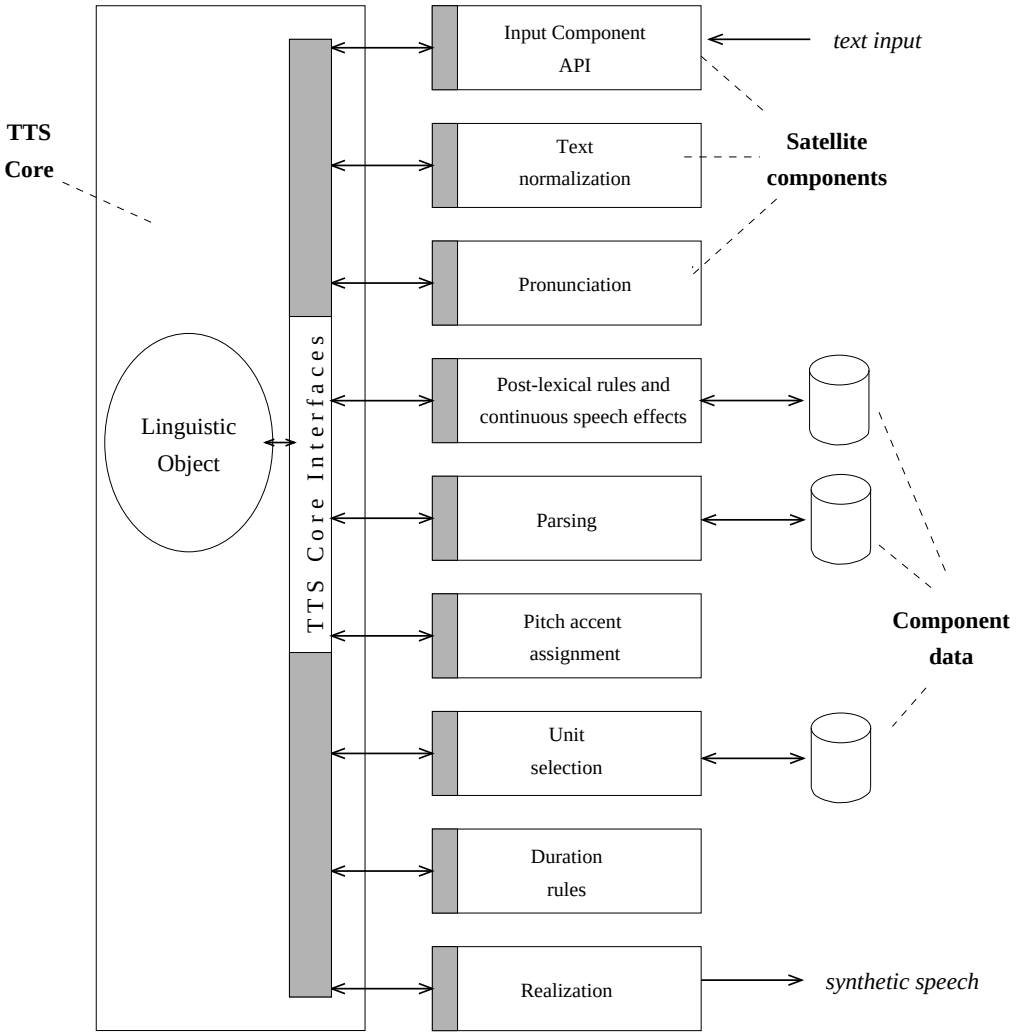

Figure 4.1: The LAUREATE II text-to-speech system (after [5])

.

model parameters for each unit from the inventory.

Unit normalization Since the units extracted from the speech inventory are generally not matched exactly in terms of energy, a normalization must be applied. This amplitude normalization is critical in reducing perceptible discontinuities in the short-term energy of the output speech.

Unit joining After normalization, the units are joined by simply adding the units to a data structure associated with the sinsoidal model for the entire synthetic utterance. The location of boundary frames must be noted so that realignment of pitch pulses and other smoothing algorithms can be applied.

Boundary smoothing Discontinuities in spectral shape, phase (i.e., pitch pulse location), energy, fundamental frequency, and other attributes will occur across the boundaries of joined segments. These arise from coarticulatory effects and other sources of variation in voice quality. Various smoothing algorithms are applied to remove or lessen the effects of these discontinuities.

Prosody modification In earlier modules shown in Figure 4.1, phonological models are used to derive a fundamental frequency contour and segmental duration information for the utterance to be synthesized. Based on the pitch and duration characteristics of the sinusoidally modeled segments, pitch modification and time-scale modification factors are derived for each frame of the sinusoidal model of the synthetic utterance.

Synthesis Given the sinusoidal model parameters and modification factors for each frame, overlap-add synthesis is performed to synthesize the desired utterance.

The rest of this section will describe in greater detail the normalization, smoothing, and prosody modification stages of the algorithm described above.

The ABS/OLA sinusoidal model analysis generates several quantities that represent each input signal frame, including (i) a set of quasi-harmonic sinusoidal param-

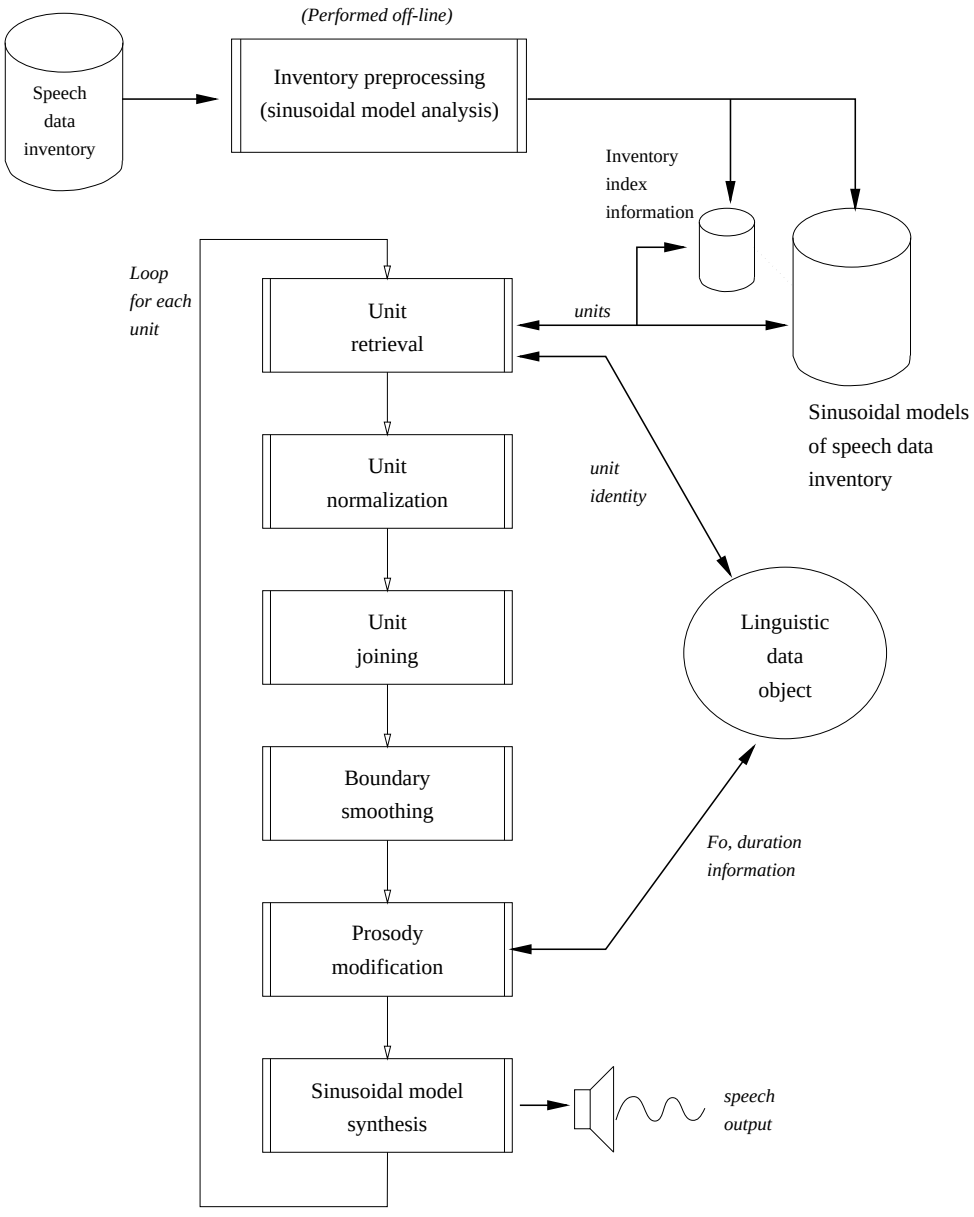

Figure 4.2: Sinusoidal model synthesis algorithm.

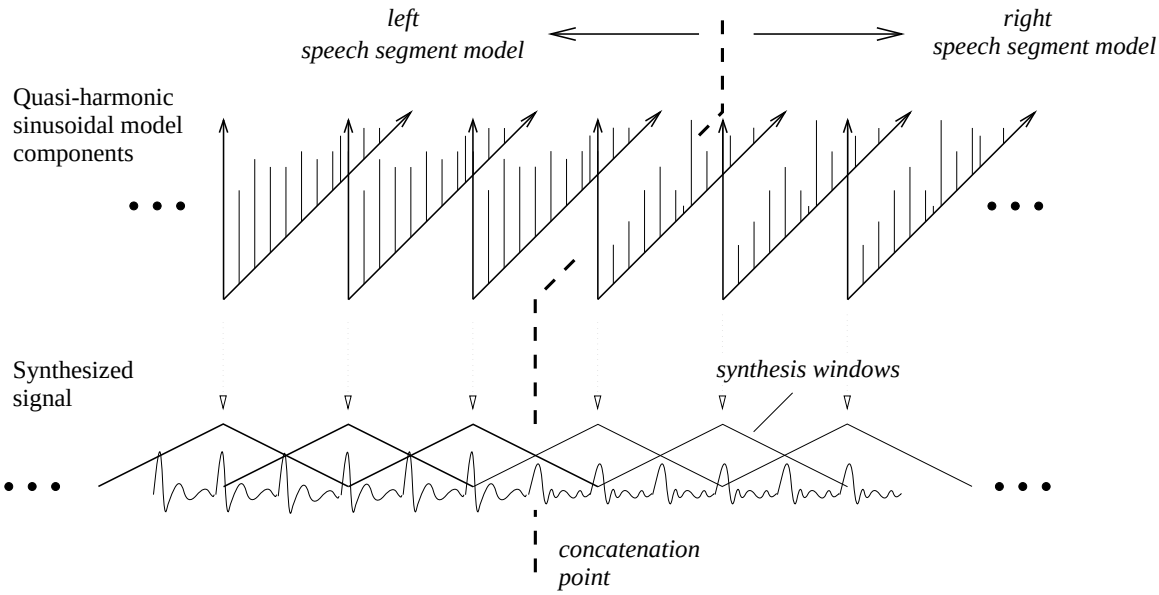

Figure 4.3: Concatenation of segments using sinusoidal model parameters.

eters for each frame (with an implied fundamental frequency estimate), (*ii*) a slowly time-varying gain envelope, and (*iii*) a spectral envelope for each frame. Disjoint modeled speech segments can be concatenated by simply stringing together these sets of model parameters and resynthesizing, as shown in Figure 4.3. However, since the joined segments are analyzed from disjoint utterances, substantial variations between the time- or frequency-domain characteristics of the signals may occur at the boundaries. These differences manifest themselves in the sinusoidal model parameters. Thus, the goal of the algorithms described here is to make discontinuities at the concatenation points inaudible by altering the sinusoidal model components in the neighborhood of the boundaries.

4.2 Unit Normalization

The units extracted from the inventory may vary in short-time signal energy, depending on the characteristics of the utterances from which they were extracted. This

variation gives the output speech a very stilted, unnatural rhythm. For this reason, it is necessary to normalize the energy of the units. However, it is not straightforward to adjust units that contain a mix of voiced and unvoiced speech and/or silence, since the RMS energy of such segments varies considerably depending on the character of the unit.

The approach taken here is to normalize only the *voiced* sections of the synthesized speech. In the analysis process, a global RMS energy for all voiced sounds in the inventory is found. Using this global target value, voiced sections of the unit are multiplied by a gain term that modifies the RMS value of each section to match the target. This can be performed by operating directly on the sinusoidal model parameters for the unit. The average energy (power) of a single synthesized frame of length N_s can be written as

$$\begin{aligned} E_{fr}^2 &= \frac{1}{N_s}\sum_{n=0}^{N_s-1}|s[n]|^2 \\ &= \frac{1}{N_s}\sum_{n=0}^{N_s-1}\left|\sigma[n]\sum_k a_k\cos(\omega_k n+\phi_k)\right|^2. \end{aligned} \tag{4.1}$$

Assuming that $\sigma[n]$ is relatively constant over the duration of the frame, Equation (4.1) can be reduced to

$$\begin{aligned} E_{fr}^2 &= \frac{\overline{\sigma}^2}{N_s}\sum_k a_k^2\sum_{n=0}^{N_s-1}|\cos(\omega_k n+\phi_k)|^2 \\ &\approx \frac{1}{2}\overline{\sigma}^2\sum_k a_k^2, \end{aligned} \tag{4.2}$$

where $\overline{\sigma}^2$ is the square of the average of $\sigma[n]$ over the frame. This energy estimate can be found for the voiced sections of the unit, and a suitable gain adjustment can be easily found. In practice, the applied gain function is smoothed to avoid abrupt discontinuities in the synthesized signal energy.

4.2.1 Voicing Decision Using Sinusoidal Parameters

In the energy normalization described above, only *voiced* segments are adjusted. This implies that a voiced/unvoiced decision must be incorporated into the analysis. Since several parameters of the sinusoidal model are already available as a byproduct of the analysis, it is reasonable to attempt to use these to make a voicing decision. For instance, the pitch detection algorithm of the ABS/OLA model (described in detail in [24]) typically defaults to a low frequency estimate below the speaker's normal pitch range when applied to unvoiced speech. Figure 4.4 shows fundamental frequency and gain contour plots for the phrase "sunshine shimmers," spoken by a female, with a plot of the two against each other to the right. It is clear from this plot (and even the ω_0 plot alone) that the voiced and unvoiced sections of the signal are quite discernible based on these values.

For this analyzed phrase, it is easy to choose thresholds of pitch or energy to discriminate between voiced and unvoiced frames, but it is difficult to choose *global* thresholds that will work for different talkers, sampling rates, etc. By taking advantage of the fact that this analysis is performed *off-line*, it is possible to choose automatically such thresholds for each utterance, and at the same time make the V/UV decision more robust (to pitch errors, etc.) by including more data in the V/UV classification.

This can be achieved by viewing the problem as a "nearest-neighbor" clustering of the data from each frame, where *feature vectors* consisting of ω_0 estimates, frame energy, and other data are defined. The centroids of the clusters can be found by employing the K-means (or LBG) algorithm commonly used in vector quantization [105], with $K = 2$ (a *voiced* class and an *unvoiced* class). This algorithm consists of two steps:

1. Each of the feature vectors is clustered with one of the K centroids to which it is "closest," as defined by a distance measure, $d(\mathbf{v}, \mathbf{c})$.

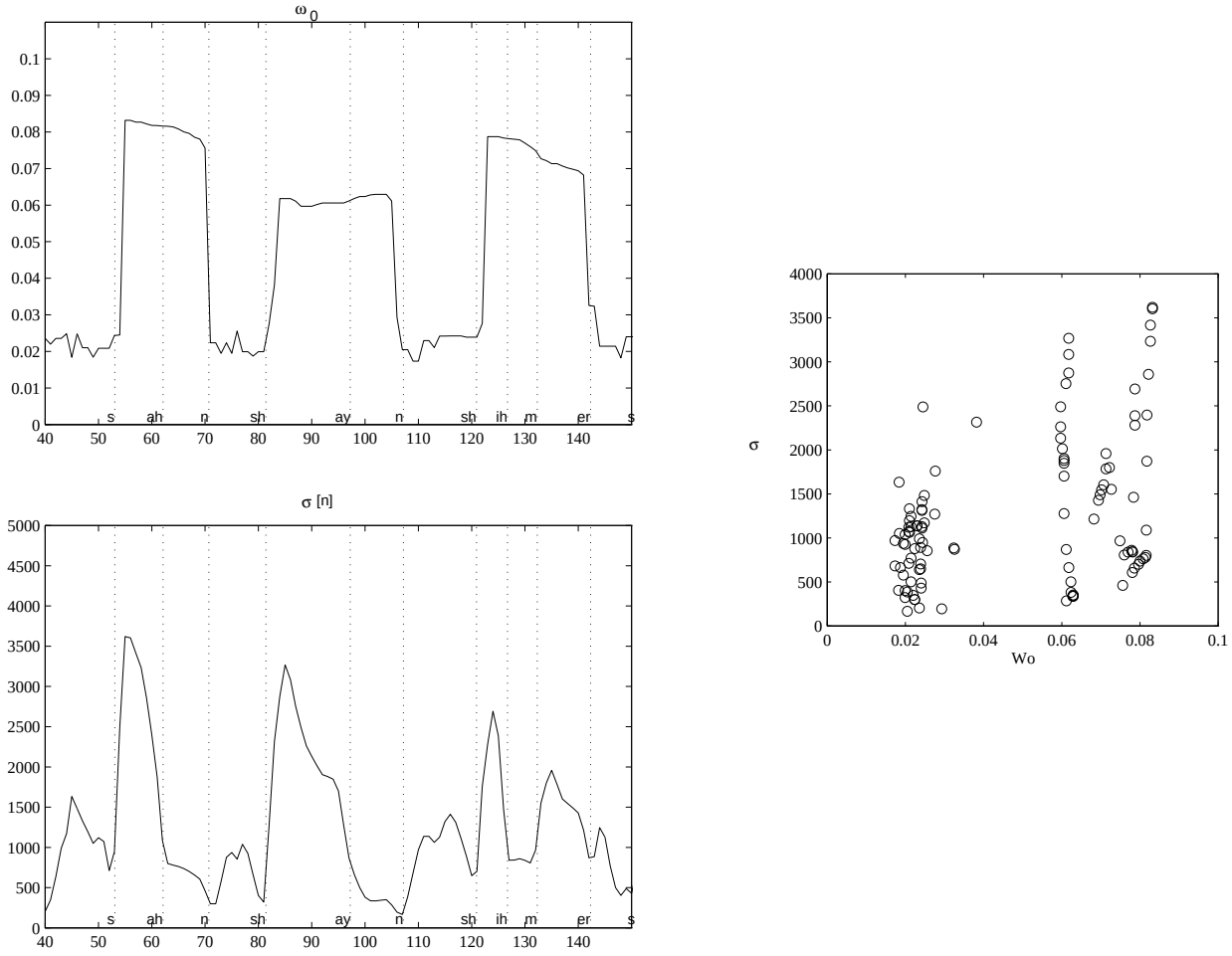

Figure 4.4: Left: Fundamental frequency and gain envelope plots for the phrase "...sunshine shimmers..." Right: a plot of these two quantities against each other. Note the clustering of data into voiced and unvoiced classes.

2. The centroids are updated by choosing as the new centroid the vector that minimizes the average distortion between it and the other vectors in the cluster (e.g., the mean if a Euclidean distance is used).

These steps are repeated until the clusters/centroids no longer change. In this case, the feature vector used in the voicing decision is

$$\mathbf{v} = \left[\begin{array}{ccc} \omega_0 & \overline{\sigma} & H_{SNR} \end{array} \right]^T,$$

where ω_0 is the fundamental frequency estimate for the frame, $\overline{\sigma}$ is the average of the time envelope $\sigma[n]$ over the frame, and H_{SNR} is the ratio of the signal energy to the energy in the difference between the "quasiharmonic" sinusoidal components in the model and the same components with frequencies forced to be harmonically related.[1] Since these quantities are not expressed in terms of units that have the same order of magnitude, a weighted distance measure is used:

$$d(\mathbf{v}, \mathbf{c}) = (\mathbf{v} - \mathbf{c})^T \mathbf{C}^{-1} (\mathbf{v} - \mathbf{c}), \tag{4.3}$$

where $\mathbf{C}$ is a diagonal matrix containing the variance of each element of $\mathbf{v}$ on its main diagonal.

This general framework for discriminating voiced and unvoiced frames has two benefits: (*i*) it eliminates the problem of manually setting thresholds that may or may not be valid across different talkers; and (*ii*) it adds robustness to the system, since several parameters are used in the V/UV discrimination. For instance, the inclusion of energy values in addition to fundamental frequency makes the method more robust to pitch estimation errors. The output of the voicing decision algorithm for an example phrase is shown in Figure 4.5.

[1] This is a measure of the degree to which the components are harmonically related to each other.

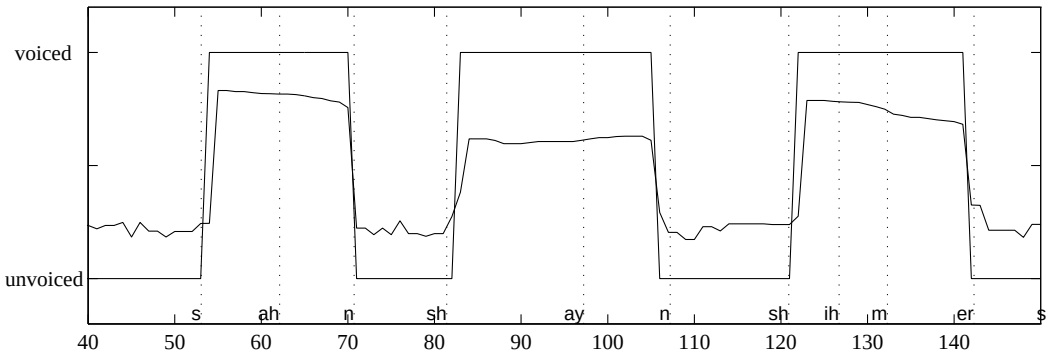

Figure 4.5: Voicing decision result, ω_0 contour, and phonetic annotation for the phrase "...sunshine shimmers..." using nearest neighbor clustering method.

4.3 Boundary Smoothing

4.3.1 Gain Smoothing

The unit normalization method described in Section 4.2 removes much of the energy variation between adjacent segments extracted from the inventory. However, since this normalization is performed on a fairly macroscopic level, perceptually significant short-time signal energy mismatches across concatenation boundaries remain.

An algorithm for smoothing the energy mismatch at the boundary of disjoint speech segments is described as follows:

1. The frame-by-frame energies of N_{smooth} frames (typically on the order of 50 ms) around the concatenation point are found using Equation (4.2).

2. The average frame energies for the left and right segments, given by E_L and E_R, respectively, are found.

3. A target value, E_{target}, for the energy at the concatenation point is determined. The average of E_L and E_R in the previous step is a reasonable assumption for such a target value.

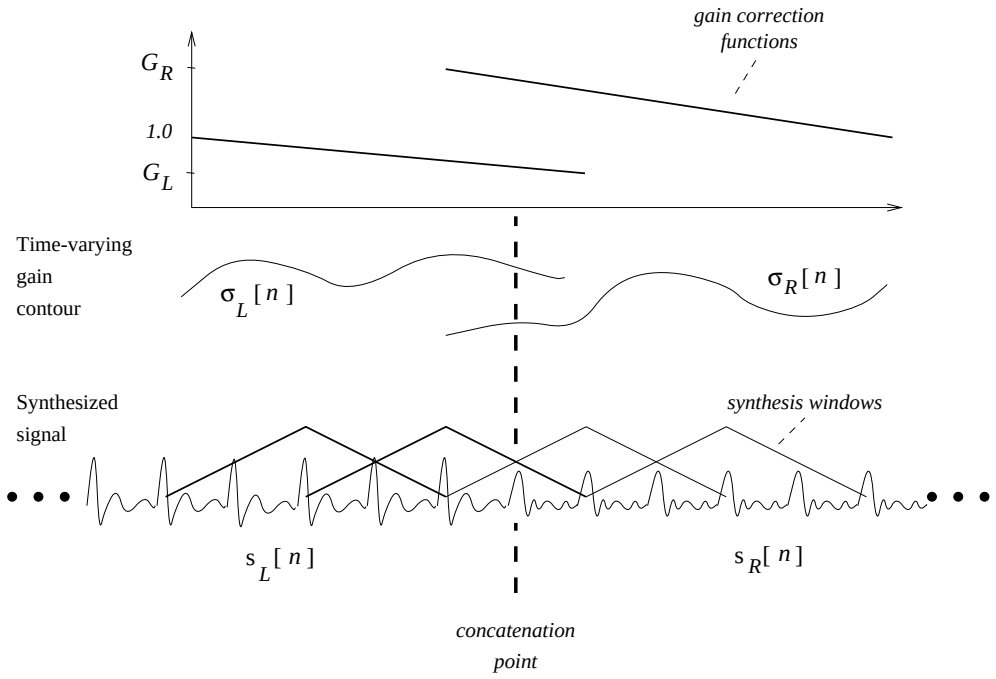

Figure 4.6: Short-time energy smoothing.

4. Gain corrections G_L and G_R are found by

$$G_L = \sqrt{\frac{E_{target}}{E_L}} \qquad\qquad G_R = \sqrt{\frac{E_{target}}{E_R}}. \tag{4.4}$$

5. Linear gain correction functions that interpolate from a value of 1 at the ends of the smoothing region to G_L and G_R at the respective concatenation points are created, as shown in Figure 4.6. These functions are then factored into the gain envelopes $\sigma_L[n]$ and $\sigma_R[n]$.

It should be noted that incorporating these gain smoothing functions into $\sigma_L[n]$ and $\sigma_R[n]$ requires a slight change in methodology. In the original model, the gain envelope $\sigma[n]$ is applied *after* the overlap-add of adjacent frames, i.e.,

$$x[n] = \sigma[n]\left(w_s[n]s_L[n] + (1 - w_s[n])s_R[n]\right), \tag{4.5}$$

where $w_s[n]$ is the window function, and $s_L[n]$ and $s_R[n]$ are the left and right synthetic contributions, respectively. However, both $\sigma_L[n]$ and $\sigma_R[n]$ should be included in the equation for the disjoint segments case. This can be achieved by splitting $\sigma[n]$ into 2 factors in the previous equation and then incorporating the left and right time-varying gain envelopes $\sigma_L[n]$ and $\sigma_R[n]$ as follows

$$x[n] = w_s[n]\sigma_L[n]s_L[n] + (1 - w_s[n])\sigma_R[n]s_R[n]. \tag{4.6}$$

This algorithm is very effective for smoothing energy mismatches in vowels and sustained consonants. However, the smoothing effect is undesirable for concatenations that occur in the neighborhood of transient portions of the signal (e.g., plosive phonemes like */k/*), since "burst" events are smoothed in time. This can be overcome by using phonetic label information available in the TTS system to vary N_{smooth} based on the phonetic context of the unit concatenation point.

4.3.2 Spectral Smoothing

Another source of perceptible discontinuity in concatenated signal segments is mismatch in spectral shape across boundaries. The segments being joined are somewhat

similar to each other in basic formant structure, due to matching of the phonetic context in unit selection. However, differences in spectral shape are often still present because of voice quality (e.g., spectral tilt) variation and other factors.

One input to the ABS/OLA pitch modification algorithm is a spectral envelope estimate represented as a set of low-order cepstral coefficients. This envelope is used to maintain formant locations and spectral shape while frequencies of sinusoids in the model are altered. An "excitation model" is computed by dividing the lth complex sinusoidal amplitude $a_l e^{j\phi_l}$ by the complex spectral envelope estimate $H(\omega)$ evaluated at the sinusoid frequency ω_l. These excitation sinusoids are then shifted in frequency by a factor β, and the spectral envelope is remultiplied by $H(\beta\omega_l)$ to obtain the pitch-shifted signal. This operation also provides a mechanism for smoothing spectral differences over the concatenation boundary, since a *different* spectral envelope may be reintroduced after pitch-shifting the excitation sinusoids.

Spectral differences across concatenation points are smoothed by adding weighted versions of the cepstral feature vector from one segment boundary to cepstral feature vectors from the other segment, and vice-versa, to compute a new set of cepstral feature vectors. Assuming that cepstral features for the left-side segment $\{..., \mathcal{L}_2, \mathcal{L}_1, \mathcal{L}_0\}$ and features for the right-side segment $\{\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, ...\}$ are to be concatenated as shown in Figure 4.7, smoothed cepstral features $\mathcal{L}_k^s$ for the left segment and $\mathcal{R}_k^s$ for the right segment are found by

$$\begin{aligned} \mathcal{L}_k^s &= w_k \mathcal{L}_k + (1 - w_k)\mathcal{R}_0 \\ \mathcal{R}_k^s &= w_k \mathcal{R}_k + (1 - w_k)\mathcal{L}_0, \end{aligned} \tag{4.7}$$

where

$$w_k = 0.5 + \frac{k}{2N_{smooth}}, \qquad k = 1, 2, ..., N_{smooth}$$

and where N_{smooth} frames to the left and right of the boundary are incorporated into the smoothing. It can be shown that this linear interpolation of cepstral features is equivalent to linear interpolation of log spectral magnitudes.

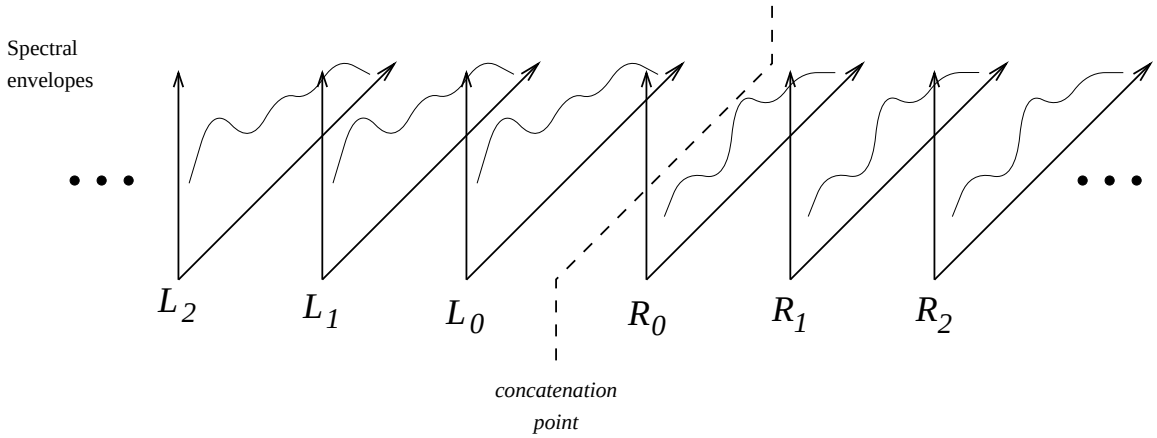

Figure 4.7: Cepstral envelope smoothing.

Once $\mathcal{L}_k^s$ and $\mathcal{R}_k^s$ are generated, they are input to the synthesis routine as an auxiliary set of cepstral feature vectors. Sets of spectral envelopes $H_k(\omega)$ and $H_k^s(\omega)$ are generated from $\{\mathcal{L}_k, \mathcal{R}_k\}$ and $\{\mathcal{L}_k^s, \mathcal{R}_k^s\}$, respectively. After the sinusoidal excitation components have been pitch-modified, the sinusoidal components are multiplied by $H_k^s(\omega)$ for each frame k to impart the spectral shape derived from the smoothed cepstral features.

4.4 Prosody Modification

One of the most important functions of the sinusoidal model in this synthesis method is as a means of performing *prosody modification* on the speech units.

It is assumed that higher levels of the system have provided the following inputs:

- a sequence of concatenated, sinusoidally-modeled speech units
- a desired pitch contour
- desired segmental durations (e.g., phone durations)

Given these inputs, a sequence of pitch modification factors $\{\beta_k\}$ for each frame can be found by simply computing the ratio of the desired fundamental frequency to the fundamental frequency of the concatenated unit. Similarly, time scale modification factors $\{\rho_k\}$ can be found by using the ratio of the desired duration of each phone (based on phonetic annotations in the inventory) to the unit duration.

The set of pitch modification factors generated in this manner will generally have discontinuities at the concatenated unit boundaries. However, when these pitch modification factors are applied to the sinusoidal model frames, the resulting pitch contour will be continuous across the boundaries.

4.5 Pitch Pulse Alignment

As mentioned in Section 2.2.4, proper alignment of adjacent frames is essential to producing high quality synthesized speech. If the pitch pulses of adjacent frames do not add coherently in the overlap-add process a "garbled" character is clearly perceivable in the resynthesized speech. There are two tasks involved in properly aligning the pitch pulses: (*i*) finding points of reference in the adjacent synthesized frames, and (*ii*) shifting frames to properly align pitch pulses, based on these points of reference.

The first of these requirements is fulfilled by the pitch pulse onset time estimation algorithm described in Section 3.4. This algorithm attempts to find the time at which a pitch pulse occurs in the analyzed frame. The second requirement, aligning the pitch pulse onset times, must be viewed differently depending on whether the frames to be aligned come from continuous speech or concatenated disjoint utterances. The time shift equation derived in [24] for continuous speech will be now be briefly reviewed in order to set up the problem for the concatenated speech case.

4.5.1 Continuous Speech Case

The diagrams in Figures 4.8 and 4.9 depict the locations of pitch pulses involved in the overlap-add synthesis of one frame. Analysis frames k and $k+1$ each contribute to the synthesized frame, which runs from 0 to $N_s - 1$. The pitch pulse onset times τ_k and τ_{k+1} describe the locations of the pitch pulses closest to the center of analysis frames k and $k+1$, respectively. In Figure 4.9, the time-scale modification factor ρ is incorporated by changing the length of the synthesis frame to ρN_s, while pitch modification factors β_k and β_{k+1} are applied to change the pitch of each of the analysis frame contributions. A time shift δ is also applied to each analysis frame. We assume that time shift δ_k has already been applied, and the goal is to find δ_{k+1} to shift the pitch pulses such that they coherently sum in the overlap-add process.

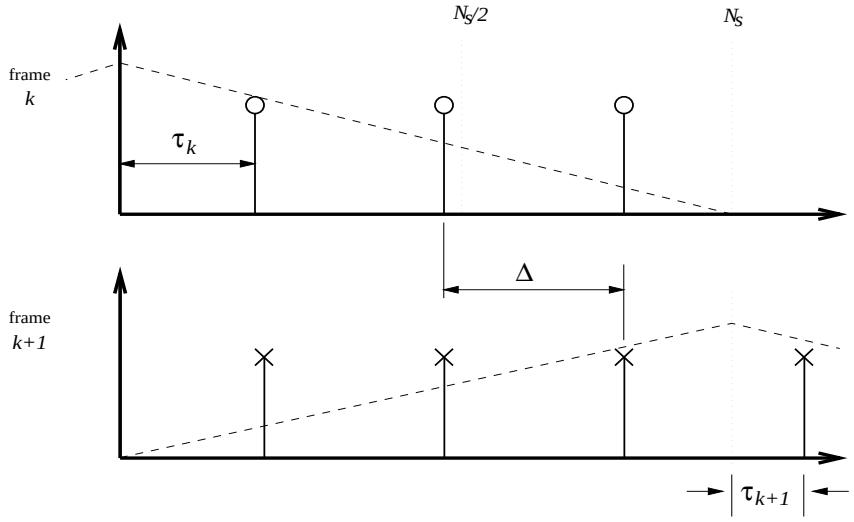

Figure 4.8: Pitch pulse alignment in absence of modification.

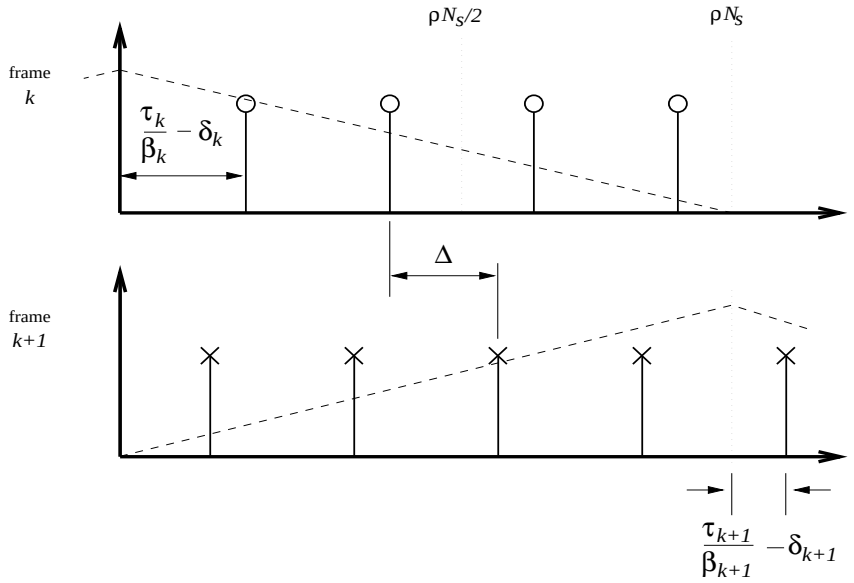

Figure 4.9: Pitch pulse alignment after modification.

From the schematic representation in Figure 4.8, an equation for the time location of the pitch pulses in the original, unmodified frames k and $k+1$ can be written as follows:

$$\begin{aligned} t_k[i] &= \tau_k + iT_0^k \\ t_{k+1}[i] &= \tau_{k+1} + iT_0^{k+1}, \end{aligned} \tag{4.8}$$

while the indices i that refer to the pitch pulses *closest to the center of the frame* are given by

$$\begin{aligned} \hat{\imath}_k &= \left\lfloor \frac{-\tau_k + \frac{N_s}{2}}{T_0^k} \right\rfloor \\ \hat{\imath}_{k+1} &= -\left\lfloor \frac{\tau_{k+1} + \frac{N_s}{2}}{T_0^{k+1}} \right\rfloor. \end{aligned} \tag{4.9}$$

Thus $t_k[\hat{\imath}_k]$ and $t_{k+1}[\hat{\imath}_{k+1}]$ are the time locations of the pitch pulses adjacent to the center of the synthesis frame.

Referring to Figure 4.9, equations for these same quantities can be found for the case where the time-scale/pitch modifications are applied:

$$t_k[i] = \frac{\tau_k}{\beta_k} - \delta_k + i\left(\frac{T_0^k}{\beta_k}\right) \tag{4.10}$$

$$t_{k+1}[i] = \frac{\tau_{k+1}}{\beta_{k+1}} - \delta_{k+1} + i\left(\frac{T_0^{k+1}}{\beta_{k+1}}\right) \tag{4.11}$$

$$\hat{\imath}_k = \left\lfloor \frac{-\tau_k + \beta_k(\delta_k + \frac{\rho N_s}{2})}{T_0^k} \right\rfloor \tag{4.12}$$

$$\hat{\imath}_{k+1} = -\left\lfloor \frac{\tau_{k+1} + \rho\beta_{k+1}\frac{N_s}{2}}{T_0^{k+1}} \right\rfloor. \tag{4.13}$$

Since the analysis frames k and $k+1$ were analyzed from continuous speech, we can assume that the pitch pulses will naturally line up coherently when $\beta = \rho = 1$. Thus the time difference Δ in Figure 4.8 will be approximately the average of the pitch periods T_0^k and T_0^{k+1}. To find δ_{k+1} after modification, then, it is reasonable to assume that this time shift should become $\hat{\Delta} = \Delta/\beta_{av}$, where β_{av} is the average of β_k and β_{k+1}.

Letting $\hat{\Delta} = \Delta/\beta_{av}$ and using Equations (4.10) through (4.13) to solve for δ_{k+1} results in the time shift equation [24]

$$\delta_{k+1} = \delta_k + (\rho_k - 1/\beta_{av})N_s + \frac{\beta_k - \beta_{k+1}}{2\beta_{av}}\left(\frac{\tau_k}{\beta_k} + \frac{\tau_{k+1}}{\beta_{k+1}}\right) - \frac{\hat{\imath}_k}{\beta_k}T_0^k + (i_k T_0^k - i_{k+1}T_0^{k+1})/\beta_{av}. \tag{4.14}$$

It can easily be verified that Equation (4.14) results in $\delta_{k+1} = \delta_k$ for the case $\rho = \beta_k = \beta_{k+1} = 1$. In other words, the frames will naturally line up correctly in the no-modification case since they are overlapped and added in a manner equivalent to that of the analysis method. This behavior is advantageous, since it implies that even if the pitch pulse onset time estimate is in error, the speech will not be significantly affected when the modification factors ρ, β_k, and β_{k+1} are close to 1.

4.5.2 Concatenation Case

The approach to finding δ_{k+1} given above is not valid, however, when finding the time shift necessary for the frame occurring just after a concatenation point, since even the condition $\rho = \beta_k = \beta_{k+1} = 1$ (no modification) does not assure that the adjacent frames will naturally overlap correctly. This is, again, due to the fact that the locations of pitch pulses (hence, onset times) of the adjacent frames across the boundary are essentially unrelated. In this case, a new derivation is necessary.

The goal of the frame alignment process is to shift frame $k+1$ such that the pitch pulses of the two frames line up and the waveforms add coherently. A reasonable way to achieve this is to force the time difference Δ between the pitch pulses adjacent to the frame center to be the average of the modified pitch periods in the two frames. It should be noted that this approach, unlike that above, makes no assumptions about the coherence of the pulses prior to modification. Typically, the modified pitch periods T_0^k/β_k and T_0^{k+1}/β_{k+1} will be approximately equal,[2] thus,

$$\Delta = \tilde{T}_o^{avg} = t_{k+1}[\hat{\imath}_{k+1}] + \rho N_s - t_k[\hat{\imath}_k], \tag{4.15}$$

[2]The desired fundamental frequency contour has already been imposed on the speech, as described in the previous section.

where

$$\tilde{T}_o^{avg} = \left(\frac{T_0^k}{\beta_k} + \frac{T_0^{k+1}}{\beta_{k+1}} \right) /2.$$

Substituting Equations (4.10) through (4.13) into Equation (4.15) and solving for δ_{k+1}, we obtain

$$\delta_{k+1} = \delta_k + \frac{\tau_{k+1}}{\beta_{k+1}} - \frac{\tau_k}{\beta_k} + \hat{\imath}_{k+1} \left(\frac{T_0^{k+1}}{\beta_{k+1}} \right) - \hat{\imath}_k \left(\frac{T_0^k}{\beta_k} \right) + \rho N_s - \tilde{T}_o^{avg}. \tag{4.16}$$

This gives an expression for the time shift of the sinusoidal components in frame $k+1$. This time shift (which need not be an integer) can be implemented directly in the frequency domain by modifying the sinusoid phases ϕ_i prior to resynthesis:

$$\tilde{\phi}_i = \phi_i + i\beta\omega_o\delta. \tag{4.17}$$

Reliance on pitch pulse onset time estimates It has been confirmed experimentally that applying Equation (4.16) does indeed result in coherent overlap of pitch pulses at the concatenation boundaries in speech synthesis. However, it should be noted that this method is *critically dependent* on the pitch pulse onset time estimates τ_k and τ_{k+1}. If either of these estimates is in error, the pitch pulses will not overlap correctly, distorting the output waveform. This underscores the importance of the onset estimation algorithm developed in Section 3.4. For modification of continuous speech, the onset time accuracy is less important, since poor frame overlap only occurs due to an onset time error when β is not close to 1.0, and only when the *difference* between two onset time estimates is not an integer multiple of a pitch pulse. However, in the concatenation case, onset errors nearly always result in audible distortion, since Equation (4.16) is completely reliant on the correct estimation of pitch pulse onset times to either side of the concatenation point.

In the TTS system developed for this research, the speech inventory was spoken by a fairly breathy female speaker. Because of the characteristics of her voice, the onset time algorithm made frequent errors. To circumvent this problem, auxiliary

information was used. *Pitchmarks* derived from an electroglottograph[3] were used as initial estimates of the pitch onset time. Instead of relying on the onset time estimator to search over the entire range $[-T_0/2, T_0/2]$, the pitchmark closest to each frame center was used to derive a rough estimate of the onset time, which was then refined using the estimator function described earlier. This rough estimate dramatically improved the performance of the onset estimator and the output speech quality.

4.6 Results

In order to evaluate the quality of speech produced by the sinusoidal model-based text-to-speech system, a listener evaluation of the algorithm in comparison to an implementation of the time-domain PSOLA synthesis method (Section 2.2.4) was performed.

4.6.1 Subjective comparison

A subjective comparison experiment was performed using the same 25 subjects as in the experiment described in Section 3.3.3. The subjects were asked to compare 30 pairs of sentences: one synthesized using PSOLA and one synthesized using the sinusoidal model synthesis method developed in this research. Synthesis units were selected from an inventory of continuous speech using a nonuniform unit selection procedure similar to the techniques described in Section 2.2.4, and the same units were used in each method. Intonation was generated by a phonological model, and was the same in both cases. No explicit duration model was used—the durations of units selected from the inventory were *not changed* in the final synthesized sentence. The phone-level durations were, however, the same for the PSOLA and sinusoidal model outputs.

[3]The electroglottograph produces a measurement of glottal activity that can be used to find instants of glottal closure.

The order of the sentence pairs and the order of the elements of each pair were selected randomly for each subject. The text items used as input to the synthesizers were three sets of 10 sentences taken from the "Harvard sentences" – the 30 sentences used are shown in Table 4.1. For each trial, the text representation of the sentence was displayed for the subject, and the two synthesized sentences were presented via headphones. The subject was then asked to select sentence "A" or "B" according to his or her preference "in terms of overall sound quality."

4.6.2 Comparison results

The results of the comparison test are shown in a bar graph in Figure 4.10, and histograms of by-sentence and by-subject results are shown in Figures 4.11(a) and 4.11(b). Across all subjects and test cases, the results were as follows:

Prefer sinusoidal model method	52 %
Prefer PSOLA method	48 %

This result fails to show a statistically significant preference for the sinusoidal model method. (Based on 750 binomial trials, the standard deviation is $\sigma = 13.7$, and the probability that this result is due to chance is 0.2 [106].) In informal questioning after the test, most subjects reported having difficulty in distinguishing between the sentences in the comparison.

By-subject and by-sentence breakdown The breakdown by sentence in Figure 4.11(a) shows that for one particular sentence (Sentence 4.7), the sinusoidal model was preferred by a statistically significant 92% of subjects ($p < 0.001$). Upon review of the PSOLA exemplar for this sentence, a "crackling" distortion was noted during one brief section of the file. Since this artifact was *not* present in the sinusoidal model output (which used the same source material), it was assumed that this was produced by a failure of the PSOLA implementation for this input. For two sentences (Sentences 3.6 and 4.3), a significant number of subjects preferred the PSOLA output.

Table 4.1: Sentences used in TTS system subjective comparison test.

1.1. *The birch canoe slid on the smooth planks.*
1.2. *Glue the sheet to the dark blue background.*
1.3. *It's easy to tell the depth of a well.*
1.4. *These days a chicken leg is a rare dish.*
1.5. *Rice is often served in round bowls.*
1.6. *The juice of lemons makes fine punch.*
1.7. *The box was thrown beside the parked truck.*
1.8. *The hogs were fed chopped corn and garbage.*
1.9. *Four hours of steady work faced us.*
1.10. *A large size in stockings is hard to sell.*

3.1. *The small pup gnawed a hole in the sock.*
3.2. *The fish twisted and turned on the bent hook.*
3.3. *Press the pants and sew a button on the vest.*
3.4. *The swan dive was far short of perfect.*
3.5. *The beauty of the view stunned the young boy.*
3.6. *The blue fish swam in the tank.*
3.7. *Her purse was full of useless trash.*
3.8. *The colt reared and threw the tall rider.*
3.9. *It snowed, rained, and hailed the same morning.*
3.10. *Read verse out loud for pleasure.*

4.1. *Hoist the load to your left shoulder.*
4.2. *Take the winding path to reach the lake.*
4.3. *Note closely the size of the gas tank.*
4.4. *Wipe the grease off his dirty face.*
4.5. *Mend the coat before you go out.*
4.6. *The wrist was badly strained and hung limp.*
4.7. *The stray cat gave birth to kittens.*
4.8. *The young girl gave no clear response.*
4.9. *The meal was cooked before the bell rang.*
4.10. *What joy there is in living.*

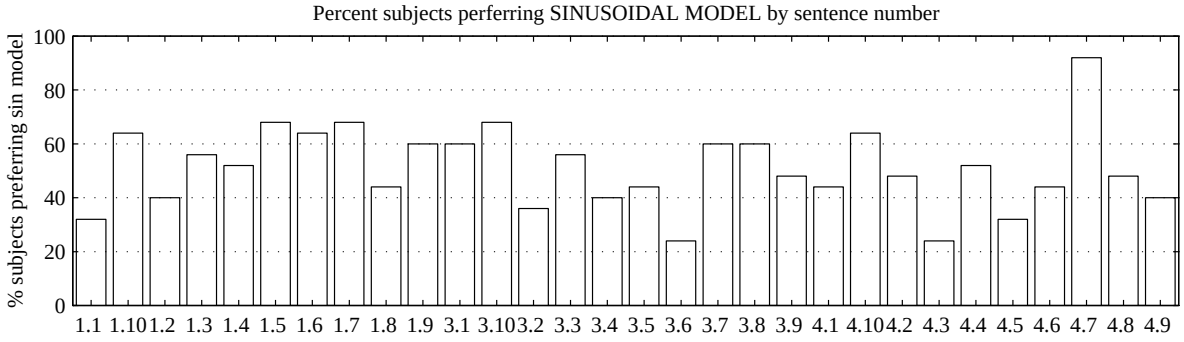

Figure 4.10: Subjective comparison responses by sentence. Bars indicate percentage of subjects preferring sinusoidal synthesis method. Sentence numbers correspond to those in Table 4.1.

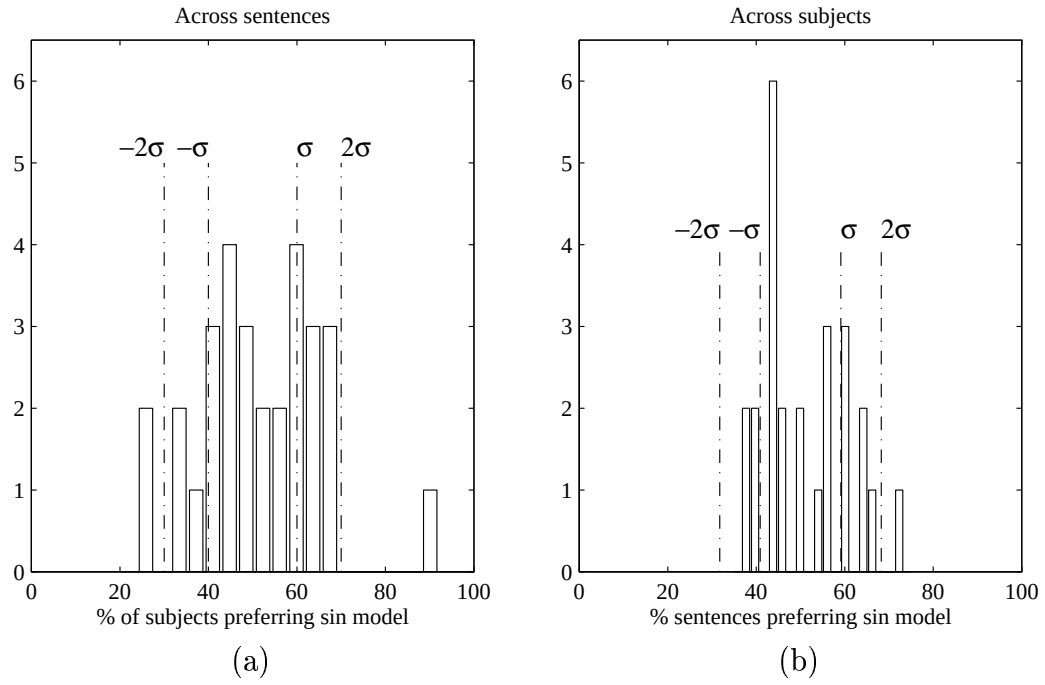

Figure 4.11: Histograms of listener preference for sinusoidal method (a) by sentence (b) by subject. Labeled dashed lines indicate significance levels for (a) 25 subjects ($\sigma = 10.0\%$) and (b) 30 test sentences ($\sigma = 9.1\%$).

In Sentence 3.6, a misalignment between frames occurred in the sinusoidal model synthesis, causing a phase discontinuity (pitch pulse misalignment) between two concatenated segments in one word. No obvious differences were audible in Sentence 4.3.

The breakdown by subject in Figure 4.11(a) shows that one particular subject preferred a significant number of sentences produced by the sinusoidal model method ($p < 0.006$). (It is interesting to note that this subject is a student researching high-fidelity audio coding methods, and he often performs critical listening assessments.) No subjects preferred the PSOLA method at a significance higher than ($p < 0.07$).

4.6.3 Discussion

The results of the subjective comparison indicate that, taken as a whole, the pool of 25 listeners did not prefer one algorithm over the other. This suggests that for this particular application, there is no clear advantage in speech quality to be gained by using one method or the other.

It should be emphasized that the algorithms were tested as part of a full TTS system with several interdependent modules. Several factors may help explain why the two synthesis methods produce results of similar subjective quality in this application:

- Upon review of the synthesized audio files, it was clear that the synthesis results for each sentence were either both very good or both very poor in terms of naturalness and overall quality. The nature of the distortions suggests that this quality difference was not due to characteristics of the prosody modification algorithms, but rather the set of concatenated units selected from the inventory during unit selection. The unit selection method was the same for both the sinusoidal model and PSOLA cases.

  In cases where the synthesis results were of very high quality and naturalness, the units selected were matched very well to each other in terms of spectral

shape, pitch, etc. Thus the responsibilities placed on the prosody modification algorithm in these cases were very slight (pitch modification factors close to 1.0.). As would be expected, the superiority of one algorithm over the other was thus less apparent, since both were essentially resynthesizing the original speech. Both the PSOLA and ABS/OLA models are capable of nearly-exact reconstruction of the original, unmodified speech.

In cases where both exemplars were of low quality and naturalness, this seemed to be due to a choice of synthesis units that produced barely-intelligible speech. Since 23 of the 25 listeners were naïve to subjective tests of speech quality, these listeners may have considered both exemplars to be of "equally poor" quality and simply chose one or the other by chance.

- Another factor that reduced the necessity for a sophisticated prosody modification technique was the fact that the duration model in the synthesis system was quite simplistic: the units extracted from the inventory were used with *no duration modification*. Exceptions to this were "silence" phonemes, for which durations were changed in the PSOLA implementation by simply adding or cutting zero samples.

  This fact may eliminate a *weakness* of PSOLA and a *strength* of ABS/OLA from consideration in the test. Empirical results of time-scale expansion and contraction experiments using the ABS/OLA model have shown this application to be one of its strong points. In contrast, time-scale expansion using PSOLA has been cited as one of its weaknesses [11]. In this method, the signal is time-expanded by repeating windowed waveform segments, which introduces unwanted periodicities into the output signal, perceivable as tonal noise. Subjective comparison experiments using a TTS system that employs an explicit duration model would perhaps show an increased preference for the ABS/OLA method.

In addition to speech quality, other factors should be considered in the comparison between the two algorithms:

Computational complexity It is possible to implement PSOLA using approximately 7 operations per sample. The ABS/OLA model, on the other hand, requires greater than 40 operations per sample [24]. Although reduced complexity is clearly one advantage of the PSOLA method, the synthesis requirements of the ABS/OLA model are still quite reasonable for most applications. This is especially true given the fact that a concatenation-based TTS system requires significant storage requirements: such a system will most often be implemented on a PC or workstation, rather than in a low-power embedded application.

Disk storage requirements In the implementation of the sinusoidal model system developed in this research, little effort was made to compress the inventory speech data to conserve disk space. Most model parameters were stored as full-precision floating point numbers. This resulted in an inventory that required roughly twice the storage of the uncompressed, 16 bit, 16 kHz–sampled speech used by the PSOLA algorithm. Since the sinusoidal model was originally developed as a speech coding algorithm [107, 108], it is reasonable to assume that these data could be compressed significantly with little or no perceivable loss of quality. However, exploration of such compression algorithms was beyond the scope of this research.

Algorithmic flexibility The results of the subjective comparison test do not conclusively demonstrate superiority of the sinusoidal model method. However, the added flexibility of the sinusoidal model over PSOLA-type methods is easy to demonstrate, since PSOLA offers no convenient control over signal properties other than pitch and time-scale evolution, while sinusoidal methods offer very precise control of time- and frequency-domain signal characteristics.

To show that the sinusoidal model is capable of high quality synthesis and

manipulation of subtle aspects of the speech signal, a further application of the method was explored—synthesis of *singing voice*. Although the synthesis method was not compared formally with other singing voice systems (of which there are very few), many unique capabilities of the model were explored in this application, as described in the next chapter.

CHAPTER 5

SYNTHESIS OF THE SINGING VOICE

In a joint research project between the Georgia Tech School of Music, the Center for Signal and Image Processing, and the Texas Instruments DSP Research and Development Center, the sinusoidal model text-to-speech synthesis framework described in Chapter 4 was extended to the synthesis of *singing voice* [109]. The work described in this chapter demonstrates that the sinusoidal model is capable of controlling subtle details of the voice, in addition to synthesizing intelligible speech.

5.1 Acoustic and Physiological Analysis of the Singing Voice

This section gives a brief survey of existing literature describing analysis of the human singing voice, with particular attention given to those properties that distinguish singing from the better-known properties of speech. Much of the existing literature deals with the singing style found in *opera*, since the operatic style departs most widely from ordinary speech, in comparison to other popular singing styles.

Singing voice and speech differ in these and other respects [110]:

1. In singing, the identity of an individual vowel is often secondary to its intonation.

2. Speech sounds are not sustained as in singing.

3. The pitch range of speech is much smaller and lower than that of singing. This means that in speech there are almost always an ample number of partials (i.e., harmonics) at which the spectrum is "sampled" to identify a given vowel, and the fundamental usually lies below the first formant.

4. Subglottal pressure and laryngeal positioning are different. In speech, this pressure is mainly used for loudness control, which is highly correlated with pitch. In singing, loudness and pitch must be controlled much more independently.

5. Studies have shown that as voice training is undertaken, a vocalist is capable of producing greater overall vocal intensity, and is able to control this intensity more precisely than an untrained vocalist is [111].

Formant shifting

One of the most important differences between speech and singing voice involves the tendency of singers (female singers, especially) to shift formant locations in response to pitch changes. In [112], Sundberg describes a study of formant modification in a female vocalist (reviewed along with other literature in [113, 110]).

In the female voice, especially soprano voices, the number of partials is few, and the spectral envelope of vocal tract resonances is "sampled" very sparsely. In fact, when a vowel with a low first formant is sung at a high pitch, the fundamental may often lie *well above* the first formant. In addition, variations in pitch can cause wide variations in the vowel quality and intelligibility, the overall loudness, and the physical effort needed to produce the sound. For this reason, vocalists very often modify the locations of formants to coincide with various partials in the source spectrum, especially the fundamental. This is accomplished by a learned method of varying the lip, tongue, and jaw positions when singing.

In [113, 112], it is shown that female formant locations in singing are much like those of speech when the fundamental frequency lies below the first formant.

However, when the fundamental rises above the first formant, trained singers shift the first formant to follow F_0. Another observed effect is that the second formant tends to decrease as F_0 increases, gravitating towards a "neutral" value of approximately 1500 Hz for all vowels. Because of these effects, quality differences between vowels tend to disappear as pitch is raised. Some vowels, such as those that involve lip rounding, are difficult for most vocalists to produce at high fundamental frequencies. Modification of formants has the following effects:

- Vocal tract resonance is optimally matched to the source by providing maximum energy throughput for the fundamental, and possibly by strengthening the source amplitude through optimal loading of the glottal oscillator.
- Strong sounds are produced with minimal muscular effort.
- Variation of tone quality and loudness are minimized when the formants follow pitch changes.
- Intelligibility of many sung vowels is increased in comparison to the unmodified formant case.

Men modify formants to some degree as well, but this is much less necessary, since the density of partials is much greater in the male source spectrum. Obviously, tenor voices are much more likely candidates than basses for use of this technique.

The "singer's formant"

One major motivation for the unique features of the singing voice is the desire of the solo singer to be differentiated from the musical accompaniment. Formant shifting in females plays a part in accomplishing this. A counterpart to this in male voices is the existence of the so called "singer's formant," a fixed resonance which appears in the 2500 to 3000 Hz range of the spectra of many male voices. This resonance is *independent* of fundamental frequency and vocal tract shape, and has been associated

with a resonant mode of the laryngeal collar lying just above the vocal folds [113, 110, 114]. Trained vocalists seem to exaggerate the impedance mismatch between this laryngeal tube and the lower pharynx, creating a strong resonance that is independent of the rest of the vocal tract shape. This resonance helps to differentiate the singer from the accompaniment. In a series of perception experiments [115], the subjective description of music passages as being more "colorful" was found to correlate well with the existence of the singer's formant.

Female vocalists are less likely to produce a singer's formant, but solo female opera vocalists do tend to produce more high frequency energy than vocalists trained in choir singing [116].

Vibrato

Vibrato is another attribute that increases timbral variety and allows the soloist to stand out from his or her orchestral accompaniment. The physiological mechanism of the pitch, amplitude, and timbral variation caused by vibrato is somewhat in debate. Pure frequency modulation of the glottal source waveform is capable of producing many of the observed effects of vibrato [113, 117]. As the source harmonics are swept across the vocal tract resonances, timbre and amplitude modulations as well as frequency modulation take place. Another plausible theory is that the the source spectrum remains relatively constant, and the vocal tract resonances are modulated, causing timbral changes that result in variation of *perceived* pitch [111]. Several sources (cited in [111]) also show that auditory feedback plays a vital role in the production of a controlled vibrato.

Glottal source characteristics

Differences in the source spectrum also distinguish the singing voice from speech. The upper harmonics of the singing voice spectrum become more prominent as a vocalist undergoes training, caused by changes in the glottal "closed quotient" [118]. This

spectral richness comes with little or no extra vocal effort, but rather more efficient use of the vocal cords [111].

Another important aspect of the vocal source is the variation of spectral tilt with loudness. Crescendo of the voice is accompanied by a leveling of the usual downward tilt of the spectrum [119, 120, 121] (in other words, an increase in energy of high frequency partials). This timbral difference is also apparent in the comparison of solo and choir-style singing [116]. In contrast, Klatt [102] has demonstrated that speech perceived as *breathy* (often associated with soft speech and singing) has a higher level of aspiration noise at higher frequencies than fully-phonated speech, and this is an important cue to naturalness in synthetic speech. Likewise, the existence of *pulsed noise* resulting from turbulence in the glottal airstream has been investigated by Cook in the development of the SPASM singing synthesis system section [41, 120].

Pulse *jitter* and *drift* are also important attributes of the glottal waveform. *Jitter* relates to variations in fundamental frequency at rates higher than the typical vibrato frequency of a singer, typically associated with involuntary random neural firing in the auditory feedback chain of the human pitch control mechanism. *Drift* refers to slow variations of pitch due to tuning, a somewhat more controllable attribute. Cook [120] studied the spectra of pitch jitter and drift signals computed from vocalists singing in various ranges of pitch and dynamic range. In general, he found that minimum jitter was found when singers tried to produce a signal with no vibrato, at high pitch, and at a low dynamic level. Maximum jitter occurred with production of vibrato, at low pitch, and at a high dynamic level.

5.2 Previous Approaches to Singing Voice Synthesis

This section gives a brief overview of existing literature dealing with computer synthesis of the singing voice. In comparison to speech synthesis, a relatively small amount

of work on the topic has been undertaken by researchers.

SPASM The "Singing Physical Articulatory Synthesis Model" (SPASM) was developed by Perry Cook at the Stanford Center for Computer Research in Music and Acoustics (CCRMA) [120, 122, 123]. SPASM is a graphical, interactive singing synthesis system that allows the user to manipulate the various articulators of the vocal tract and synthesize the resulting sounds in real-time on a NeXT workstation.

The underlying model is that of a "waveguide" or one-dimensional tube representation of the vocal tract. Coupling of the vocal tract to the nasal cavity is also modeled, as is radiation of sound through the throat wall. Various glottal pulse shapes and turbulence (noise) sources can be used to excite the vocal tract model. System identification tools are also provided to allow the user to obtain synthesis parameters automatically from a prerecorded sound file. The system is also capable of interpolating parameters in various domains, such as a "shape space" that correlates more closely with perceptual vowel attributes than linear interpolation of vocal tract model parameters. A software synthesis system called "Singer," which uses SPASM parameters to synthesize sounds from simple mouse and MIDI[1] controls, has also been created. This system has been applied to the synthesis of Ecclesiastical Latin (which has very rigid phonetic structure).

CHANT The CHANT system [119] was developed by Xavier Rodet and his colleagues at the *Institut de Recherche et Coordination Acoustique/Musique* (IRCAM) in Paris. This system, which has also been used in more general musical instrument synthesis, relies on direct time-domain synthesis of so called "formant waveforms." No direct control of the glottal source is available. A spectral tilt/loudness scaling algorithm is used to control high frequency partial amplitudes as described in the previous section. Since formant locations are explicitly controlled, the pitch-dependent

[1] The Musical Instrument Digital Interface (MIDI) standard is an industry-standard protocol for controlling electronic music instruments.

formant shifting of the female voice, as described above, can be conveniently implemented. Consonants are synthesized by trial and error stylization of formant contours that resemble formant transitions in natural singing.

Sundberg has also investigated formant synthesis methods for singing [121].

FM synthesis John Chowning of CCRMA has experimented with frequency modulation (FM) synthesis of the singing voice [124, 125]. This technique, which has been a popular method of music synthesis for over 20 years, relies on creating complex spectra with a small number of simple FM oscillators. Although this method offers a low-complexity method of producing rich spectra and musically interesting sounds, it has little or no correspondence to the acoustics of the voice, and seems difficult to control. The methods Chowning has devised resemble the "formant waveform" synthesis method of CHANT, where each formant waveform is created by an FM oscillator.

Other work Maher and Beauchamp have experimented with *wavetable synthesis* of singing voice [117]. Wavetable synthesis is a low-complexity method that involves filling a buffer with one period of a periodic waveform, and then cycling through this buffer to choose output samples. Pitch modification is made possible by cycling through the buffer at various rates. The waveform evolution is handled by updating samples of the buffer with new values as time evolves. Experiments were conducted to determine the perceptual necessity of the amplitude modulation which arises from frequency modulating a source that excites a fixed-formant filter—a more difficult effect to achieve in wavetable synthesis than in source/filter schemes. They found that this timbral/amplitude modulation *was* a critical component of naturalness, and should be included in the model.

5.3 LYRICOS: Singing Voice Synthesis Based on a Sinusoidal Model

5.3.1 System Overview

The system developed in this work, called LYRICOS, is shown in block diagram form in Figure 5.1. It uses a commercially-available, MIDI-based music composition software as a user interface. Using this interface, the user specifies a musical score and lyrics, as well as other musically-interesting control parameters such as vibrato and vocal intensity. This control information is stored in a standard MIDI file format that contains all information necessary to synthesize the vocal passage.

Based on this input MIDI file, the system selects synthesis model parameters from an inventory of voice data that has been analyzed offline using the sinusoidal model. As in the TTS system described previously, units are selected to represent segmental phonetic characteristics of the utterance, including coarticulation effects caused by the context of each phoneme. Algorithms described in Chapter 4 are then applied to the modeled segments to remove disfluencies in the signal at the joined boundaries. The sinusoidal model is then used to modify the pitch, duration, and spectral characteristics of the concatenated voice units as specified by the input musical score and control information. Finally, the output waveform is synthesized using OLA sinusoidal synthesis, as in the text-to-speech application.

This application of the synthesis framework is particularly interesting with respect to the work described in this thesis because it diminishes the importance of text analysis, prosody models, and other linguistically-motivated front-end elements of a TTS system. Musically, each syllable of the input text is associated with one or more notes, and the pitch and durations of each note specify much of the "prosody" of the utterance.

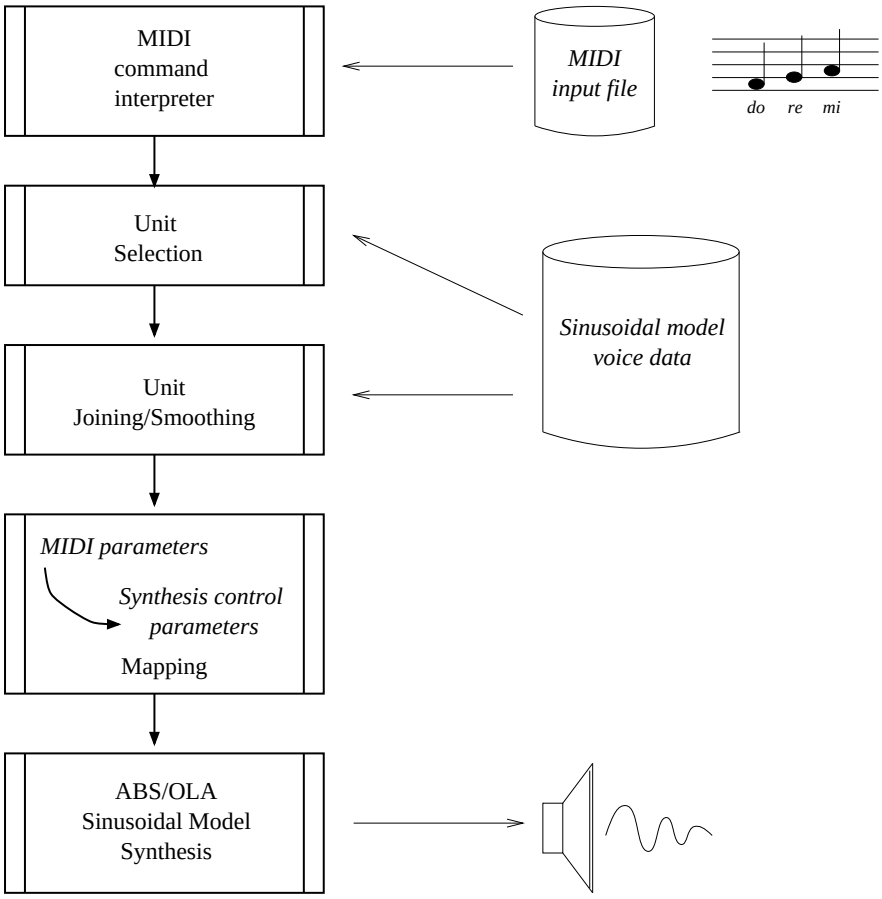

Figure 5.1: LYRICOS synthesis system block diagram.

5.3.2 Voice Corpus Collection

To provide an inventory of singing voice data for use by the synthesis algorithm, a script of nonsense words was designed, and a trained vocalist was recruited to sing the script. The design of the voice corpus was based on the following assumptions:

1. As the number of "phonetic contexts" represented in the inventory increases, better synthesis results will be obtained, since more accurate modeling of coarticulatory effects will occur.

2. For any given voiced speech segment, resynthesis with pitch modification factors close to 1.0 produces the most natural-sounding result. Thus, using an inventory containing vowels sung at several pitches will result in better-sounding synthesis, since units close to the desired pitch will usually be found.

3. Accurate modeling of transitions to and from *silence* contributes significantly to naturalness of the synthesized segments.

4. Consonant clusters are difficult to model using concatenation, due to coarticulation and rapidly varying signal characteristics.

5. In synthesis of singing, the *musical quality* of the voice is more critical than the *intelligibility* of the lyrics. Thus the fidelity of sustained vowels is more important than that of consonants.

6. Based on features such as place of articulation, voicing, nasality, etc., phonemes can be grouped into "classes." Phonemes in each class have somewhat similar coarticulatory effects on neighboring phonemes.

Assumptions 1 – 4 above suggest that the inventory should be made as large as possible, and incorporate units with consonant clusters, transitions to and from silence, and vowels sung at several pitches. This goal, however, must be balanced with the

facts of (a) time and expense of collecting and annotating the inventory and (b) fatigue of the vocalist. Assumptions 5 and 6 above make it possible to intelligently make the size of the inventory smaller, but with a minimal loss of quality in the resulting synthesis.

The script used to generate the inventory was designed as follows: For each vowel V, the set of all possible C_LV and VC_R units was created, where C_L and C_R represent "classes" of consonsants and consonant clusters located to the left and right of the vowel, respectively, as listed in Table 5.1. The actual phonemes selected from each class were chosen sequentially such that each consonant appeared roughly an equal number of times across all tokens. These C_LV and VC_R units were then paired arbitrarily to form C_LVC_R units. (Note that this represents only a subset of the possible C_LVC_R units, because only the pairwise possible choices are enumerated.)

These C_LVC_R units were then embedded in a "carrier" phonetic context to avoid word boundary effects. This carrier context consisted of the neutral vowel */ax/* (in ARPAbet notation), resulting in units of the form */ax/* C_LVC_R */ax/*. Two nonsense word tokens for each */ax/* C_LVC_R */ax/* unit were generated, and sung at high and low pitches within the vocalist's natural range.

Transitions of each phoneme to and from silence were generated as well. For vowels, these units were sung at both high and low pitches. The affixes — */s/* and — */z/* were also generated in the context of all valid phonemes. The complete list of nonsense words is given in Tables 5.2 and 5.3.

To generate the voice data, a classically-trained male vocalist sang 500 of the tokens described above. The recording studio time was obtained at minimal cost through the courtesy of RKM Studios in Atlanta. The singer was placed in an isolation booth, and was wearing headphones to communicate with others in a control booth. The recording took approximately 45 minutes, after initial set-up time. The voice data files were trimmed to remove silences, mistakes, etc. using Entropic `xwaves` and a simple file cutting program, resulting in approximately 10 minutes of contin-

Table 5.1: Classifications of consonants and clusters used in inventory design and unit selection. Clusters fall into different classes based on whether they appear before or after the vowel of interest (ARPAbet symbols used).

located to the LEFT of the vowel	
nasals	M, N, NG
whisper	HH
voiced fricatives	V, DH, Z, ZH, JH
unvoiced fricatives	F, TH, S, SH, CH
semivowels	R, L, W, Y, BR, DR, GR, PR, TR, KR, FR, THR, SHR, BL, GL, PL, KL, FL, SL, SHL
voiced stops	B, D, G
unvoiced stops	P, T, K, SP, ST, SK

located to the RIGHT of the vowel	
nasals	M, N, NG
whisper	HH
voiced fricatives	V, DH, Z, ZH, JH
unvoiced fricatives	F, TH, S, SH, CH, FR, THR, SHR, SL, FL, SHL
semivowels	R, L, W, Y
voiced stops	B, D, G, BR, DR, GR, BL, GL
unvoiced stops	P, T, K, PR, TR, KR, PL, KL

Table 5.2: Nonsense words sung in inventory data collection.

```
=== low pitch ===
1>   ax M IY M ax
2>   ax HH IY V ax
3>   ax V IY F ax
4>   ax F IY R ax
5>   ax R IY B ax
6>   ax B IY P ax
7>   ax P IY N ax
8>   ax BR IY DH ax
9>   ax N IH TH ax
10>   ax HH IH L ax
11>   ax DH IH D ax
12>   ax TH IH T ax
13>   ax L IH NG ax
14>   ax D IH Z ax
15>   ax T IH S ax
16>   ax DR IH W ax
17>   ax NG EY G ax
18>   ax HH EY K ax
19>   ax Z EY M ax
20>   ax S EY ZH ax
21>   ax W EY SH ax
22>   ax G EY Y ax
23>   ax K EY B ax
24>   ax GR EY SP ax
25>   ax M EH N ax
26>   ax HH EH JH ax
27>   ax ZH EH CH ax
28>   ax SH EH R ax
29>   ax Y EH D ax
30>   ax B EH ST ax
31>   ax SP EH NG ax
32>   ax PR EH V ax
33>   ax N AE F ax
34>   ax HH AE L ax
35>   ax JH AE G ax
36>   ax CH AE SK ax
37>   ax R AE M ax
38>   ax D AE DH ax
39>   ax ST AE TH ax
40>   ax TR AE W ax
41>   ax NG ER B ax
42>   ax HH ER P ax
43>   ax V ER N ax
44>   ax F ER Z ax
45>   ax L ER S ax
46>   ax G ER Y ax
47>   ax SK ER D ax
48>   ax KR ER T ax
49>   AH M ax NG AH
50>   AH HH ax ZH AH
51>   AH DH ax SH AH
52>   AH TH ax R AH
53>   AH W ax G AH
54>   AH B ax K AH
55>   AH P ax M AH
56>   AH FR ax JH AH
57>   ax N AH CH ax
58>   ax HH AH L ax
59>   ax Z AH B ax
60>   ax S AH SP ax
61>   ax Y AH N ax
62>   ax D AH V ax
63>   ax T AH F ax
64>   ax THR AH W ax
65>   ax NG UW D ax
66>   ax HH UW ST ax
67>   ax ZH UW NG ax
68>   ax SH UW DH ax
69>   ax R UW TH ax
70>   ax G UW Y ax
71>   ax K UW G ax
72>   ax SHR UW SK ax
73>   ax M UH M ax
74>   ax HH UH Z ax
75>   ax JH UH S ax
76>   ax CH UH R ax
77>   ax L UH B ax
78>   ax B UH P ax
79>   ax SP UH N ax
80>   ax BL UH ZH ax
81>   ax N OW SH ax
82>   ax HH OW L ax
83>   ax V OW D ax
84>   ax F OW T ax
85>   ax W OW NG ax
86>   ax D OW JH ax
87>   ax ST OW CH ax
88>   ax GL OW W ax
89>   ax NG AO G ax
90>   ax HH AO K ax
91>   ax DH AO M ax
92>   ax TH AO V ax
93>   ax Y AO F ax
94>   ax G AO Y ax
95>   ax SK AO B ax
96>   ax PL AO SP ax
97>   ax M AA N ax
98>   ax HH AA DH ax
99>   ax Z AA TH ax
100>   ax S AA R ax
101>   ax R AA D ax
102>   ax B AA ST ax
103>   ax P AA NG ax
104>   ax KL AA Z ax
105>   ax N AY S ax
106>   ax HH AY L ax
107>   ax ZH AY G ax
108>   ax SH AY SK ax
109>   ax L AY M ax
110>   ax D AY ZH ax
111>   ax T AY SH ax
112>   ax FL AY W ax
113>   ax NG AW B ax
114>   ax HH AW P ax
115>   ax JH AW N ax
116>   ax CH AW JH ax
117>   ax W AW CH ax
118>   ax G AW Y ax
119>   ax K AW D ax
120>   ax SL AW T ax
121>   ax M OY NG ax
122>   ax HH OY V ax
123>   ax V OY F ax
124>   ax F OY R ax
125>   ax Y OY G ax
126>   ax B OY K ax
127>   ax SP OY M ax
128>   ax SHL OY DH ax
=== high pitch ===
129>   ax N IY TH ax
130>   ax HH IY L ax
131>   ax DH IY B ax
132>   ax TH IY SP ax
133>   ax R IY N ax
134>   ax D IY Z ax
135>   ax ST IY S ax
136>   ax BR IY W ax
137>   ax NG IH D ax
138>   ax HH IH ST ax
139>   ax Z IH NG ax
140>   ax S IH ZH ax
141>   ax L IH SH ax
142>   ax G IH Y ax
143>   ax SK IH G ax
144>   ax DR IH SK ax
145>   ax M EY M ax
146>   ax HH EY JH ax
147>   ax ZH EY CH ax
148>   ax SH EY R ax
149>   ax W EY B ax
150>   ax B EY P ax
151>   ax P EY N ax
152>   ax GR EY V ax
153>   ax N EH F ax
154>   ax HH EH L ax
155>   ax JH EH D ax
156>   ax CH EH T ax
157>   ax Y EH NG ax
158>   ax D EH DH ax
159>   ax T EH TH ax
160>   ax PR EH W ax
161>   ax NG AE G ax
162>   ax HH AE K ax
163>   ax V AE M ax
164>   ax F AE Z ax
165>   ax R AE S ax
166>   ax G AE Y ax
167>   ax K AE B ax
168>   ax TR AE SP ax
169>   ax M ER N ax
170>   ax HH ER ZH ax
171>   ax DH ER SH ax
172>   ax TH ER R ax
173>   ax L ER D ax
174>   ax B ER ST ax
175>   ax SP ER NG ax
176>   ax KR ER JH ax
177>   AH N ax CH AH
178>   AH HH ax L AH
179>   AH Z ax G AH
180>   AH S ax SK AH
181>   AH W ax M AH
182>   AH D ax V AH
183>   AH ST ax F AH
184>   AH FR ax W AH
185>   ax NG AH B ax
186>   ax HH AH P ax
187>   ax ZH AH N ax
188>   ax SH AH DH ax
189>   ax Y AH TH ax
190>   ax G AH Y ax
191>   ax SK AH D ax
192>   ax THR AH T ax
193>   ax M UW NG ax
194>   ax HH UW Z ax
195>   ax JH UW S ax
196>   ax CH UW R ax
197>   ax R UW G ax
198>   ax B UW K ax
199>   ax P UW M ax
200>   ax SHR UW ZH ax
201>   ax N UH SH ax
202>   ax HH UH L ax
203>   ax V UH B ax
204>   ax F UH SP ax
205>   ax L UH N ax
206>   ax D UH JH ax
207>   ax T UH CH ax
208>   ax BL UH W ax
209>   ax NG OW D ax
210>   ax HH OW ST ax
211>   ax DH OW NG ax
212>   ax TH OW V ax
213>   ax W OW F ax
214>   ax G OW Y ax
215>   ax K OW G ax
216>   ax GL OW SK ax
217>   ax M AO M ax
218>   ax HH AO DH ax
219>   ax Z AO TH ax
220>   ax S AO R ax
221>   ax Y AO B ax
222>   ax B AO P ax
223>   ax SP AO N ax
224>   ax PL AO Z ax
225>   ax N AA S ax
226>   ax HH AA L ax
227>   ax ZH AA D ax
228>   ax SH AA T ax
229>   ax R AA NG ax
230>   ax D AA ZH ax
231>   ax ST AA SH ax
232>   ax KL AA W ax
233>   ax NG AY G ax
234>   ax HH AY K ax
235>   ax JH AY M ax
236>   ax CH AY JH ax
237>   ax L AY CH ax
238>   ax G AY Y ax
239>   ax SK AY B ax
240>   ax FL AY SP ax
241>   ax M AW N ax
242>   ax HH AW V ax
243>   ax V AW F ax
244>   ax F AW R ax
245>   ax W AW D ax
246>   ax B AW ST ax
247>   ax P AW NG ax
248>   ax SL AW DH ax
249>   ax N OY TH ax
250>   ax HH OY L ax
```

Table 5.3: Nonsense words sung in inventory data collection (cont'd).

```
251>   ax DH OY G ax
252>   ax TH OY SK ax
253>   ax Y OY M ax
254>   ax D OY Z ax
255>   ax T OY S ax
256>   ax SHL OY W ax
=== low pitch ===
257>   ## IY F ax
258>   ## IH F ax
259>   ## EY F ax
260>   ## EH F ax
261>   ## AE F ax
262>   ## ER F ax
263>   ## ax F AH
264>   ## AH F ax
265>   ## UW F ax
266>   ## UH F ax
267>   ## OW F ax
268>   ## AO F ax
269>   ## AA F ax
270>   ## AY F ax
271>   ## AW F ax
272>   ## OY F ax
=== high pitch ===
273>   ## IY F ax
274>   ## IH F ax
275>   ## EY F ax
276>   ## EH F ax
277>   ## AE F ax
278>   ## ER F ax
279>   ## ax F AH
280>   ## AH F ax
281>   ## UW F ax
282>   ## UH F ax
283>   ## OW F ax
284>   ## AO F ax
285>   ## AA F ax
286>   ## AY F ax
287>   ## AW F ax
288>   ## OY F ax
=== high pitch  ===
289>   ## M AE F ax
290>   ## N AE F ax
291>   ## NG AE F ax
292>   ## HH AE F ax
293>   ## V AE F ax
294>   ## DH AE F ax
295>   ## Z AE F ax
296>   ## ZH AE F ax
297>   ## JH AE F ax
298>   ## F AE F ax
299>   ## TH AE F ax
300>   ## S AE F ax
301>   ## SH AE F ax
302>   ## CH AE F ax
303>   ## R AE F ax
304>   ## L AE F ax
305>   ## W AE F ax
306>   ## Y AE F ax
307>   ## B AE F ax
308>   ## D AE F ax
309>   ## G AE F ax
310>   ## P AE F ax
311>   ## T AE F ax
312>   ## K AE F ax
313>   ## SP AE F ax
314>   ## ST AE F ax
315>   ## SK AE F ax
316>   ## BR AE F ax
317>   ## DR AE F ax
318>   ## GR AE F ax
319>   ## PR AE F ax
320>   ## TR AE F ax
321>   ## KR AE F ax
322>   ## FR AE F ax
323>   ## THR AE F ax
324>   ## SHR AE F ax
325>   ## BL AE F ax
326>   ## GL AE F ax
327>   ## PL AE F ax
328>   ## KL AE F ax
329>   ## FL AE F ax
330>   ## SL AE F ax
331>   ## SHL AE F ax
== high pitch  ===
332>   ax F AE M ##
333>   ax F AE N ##
334>   ax F AE NG ##
335>   ax F AE V ##
336>   ax F AE DH ##
337>   ax F AE ZH ##
338>   ax F AE JH ##
339>   ax F AE F ##
340>   ax F AE TH ##
341>   ax F AE SH ##
342>   ax F AE CH ##
343>   ax F AE R ##
344>   ax F AE L ##
345>   ax F AE W ##
346>   ax F AE Y ##
347>   ax F AE B ##
348>   ax F AE D ##
349>   ax F AE G ##
350>   ax F AE P ##
351>   ax F AE T ##
352>   ax F AE K ##
353>   ax F AE SP ##
354>   ax F AE ST ##
355>   ax F AE SK ##
=== low pitch ===
356>   ax F IY ##
357>   ax F IH ##
358>   ax F EY ##
359>   ax F EH ##
360>   ax F AE ##
361>   ax F ER ##
362>   AH F ax ##
363>   ax F AH ##
364>   ax F UW ##
365>   ax F UH ##
366>   ax F OW ##
367>   ax F AO ##
368>   ax F AA ##
369>   ax F AY ##
370>   ax F AW ##
371>   ax F OY ##
372>   ax F EL ##
373>   ax F EN ##
374>   ax F EM ##
=== high pitch ===
375>   ax F IY ##
376>   ax F IH ##
377>   ax F EY ##
378>   ax F EH ##
379>   ax F AE ##
380>   ax F ER ##
381>   AH F ax ##
382>   ax F AH ##
383>   ax F UW ##
384>   ax F UH ##
385>   ax F OW ##
386>   ax F AO ##
387>   ax F AA ##
388>   ax F AY ##
389>   ax F AW ##
390>   ax F OY ##
391>   ax F EL ##
392>   ax F EN ##
393>   ax F EM ##
== high pitch  ===
394>   ax F AE F S ##
395>   ax F AE TH S ##
396>   ax F AE R S ##
397>   ax F AE L S ##
398>   ax F AE W S ##
399>   ax F AE B S ##
400>   ax F AE D S ##
401>   ax F AE G S ##
402>   ax F AE P S ##
403>   ax F AE T S ##
404>   ax F AE K S ##
405>   ax F AE SP S ##
406>   ax F AE ST S ##
407>   ax F AE SK S ##
=== low pitch ===
408>   ax F IY S ##
409>   ax F IH S ##
410>   ax F EY S ##
411>   ax F EH S ##
412>   ax F AE S ##
413>   ax F ER S ##
414>   AH F ax S ##
415>   ax F AH S ##
416>   ax F UW S ##
417>   ax F UH S ##
418>   ax F OW S ##
419>   ax F AO S ##
420>   ax F AA S ##
421>   ax F AY S ##
422>   ax F AW S ##
423>   ax F OY S ##
=== high pitch ===
424>   ax F IY S ##
425>   ax F IH S ##
426>   ax F EY S ##
427>   ax F EH S ##
428>   ax F AE S ##
429>   ax F ER S ##
430>   AH F ax S ##
431>   ax F AH S ##
432>   ax F UW S ##
433>   ax F UH S ##
434>   ax F OW S ##
435>   ax F AO S ##
436>   ax F AA S ##
437>   ax F AY S ##
438>   ax F AW S ##
439>   ax F OY S ##
=== high pitch  ===
440>   ax F AE M Z ##
441>   ax F AE N Z ##
442>   ax F AE NG Z ##
443>   ax F AE V Z ##
444>   ax F AE DH Z ##
445>   ax F AE ZH Z ##
446>   ax F AE JH Z ##
447>   ax F AE F Z ##
448>   ax F AE TH Z ##
449>   ax F AE SH Z ##
450>   ax F AE CH Z ##
451>   ax F AE R Z ##
452>   ax F AE L Z ##
453>   ax F AE W Z ##
454>   ax F AE B Z ##
455>   ax F AE D Z ##
456>   ax F AE G Z ##
457>   ax F AE P Z ##
458>   ax F AE T Z ##
459>   ax F AE K Z ##
460>   ax F AE SP Z ##
461>   ax F AE ST Z ##
462>   ax F AE SK Z ##
=== low pitch ===
463>   ax F IY Z ##
464>   ax F IH Z ##
465>   ax F EY Z ##
466>   ax F EH Z ##
467>   ax F AE Z ##
468>   ax F ER Z ##
469>   AH F ax Z ##
470>   ax F AH Z ##
471>   ax F UW Z ##
472>   ax F UH Z ##
473>   ax F OW Z ##
474>   ax F AO Z ##
475>   ax F AA Z ##
476>   ax F AY Z ##
477>   ax F AW Z ##
478>   ax F OY Z ##
479>   ax F EL Z ##
480>   ax F EN Z ##
481>   ax F EM Z ##
=== high pitch ===
482>   ax F IY Z ##
483>   ax F IH Z ##
484>   ax F EY Z ##
485>   ax F EH Z ##
486>   ax F AE Z ##
487>   ax F ER Z ##
488>   AH F ax Z ##
489>   ax F AH Z ##
490>   ax F UW Z ##
491>   ax F UH Z ##
492>   ax F OW Z ##
493>   ax F AO Z ##
494>   ax F AA Z ##
495>   ax F AY Z ##
496>   ax F AW Z ##
497>   ax F OY Z ##
498>   ax F EL Z ##
499>   ax F EN Z ##
500>   ax F EM Z ##
```

uous singing data. This material was then phonetically annotated, which required approximately 40 hours for a relatively inexperienced labeler.

5.3.3 Non-Uniform Unit Selection

Philosophy

In order to take maximum advantage of the phonetic contexts in the recorded voice data, a unit selection method was designed, based on the following principles.

- Instead of using fixed-size units that are prepared from the recorded voice data ahead of time, units should be extracted from the phonetically-annotated inventory *during synthesis*. This permits the use of variable-size units and better preserves context information contained in the inventory.
- If desired strings of more than one phoneme are found in the inventory, these should be used directly in synthesis, instead of being composed from concatenation of several smaller units. The number of concatentations should be made as small as possible without compromising the phonetic content.
- Selection of the "optimal" unit at runtime should be performed with minimal computational requirements.

Context decision tree/variable size unit selection

Although it is possible to formulate unit selection as a dynamic programming problem with "unit costs" and "concatenation costs," and find an optimal path through the lattice of all possible units in the inventory [80], the approach taken here is simpler. The unit selection procedure allows variable-size units to be extracted from the inventory to represent the phonetic sequence specified by the input. Since longer units generally result in improved speech quality at the output, the method places a priority on finding longer units that match the desired phonetic context.

For a given phoneme P in the input phonetic string and its left and right neighbors, P_L and P_R, the selection algorithm attempts to find P in a context most closely matched to $P_L\,P\,P_R$. When exact context matches are found, the algorithm extracts the matching adjacent phoneme(s) as well, to preserve the transition between these phonemes. Thus, each extracted unit consists of an instance of the target phoneme and one or both of its neighboring phonemes (i.e., it extracts a *monophone, diphone, or triphone*). Figure 5.2 shows a catalog of all possible combinations of monophones, diphones, and triphones, including class match properties, ordered by their preference for synthesis.

In addition to searching for phonemes in an exact phonemic context, however, the system also is capable of widening its search to find phonemes that may have a context *similar*, but not identical, to the desired context. For example, if the algorithm is searching for */ae/* in the context */d/-/ae/-/d/*, but this triphone cannot be found in the inventory, the monophone */ae/* taken from the context */b/-/ae/-/b/* can be used instead, since */b/* and */d/* have a similar effect on the neighboring vowel. The notation of Figure 5.2 indicates the resulting unit output, along with a description of the context rules satisfied by the units. In the notation of this figure, $x_L P_1 x_R$ indicates a phoneme with an exact triphone context match (as */d/-/ae/-/d/* would be for the case described above). The label $c_L P_1 c_R$ indicates a match of phoneme *class* on the left and right, as for */b/-/ae/-/b/* above. Labels with the symbol P_2 indicate a second unit is used to provide the final output phonemic unit. For example, if */b/-/ae/-/k/* and */k/-/ae/-/b/* can be found, the two */ae/* monophones can be joined to produce an */ae/* with the proper class context match on either side.

In order to find the unit with the most appropriate available context, a binary decision tree was used (shown in Figure 5.3). Nodes in this tree indicate a test defined by the context label next to each node. The right branch out of each node indicates a "no" response; downward branches indicate "yes." Terminal node numbers correspond to the outputs defined in Figure 5.2. Diamonds on the node branches indicate

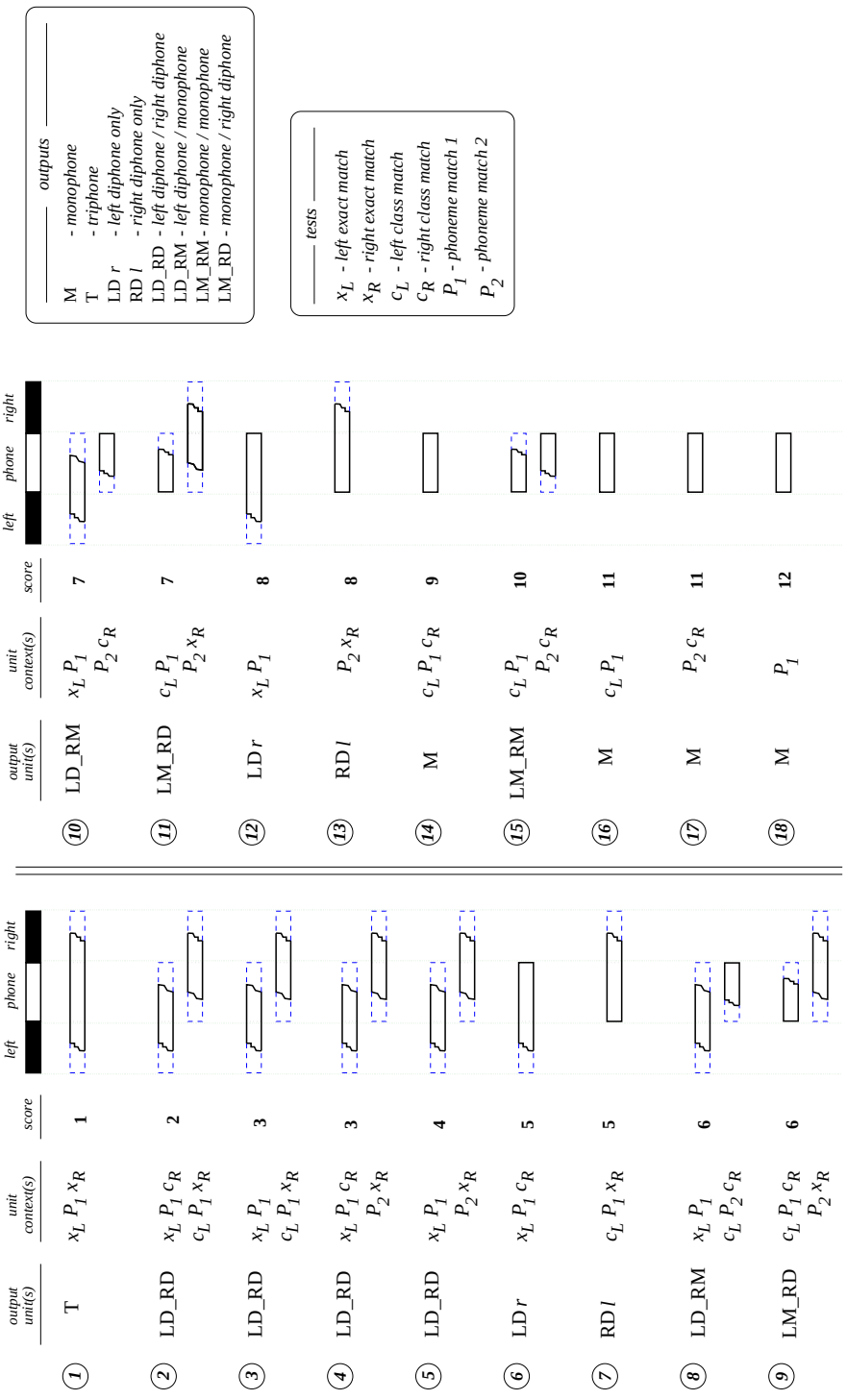

Figure 5.2: Catalog of variable-size units available to represent a given phoneme.

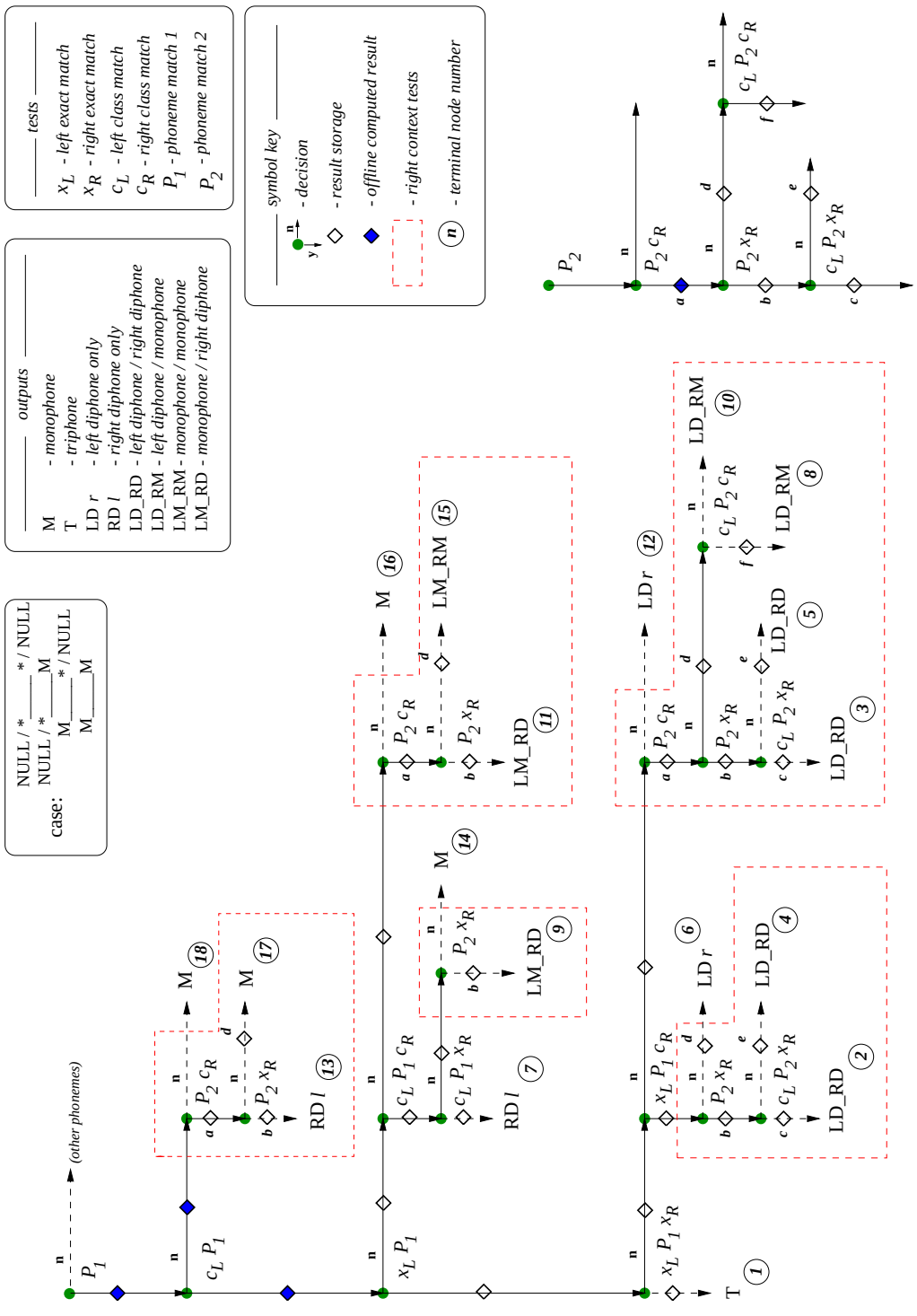

Figure 5.3: Decision tree for context matching.

storage arrays that must be maintained during the processing of each phoneme. Regions enclosed in dashed lines refer to a second search for phonemes with a desired *right* context to supplement the first choice (the case described at the end of the previous paragraph). The smaller tree at the bottom right of the diagram describes all tests that must be conducted to find this second phoneme. The storage locations here are computed once and used directly in the dashed boxes. To save computation at runtime, the first few tests in the decision tree are performed offline and stored in a file. The results of the precomputed branches are represented by filled diamonds on the branches.

After the decision tree is evaluated for every instance of the target phoneme, the (nonempty) output node representing the lowest score in Figure 5.2 is selected. All units residing in this output node are then ranked according to their closeness to the desired pitch (as input in the MIDI file). A rough pitch estimate is included in the phonetic labeling process for this purpose. Thus the unit with the best phonetic context match and closest pitch to the desired unit is selected.

The decision to develop this method instead of implementing the dynamic programming method [80] was based on the following rationale: Because the inventory was constructed with emphasis on providing a good coverage of the necessary vowel contexts, "target costs" of phonemes in dynamic programming should be biased such that units representing vowels will be chosen more or less independently of each other. Thus a slightly suboptimal, but equally effective, method is to choose units for *all vowels* first, then go back to choose the remaining units, leaving the already-specified units unchanged. Given this, three scenarios must be addressed to "fill in the blanks":

1. *Diphones or triphones have been specified on* both *sides of the phoneme of interest.*

   Result: a complete specification of the desired phoneme has already been found, and *no units* are necessary.

2. *A diphone or triphone has been specified on the* left *side of the phoneme of*

interest.

Result: The pruned decision tree in Figure 5.4 is used to specify the remaining portion of the phoneme.

3. *A diphone or triphone has been specified on the* right *side of the phoneme of interest.*

   Result: The pruned decision tree in Figure 5.5 is used to specify the remaining portion of the phoneme.

If no units have been specified on either side, or if *monophones only* have been specified, then the general decision tree in Figure 5.3 can be used.

Concatenation of units

Once the units necessary to cover the entire phonetic sequence have been specified, concatenation of the units can take place. Each pair of units is joined by either a cutting/smoothing operation or an "abutting" of one unit to another. The type of unit-to-unit transition uniquely specifies whether units are joined (cut and smoothed) or abutted. Figure 5.6 shows a "transition matrix" of possible unit–unit sequences and their proper join method. It should be noted that the NULL unit has zero length – it serves as a mechanism for altering the type of join in certain situations.

5.3.4 Musical Control Parameters

Pitch variation

Since the prosody modification step in the sinusoidal synthesis algorithm transforms the pitch of every frame to match a target, the result is a signal that does not exhibit the natural pitch fluctuations of the human voice.

In [102], a simple equation for "quasirandom" pitch fluctuations in speech is proposed:

$$\Delta F_0 = \frac{F_0}{100}\left(\sin(12.7\pi t) + \sin(7.1\pi t) + \sin(4.7\pi t)\right)/3. \tag{5.1}$$

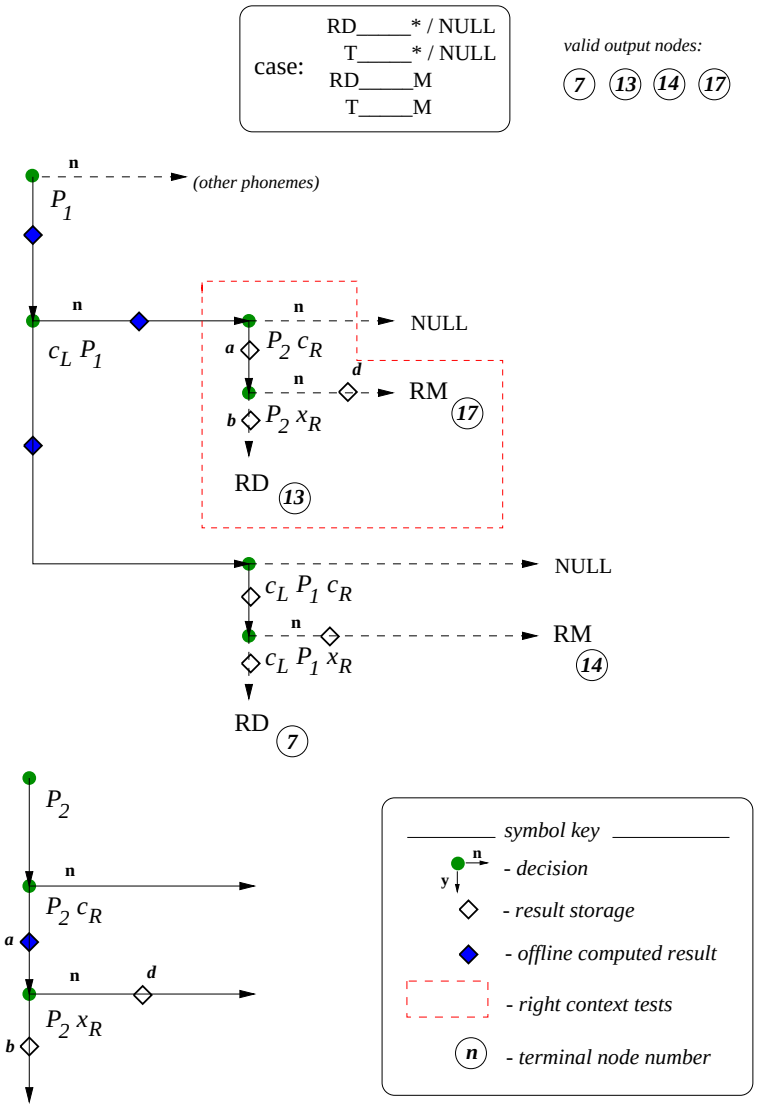

Figure 5.4: Decision tree for phonemes preceded by an already-chosen diphone or triphone.

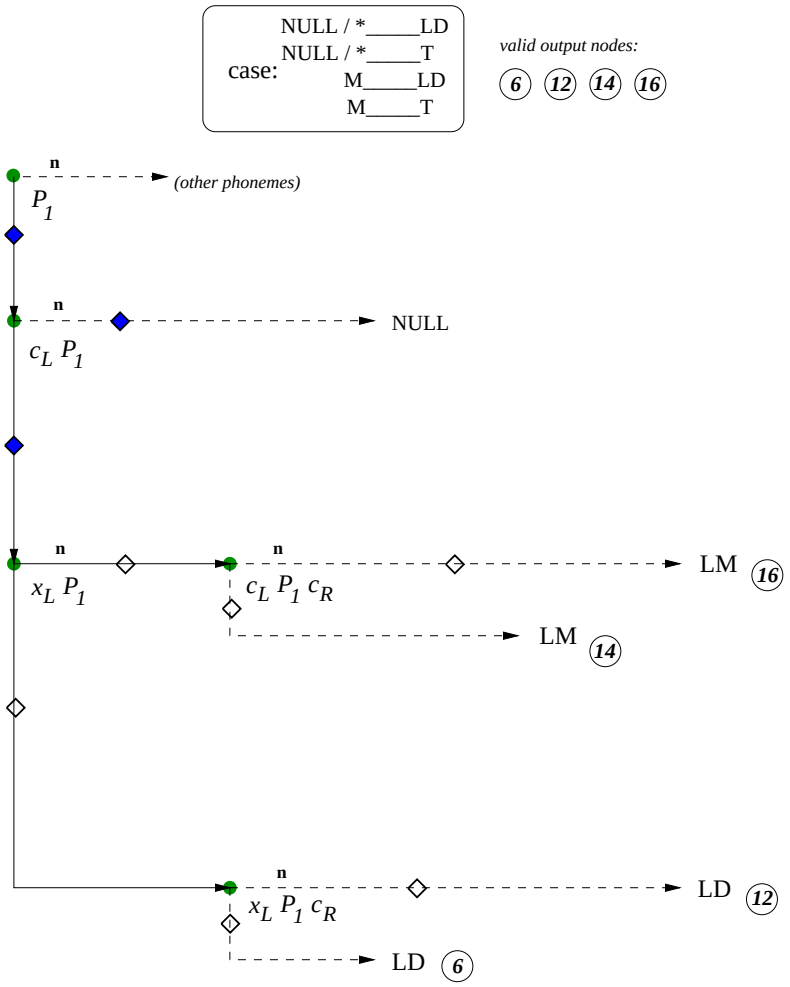

Figure 5.5: Decision tree for phonemes followed by an already-chosen diphone or triphone.

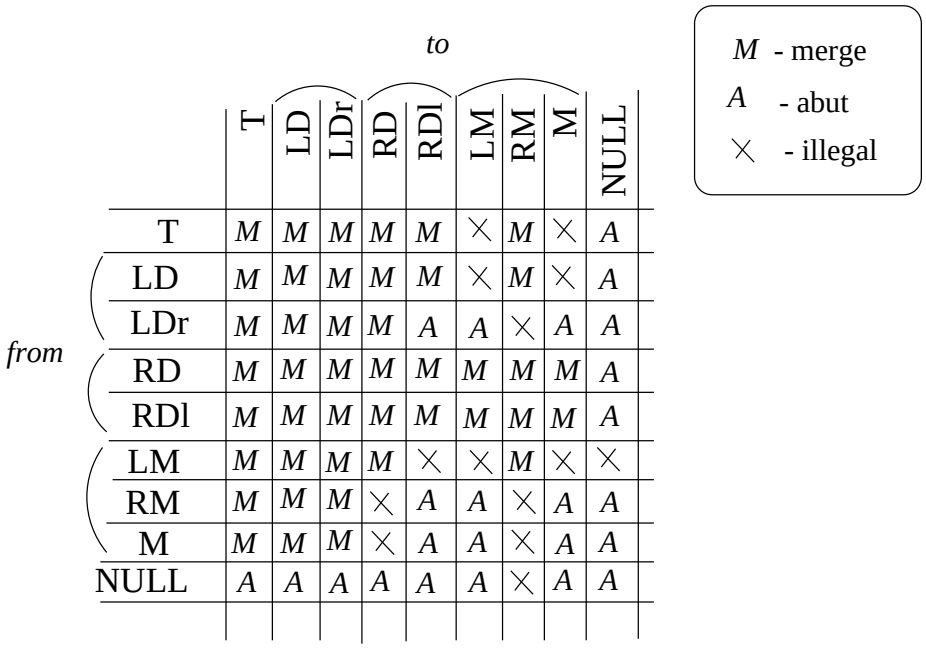

Figure 5.6: Transition matrix for all possible unit–unit combinations.

The addition of this fluctuation to the desired pitch contour gives the voice a more "human" feel, since a slight wavering is present in the voice. Bennett and Rodet [119] propose a similar model. A global scaling of ΔF_0 is incorporated as a controllable parameter to the user, so that more or less fluctuation can be synthesized.

Abrupt transitions of one note to another at a different pitch are not a natural phenomena. Rather, singers tend to transition somewhat gradually from one note to another. This effect can be modeled by applying a smoothing at note-to-note transitions in the target pitch contour. Timing of the pitch change by human vocalists is usually such that the transition between two notes takes place *before* the onset of the second note, rather than dividing evenly between the two notes [121].

Rhythmic characteristics

The natural "quantal unit" of rhythm in vocal music is the *syllable*—each syllable of lyric is associated with one or more notes of the melody. However, it is easily demonstrated that vocalists do not execute the onsets of notes at the beginnings of the leading *consonant* in a syllable, but rather at the beginning of the *vowel*. This effect has been cited in the study of rhythmic characteristics of singing [121] and speech [61]. LYRICOS employs rules that align the beginning of the first note in a syllable with the onset of the vowel in that syllable.

In this work, a simple model for scaling durations of syllables is used. First an average time scaling factor ρ_{syll} is computed:

$$\rho_{syll} = \frac{\sum_{n=1}^{N_{notes}} D_n}{\sum_{m=1}^{N_{phon}} D_m}, \tag{5.2}$$

where the values D_n are the desired durations of the N_{notes} notes associated with the syllable and D_m are the durations of the N_{phon} phonemes extracted from the inventory to compose the desired syllable. If $\rho_{syll} > 1$, then the vowel in the syllable is *looped* by repeating a set of frames extracted from the stationary portion of the vowel, until $\rho_{syll} \approx 1$. This preserves the duration of the consonants, and avoids unnatural time-

stretching effects. If $\rho_{syll} < 1$, the entire syllable is compressed in time by setting the time-scale modification factor ρ for all frames in the syllable equal to ρ_{syll}.

A more sophisticated approach to the problem would involve phoneme- and context-dependent rules for scaling phoneme durations in each syllable to more accurately represent the manner in which humans perform this adjustment.

Vibrato

Most trained vocalists produce a 5–6 Hz near-sinusoidal vibrato. As mentioned, pure frequency modulation of the glottal source can represent many of the observed effects of vibrato, since amplitude modulation will automatically occur as the partials "sweep by" the formant resonances. This effect is also easily implemented within the sinusoidal model framework by adding a sinusoidal modulation to the target pitch of each note. Vocalists usually are not able to vary the *rate* of vibrato, but rather modify the *modulation depth* to create expressive changes in the voice [120].

Using the graphical MIDI-based input to LYRICOS, users can draw contours that control vibrato depth over the course of the musical phrase, thus providing a mechanism for adding expressiveness to the vocal passage. A global setting of the vibrato rate is also possible.

Vocal tract length scaling

In synthesis of bass voices using the male voice inventory (recorded from a baritone vocalist), it was found that the voice took on an artificial-sounding buzzy quality. Through analysis of a simple tube model of the human vocal tract, it can be shown that the nominal formant frequencies associated with a longer vocal tract are lower than those associated with a shorter vocal tract [126]. Because of this, larger people usually have voices with a "deeper" quality; bass vocalists are typically males with voices possessing this characteristic.

To approximate the differences in vocal tract configuration between the recorded

and "desired" vocalists, a frequency-scale warping of the spectral envelope in each frame was performed, such that

$$\hat{H}(\omega) = H(\omega/\mu), \tag{5.3}$$

where $H(\omega)$ is the spectral envelope fit to the sinusoidal components in the frame and μ is a global frequency scaling factor dependent on the average pitch modification factor. The factor μ typically lies in the range $0.75 < \mu < 1.0$. Values of $\mu > 1.0$ could be used to simulate a more child-like voice, as well. In tests of this method, it was found that this frequency warping gives the synthesized bass voice a much more rich-sounding, realistic character.

Loudness scaling

Simply scaling the overall amplitude of the signal to produce changes in loudness has the same perceptual effect as turning the "volume knob" of an amplifier; it is quite different from a change in *vocal effort* by the vocalist. Nearly all studies of singing mention the fact that the downward tilt of the vocal spectrum increases as the voice becomes softer (e.g., [119, 120, 121]). This effect is conveniently implemented in a frequency-domain representation such as the sinusoidal model, since scaling of the sinusoid amplitudes can be performed. In LYRICOS, an amplitude scaling function based on the work in [119] is used:

$$G_{dB} = \frac{T_{in} \log_{10}(F_l/500)}{\log_{10}(3000/500)}, \tag{5.4}$$

where F_l is the frequency of the lth sinusoidal component and T_{in} is a spectral tilt parameter controlled by a MIDI "vocal effort" control function input by the user. This function produces a frequency-dependent gain scaling function parameterized by T_{in}, as shown in Figure 5.7.

In studies of acoustic correlates of perceived voice qualities [127, 102], it has been shown that utterances perceived as "soft" and "breathy" also exhibit a higher

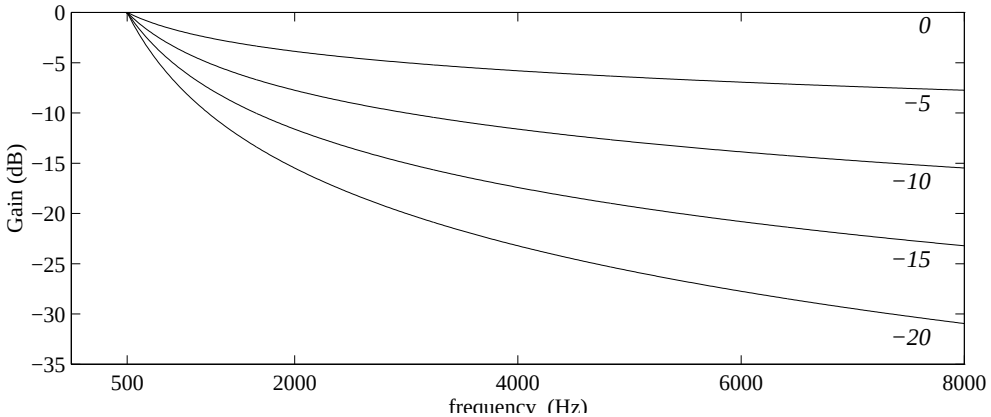

Figure 5.7: Spectral tilt modification as a function of frequency and parameter T_{in}.

level of high frequency aspiration noise than fully phonated utterances, especially in females. This effect on glottal pulse shape and spectrum is shown in Figure 5.8. As discussed in Section 3.3, it is possible to introduce a frequency-dependent noise-like character to the signal by employing the subframe phase randomization method. In LYRICOS, this capability has been used to model aspiration noise. The degree to which the spectrum is made noise-like is controlled by a mapping from the MIDI-controlled vocal effort parameter to the amount of phase dithering introduced.

Informal experiments with mapping the amount of randomization to (*i*) a cut-off frequency above which phases are dithered, and (*ii*) the scaling of the amount of dithering within a fixed band, have been performed. Employing either of these strategies results in a more natural, breathy, soft voice, although careful adjustment of the model parameters is necessary to avoid an unnaturally noisy quality in the output. A refined model that more closely represents the acoustics of loudness scaling and breathiness in singing is a topic for more extensive study in the future.

5.4 Qualitative Results

Because the field of singing synthesis is relatively immature in comparison to other areas of speech synthesis research, the value of a direct comparison of one system to another is somewhat questionable, and competing synthesis systems are difficult or impossible to obtain. For these reasons, only a qualitative assessment of the LYRICOS system is given here; demonstrations are available from the author upon request.

Phonetic modeling

As described in Section 5.3.3, the voice inventory design and unit selection methods were created with the intent of modeling vowels and classes of CV (consonant-vowel) and VC transitions well, at the expense of a more sparse representation of other phonetic contexts. Empirical evaluation of the system verifies that this objective was

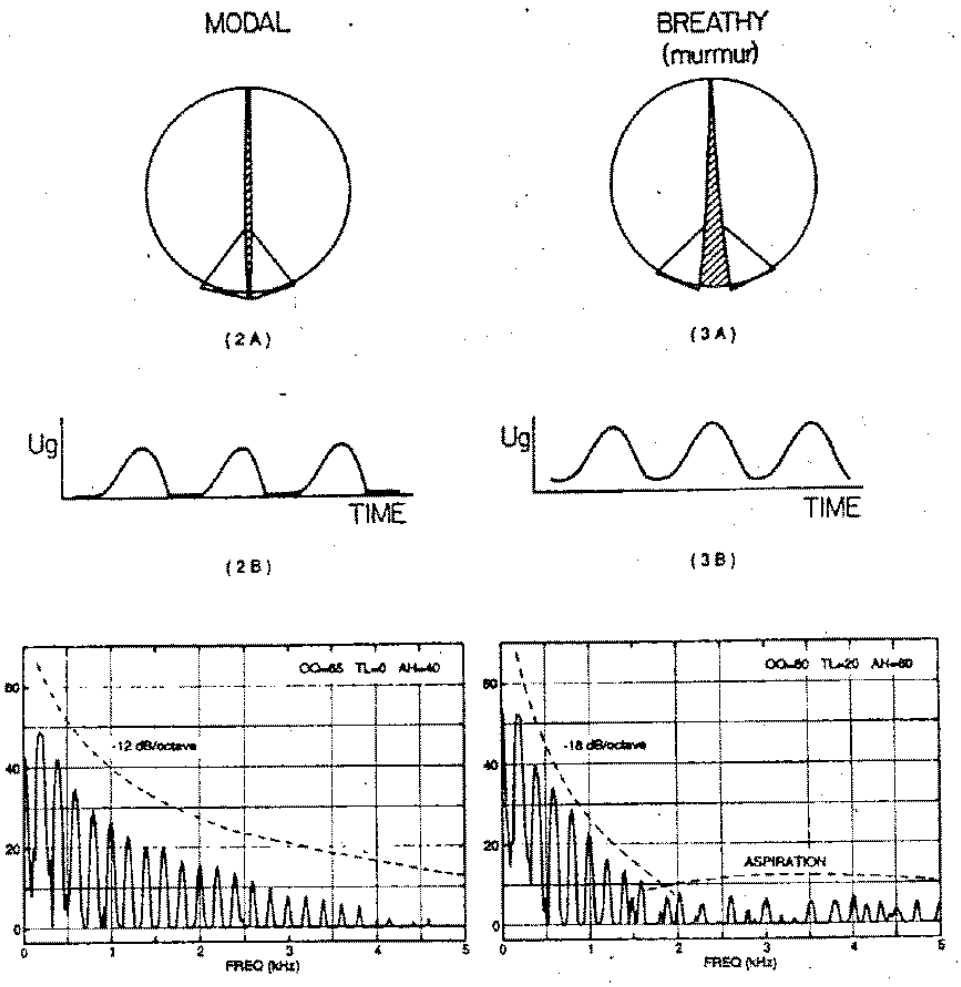

Figure 5.8: Spectral characteristics of the glottal source in modal (normal) and breathy speech (from [102]). *Top:* vocal fold configuration; *Middle:* time-domain waveforms; *Bottom:* short-time spectrum.

achieved. In general, sustained vowels and many simple CV and VC contexts sound very natural, and the perceived identity of the synthesized voice is very close to that of the original vocalist.

In contrast, intricate clusters of consonants (other than clusters explicitly included in the inventory) are not modeled well. Because the inventory contains relatively few instances of consonants in groups, the unit selection procedure selects small signal segments that are not representative of natural coarticulatory phenomena, and the voice sounds discontinuous and unnatural.

Alignment of the vowel portion of each syllable with note onsets leads to a fairly natural rhythm in songs with a moderate to slow tempo, but the rhythm of the voice becomes somewhat choppy and stilted at faster tempos. The lack of a sophisticated duration model results in an abrupt and "over-articulated" effect in rapidly articulated passages.

In general, the system is able to synthesize more natural sounding classical vocal styles than other popular styles. This is due to the fact that a much more precise articulation is required in classical music, rather than the more relaxed pronunciation of other styles. Recording of an inventory sung by a vocalist singing in a more relaxed style would possibly alleviate this problem, as would a more sophisticated duration model that controls phoneme durations by context-dependent rules.

Concatenation and synthesis

In general, the algorithms for concatenation and smoothing of joined segments were successful in producing a continuous, natural-sounding signal. However, one major problem in the synthesis algorithm is the accurate estimation of pitch pulse onset times. Frequent errors in onset time estimation occur, resulting in severe distortions of the signal. As a result, much of the inventory voice data required manual correction of pitch onset locations, a time-consuming and tedious procedure. The reason behind the unreliability of the onset estimation methods described previously in this thesis is

somewhat of a mystery. It may be that the characteristics of the particular vocalist used or of the *singing voice* excitation source in general are not well-suited to the onset estimation method. The development of a more robust analysis method is a necessity for further enhancement of the system's capabilities.

Another potential area of improvement relates to the spectral smoothing algorithms described in Section 4.3.2. These methods were designed to smooth discontinuities in only the vocal tract-related spectral envelope. However, differences in the shape of the excitation model remaining after removal of the spectral envelope still remain at the segment boundaries, and these are perceptible as abrupt changes in the voice timbre. A more sophisticated level of smoothing would attempt to normalize or smooth over differences in the spectral shape of the excitation as well.

Musical expression

A significant amount of time was spent on incorporating MIDI-controllable parameters to enable *expressiveness* in the synthesized voice. The availability of control over pitch variation and vocal effort correlates increased the naturalness and musicality of the results dramatically.

A quasi-random drift of the pitch, coupled with a sinusoidal frequency modulation of 5 to 6 Hz resulted in a natural-sounding vibrato. Control of the depth of vibrato enabled incorporation of effects such as a gradual increase in vibrato over the duration of sustained notes. Incorporating a smooth pitch transition prior to the onset of a new note was also effective.

A combination of amplitude and spectral tilt change was successfully used to produce the effect of *vocal effort* scaling. This produced a voice with a scale of perceived *loudness* levels, rather than giving the impression of a source changing *distance* to the listener, which is the result of simple amplitude scaling only. A model for incorporating *breathiness* in soft voice was also included. Although it was possible to add frequency-dependent noise to the signal using this method, it was difficult to

adjust the noise to a precise level that was perceivable but not distracting.

CHAPTER 6

CONCLUSIONS

In this research, the application of the *Analysis-by-Synthesis/Overlap-Add* sinusoidal model to synthesis of speech and singing voice was investigated, and a set of basic extensions and improvements of the capabilities of the model were developed.

First, the application of the model to concatenation-based text-to-speech (TTS) synthesis was described. Methods for concatenating segments extracted from a corpus of recorded speech were presented, and challenges associated with removing perceptible mismatches in time/frequency structure around the segment boundaries were identified. Methods for smoothing the signal near these boundaries using the sinusoidal model were presented. Results of a comparison between the new method and the commonly-used *Pitch-Synchronous Overlap Add* (PSOLA) method indicated that the method performs equally as well as an implementation of the PSOLA method for the cases tested. It was argued that, although the sinusoidal model method performance is similar to PSOLA for high-quality TTS, it presents a flexible framework for exploring other effects in synthesis.

To demonstrate this capability, the text-to-speech synthesis method was extended to the synthesis of singing. It was shown that the sinusoidal model approach enables the incorporation of various musically-interesting effects in the synthesized signal. These effects include vibrato, pitch variation and transition effects, and signal changes correlated with variation of vocal effort. Also in this work, methods of corpus design and unit selection specifically designed for singing synthesis were developed.

Despite the fact that a relatively small voice inventory is used, the system is capable of synthesizing a musically-pleasing singing voice that maintains the perceived identity of the vocalist recorded to create the unit inventory.

Finally, several improvements to the sinusoidal model itself were developed. One pervasive artifact in speech modified by the original model is a "choppiness" that occurs in unvoiced speech after downward pitch shifting of an utterance. The source of this modulation was found by deriving a time-domain interpretation of the pitch modification algorithm, and a method for eliminating the modulation effect was derived and implemented. Another commonly-cited artifact referred to as "tonal noise" occurs during unvoiced speech in utterances that are pitch-raised or time-expanded. This artifact was mitigated by proposing a method for phase randomization based on subframe synthesis of the signal. This method was also applied to synthesis of *breathiness* in voiced speech and applied within the singing voice synthesis system. Finally, artifacts due to errors in the estimation of pitch pulse onset times within each frame were analyzed, and a method for mitigating these errors was developed.

6.1 Contributions

Contributions of the proposed work include the following:

- Development of a framework for text-to-speech synthesis using the ABS/OLA sinusoidal model as a waveform synthesis engine, including methods for concatenating and smoothing disjointly-analyzed speech segments.
- Implementation and testing of the proposed algorithms as integrated with a commercial TTS system.
- Development of a singing voice synthesis system based on the sinusoidal model framework, including methods for controlling musically-interesting parameters such as vibrato, pitch variation, and vocal effort.

- Theoretical analysis of the ABS/OLA sinusoidal model behavior, explaining its limitations.
- Extensions and improvements of the capabilities of the ABS/OLA sinusoidal model for speech prosody modification, including
  - development of a method for frequency-dependent synthesis of noise-like signals using the sinusoidal model;
  - improvement in modification of unvoiced speech, overcoming or diminishing common "tonal" noise artifacts;
  - improvement in pitch modification methods to mitigate undesirable modulation effects;
  - development of a method for correcting pitch pulse onset time errors in the ABS/OLA model analysis algorithm.

6.2 Future work

Pitch pulse onset time estimation Perhaps the best way in which the ABS/OLA sinusoidal model could be universally improved for all applications is by development of more robust methods for estimation of the pitch pulse onset time. Although a method for masking the effects of isolated errors in this parameter was developed here (Section 3.4), this method is not capable of solving the more serious problems associated with its role in concatenation of segments. Further improvement of this analysis capability may involve viewing the problem in the time-domain rather than the sinusoidal domain.

Transient speech modification The sinusoidal model is most effective in modifying signals that are stationary, due to its frame-based nature. Although the ABS/OLA model can resynthesize nearly *any* signal accurately with *no modification*, time-scale and pitch modification disrupt transient speech events

significantly. Methods for identifying such events and applying models more consistent with the nature of the signal are necessary.

Voice quality changes In the singing synthesis work, attempts to synthesize *breathy* speech were made. Although it was observed that the phase randomization model (Section 3.3) was capable of introducing noise-like energy in specific frequency bands, it was found to be difficult to synthesize speech that was perceived as *natural* with such a method. A study of the acoustic characteristics of breathy speech and voice could be undertaken, with a focus on introducing these effects in speech modification. Also, implentation of other glottal source effects such as *laryngealization* [102] should be explored.

Excitation smoothing in concatenation The spectral smoothing algorithms developed for concatenation in Section 4.3.2 were effective in smoothing differences in formant location to produce smoother-sounding synthetic speech. However, further perceptible spectral differences (due to glottal source differences) often still remain at the boundary of joined segments. Further methods of smoothing spectral differences at concatenation boundaries should be explored to alleviate this problem.

Rule-driven coarticulatory effects in synthesis In both singing synthesis and TTS, concatenated segments often do not model all necessary coarticulatory effects, especially in rapidly-spoken speech, where such effects may extend over several adjacent phonemes. In the development of rule-based synthesis systems over the last few decades, models that describe many of these phonological effects have been implemented. However, this type of rule-based knowledge has not been applied in concatenation-based systems. Using a flexible framework such as the sinusoidal model, it may be possible to develop speech modification strategies that can implement supra-segmental coarticulatory effects dictated by knowledge derived from such rules.

Duration models in singing synthesis As mentioned in Section 5.4, the duration scaling strategies employed in the singing synthesis system are fairly rudimentary. The naturalness of the vocal rhythm produced by the synthesizer can be improved by incorporating ideas from the more sophisticated models of duration scaling commonly used in TTS synthesis.

Bibliography

[1] J. Webster, A. Cook, W. Tompkins, and G. Vanderheiden, eds., *Electronic Devices for Rehabilitation*, New York, NY: John Wiley and Sons, 1985.

[2] N. Alm, J. Todman, L. Elder, and A. Newell, "Computer-aided conversation for severely physically impaired non-speaking people," *Proceedings Interchi '93*, pp. 236–241, April 1993.

[3] A. Thomas, "Communication devices for the nonvocal disabled," *IEEE Computer Magazine*, pp. 25–30, January 1981.

[4] M. W. de Kleijn-de Vrankrijker, *The Use of Technology in the Care of the Elderly and the Disabled*, Westport, CT: Greenwood Press, 1980.

[5] J. H. Page and A. P. Breen, "The Laureate text-to-speech system: Architecture and applications," *BT Technology Journal*, vol. 14, January 1996.

[6] D. H. Klatt, "Review of text-to-speech conversion for English," *Journal of the Acoustical Society of America*, vol. 82, pp. 737–793, September 1987.

[7] G. Fant, *Acoustic Theory of Speech Production*, The Netherlands: Mouton, 's-Gravenhage, 1960.

[8] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America*, vol. 67, pp. 971–995, 1980.

[9] Lernout & Hausbie Speech Products U.S.A., Inc., Woburn, MA, *The Lernout & Hauspie Text-to-Speech System.*

[10] C. Hamon, E. Moulines, and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modificaitons of speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 238–241, 1989.

[11] E. Moulines and F. Charpentier, "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, pp. 453–467, December 1990.

[12] J. Olive, "A scheme for concatenating units for speech synthesis," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 568–571, 1980.

[13] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, pp. 561–580, April 1975.

[14] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 614–617, May 1982.

[15] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 937–940, April 1985.

[16] T. F. Quatieri and R. J. McAulay, "Speech transformations based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1449–1464, December 1986.

[17] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Transactions on Signal Processing*, vol. 40, pp. 497–510, March 1992.

[18] T. F. Quatieri and R. G. Danisewicz, "An approach to co-channel talker interference suppression using a sinusoidal model for speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 56–69, January 1990.

[19] J. C. Rutledge, *Time-Varying, Frequency Dependent Compensation for Recruitment of Loudness*, Ph.D. thesis, Georgia Institute of Technology, December 1989.

[20] J. C. Rutledge and M. A. Clements, "Compensation for recruitment of loudness in sensorineural hearing impairments using a sinusoidal model of speech," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 3641–3644, April 1991.

[21] T. F. Quatieri and R. J. McAulay, "Noise reduction using a soft-decision sine-wave vector quantizer," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 821–824, April 1990.

[22] M. W. Macon and M. A. Clements, "Sinusoidal modeling and coding of audio signals," tech. rep., Georgia Center for Advanced Telecommunications Technology, June 1994.

[23] D. P. W. Ellis, "Hierarchic models of hearing for sound separation and reconstruction," Technical Report 219, MIT Media Lab Perceptual Computing Group, March 1993.

[24] E. B. George, *An Analysis-by-Synthesis Approach to Sinusoidal Modeling Applied to Speech and Music Signal Processing*, Ph.D. thesis, Georgia Institute of Technology, November 1991.

[25] E. B. George and M. J. T. Smith, "An analysis-by-synthesis approach to sinusoidal modeling applied to the analysis and synthesis of musical tones," *Journal of the Audio Engineering Society*, vol. 40, pp. 497–516, June 1992.

[26] E. B. George and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," accepted for publication in *IEEE Transactions on Speech and Audio Processing*, 1996.

[27] R. J. McAulay and T. F. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech model," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 27.6.1–27.6.4, April 1984.

[28] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 744–754, August 1986.

[29] R. J. McAulay and T. F. Quatieri, "Phase modeling and its application to sinusoidal transform coding," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1713–1715, April 1986.

[30] T. F. Quatieri and R. J. McAulay, "Phase coherence in speech reconstruction for enhancement and coding applications," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 207–210, April 1989.

[31] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1223–1235, August 1988.

[32] X. Serra, *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*, Ph.D. thesis, Stanford University, CCRMA Department of Music, October 1989.

[33] X. Serra and J. S. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–23, 1990.

[34] H. Carl and B. Kolpatzik, "Speech coding using nonstationary sinusoidal modeling and narrow-band basis functions," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 581–584, April 1991.

[35] J. S. Marques and A. J. Abrantes, "Hybrid harmonic coding of speech at low bit-rates," *Speech Communication*, vol. 14, pp. 231–247, June 1994.

[36] J. S. Marques and L. B. Almeida, "New basis functions for sinusoidal decompositions," *EUROCON 88: 8th European Conference on Electrotechnics*, pp. 48–51, June 1988.

[37] J. S. Marques and L. B. Almeida, "Sinusoidal modeling of speech: Representation of unvoiced sounds with narrow-band basis functions," *Signal Processing IV: Theories and Applications. Proceedings of EUSIPCO-88: Fourth European Signal Processing Conference*, pp. 891–894, 1988.

[38] K. Fitz and L. Haken, "Bandwidth enhanced sinusoidal modeling in Lemur," in *Proceedings of the International Computer Music Conference*, pp. 154–157, 1995.

[39] C. D'Alessandro, B. Yegnanarayana, and V. Darsinos, "Decomposition of speech signals into deterministic and stochastic components," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, May 1995.

[40] V. Darsinos, C. D'Alessandro, and B. Yegnanarayana, "Evaluation of a periodic/aperiodic speech decomposition algorithm," in *Proceedings of EUROSPEECH*, (Madrid), pp. 393–396, September 1995.

[41] P. R. Cook, "Noise and aperiodicity in the glottal source: A study of singer voices," Tech. Rep. Stan-M-75, Stanford University Department of Music, Stanford, CA, August 1991. also published in *Proceedings of the Twelfth International Congress of Phonetic Sciences*, Aix-en-Provence, France.

[42] C. Chafe, "Pulsed noise in self-sustained oscillations of musical instruments," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1157–1160, IEEE, April 1990.

[43] S. Grau-Grovel, C. D'Alessandro, and G. Richard, "A speech formant synthesizer based on harmonic + random formant-waveforms representations," in *Proceedings of EUROSPEECH*, (Berlin), pp. 1697–1700, 1993.

[44] G. Richard, *Modélisation de la Composante Stochastique de la Parole*, Ph.D. thesis, l'Université Paris XI, April 1994.

[45] G. Richard, C. d'Alessandro, and S. Grau, "Unvoiced speech analysis and synthesis using Poissonian random formant-wave-functions," in *Signal Processing IV: Theories and Applications, Proceedings of EUSIPCO-92*, pp. 347–350, Elsevier Science Publishers B.V., 1992.

[46] J. Laroche, Y. Stylianou, and E. Moulines, "HNS: Speech modification based on a harmonic + noise model," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. II–550–553, April 1993.

[47] Y. Stylianou, J. Laroche, and E. Moulines, "High quality speech modification based on a harmonic + noise model," *Proceedings of EUROSPEECH*, pp. 451–454, September 1995.

[48] J. Allen, S. Hunnicutt, and D. H. Klatt, *From Text to Speech: The MITalk System*, Cambridge, UK: Cambridge Univ. Press, 1987.

[49] E. Charniak, *Statistical Language Learning*, Cambridge, MA: MIT Press, 1993.

[50] M. Edgington, A. Lowry, P. Jackson, A. P. Breen, and S. Minnis, "Overview of current text-to-speech techniques: Part I - Text and linguistic analysis," *BT Technology Journal*, vol. 14, January 1996.

[51] H. C. Nusbaum, A. L. Francis, and A. S. Henly, "Measuring the naturalness of synthetic speech," *International Journal of Speech Technology*, vol. 1, no. 1, pp. 7–19, 1995.

[52] J. Hirschberg and P. Prieto, "Training intonational phrasing rules automatically for English and Spanish text-to-speech," *Speech Communication*, vol. 18, pp. 283–292, May 1996.

[53] M. Halle and J. R. Vergnaud, *An Essay on Stress*, Cambridge, MA: MIT Press, 1987.

[54] B. Hayes, "Extrametricality and English stress," *Linguistic Inquiry*, vol. 13, pp. 227–276, 1982.

[55] M. Liberman and A. Prince, "On stress and linguistic rhythm," *Linguistic Inquiry*, vol. 8, pp. 249–336, 1977.

[56] M. Edgington, A. Lowry, P. Jackson, A. P. Breen, and S. Minnis, "Overview of current text-to-speech techniques: Part II - Prosody and speech generation," *BT Technology Journal*, vol. 14, January 1996.

[57] J. B. Pierrehumbert, *The Phonology and Phonetics of English Intonation*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1980.

[58] J. B. Pierrehumbert, "Synthesizing intonation," *Journal of the Acoustical Society of America*, vol. 70, pp. 985–995, October 1981.

[59] B. Möbius, "A quantitative model of German intonation and its application to speech synthesis," in *Proceedings of the Second ESCA/IEEE Workshop on Speech Synthesis*, pp. 139–142, September 1994.

[60] K. Ross and M. Ostendorf, "A dynamical system model for generating F_0 for synthesis," in *Proceedings of the Second ESCA/IEEE Workshop on Speech Synthesis*, pp. 131–134, September 1994.

[61] P. Barbosa and G. Bailly, "Characterisation of rhythmic patters for text-to-speech synthesis," *Speech Communication*, vol. 15, pp. 127–137, October 1994.

[62] A. P. Breen, "A comparison of statistical and rule-based methods of determining segmental durations," in *Proceedings of the International Conference on Spoken Language Processing*, pp. 1199–1202, 1992.

[63] G. Bailly, E. Castelli, and B. Gabioud, "Building prototypes for articulatory synthesis," in *Proceedings of the Second ESCA/IEEE Workshop on Speech Synthesis*, pp. 9–11, 1994.

[64] C. L. Smith, C. P. Browman, R. S. McGowan, and B. Kay, "Extracting dynamic parameters from speech movement data," *Journal of the Acoustical Society of America*, vol. 93, pp. 1580–1588, March 1993.

[65] C. H. Coker, "A model of articulatory dynamics and control," *Proceedings of the IEEE*, vol. 64, pp. 452–459, 1976.

[66] J. N. Larar, J. Schroeter, and M. M. Sondhi, "Vector quantization of the articulatory space," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1812–1818, December 1988.

[67] K. Cummings and M. Clements, "Modelling speech production using finite difference techniques," *Presented at the Meeting of the Acoustical Society of America*, June 1994.

[68] I. Titze, "A four-parameter model of the glottis and vocal fold contact area," *Speech Communication*, vol. 8, pp. 191–201, September 1989.

[69] J. N. Holmes, "Formant synthesizers: cascade or parallel," *Speech Communication*, vol. 2, pp. 251–273, 1983.

[70] H. Kaeslin, "A systematic approach to the extraction of diphone elements from natural speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 264–271, April 1986.

[71] O. Fujimura, "Syllables as concatenated demisyllables and affixes," *Journal of the Acoustical Society of America*, p. S55, Spring 1976. Supplement No. 1.

[72] S. Nakajima and H. Hamada, "Automatic generation of synthesis units based on context oriented clustering," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 659–662, 1988.

[73] K. Itoh, S. Nakajima, and T. Hirokawa, "A new waveform speech synthesis approach based on the coc speech spectrum," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. I, pp. 577–580, May 1994.

[74] S. Nakajima, "Automatic synthesis unit generation for English speech synthesis based on multi-layered context oriented clustering," *Speech Communication*, vol. 14, pp. 313–324, September 1994.

[75] N. Iwahashi, N. Kaiki, and Y. Sagisaka, "Concatenative speech synthesis by minimum distortion criteria," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. II, pp. 65–68, 1992.

[76] Y. Sagisaka, "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 679–682, 1988.

[77] W. J. Wang, W. N. Campbell, N. Iwahashi, and Y. Sagisaka, "Tree-based unit selection for English speech synthesis," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. II, pp. 191–194, 1993.

[78] A. Black and W. N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proceedings EUROSPEECH*, (Madrid, Spain), pp. 581–584, ESCA, September 1995.

[79] N. Campbell, "Prosody and the selection of units for concatenation synthesis," in *Proceedings of the Second ESCA/IEEE Workshop on Speech Synthesis*, (New Paltz, NY), pp. 61–64, September 1994.

[80] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 373–376, 1996.

[81] Y. Itoh, M. Hashimoto, and N. Higuchi, "Sub-phonemic optimal path search for concatenative speech synthesis," in *Proceedings EUROSPEECH*, (Madrid, Spain), pp. 577–580, ESCA, September 1995.

[82] T. Dutoit, "High quality text-to-speech synthesis: A comparison of four candidate algorithms," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. I–565–568, April 1994.

[83] D. Bigorgne, O. Boeffard, B. Cherbonnel, F. Emerard, D. Larreur, J. Le Saint-Milon, I. Metayer, C. Sorin, and S. White, "Multilingual PSOLA text-to-speech system," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. II–187–190, April 1993.

[84] H. Kawai, N. Higuchi, T. Simizu, and S. Yamamoto, "Development of a text-to-speech system for Japanese based on waveform splicing," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. I, pp. 569–572, 1994.

[85] E. Moulines, F. Emerard, D. Larreur, J. L. Le Saint Milon, L. Le Faucheur, F. Marty, F. Charpentier, and C. Sorin, "A real-time French text-to-speech system generating high-quality synthetic speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 309–312, 1990.

[86] B. Atal and N. David, "On synthesizing natural sounding speech by linear prediction," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 44–47, 1979.

[87] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Communication*, vol. 16, pp. 175–206, February 1995.

[88] B. Caspers and B. S. Atal, "Role of multipulse excitation in synthesis of natural sounding voiced speech," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 2388–2391, 1987.

[89] A. Varga and F. Fallside, "A technique for using multipulse linear predictive speech synthesis in test-to-speech type systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-35, pp. 586–587, April 1987.

[90] F. Charpentier and E. Moulines, "Text-to-speech algorithms based on FFT synthesis," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 667–670, 1988.

[91] E. R. Banga, E. López-Gonzalo, and C. García-Mateo, "A text-to-speech system for Spanish with frequency domain based prosodic modification algorithm," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. II–183–186, April 1993.

[92] M. A. Rodríguez-Crespo, P. Sanz-Velasco, and L. M.-S. amd J. G. Escalada-Sardina, "On the use of a sinusoidal model for speech synthesis in TTS," in *Proceedings of the Second ESCA/IEEE Workshop on Speech Synthesis*, pp. 21–24, ESCA/IEEE, September 1994.

[93] E. R. Banga and C. García-Mateo, "Shape-invariant pitch-synchronous text-to-speech conversion," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 656–659, May 1995.

[94] T. Dutoit and H. Leich, "Improving the TD-PSOLA text to-speech synthesizer with a specially desgined MBE resynthesis of the segments database," in *Proceedings EUSIPCO*, pp. 343–347, August 1992.

[95] T. Dutoit and H. Leich, "MBR-PSOLA: Text-to-speech synthesis based on an MBE re-synthesis of the segments database," *Speech Communication*, vol. 13, pp. 435–440, December 1993.

[96] M. W. Macon and M. A. Clements, "Sinusoidal modeling and modification of unvoiced speech," submitted to *IEEE Transactions on Speech and Audio Processing*, 1996.

[97] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, Academic Press Limited, 3rd ed., 1989.

[98] R. J. McAulay and T. F. Quatieri, "Low-rate speech coding based on the sinusoidal model," in *Advances in Speech Signal Processing* (S. Furui and M. M. Sondhi, eds.), New York: M. Dekker, 1992.

[99] W. M. Fisher and G. R. Doddington, "The DARPA speech recognition research database: specification and status," in *Proceedings of the DARPA Speech Recognition Workshop*, (Palo Alto, CA), pp. 93–99, 1986.

[100] T. F. Quatieri and R. J. McAulay, "Mixed-phase deconvolution of speech based on a sine-wave model," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 649–652, April 1987.

[101] M. W. Macon and M. A. Clements, "Improvements to the ABS/OLA sinusoidal model." Final Report, Sanders Inc., a Lockheed-Martin Company, June 1995.

[102] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *Journal of the Acoustical Society of America*, vol. 87, pp. 820–857, February 1990.

[103] M. W. Macon and M. A. Clements, "Speech synthesis based on an overlap-add sinusoidal model," *Journal of the Acoustical Society of America*, vol. 97, p. 3246, May 1995. (A).

[104] M. W. Macon and M. A. Clements, "Speech concatenation and synthesis using an overlap-add sinusoidal model," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 361–364, May 1996.

[105] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4–29, April 1984.

[106] W. Mendenhall and T. Sincich, *Statistics for the Engineering and Computer Sciences*, San Francisco: Dellen–Macmillan, 2 ed., 1988.

[107] E. B. George and M. J. T. Smith, "A new speech coding model based on a least-squares sinusoidal representation," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 1641–1644, April 1987.

[108] E. B. George and M. J. T. Smith, "Perceptual considerations in a low bit rate sinusoidal vocoder," in *Proceedings of the IEEE International Phoenix Conference on Computers in Communications*, pp. 268–275, March 1990.

[109] M. W. Macon, L. Jensen-Link, J. Oliverio, M. A. Clements, and E. B. George, "A singing voice synthesis system based on sinusoidal modeling." submitted to *ICASSP-97*.

[110] B. M. Doscher, *The Functional Unity of the Singing Voice*, Metuchen, NJ: Scarecrow Press, 2nd ed., 1994.

[111] G. J. Troup, "The physics of the singing voice," *Physics Letters*, vol. 74, no. 5, pp. 379–401, 1981.

[112] J. Sundberg, "Formant technique in a professional female singer," *Acustica*, vol. 32, pp. 89–96, 1975.

[113] A. H. Benade, *Fundamentals of Musical Acoustics*, ch. 19, New York: Dover Publications, Inc., 1990.

[114] J. Sundberg, "The acoustics of the singing voice," *Scientific American*, vol. 236, pp. 82–91, March 1977.

[115] G. Bloothooft and R. Plomp, "The timbre of sung vowels," *Journal of the Acoustical Society of America*, vol. 84, pp. 847–860, September 1988.

[116] T. D. Rossing, J. Sundberg, and S. Ternström, "Acoustic comparison of soprano solo and choir singing," *Journal of the Acoustical Society of America*, vol. 82, pp. 830–836, September 1987.

[117] R. Maher and J. Beauchamp, "An investigation of vocal vibrato for synthesis," *Applied Acoustics*, vol. 30, pp. 219–245, 1990.

[118] D. Rossiter and D. M. Howard, "Voice source and acoustic output qualities for singing synthesis," in *Proceedings of the International Computer Music Conference*, pp. 483–484, 1994.

[119] G. Bennett and X. Rodet, "Synthesis of the singing voice," in *Current Directions in Computer Music Research* (M. V. Mathews and J. R. Pierce, eds.), pp. 19–44, MIT Press, 1989.

[120] P. R. Cook, *Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing*, Ph.D. thesis, Stanford University Department of Music, Stanford, CA, December 1990. Technical Report Stan-M-68.

[121] J. Sundberg, "Synthesis of singing by rule," in *Current Directions in Computer Music Research* (M. V. Mathews and J. R. Pierce, eds.), pp. 45–56, MIT Press, 1989.

[122] P. R. Cook, "Synthesis of the singing voice using a physically parameterized model of the human vocal tract," Tech. Rep. Stan-M-57, Stanford University Department of Music, August 1989. Also published in *Proceedings of the International Computer Music Conference*, Ohio, 1989.

[123] P. R. Cook, "SPASM, a real-time vocal tract physical model controller and Singer, the companion software synthesis system," *Computer Music Journal*, vol. 17, pp. 30–43, Spring 1993.

[124] J. M. Chowning, "Computer synthesis of the singing voice," in *Sound Generation in Winds, Strings, Computers* (J. Sundberg, ed.), pp. 4–13, Stockholm: Royal Swedish Academy of Music, 1980.

[125] J. M. Chowning, "Frequency modulation synthesis of the singing voice," in *Current Directions in Computer Music Research* (M. V. Mathews and J. R. Pierce, eds.), pp. 57–64, MIT Press, 1989.

[126] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, New Jersey: Prentice-Hall, 1978.

[127] H. M. Hanson, *Glottal Characteristics of Female Speakers*, Ph.D. thesis, Harvard University, Cambridge, MA, May 1995.

Vita

Michael W. Macon was born October 11, 1968 in Milwaukee, Wisconsin, where he graduated from Thomas More High School in 1986. He attended the University of Dayton (Dayton, OH), where he graduated *magna cum laude* with a Bachelor of Electrical Engineering degree in April 1991. In September 1991, he began graduate studies at the Georgia Institute of Technology (Atlanta, GA). In 1993, he spent two quarters studying at Georgia Tech Lorraine (Metz, France), where he completed the requirements for the degree of Master of Science in Electrical Engineering. He received the Ph.D. degree from Georgia Tech in 1996.

His research interests include text-to-speech synthesis, sinusoidal modeling, speech and audio coding, human auditory perception, and music synthesis via computer and accordion.

Speech Synthesis Based on Sinusoidal Modeling

Michael W. Macon

154 pages

Directed by: Dr. Mark A. Clements

In this research, the application of the *Analysis-by-Synthesis/Overlap-Add* sinusoidal model to synthesis of speech and singing voice is investigated, and a set of basic extensions and improvements of the capabilities of the model are developed. First, the application of the model to concatenation-based text-to-speech (TTS) synthesis is described. Methods for concatenating segments extracted from a corpus of recorded speech are presented, and challenges associated with removing perceptible mismatches in time/frequency structure around the segment boundaries are identified. Methods for smoothing the signal near these boundaries using the sinusoidal model are presented. The implementation of this model within a commercial TTS system serves as a research testbed. Results of a comparison between the new method and the commonly-used *Pitch-Synchronous Overlap Add* (PSOLA) method indicate that the method performs equally as well as the PSOLA method in the cases tested.

Next, through the extension of the text-to-speech synthesis method to the synthesis of singing, it is shown that the flexibility of the sinusoidal model approach enables the incorporation of various musically-interesting effects into the synthesized signal. These effects include vibrato, pitch variation and transition effects, and changes correlated with change in vocal effort. Also in this system, methods of corpus design and unit selection specifically designed for singing synthesis are developed. Despite the fact that a relatively small voice inventory is used, the system is capable of synthesizing a musically-pleasing singing voice that maintains the perceived identity of the vocalist recorded to create the unit inventory.

Finally, several improvements to the sinsusoidal model itself are detailed. The

causes of artifacts present in the original ABS/OLA model are found by theoretical and empirical analysis, and methods for eliminating or diminishing these artifacts are presented. Among the innovations is a method for phase randomization based on subframe synthesis of the signal. It is shown through the results of a subjective comparison test that the method improves the quality of unvoiced speech synthesized using the model.