IMPROVING HIGH QUALITY CONCATENATIVE TEXT-TO-SPEECH SYNTHESIS USING THE CIRCULAR LINEAR PREDICTION MODEL

A Thesis
Presented to
The Academic Faculty

by

Sunil Ravindra Shukla

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2007

IMPROVING HIGH QUALITY CONCATENATIVE TEXT-TO-SPEECH SYNTHESIS USING THE CIRCULAR LINEAR PREDICTION MODEL

Approved by:

Professor David Anderson,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Thomas P. Barnwell III,
Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mark A. Clements
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Bruce Walker
School of Psychology
Georgia Institute of Technology

Professor Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 5 January 2007

To my family for their unconditional love and support,

to my beloved wife, Minu, for her untiring faith and endless patience,

and to Dr. Thomas P. Barnwell III for helping me achieve my dreams.

ACKNOWLEDGEMENTS

Though I have been away from the Georgia Tech Center for Signal and Image Processing (CSIP) for many years now, my experiences there during the early years of my Ph.D. have been memorable and life-changing. I have had the privilege opportunity to work among the brightest and most generous individuals in this field. Even while working remotely on my thesis from Michigan, I have received their support and guidance regularly.

First and foremost, I thank my advisor, Dr. Thomas P. Barnwell III, for his tireless support, patience, guidance, and encouragement. We knew that to solve all of the problems proposed for this thesis, working from a distant location, was always going to be a difficult task. I clearly recall Dr. Clements warning me, quite accurately, that if I work on the thesis remotely while keeping a full-time position, 1 year of work will be the equivalent of 1-3 months at CSIP. We finally completed the work following years of teleconference meetings during weekends, evening hours, and even holidays. I believe that for a professor at his level to be so generous of his time is very rare. This thesis would never have been possible without his insightful observations and advice. He challenged me to find my own path in research and taught me to demonstrate the research contributions with accuracy and integrity. It has been a privilege and pleasure to have worked with Dr. Barnwell.

I would like to express my sincere appreciation to Dr. Mark A. Clements and Dr. David V. Anderson for serving on the thesis reading committee. Dr. Clements has given me valuable advice throughout my time at CSIP. I especially enjoyed his group meetings, where I had the opportunity to learn about all of the research conducted by the students. I would like to especially thank Dr. Anderson for being instrumental

in assisting with the scheduling of the defense. It has also been a pleasure to work with him during my early years as a fellow student. I express my sincere gratitude to Dr. Aaron Lanterman for serving on the thesis defense committee though we have not met in the past. I also express sincere thanks to Dr. Bruce Walker for serving on the thesis defense committee as the outside faculty member on a last minute notice.

I thank the staff at CSIP and ECE Graduate Affairs for their often overlooked, but extremely important, administrative support. Sam Smith, Keith May, Kay Gilstrap, Charlotte Doughty, and Marilou Mycko have all been extremely helpful and considerate in assisting me with various problems that arose. I would like to thank my fellow colleagues at CSIP, especially Dr. Ali Erdem Ertan. Though we developed much of the ground work jointly, his insights and detailed work in some of the areas have been valuable for completing my thesis. I thank my friends and colleagues at my employers, Visteon and CSR, for serving as subjects for the tests.

My thesis topic was developed during an internship at Texas Instruments under the supervision of Dr. Tandhoni Rao and Dr. Vishu Vishwanathan. Much of the funding for my work was also provided by Texas Instruments. For this, I owe Texas Instruments and the research staff my gratitude.

Most importantly, without the commitment and support from my loving wife, Minu Shukla, this thesis could never have been completed. She has been remarkably patient and encouraging throughout this undertaking. I would like to express my deepest love and appreciation for her immense sacrifice in this effort. Minu, you are equally deserving of the credit for this thesis and the Ph.D.

Finally, I wish to express my appreciation and love to my family; my grandmother, Durgaben Shukla, my father and mother, Dr. and Mrs. Ravindra and Charugita Shukla, my brother and sister-in-law, Dr. and Mrs. Aseem and Suhag Shukla. They have been an inspiration for me and instilled in me a desire to achieve the highest goals. Thank you for always standing by me in all my endeavors with encouragement,

support, and unconditional love. I am thankful for the tremendous love and support that I have received from my father-in-law and mother-in-law, Mr. and Mrs. Arun and Mohini Upadhyay. This thesis is dedicated to my family and I hope that I have made them proud. I also thank my extended family and friends for their good wishes and blessings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Current high quality text-to-speech (TTS) systems are based on unit selection from a large database that is both contextually and prosodically rich. These systems, albeit capable of natural voice quality, are computationally expensive and require a very large footprint. Their success is attributed to the drammatic reduction of storage costs in recent times. However, for many TTS applications a smaller footprint is becoming a standard requirement. Reducing the footprint in unit selection based TTS systems predictably compromises the quality. This thesis presents a new method for representing speech segments to improve current concatenative TTS systems.

The circular linear prediction (CLP) model is revisited and combined with the constant pitch transform (CPT) to provide a robust representation of speech signals that allows for limited prosodic movements without a perceivable loss in quality. The CLP model assumes that each frame of voiced speech is an infinitely periodic signal. This assumption allows for LPC modeling using the covariance method, with the efficiency of the autocorrelation method, as the two become identical. In implementation, the periodicity requirement is satisfied by using a highly precise fractional pitch detector to determine frame boundaries. For unvoiced speech, a constant pitch period is used as the periodicity requirement is not relevant. The CPT is combined with this model to provide a database that is uniform in pitch for matching the target prosody during synthesis. Since the frames are pitch synchronous with fractional resolution of the pitch periods, unit concatenation can be performed without introducing errors caused by interpolation of the waveforms or modeling parameters. For resolving artifacts caused by pitch modifications in regions of voicing transitions, a method has been introduced for reducing peakiness in the LP spectra by constraining

the line spectral frequencies (LSF). Additionally, the problem of optimal unit size is investigated and a new method for defining concatenative speech units is presented. This method involves analysis of speech and text corpora to define concatenation units based on junctural characteristics.

Two experiments have been conducted to demonstrate the potential for the CLP/CPT representation to enhance current systems in terms of voice quality and scalability. The first is a listening test to determine the ability of this model to realize prosody modifications without perceivable degradation. In this test, utterances are resynthesized using the CLP/CPT method with emphasized prosodics to increase intelligibility in harsh environments. The second experiment compares the quality of utterances synthesized by unit-selection based limited-domain TTS against the CLP/CPT method. The CLP/CPT method uses only a limited number of units from the database of the unit-selection TTS system. The results demonstrate that the CLP/CPT representation, applied to current concatenative TTS systems, can reduce the size of the database and increase the prosodic richness without noticeable degradation in voice quality.

CHAPTER I

INTRODUCTION

1.1 Problem Statement

For most people, speech is the most natural form of communication and an ideal medium for interfacing with the environment to obtain information. Today, the most common interfaces for human-machine interaction are still keyboards, keypads, and mice. However, an increasing necessity to interface with machines in mobile environments is leading to speech becoming a required means to interface with machines and automated information services. For this reason the automatic generation of speech from text, referred to as text-to-speech (TTS) synthesis, which has been extensively researched and improved upon over the last two decades, has been gaining significant interest in commercial applications over the last 5-7 years. TTS is a complex problem that has made significant progress in the realm of concatenative systems in the last few years. Since the first practical TTS systems of the late 1970s and early 1980s (MITalk in 1979, Klattalk in 1981, Prose-2000 in 1982, and DECTalk and Infovox in 1983) [47], a number of different techniques have been developed and implemented to produce systems capable of synthesizing speech that is of high intelligibility and telephone quality. Notably, recent developments in unit-selection based concatenative TTS for limited vocabularies has demonstrated "natural quality" speech. For limited-domain applications, TTS is starting to gain popularity in information services such as kiosks, telemarketing, customer service, airline reservation systems, etc. However, for applications that require an unlimited or very large vocabulary, it is still far from becoming a widely used interface due to a number of factors including quality and the large storage requirements.

The overall text-to-speech process consists of a number of complex steps that can be classified under two general categories: natural language processing (NLP) and signal processing. This research is concerned with improvements to voice quality, which is primarily affected by the signal processing phase. Important methods and advances of the NLP phase are introduced as a reference. However, the goal of this research is to introduce new methods for implementing the signal processing stage of TTS to produce very high voice quality speech. For implementing the necessary NLP stages of the TTS process, the modular Festival TTS system [16], developed at the University of Edinburgh, was used.

The most significant advances in voice quality and intelligibility, in recent times, have been in the area of concatenative text-to-speech (TTS) synthesis. These system produce speech by modifying and concatenating recorded segments with or without the use of a speech model. Commercial and research systems by AT&T, SVOX, Cepstral, Festival, MBROLA Project, and others provide viable solutions for interactive applications that would otherwise require a real human voice. These systems can be classified under three general categories, each with merits in different aspects of the technology: diphone synthesis, unit selection synthesis, and limited-domain synthesis.

Diphone synthesis is based on the concatenation of recorded units at the midpoint of each phoneme. This has been a preferred method for many years due to its ability to synthesize an unlimited vocabulary at the cost of a very small footprint (1000 to 2500 units). Over the years TTS based on diphone synthesis has improved significantly to produce speech with good intelligibility [30]. However, since generally only one instance of each unit is stored in the database, prosodic (segmental pitch and duration) modifications are required for intelligibility. Prosodic modifications are applied by using a speech model such as residual-excited linear prediction (RELP) [5][55] or multi-band resynthesis pitch synchronous overlap-add (MBR-PSOLA) [32] to parameterize the units. For most current speech models, prosody modifications

introduce artifacts of varying degree depending on the extent of modification, class of unit (vowel, consonant, fricative, etc.), and speech model. This coupled with the large number of segment boundaries inherent in diphone synthesis, results in speech that is unnatural in sound quality.

Over the last few years TTS based on unit-selection synthesis [13] has gained wide acceptance due to its ability to produce "customer quality" speech. In this method a tree-structured database is created with numerous instances of units. Units can vary in size ranging from "half phones" to multi-phones (diphones, triphones, syllables, etc.) to phrases. The database is created by choosing optimal units from a speech corpus based on join cost, spectral variations, and target cost, prosodic variations. Hence, the database is rich in context, spectral characteristics and prosody, reducing or even eliminating the necessity for prosodic modifications and boundary smoothing. Unit-selection synthesis has been investigated for many years, however its recent success is attributed to the availability of large computational power, storage capabilities, speech corpus, and automated labelling techniques. For achieving "customer quality" speech, a labelled corpus of over 10 hours of speech is required [81]. Currently available systems have storage requirements ranging from 60 to 500 megabytes with a minimum of 300 megahertz of CPU speed for achieving the highest quality. Though these systems are scalable, the quality is predictably compromised as the database is reduced.

Limited-domain TTS is a popular version of unit-selection synthesis for applications such as telephone banking, telemarketing, information kiosks, etc. Since the vocabulary, context, and subject are limited, the database can consist of larger units of varying prosodic movements. Limited-domain TTS can potentially achieve "natural quality" with a relatively smaller footprint than unlimited unit-selection synthesis. The CMU Communicator [76] is an example of a flight reservation system using limited-domain TTS.

Current implementations of the above methods use either RELP, MBR-PSOLA, or the Harmonic Plus Noise Model (HNM), a variant of the Sinusoidal Model, for representing the speech and applying prosodic variations, if necessary. Though perceptually high quality synthesis is achievable, these methods have inherent modeling errors. These errors can produce audible artifacts at segment boundaries and when applying prosodic movements. Circular Linear Prediction [7] combined with the Constant Pitch Transform (CLP/CPT) provides a more robust model for representing speech that is, theoretically, free of modeling errors. As presented in this thesis, this model can enhance the performance of the current TTS systems by providing a method for high quality prosodic variations. Specifically for unit-selection and limited-domain TTS, this method can reduce the storage requirements by reducing the number of prosodic variations necessary for each unit.

1.2 Contributions of the thesis

To develop a foundation for the research in this thesis, initially, a currently successful synthesis model known as pitch-synchronous residual-excited linear prediction (PS-RELP) [54] model was implemented as a TTS engine. Utterances were synthesized using a limited-domain TTS database to understand the limitations of this model. Due to the poor correlation of parameters at the junctures of concatenated units, this model was found to result in artifacts at these junctures. Prosody modifications can increase the effect of the artifacts resulting in highly audible pops and clicks in the synthesized speech. Based on the idea of pitch-synchronous modeling of speech in PS-RELP, circular linear prediction (CLP) has been implemented as a potentially more robust method for concatenative synthesis. This method determines the LPC parameters by circular autocorrelation removing the need to apply a window to the LP analysis frames by assuming that the signal is exactly periodic. This results in LPC parameters that exactly model the entire analysis frame. The theory for

a fractional pitch detection and analysis algorithm has been developed for accurate implementation of the CLP model for TTS. New prosody matching techniques have been implemented to support this model. Specifically, an efficient pitch modification algorithm is introduced. In addition, a database equalization method has been developed to simulate a uniform recording environment for all units in the database. Finally, the advantages and disadvantages of various types of units have been studied. This has resulted in the use of cepstral distance measures for defining dynamic unit sizes that are based on the junctural phonemes.

The following is a summary of the major contributions of this thesis:

1- Application of the circular linear prediction model with fractional pitch resolution to improve the synthesis of residual-excited LP TTS method.

2- Introduction of the constant pitch transform as a method to create a uniform pitch synthesis database.

3- Application of the inverse constant pitch transform for applying pitch modifications for prosody matching.

4- Introduction of a fractional pitch boundary estimation algorithm for precise pitch-synchronous CLP analysis.

5- A method for spectral normalization to match varying spectral characteristics of speech segments in a TTS database.

6- A new method for reducing artifacts resulting from prosodic modifications by constraining the movement of line spectral frequency (LSF) tracks.

7- Application of the CLP/CPT model to an existing unit-selection based limited-domain TTS system to realize prosodics with varying degrees of emphasis.

8- An analysis method for defining new concatenative units based on spectral characteristics of junctural phonemes and determining the feasibility of candidate unit definitions.

1.3 Organization of the thesis

The thesis begins with the background in Chapter 2, which presents an overview of TTS including and prosody generation and advantages and disadvantages of current concatenative synthesis techniques. Additionally, Chapter 2 includes brief summaries of the linear prediction model, other synthesis models, and a detailed overview of residual-excited LP synthesis, which has been implemented by existing systems. This background forms a basis for the circular linear prediction model, detailed in Chapter 4. In Chapter 3, the problem of optimal unit size is investigated and a method for defining variable size units is presented. The method is applied to text corpora to determine the feasibility of the variable-size units. In addition to an in-depth discussion of the analysis/synthesis methods of CLP, Chapter 4 details the constant pitch transform. Chapter 5 details the implementation of TTS using the CLP model, including database normalization and LSF track constraints. Chapter 6 describes the subjective tests conducted for comparing TTS using CLP versus unit-selection based TTS and TTS with emphasis. Finally, the conclusions drawn from this thesis are presented in Chapter 7.

CHAPTER II

BACKGROUND

2.1 Overview of TTS Synthesis

Figure 1 gives a high-level block diagram of the TTS synthesis process. The purpose of the natural language processing (NLP) phase is to provide the synthesizer with the necessary prosodic and phonemic information. First the incoming text must be accurately converted to its phonemic and stress level representations. This includes determination of word boundaries, syllabic boundaries, syllabic accents, and phonemic boundaries. There are numerous methods that have been proposed and implemented for the text processing. [2] [52] [87]. The next step, referred to as syntactic prosodic parsing, involves the determination of phrase boundaries and phrase level accents. A number of statistical methods for achieving this step have been presented by [93] [19]. These methods involve developing probabilistic models, based on a set of features (i.e. part-of-speech sequences, distance from last phrase boundary, etc.), by parsing a large text corpus. The final step for the NLP phase of synthesis is the determination of prosody values. Prosodic features (i.e. intonation and phonemic duration) are determined based on the phrase accents, syllabic accents, and phoneme location. These features are represented by actual pitch and duration values for each phoneme. Accurate determination of the pitch and duration values is essential for producing more natural sounding speech [33, pages 129–130]. Though, numerous models and methods have been proposed, the problem of realizing correct prosody is far from being solved. A more detailed discussion on prosody is provided in section (2.3). As previously noted, the natural language processing steps discussed above are beyond the scope of this proposal and are not discussed any further here. They have been

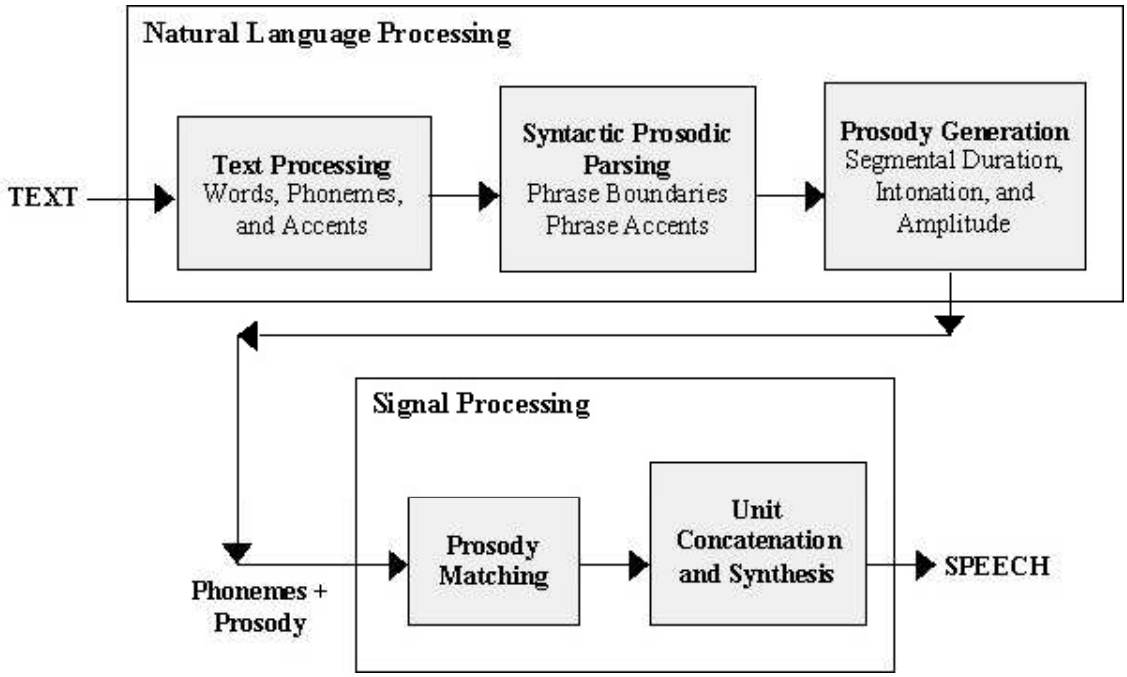

Figure 1: General block diagram of TTS synthesis

introduced to give a general understanding and to provide a frame of reference for the research that is being proposed. Also, some of these algorithms have been used by the research.

The first step of the signal processing phase of TTS is prosody matching. Prosody matching refers to the actual realization of the prosodic features through modification of the pitch and duration values for each unit. This can be implemented using either spectral or time-domain techniques [33, pages 201–269], as discussed in section (2.2). For the initial implementation, this research uses the time-domain pitch-synchronous overlap-add (TD-PSOLA) algorithm for prosody modification, which is detailed in section (2.2.2). Concatenation of the synthesis parameters of the units can be implemented either before or after the prosody matching depending on the method used for synthesis. Finally, after all of the parameters have been modified and concatenated, the waveform is synthesized. The method used for synthesis is essential to the quality of the output speech. The next section gives brief background on waveform synthesis strategies that have been implemented in the past. This research has implemented

Table 1: The relationships between acoustical, perceptual, and linguistical prosody [33, page 130].

Acoustic	**Perceptual**	**Linguistic**
Fundamental Frequency (F0)	Pitch	Tone, intonation, aspect of stress
Amplitude, energy, intensity	Loudness	Aspect of stress
Duration	Length	Aspect of stress
Amplitude dynamics	Strength	Aspect of stress

residual-excited linear prediction as the basic synthesis method.

2.2 Prosody Generation

Prosody refers to the characteristics of speech that make sentences flow in a perceptually natural, intelligible manner. Without these features, speech would sound like a reading of a list of words. The major components of prosody that can be recognized perceptually are fluctuations in the pitch, loudness of the speaker, length of syllables, and strength of the voice. These perceptual qualities are a result of variations in the acoustical parameters of fundamental frequency (F_0), intensity (amplitude), phonemic duration, and amplitude dynamics [33, page 130]. Table 1 summarizes the correlation between prosodic features at the acoustic, perceptual, and linguistic levels. In implementation, however, fundamental frequency and duration are considered the most important prosody parameters.

In spoken language, to increase intelligibility, sentences are usually divided into phrases, which can be perceived as independent prosodic units. Furthermore, levels of stress are dependent on the syntactic structure such as parts-of-speech and phrase/clause boundaries of sentences [2]. Thus, before the acoustic prosodic parameters (i.e F0, duration) can be generated from a given segment of text, the prosodic boundaries of the segment need to be located. Prosodic boundaries are often realized

as short and long pauses in speech. Following this, the stress markers (accents) are then placed on the appropriate syllables. Since all of the prosodic features are related to aspects of stress (see Table 1), the prosody can then be generated from the stress markers.

It is important to note that prosody is not dependent on syntax alone. The semantics, context, intent, and emotion of the speaker/listener all have a significant effect on prosody. It is impossible, however, to determine the intent and emotional state of the speaker, and determining the meaning of the text being parsed would be too complex to implement for real-time systems. It is believed that as long as the synthetic speech is constrained to be "acceptably neutral", syntax-based parsing is sufficient [33, page 129].

2.2.1 Syntactic-Prosodic Parsing

The process of determining prosodic boundaries and the linguistic prosodic parameters is referred to as syntactic-prosodic parsing. An efficient early algorithm to acheive this, which is implemented by MITalk, parses text into noun phrases, verb phrases, and prepositional phrases [3]. These phrases are defined by a set of grammar rules. It was observed that there are many more noun phrases than verb phrases in the English language, and that detecting verb phrases is more difficult than detecting noun phrases. Hence, the algorithm searches for noun phrases, first, followed by verb phrase detection. Simple clause-level tests are applied to verify the verb phrases. This system is useful for real-time systems because of its low complexity. However, the dependence on parts-of-speech causes instances where a verb is detected as a noun, since many verbs in the English Language can also be nouns. Accentuation at the phrase level is simply achieved by accenting all content words.

Liberman and Church proposed an even simpler approach, which defines a prosodic phrase by at least one function word followed by atleast one content word. The parser

begins by searching for only a function word ignoring all other words. Once a function word is found, the parser searches for only a content word. After the content word is identified, the search continues for the next function word. The prosodic phrase constitutes of all words before the next function word. The performance of this algorithm was improved by classifying objective pronouns such as "him" and "them" as content words and tensed verb forms (i.e. produced, helped) as function words. Table 2 shows the difference in the way the two phrasing methods discussed divide the same sentence.

Methods for automatically generating the rules to determine prosodic boundaries are based on probabilistic measures, which are derived from contextual and categorical factors from a large training speech corpus. After the probabilities are obtained, a type of decision tree, known as a classification and regression tree (CART), is generated so that each node has a probability and a set of parameters associated to it. The nodes of the tree are then traversed optimizing the prediction score. Implementation of this method in the AT&T TTS system resulted in detection of 90% of the boundaries [87]. Another probabilistic method for automatic determination of phrase boundaries, which has been utilized by this research, is detailed in [93]. This method uses a Markov model to give the most likely sequence of phrase break locations, based on a sequence of part-of-speech (POS) tags for a given utterance. The model is trained on a large corpus of labeled text. The accuracy of this model, however, is dependent on the accuracy of the automatic POS tagging algorithm. The method used in this research for automatically generating pitch contours for each phrase makes use of linear regression and classification and regression trees (CART) [12]. Another method that is based on a set of phrasing rules determined by a professional speaker's phrasing behavior is given by [79]. The rule set can make phrasing decisions with five levels of boundary strengths.

For determining intonation and locations of stress, the methods by Allen and

Table 2: A comparison of two prosodic parsing methods. The left side shows the result of the method implemented by [3] and the right side shows the result of the method implemented by [52]

Before	**After**
A convicted murderer and rapist whose parole several years ago provoked public outrage has been charged with attacking a woman.	A convicted murderer and rapist whose parole several years ago provoked public outrage has been charged with attacking a woman
I asked them if they were going home to Idaho and they said yes and anticipated one more stop before getting home	I asked them if they were going home to Idaho and they said yes and anticipated one more stop before getting home
and a Kansas state trooper helped them on Interstate 70 near the Colorado border	and a Kansas state trooper helped them on Interstate 70 near the Colorado border
Ellsberg testified Friday that a protester much like those on trial persuaded him to leak the Pentagon Papers	Ellsberg testified Friday that a protester much like those on trial persuaded him to leak the Pentagon Papers

Liberman and Church described earlier implement a simple method of applying stress to the content words in a mildly alternating pattern to prevent the speech from sounding monotonic [52]. This method for phrasal accentuation does not account for the perceptual quality of prominence. For multiple accented words within a sentence, certain words are perceived to have a greater level of stress than others. A number of tests have been performed by [94] [42] to determine the effects of perceptual prominence on the pitch and duration of syllables. It was observed that prominence is an audible quality which can enhance the resulting speech.

A relatively more complex procedure, which accounts for prominence by assigning four different levels of stress, is introduced by [65]. In addition to noun phrases and verb phrases, this method also considers independent phrases, which are phrases that do not modify the subject, verb, or object of a clause or that modify the entire clause. After parsing, each phrase is analyzed to identify patterns that can affect prosody, such as parallel contrast between phrases, conjoined phrases, comparative structures within phrases, and question phrases. There are numerous rules that have been defined for identifying each type of phrase and prosodic characteristics. An additional feature is that this algorithm uses only a 300 word dictionary consisting of function words, prefixes, and suffixes. The suffixes and prefixes are used with a number of rules to identify the POS of the content words. Though, the smaller dictionary calls for a slightly more complex algorithm, the memory requirements are reduced significantly.

2.2.2 Realization of Pitch Contours

Following the syntactic-prosodic parsing to determine phrase boundaries and accentuation (stess assignment) of the text, the acoustic prosodic features can be generated. In terms of intonation, stressed syllables are generally realized by a rise in the pitch

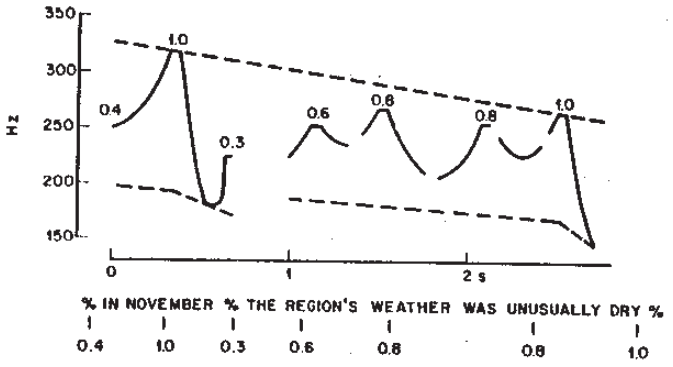

Figure 2: Pierrehumbert's method of interpolating pitch targets for realizing F_0 contours [70]

while unstressed syllables are perceived to be lower in pitch. The pitch (F_0) and duration models developed for earlier TTS systems were based on compilations of rules, which are developed from contextual and grammatical patterns. In 1981, Pierrehumbert [70] introduced a rule- driven method to realize pitch contours by interpolating a set of pitch targets. The target pitch values are based on varying levels of stress and are determined by a set of rules. Pierrehumbert's method used quadratic interpolation between pitch marks resulting in pitch contours consisting of peaks (pitch marks) separated by "sagging" arcs. The resulting pitch contour for a given sentence is shown in Figure 2. The AT&T TTS system was implemented with this method for intonation [95]. It is worth mentioning that a general rule, which has become standard for almost all intonational models, is a slight decline in the envelope of the pitch contour as shown in Figure 2 [95] [96].

A second model, introduced by Fujisaki around the same time, generates contours from two sets of unit functions passed through second order linear smoothing filters (See Figure 3) [36] [37]. The unit functions, representing accent commands, are of varying length and amplitude to represent different levels and durations of stress. There are separate inputs for word accents and phrasal stress, which are filtered

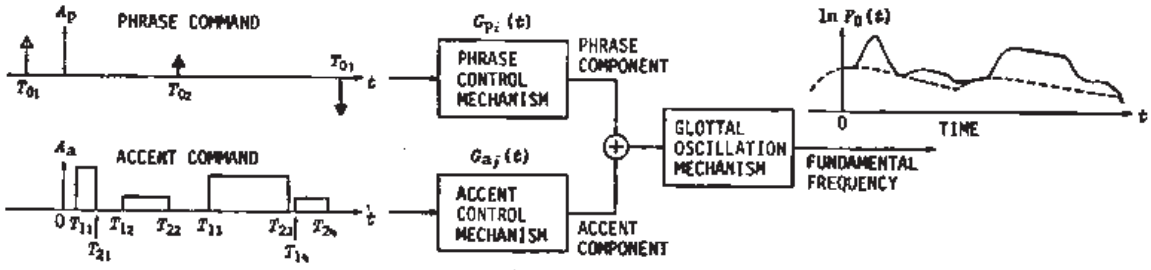

Figure 3: Fujisaki's model for realizing pitch contours [36]

separately and summed. This method results in fast contour generation without the need for too many rules. Due to the smooth contours produced by the Fujisaki model, it has been implemented in numerous TTS systems including Klattalk [47] [37]. Later Pierrehumbert's model was modified to use a filter with the targets for generating F_0 contours [4]. The Fujisaki model is dependent on the method for which the parameters for the accent commands are determined.

Another widely used approach is to construct the F_0 contour from a set of stylized contours [104]. In this method, an inventory of standard pitch movements along with their grammatical cues are extracted from a corpus of speech. Rules are then derived to determine the extent of rise/fall and duration of the contours. The final F_0 contour is then generated by placing a sequence of stylized contours on a declination envelope. It was shown by [96] that for this method, a simple p iecewise linear approximation of the actual F_0 contour "is, perceptually, not inferior to an approximation by means of fragments of parabolas."

One of the F_0 contour determination models implemented by the Festival Speech Synthesis system, and used in this research, consists of building a linear regression model, which outputs target F_0 values for each syllable based on a set of features. For example, features include accent types of current, previous, and next syllables, number of stressed and accented syllables, position in phrase, etc. Three linear regression models (start, mid-vowel, and end F_0 target values) are determined for each syllable

from a large training speech corpus. Each model is of the form:

$$target = I + w_1 f_1 + w_2 f_2 + ... + w_n f_n \quad (1)$$

where f_i are the features and I and $w_1 ... w_n$ are coefficients estimated by the regression analysis on the training corpus. All optimal target values are then determined from the corpus and stored in CART trees for implementation during synthesis. This method has the advantage of not requiring the tedious task of compiling rules by expert linguists.

Though rule-driven and data-driven methods for realizing the pitch contours discussed above are effective and used widely, they are not ideal for natural speech. Compiling a rule database that accounts for all prosodic inflections in a language can be a tedious task, if at all possible. The linear regression model is limited by the database used derive the model. Often the target pitch tracks created by rules or parameters only consist of general pitch movements. These methods cannot create the fine-grained pitch inflections that contribute to natural of speech, resulting in generally monotonic intonation [73].

Recently, corpus-based intonation models have been introduced with success in acheiving "microprosodic" inflections. In this method the F_0 contours are extracted from a database without any modifications to keep the target pitch as natural as possible. The pitch contours, known as pitch templates, are classified based on syntax and/or intonational events. The intonation of an utterance is then created by concatenating the templates that best match the syntax or events of the phrases in the utterance [43]. A fully data-driven approach for corpus-based intonation was proposed by [73]. This method derives the F_0 templates automatically based on a set of parameters influencing intonation, similar to the linear regression approach mentioned earlier. Unique to this method is that the templates can be selected at the segment (diphone or phoneme) level. This enables the model to generate target intonation with microprosodic and macroprosodic movements, resulting in increased

naturalness.

2.2.3 Realization of Phonemic Duration

Unlike pitch rules, which are primarily dependent on levels of stress, duration is dependent on a number of factors, such as syllable location, phonetic identity, surrounding segments, etc. [99] In 1979, Klatt proposed a set of duration rules for syllables, phonemes, and pauses based on observation made by various researchers. Crystal and House performed a number of quantitative tests with various speakers to derive a similar set of rules [23] [24]. These rules have been implemented in the Klattalk synthesizer and other systems with minor modifications. Often, a given phoneme, p, is affected by more than one rule, and the duration, $DURout(p)$, resulting from the application of each rule, r, is given in [47] by

$$DURout(p) = \frac{[PRCNT(r)][DURin(p) - MINDUR(p)]}{100} + MINDUR(p) \qquad (2)$$

where $PRCNT(r)$ is the percentage of lengthening/shortening as prescribed by the rule, $DURin(p)$ is the resulting duration from the application of the previous rule, and MINDUR(p) is the minimum duration threshold for the phoneme. $DURin(p)$ is initialized to the inherent duration of the phoneme. MITalk is also based on a similar rule system for duration with slight differences in the manner and extent of the effect of each rule [3].

An inherent drawback to this type of rule system is the inability to globally optimize the effect of each rule, $PRCNT(r)$. It is difficult, if at all possible, to determine the optimum percentages while simultaneously varying all of the other factors [99]. Similar to the linear regression method for intonation modeling discussed in 2.2.2, duration values can also be found based on statistical data-driven methods. For each phoneme, a probabilistic model is developed by training a classification and regression tree from a large labeled speech corpus. The CARTs are based on a given set of features, such as context (a number of adjacent phonemes to the left and right),

stress levels, lexical position, etc. [19] [45]. The CART tree consists of values representing the number of standard deviations from the mean duration value for each phoneme. This type of model does not place any constraints on the data, and is based completely on statistics derived from a large training corpus. A tree is computed by successively partitioning the space of duration factors until a node exists for all probable sets of factors. Mean durations are associated with each node. For a given input, consisting of a vector of duration factors, the tree is traversed by selecting the branches that correspond to the least variance. Though the use of CARTs is attractive because they remove the long tedious processthe need for rule determination, the performance of this type of system is completely dependent on the training data and its accountability for all the different factors.

A more accurate, though computationally expensive, duration model introduced by Van Santen estimates optimal duration parameters from sum of products type equations based on a number of inequality constraints, such as:

$$DUR(/I/, 1-stressed, f3, ..., fN) \geq DUR(/a/, 2-stressed, f3, ..., fN) \quad (3)$$

where the first two duration factors are vowel identity, V, and syllabic stress, S, and f3,...,fN represent other factors. This inequality above, for example, states that the duration of the vowel phoneme /I/ inside a primary stressed syllable, is always greater than or equal to the vowel phoneme /a/ inside a secondary stressed syllable. In this manner all of the inequalities all of the inequalities for a set of related factors are determined manually based on phonological and phonetic distinctions. From these inequalities and a corpus of training data, a sum-of-products model for determining the duration can be derived which best fits the data. For example, for the inequality above, a possible model is

$$DUR(V, S, f3, ..., fN) = S1(V)S1(S)S1(f3)S1(f4) + S2(f4) + S3(f5)S3(f5) \quad (4)$$

where S1,S2, and S3 are parameters that represent the effects of the duration factors, V, S, and f1,...,fN. The complexity of this procedure is of concern, and it can be reduced by placing constraints that eliminate entire subsets of the space of possible models. Once the best-fitting sums-of-products model has been determined, the optimum parameters can be determined easily using the training data. The key to this method is correct determination of the inequality constraints. Since this is done manually, it is prone to human judgement errors. This duration model has been implemented in the AT&T Bell Labs TTS system.

The problem of accurate prosody modeling is far from being solved. The prosody prediction models discussed above result in speech that is noticeably artificial and sometimes unintelligible. Often prosodics are either under-varied or over-varied when compared to natural speech [12]. Much research in this area is needed before truly natural TTS can become a reality. However, the topics of prosodic phrasing, intonation modeling, and the determination of duration values fall under the category of natural language processing. Since, this research is concerned with the signal processing aspects of TTS relating to voice quality, prosody prediction is beyond its scope and will not be discussed further.

2.3 The Source/Filter Model for Speech Production

The purpose of this section is to give a brief introduction to the source/filter model for speech production, which forms the basis for many speech production models, including the method used for this research. Key details of this method, also well-known as the linear prediction (LP) model, are provided here because it forms the basis of many existing speech synthesis methods including the model presented in this thesis. This model, first introduced by Fant [35], in 1960, has become a very useful tool for speech analysis and applications such as vocoders, speech synthesis, speech modification, speech enhancement, etc. Although, this model is well established

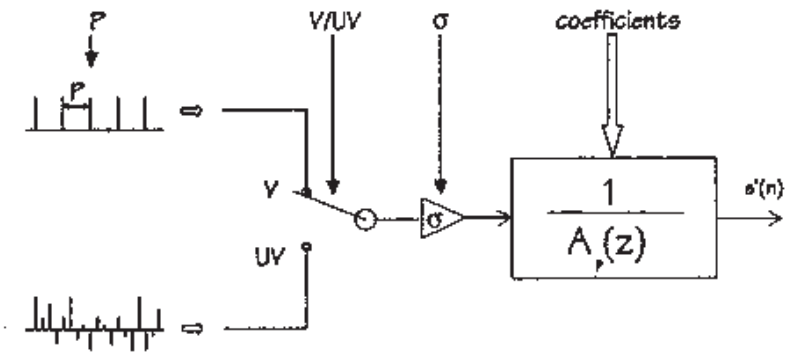

Figure 4: Linear prediction model for speech [6]

and detailed throughout the literature [33] [8] [40], the major concepts have been summarized as a reference for later discussions on derivations of this model. The line spectral pair (LSP) representation of the LP parameters is also summarized here because it has a very useful application in this thesis.

Fant demonstrated that by modeling the vocal tract as a series of concatenated, lossless acoustic tubes, a linear model for speech production can be derived. With the assumption that speech is quasi-stationary, it was found that the lossless acoustic tubes could be modeled by a slowly varying all-pole filter, $H(z)$, of the form:

$$H(z) = \frac{\sigma}{1 + \sum_{i=1}^{p} a_i z^{-i}} = \frac{\sigma}{A_p(z)} \tag{5}$$

where the order, p, refers to the number of acoustic tubes in the vocal-tract model [72]. As shown in the block diagram in Figure 4, the excitation of this filter is either an impulse train for producing voiced speech and zero mean, unit variance, Gaussian noise for unvoiced speech. For voiced speech, the period of the impulse train corresponds to the pitch, T_0, of the speaker. The source/filter model is often referred to as the autoregressive (AR) model of speech production due to the all-pole nature of the filter. For speech synthesis applications, this model is more commonly referred to as the linear prediction (LP) model, since it takes advantage of the linear predictability of speech. It is worth mentioning that this model does not account for

all types of speech production. In reality, the vocal tract is not lossless. Additionally, sounds produced by the nasal tract are not provided for in this model. Also, there are certain forms of speech, such as voiced fricatives, which require dual excitation modes. However, this model forms the foundation for later models which have attempted to account for additional factors.

2.3.1 LP Coefficients

The coefficients, a_i, can be determined by exploiting the fact that speech is a relatively slow time-varying signal that has a smooth linearly predictable waveform. The equation for the prediction of $s(n)$ from p previous samples of speech is given by

$$s(n) = \sum_{i=1}^{p} -a_i s(n-i) + e(n) \tag{6}$$

where $e(n)$ is the prediction error. The term, p is often referred to as the LP order since it represents the order of the inverse filter $A_p(z)$. The coefficients are determined such that

$$\min_{a_0,..,a_p} \sum_{n=1}^{N} e(n)^2 \tag{7}$$

where N is the length of a stationary frame of speech in samples. Combining equations (2.3.1) and (7) results in,

$$\sum_{i=1}^{p} a_i \sum_{n=1}^{N} s(n-i)s(n-j) = -\sum_{n=1}^{N} s(n)s(n-j) \tag{8}$$

where $j = 1, 2, ..., p$. In order to solve for the coefficients, a_i, the autocorrelation method is usually implemented, which assumes that the values of $s(n) = 0$ outside the interval $[0, N]$. Equation (8) can then be rewritten for simplification as:

$$\sum_{i=1}^{p} a_i r(j-i) = -r(j) \tag{9}$$

where $j = 1, 2, ..., p$ and the expression $r(k)$ is defined as:

$$r(k) = r(-k) = \sum_{n=k}^{N} s(n)s(n-k) = \sum_{n=0}^{N-k} s(n)s(n+k) \tag{10}$$

for $k \geq 0$. In matrix form, equation (10) becomes

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ r(2) & r(1) & r(0) & \cdots & r(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(p) \end{bmatrix} \tag{11}$$

Equation 11 shows that equations (9) and (10) result in a Toeplitz structured autocorrelation matrix, which can be efficiently inverted using the Levinson recursion to solve for the coefficients $(a_1, ..., a_p)$ [40]. In addition, the Levinson recursion guarantees a stable prediction filter, $H(z)$. Once the coefficients have been found, the gain, σ, can be easily found by calculating the energy of the residual signal, $e(n)$, from equation ():

$$\sigma^2 = |e(n)|^2 = r(0) + \sum_{i=1}^{p} a_i r(i) \tag{12}$$

A major drawback to the autocorrelation method is the inaccuracy in modeling due to the assumption that $s(n) = 0$ outside the interval $[0, N]$. Multiplying the frame, $s(n)$, by a tapered window $w(n)$, such as a Hanning window, can reduce the distortion caused by this inaccuracy. The autocorrelation, $r(j)$ now becomes

$$r(j) = \sum_{n=j}^{N} w(n)s(n)w(n-j)s(n-j) \tag{13}$$

However, any kind of windowing will introduce distortion due to convolution of the speech spectrum with the transform of the window function. To better understand the advantages and limitations of the LP model, the LP generated spectra for different modes of speech were generated and compare to the FFT spectra as shown in Figures 5, 6, and 7. The LP spectra was generated from speech segments sampled at 16kHz and 16th order LP analysis. In Figure 5, the LP spectrum of the vowel, /aw/, can be seen to be a very good approximation of the actual FFT of the speech segment. However, for the unvoiced fricative, /sh/, Figure 6 shows that the estimated spectrum

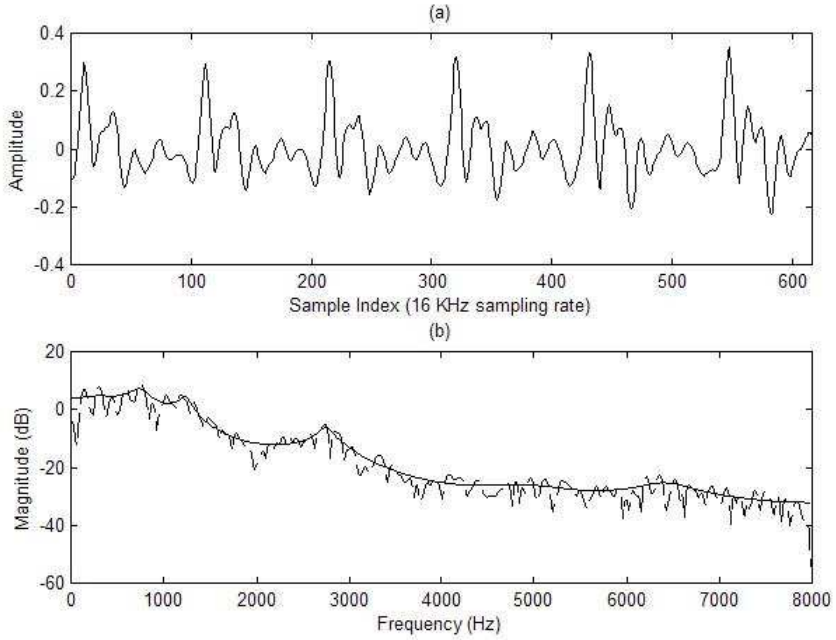

Figure 5: (a) Speech segment of the vowel phoneme /aw/ and (b) a comparison of the FFT spectrum (dashed line) to the LP spectrum (solid line)

has many errors. As mentioned earlier, the prominent zeros of the FFT spectra are not modeled well by linear prediction. For nasal phonemes such as /n/, (Figure 7), the prominent zeros are an integral part of the spectra for natural speech.

A more accurate solution for determining the coefficients a_i is called the covariance method [40]. This method does not window the signal resulting in a covariance matrix, which is a better representation of the signal, in place of the autocorrelation matrix. However, as shown in equation 14 below, the covariance matrix is no longer Toeplitz.

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

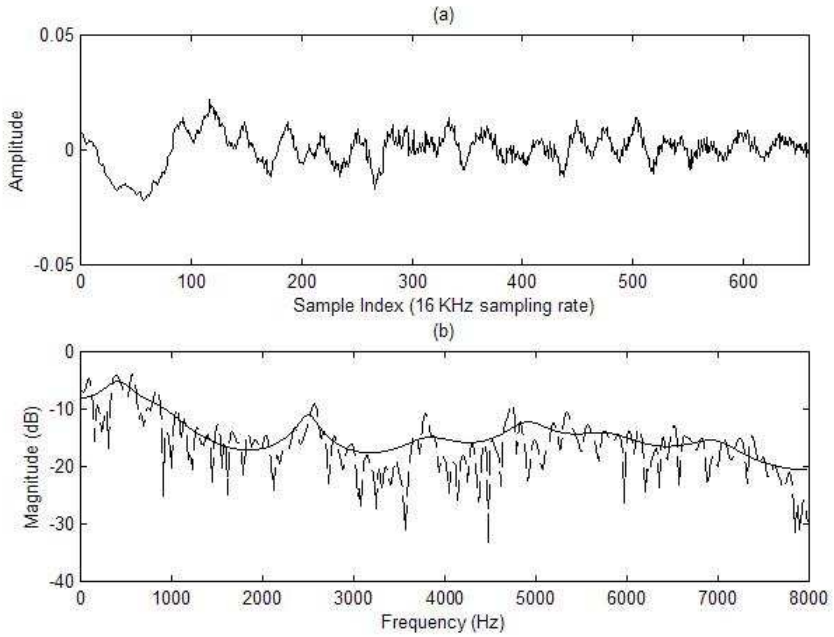

Figure 6: (a) Speech segment of the fricative phoneme /sh/ and (b) a comparison of the FFT spectrum (dashed line) to the LP spectrum (solid line)

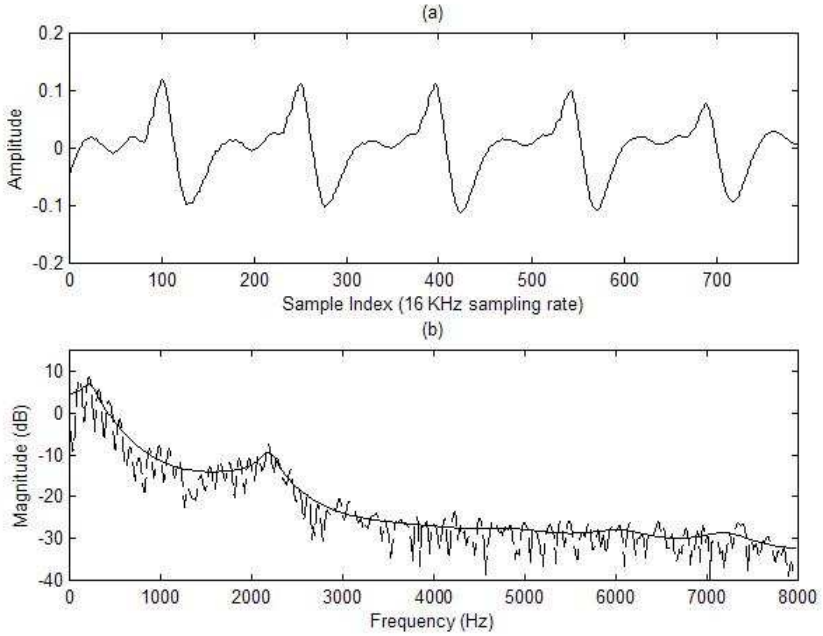

Figure 7: (a) Speech segment of the nasal phoneme /n/ and (b) a comparison of the FFT spectrum (dashed line) to the LP spectrum (solid line)

The non-Toeplitz characteristic of the covariance matrix greatly increases the complexity for matrix inversion. More importantly, the resulting prediction filter is not guaranteed to be stable. Since the matrix inversion cannot be achieved efficiently using the Levinson recursion, the autocorrelation method is generally preferred over this method.

2.3.2 Line Spectral Pair Representation

The complex roots of the coefficients of the linear prediction inverse filter, $A(z)$, lie within the unit circle and correspond to the frequency location of the prominent poles, or formants, of the speech spectra. Since they consist of both magnitude and frequency information, their location in the z-plane is not predictable. There is an alternative representation of the coefficients that consists solely of frequency information that is more useful in many applications. This representatation, called the line spectral pairs (LSP), consists of a pair of polynomials, $P(z)$ and $Q(z)$, which are directly derived from the LP coefficients by modifying the vocal tract model to, first, have only an open glottis, $P(z)$, and, then only a closed glottis, $Q(z)$. However, as opposed to the roots of LP coefficients, the roots of LSPs lie only on the unit circle [49] [44]. In turn, their movements in time are more predictable and gradual compared to LP coefficients. For speech coding applications, these properties lead to a more efficient method for quantization of the LP coefficients. Furthermore, the LSPs corresponds to the bandwidth and approximate location of the formant frequencies. Hence, the LSPs provide an efficient means of obtaining information about formant locations and bandwidths directly from the LP coefficients [22].

2.4 Speech Synthesis Methods

This sections briefly discusses various speech synthesis models that have been implemented for TTS applications. Implementations over the last 2 decades of TTS based on linear prediction, sinusoidal model, harmonic plus noise model, and time domain

synthesis are presented. Finally, the highly successful unit-selection based concatenative synthesis technique, which has been implemented with and without speech models, is detailed.

The various speech synthesis models that have been implemented in the past can be classified into two main subcategories: synthesis by rule and synthesis by concatenation. Rule-based synthesis refers to the generation of speech production parameters for a given set of phonemes and prosodic features based on a set of predetermined rules. Parameters generally include formants, antiformants, and their respective bandwidths. These parameters are found by performing LP analysis on a large speech corpus consisting of all the phonemes in a language with various prosodic modes. The prominent poles and zeros of the linear prediction inverse filter (section 2.3) correspond to the formants and antiformants, respectively. The parameters are analyzed to determine an optimal set of rules for generating them with varying prosodics. Rule determination is a tedious process conducted by experts by studying the effects prosodic variations to every phoneme. Additionally, the transition from one phoneme to the next, known as coarticulation effects, need to be accounted for. A unique set of rules are required for every phoneme transition. To resolve this problem of coarticulation, a rule-based synthesizer using diphones rather than phonemes was presented by [74]. Since diphones begin and end at the middle of phonemes, the coarticulation is usually contained within the unit. The early synthesizers of the late 1970s and early 1980s, such as the MITalk, Klattalk, and INFOVOX systems fall in the category of rule-based synthesizers. These synthesizers, though intelligible, heavily compromised the voice quality due to the difficulty in determining an optimal set of rules. [47]

As opposed to producing speech from completely artificial means, concatenation-based synthesizers start with parametrically coded segments of prerecorded speech. As shown in the block diagram in Figure 8, the analysis stage requires segmenting the

speech corpus into appropriate units and determination of contextual and prosodic parameters of each segment. These parameters are the key to accurate unit selection and prosody matching during the synthesis phase. The parameteric segment database is, then, equalized, coded base on the speech model used, stored in a segment database for synthesis [33]. During synthesis, the segments that best match the the phonemes and prosodic features to be synthesized are selected. The selected segments are, then, appropriately modified for prosody matching, concatenated, and synthesized for speech production. The decoding block shown in the figure can occur either before prosody matching or during synthesis, depending on the speech model used. The quality of speech produced is largely dependent on the method of preparation of the segment database and the model used for synthesis.

Compared to rule-based TTS, concatenative TTS simplifies synthesis to an extent since the rules for speech production do not need to be determined. However, this method introduces the challenges of prosodic modifications to speech segment and resolving discontinuities at segment boundaries. Numerous methods for concatenation-based synthesis have been implemented in the past including linear prediction models, the sinusoidal model, hybrid harmonic models, and time-domain methods [83]. This research has implemented a form of pitch-synchronous residual-excited linear prediction (PS-RELP). This method combines both the linear prediction models and time-domain methods for synthesis. The next sections give a summary of earlier concatenation-based synthesis models that have been implemented for TTS.

2.4.1 TTS Based on LP

The speech production model described in 2.3, also referred to in speech applications as the linear prediction (LP) model, was implemented by early concatenation-based synthesizers [21] because of its efficiency in terms of storage and computation. In the TTS implementation of this model, each speech segment is analyzed to generate the

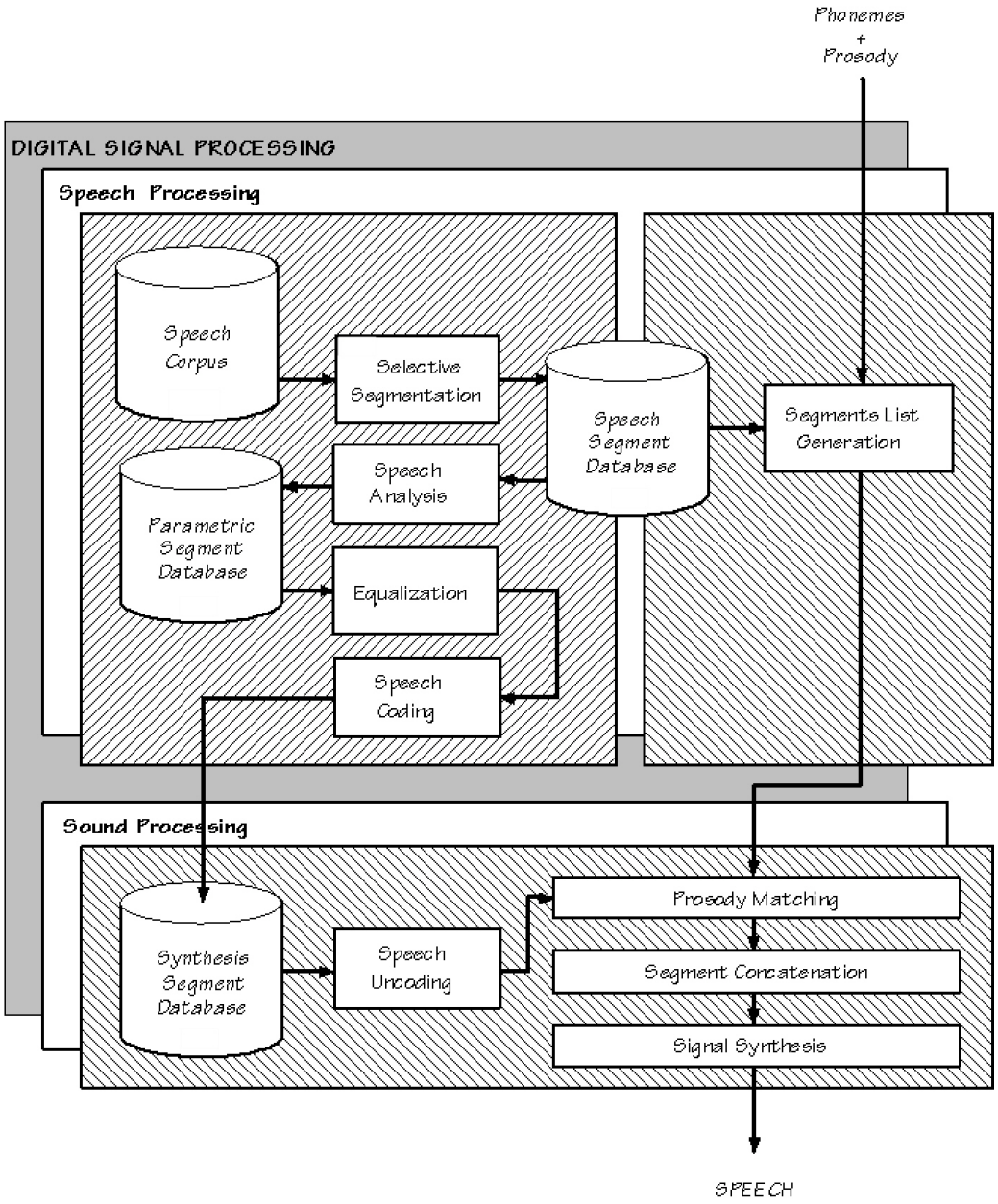

Figure 8: Block diagram of the concatenation-based synthesizers [33]

parameters for the synthesis database. Due to the stochastic, wide-sense stationary properties of speech signals, the time-varying coefficients, a_i, often remain relatively constant for a duration of 10-20 milliseconds. Hence, the input speech segments are divided into frames of this duration for analysis. Each frame of speech is represented by the set of coefficients, $(a_1,...,a_p)$, along with the V/UV switch, σ, and F_0. The number of coefficients required is related to the sampling rate of the speech segments. Generally, since the accepted bandwidth for human voice is considered to be 4 kHz, speech segements are sampled at a minimum of 8kHz (Nyquist bandwidth). At this sampling rate, 10th order LP analysis is considered sufficient. For higher quality systems, a sampling rate of 16kHz and LP filter order of 16 is also common.

Prosody matching for this model can be accomplished with relative ease. Since pitch is just the excitation parameter for voiced speech, the desired pitch, F_0, can be realized by making the period of the excitation impulse train $1/F_0$. The duration of a segment can be modified by matching the output frame length of the filter to the desired duration. For concatenation of segments, the LP parameters are interpolated at the join points. It was found that linear interpolation of the line spectral pair (LSP) representation of the LP coefficients produces smoother results than direct interpolation of the actual coefficients [33]. An inherent advantage of using the LSP representation of the coefficients is the ability to achieve a very high compression ratio [22] providing for relatively low memory requirements for the segment database.

The key advanatages of the linear prediction model for TTS is its efficiency in storage (compression) and low complexity. Initial implementations of the LP model, however, resulted in buzzy speech. The error signal, $e(n)$, is not completely whitened and still contains important spectral characteristics of the speech. This is partly because traditional LPC does not model the mixed-excitation that occurs at voiced/unvoiced transition regions [61]. In addition, as indicated earlier, the AR model does not account for nasal sounds [6]. In the case of plosives, which consist of brief bursts of

noise, the long (20 ms) frame length leads to poor performance. Finally, as mentioned earlier, the autocorrelation method used for parameter estimation has inherent spectral modeling errors, caused by windowing. Advances in linear prediction techniques for speech over the last 20 years have led to much improved performance of synthesis. The segmental quality has been improved significantly with the advances in LPC coding techniques, such as multipulse excited LP (MP-LPC). An implementation of a MP-LPC TTS synthesis system by [100] resulted in higher quality speech than the rule-based DECtalk synthesizer. The introduction of codebook-excited linear prediction (CELP), has resulted in even higher segmental quality using the linear prediction model. An implementation of speech synthesis using CELP for the Portuguese language has been detailed by [25]. By relaxing the storage constraints, recently residual-excited linear prediction (RELP) has been implemented resulting in very high quality speech [55] [54]. This is because the entire residual signal is stored as the excitation. This method achieves prosody modifications on the residual signal directly in the time-domain. This research initially implemented this approach, which is detailed in section 2.5. This method forms the foundation for the CLP/CPT method detailed in chapter 4.

2.4.2 The Sinusoidal and Hybrid Synthesis

Sinusoidal modeling of speech signals, proposed by MacAulay and Quatieri [59], has been implemented in various speech synthesis systems [58] [56]. Though, it has not been implemented for this research, some of the concepts were studied and incorporated. This section gives a brief summary of the model and its implementation.

In this method, speech is modeled as a sum of a number of sinusoids of time-varying frequencies and amplitudes. The parameters are determined by taking the short-time Fourier transform of each windowed frame of speech and choosing the instantaneous frequencies and amplitudes that correspond to a small number of spectral peaks. The

phases corresponding to the peaks for each frame are represented by a third order polynomial [58]. Prosody modifications can be achieved with relative ease since pitch and frame rate are parameters of the model. The pitch is modified by a simple linear interpolation of the harmonic peaks representing each frame to the new frequencies. The durations are modified by changing the reference time-instants for each frame.

The sinusoidal model is a more theoretically correct method for synthesis than linear prediction, since it synthesizes speech from actual formant tracks. Additionally, modeling the spectra based on the FFT peaks (harmonics) results in a better spectral estimate than LP. Figure 9, shows the FFT spectra, LP spectra, and the peak values for the sinusoidal model. It is clear that choosing the FFT peaks models the original FFT spectra more precisely than the spectral estimate from LP.

The model has resulted in synthesized speech that is of higher quality than the classical LP based systems. Prosody modifications result in fewer artifacts. An improvement to this method was proposed by [105], that combines the sinusoidal model with the techniques of LP all-pole modeling. This method, called the sinusoidal + all-pole representation, is used for resolving spectral mismatches at segment boundaries. First, a frequency warping function is derived by mapping the dominant poles of the LPC spectrum of a given frame to the LPC spectrum of a target frame. The frequency warping function is, then, applied to the sinusoidal parameters of the given frame to generate the spectrally modified paremeters of the target frame. However, some buzziness, especially in the case of voiced fricatives, has been reported for this model. [31] [1]. As can be seen in Figure 9, the poles are modeled with high precision, but the prominent zeros can be difficult to capture consistently. A significant drawback to this model is that the number of parameters (amplitudes and frequencies) required for the sinusoidal model is much greater than that of linear prediction modeling. Additionally, the required frequency domain processing results in very high computational complexity, when compared to linear prediction based methods.

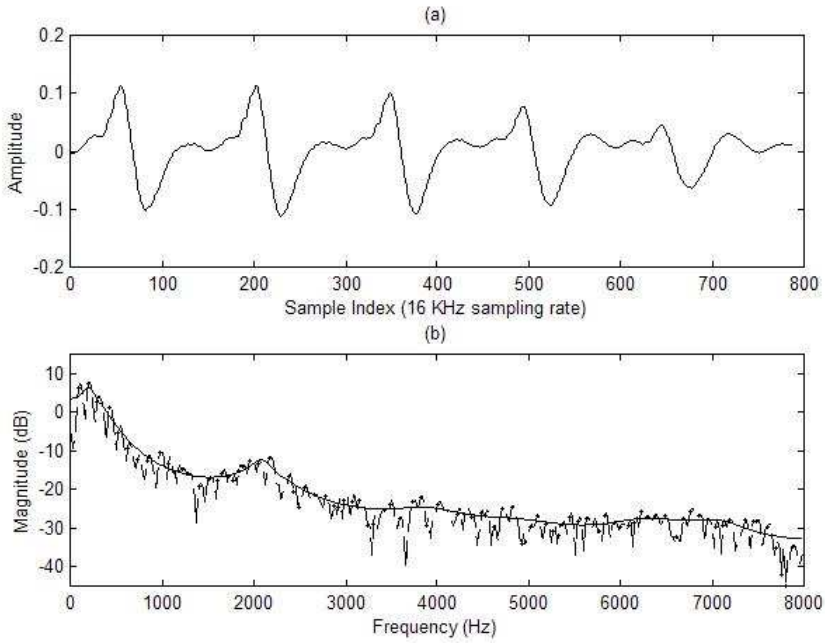

Figure 9: (a) Speech segment of the nasal phoneme /n/ and (b) a comparison of the FFT spectrum (dashed line), the LP spectrum (solid line), and the sinusoidal model peaks (dots)

The classical sinusoidal model has been extended to a number of hybrid models which introduce a time-varying, stochastic component for improving modeling of unvoiced speech[1][51]. The equation below describes the hybrid model as the sum of the classical sinusoidal model component, $s_p(n)$, and the stochastic component, $s_r(n)$:

$$s(n) = s_p(n) + s_r(n) = \sum_{i=1}^{L} a_i(n)cos(\phi_i(n)) + s_r(n) \tag{15}$$

where L is the number of harmonics accounted for and $\phi_i(n)$ is determined by

$$\phi_i(n) = w_i(n) + \phi_i(n-1) \tag{16}$$

The hybrid model from the equations can be interpreted as shown in Figure 10, where the quasi-periodic excitation, $e_p(t)$, and the random excitation, $e_r(t)$, are filtered separately to result in a harmonic and a stochastic component for speech. The parameters of the model are determined by first assuming only the sinusoidal component, $s_p(n)$,

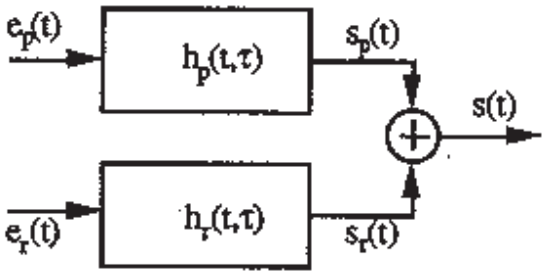

Figure 10: Block diagram of the hybrid model for speech synthesis [1]

and calculating the amplitudes and phases that minimize the weighted time-domain least squares error between the original signal, $s(n)$, and the harmonic component $s_p(n)$. A number of methods have been proposed for modeling the stochastic component, $s_r(n)$. In one method, the noise component is modeled as a sum of narrow band bandpass random signals, in which the amplitudes of the bandpass filters are determined to fit the spectral residual error, $R(\omega)$, between $S(\omega)$ and $S_r(\omega)$. An inherent feature of this model is that no voicing decision is required [1].

In an alternative method proposed by [51], which has also been implemented in the CHATR system [18], the stochastic noise component is modeled in the time domain, as white gaussian noise, $\eta(n)$, passed through an all-pole filter, $H(z)$, which shapes the noise to the error signal, $r(n)$, between $s(n)$ and $s_p(n)$.

$$s_r(n) = w(n)[h(n) * \eta(n)] \tag{17}$$

where $w(n)$ is an energy envelope function, which is based on the error between the original signal, $s(n)$, and the harmonic component, $s_p(n)$. The all-pole filter $H(z)$ is determined using the LP model by using the least squares method to minimize the residual error calculated in the time domain. The filter coefficients of $H(z)$ are then determined to fit the spectral density function of $r(n)$ using correlation methods for spectral estimation. It was reported that modeling the stochastic component in the time domain in this manner improved the performance of hybrid synthesis.

For matching the prosody during synthesis based on the hybrid model, duration

changes are achieved by changing the frame rate, as for LPC based models. However, as the duration changes, the phase needs to be propagated from the first frame to the last one. For changing the pitch, even though the fundamental frequency, F_0, is a parameter of the model, the harmonic amplitudes need to be adjusted for the new frequencies. This can be achieved through piecewise-linear interpolation of the original amplitudes [31].

Synthesis of the speech waveform from this model has been implemented with two different methods. One approach is to linearly interpolate the harmonic amplitudes, frequencies, and the stochastic component parameters from one frame to the next to maintain the continuity of the harmonics. Care needs to be taken to account for the birth and death of harmonics which most likely occur at phoneme transitions. The speech waveform is then synthesized simply by using equations (5) and (6) [58]. The number of computations required by this approach can be very large, depending on the number of harmonics and stochastic parameters for a given frame. A less computationally complex approach, introduced in [20], is the overlap-add (OLA) method. This method has gained popularity with other synthesis techniques and is detailed in the next section.

The synthesis quality of the harmonic plus noise model (HNM) is very high compared to traditional LP synthesis and an improvement on the traditional sinusoidal model. Hence, this method has been implemented in the AT&T Next Generation TTS system [92]. However, the computational complexity due to the frequency domain modeling is very high. Stylianou has presented a fast method for implementing the synthesis using delayed multi-resampled cosine functions (DMRC) [88] [90].

2.4.3 Time Domain Synthesis

The time domain pitch-synchronous overlap-add (TD-PSOLA) method, introduced by and implemented for TTS by Moulines et al.[62][38], synthesizes speech by concatenating the actual waveforms without the use of a speech model. This approach implements pitch and time-scale modifications in the time domain, pitch synchronously, based on [62]. Since the segments do not need to be synthesized from a parametric form, the resulting speech can be produced with very high quality and low computational complexity. Clearly, the segmental quality is the highest acheivable, and as good as the quality of the digital sampling.

The TD-PSOLA algorithm first analyzes the segments using a pitch extraction algorithm. Pitch marks are then placed at pitch-synchronous intervals for voiced speech and regular fixed intervals for unvoiced speech. These pitch marks indicate the center of the overlap-add (OLA) frames. The frame length, L, is adapted to the local pitch period to maintain a constant OLA factor, F_R, according to the expression

$$F_R = L/T_0 - 1 \tag{18}$$

For example, when $F_R = 1$ (a typical case), the OLA frame length will always be twice the pitch period of each frame.

The parameters for prosody matching that are stored along with the waveforms are the location of the pitch marks, M_i, phoneme durations, and segmental boundary locations. The pitch period of each frame is simply calculated from the pitch marks, $T_{0i} = M_{i+1} - M_i + 1$. The duration and pitch of a segment are modified simultaneously in the following manner:

- The analysis pitch mark locations, M_i, are mapped linearly to their new locations as specified by the target duration factor, $DURFACT$ for the phoneme as shown in equation (19) below.

$$M'_i = M'_{i-1} + T_{0i} DURFACT_i \tag{19}$$

- Next, a set of synthesis pitch mark locations, M'_j, are created on a new time axis of the same length as the modified duration. The pitch marks are derived from the target pitch, $T_{0_{TARG}}$ of the segment.

- The locations for each of the synthesis frames are determined by mapping the pitch mark locations of M'_j to those of the origianl segment, M_i that are the closest distance in time.

When the duration is shortened and/or the pitch period is increased, some of the original pitch marks may have no corresponding synthesis pitch mark. In this case the analysis frame is dropped. Likewise, when the duration is increase (slower speech) and/or the pitch period is decreased, often there are more than one synthesis pitch marks corresponding to an analysis pitch mark. In this instance the corresponding pitch marks are duplicated. The dependency of the frame length on pitch compensates for the increases and decreases in the number of frames [60].

Synthesis is then achieved simply by overlap-adding the frames as shown in Figure 11 and described by the equations below:

$$s(n) = \sum_{j=-\infty}^{\infty} s_j(n - M_j) \tag{20}$$

where

$$s_j(n - M_j) = \alpha s(n) w_j(n - M_j) \tag{21}$$

and where $w_j(n)$ is a weighting window, of modified length L'. The modified length of the synthesis window is found by multiplying the analysis window by the pitch factor, i.e. the ratio of the synthesis pitch period to the analysis pitch period:

$$L' = \frac{T_{0j}}{T_{0i}} L \tag{22}$$

The factor, α, is introduced to compensate for pitch fluctuations. Following the overlap-adding of the frames, a smoothing function is applied to allow the differences in adjacent segmental waveforms to evolve gradually.

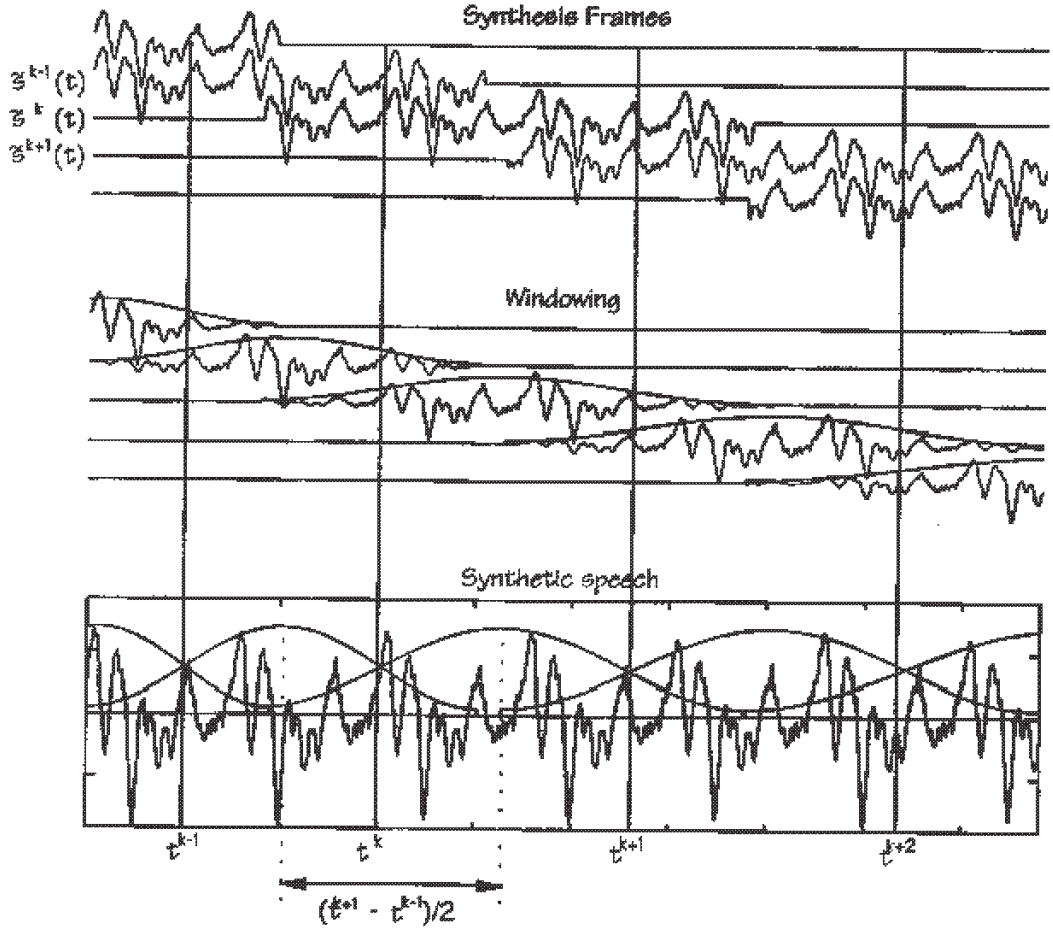

Figure 11: The overlap-add method for the synthesis of speech frames [33, page 130]

Because of its high segmental voice quality and the decreasing cost of memory, TD-PSOLA has become a popular TTS synthesis technique. The IBM Trainable Speech Synthesis System [27] [29] utilizes this approach. It has been observed that TD-PSOLA does not work well with unvoiced sounds such as fricatives and plosives [98], and with breathy or creaky voices [68]. A key drawback to the implementation of purely time-domain synthesis is that, due to the non-parametric nature of the speech segments, the resulting speech can have spectral and phase incoherencies at the boundaries between segments leading to audible aritfacts. However, combining the concepts with either the hybrid harmonic models or linear prediction can resolve these mismatches. Multi-band resynthesis pitch-synchronous overlap-add (MBR-PSOLA), introduced by [32], combines TD-PSOLA with the hybrid harmonic plus noise model. In this method, the database is resynthesized to a fixed F_0, using the HNM model, in order to eliminate phase mismatches. Prosody matching and concatenation is then performed according to the TD-PSOLA algorithm described above. The AT&T Next Generation TTS system mentioned in the previous section also uses the HNM model for synthesizing the speech segments, while using PSOLA to apply modifications and concatenation. Residual-excited linear prediction (RELP), discussed in section 2.5, combines the LP model with TD-PSOLA by applying the time-domain techniques to achieve concatenation and prosody modifications on the residual (error) signal, rather than the speech signal. A study by Dutoit analyzed the HNM, TD-PSOLA, and MBR-PSOLA models for synthesis based on intelligibility, naturalness, and fluidity [30]. Though, all three algorithms were very close in performance, the HNM model and MBR-PSOLA can be considered superior due to significantly lower storage requirements and fewer mismatches during segment concatenation.

2.4.4 Unit-Selection Based TTS

Concatenative TTS systems that synthesize speech by applying prosodic modifications to units, based on a speech model, have been successful in producing speech of high intelligibility. However, the voice quality and "naturalness" degrades with the degree of prosodic movements. This observation is the motivation for unit-selection based TTS. Many high quality text-to-speech (TTS) systems available today are based on the selection of concatenative units from a database that is both contextually and prosodically rich. Since the database consists of numerous instances of each unit with varying prosodics, modifications are either minimized or not necessary at all. Rather the unit that matches closest to the target prosody is selected and concatenated. Since the prosodics occur naturally withing the speech, these systems are capable of producing speech of "natural" quality.

This method was first introduced by Sagisaka et al., for a Japanese TTS system that synthesized speech by selecting units of varying size [77]. This system did not work well for the English language due to differences in the number of phonemes and prosodic variations. Hunt, Black, and Campbell expanded on this system with a new method for selecting units that accounted for a greater number of phonemes and more varying prosodics [11] [13]. This system, called CHATR, is designed to be language independent and was demonstrated for both English and Japanese. As mentioned earlier, the AT&T Next-Gen TTS system utilizes the CHATR unit-selection method, and combines it with the HNM model for resolving mismatches [9].

In the unit selection model of the CHATR system, each unit is labeled with a set of features during the analysis phase. The feature vectors consist of phonetic context, prosodic context, pitch, duration, and power. During synthesis, the model selects the best unit from the database based on a "unit distortion" and a "continuity distortion". The first is a measure of the differences between the features of the candidate unit and the target unit. The second is a measure of the difference between

the features of the candidate unit and the previous adjacent unit. The unit and continuity distortions are defined by a target cost function, C^t, and concatenation cost function, C^c, respectively. The target cost function given in equation (23) is the weighted sum of target sub-costs, C_j^t, which are the differences between each of the unit and target features.

$$C^t(t_i, u_i) = \sum_{j=1}^{p} w_j^t C_j^t(t_i, u_i) \tag{23}$$

where i indexes the target and candidate units, and j indexes the feature vector. Similarly, the concatenation cost function (equation 24) is the weighted sum of the concatenation sub-costs, C_j^c, which are the difference between the features of the current unit and previous unit.

$$C^c(u_{i-1}, u_i) = \sum_{j=1}^{q} w_j^c C_j^c(u_{i-1}, u_i) \tag{24}$$

The unit selection is performed by finding the set of units, $\bar{u}_1^n$ that minimizes the total cost of an utterance of n units given by:

$$\bar{u}_1^n = \min_{u_1,\ldots,u_n} [\sum_{i=1}^{n} C^t(t_i, u_i) + \sum_{i=1}^{n} C^c(u_{i-1}, u_i)] \tag{25}$$

Clearly the computational complexity of this algorithm is extremely large considering that the database required for high quality TTS consists of 50,000 to 100,000 units. The CHATR system implements a Viterbi search combined with pruning of the search space to achieve near real-time synthesis on a Sun SPARC-Station 20. The weights of the sub-costs, w_j^c and w_j^t, are determined by regression training on the synthesis database as detailed by Hunt and Black in [13].

The search methods were later improved upon by clustering similar units into a binary tree-like structure. This technique, initially called context-oriented clustering by Nakajima [64][63], is clusters the database automatically based on the feature vector. The synthesis database is continually split into smaller and smaller clusters, based on the feature with the greatest variance within each cluster. Unit selection in

the resulting clustered database, called a classification and regression tree (CART), can be performed with minimal comparisons, significantly speeding up the search procedure. Methods for automatically clustering units into a CART structured synthesis database is detailed by Black and Taylor in [15]. Ding et al further improved upon the methods for calculating concatenation cost by adding a perception-motivated, "predicted MOS", parameter to the feature vectors [26]. This work, which was based on the Japanese language, also proposed a method to determine F_0 slope discontinuities at unit boundaries and suggested making partial prosody modifications to resolve the mismatch. The work in this research can be an alternative viable solution to resolve this problem.

Limited domain TTS, which can be viewed as a specialized version of unit selection based TTS, was introduced by Black and Lenzo at the Carnegie Mellon University [14]. Motivated by the observation that many applications use a finite number of fully recorded prompts or "slot-and-filler" templates to maintain high quality, limited domain TTS was investigated as an option for reducing storage and development time, while maintaining or improving quality. Just as in unit-selection synthesis, the units are clustered and organized using CART methods. However, in this case, the database consists of phrases, words, and phonemes. Additionally, the CART will have less depth than for general unit-selection, resulting in faster synthesis. For testing purposes, a simple talking clock application was developed, which had the basic "slot-and-filler" template:

- The time is now, EXACTNESS MINUTES HOURS DAYPART.

An example utterance for this template would be:

- The time is now, a little after quarter past two in the afternoon.

This system could be developed in a fairly short time after recording the prompts and fillers, using automatic labeling techniques. The resulting speech was found to

have no notable errors 60% of the time, which improved to 90% with hand correction of the labels. This idea was expanded to improve the quality of a system for an application that is not truely limited-domain, however consists of a large amount of repetition. This system, called the CMU Communicator, is telephone based dialog system for making flight, hotel, and car reservations. In this system, the majority of common template and filler segments have been prerecorded, and out-of-domain words/phrases can be synthesized by a backup diphone synthesizer. Based on the same general idea, Donovan et al. introduced a limited domain TTS system that synthesizes high quality speech from a "splicing inventory" and a "core inventory" [27]. The splicing inventory consists of recorded phrases and words that recur often. The out-of-domain "filler" words are synthesized using the core inventory of the IBM Trainable TTS system. The CMU Communicator synthesis database has been made available for research purposes. Hence, it has been used by this research for developing the experiments to demonstrate the contributions.

It was stated by Black and Campbell in [11] and Stylianou in [88] that even the best unit selection system will produce suboptimal units in a finite data base. Hence, further signal processing to resolve concatenation and target mismatches can improve the performance. In the CHATR system, TD-PSOLA based methods were implemented for modifications. As mentioned earlier, the AT&T Next-Gen TTS system combines the unit-selection method of CHATR with the HNM model for higher quality prosodic and concatenation modifications. Interestingly, further analysis by Beutnagel et al., reported in [9], revealed that the AT&T Next-Gen TTS system performed significantly better without any prosodic modifications in a mean opinion score (MOS) subjective study. Furthermore, pruning the database to increase efficiency sacrificed synthesis quality.

Unit-selection based TTS has proved to syntehsize speech from text with the highest quality. In a limited-domain environment, it can even achieve "natural" quality.

However, the storage requirements and computational expense for realizing the highest quality is a concern. Though, in recent times, memory has become available at considerably lower prices, the size of the footprint is expensive for mobile applications that use these systems (PDAs, cell phones, navigation, etc.). The sinusoidal and HNM models can reduce the storage requirements without a significant effect on quality, however the computational complexity increases drammatically. Traditional linear prediction based synthesis, though lower in storage requirements and complexity, has not been successful in modifying prosodics without affecting quality. The CLP/CPT method presented in this thesis provides the advantages of LP based modeling and limited prosodic modifications without perceivable degradation in quality.

2.5 Residual-Excited Linear Prediction

This sections describes TTS based on RELP-PSOLA in detail. This TTS method is presented here because it is a common TTS approach that was initially implemented by this research to present a high quality TTS system using words and diphones as units. The implementation was helpful in understanding the limitations of LP based synthesis and develop a new method to address these limitations. The Festival TTS system, developed at the CSTR at University of Edinburgh, combined with the OGIresLPC synthesizer, developed at the Oregon Graduate Institute [55], was used to aid in the implementation of this method. Festival is a modular system available with source code for research purposes. Festival allows research in one area of TTS without having to redevelop all the other modules necessary for a fully functional system [16]. In this research, the front-end for the synthesis system (i.e. text preprocessing, syntactic-prosodic parsing, target prosody generation, etc.) was realized using the NLP module and prosody generation models existing in Festival [67][93]. This was useful to compare the results of implementation by this research directly to

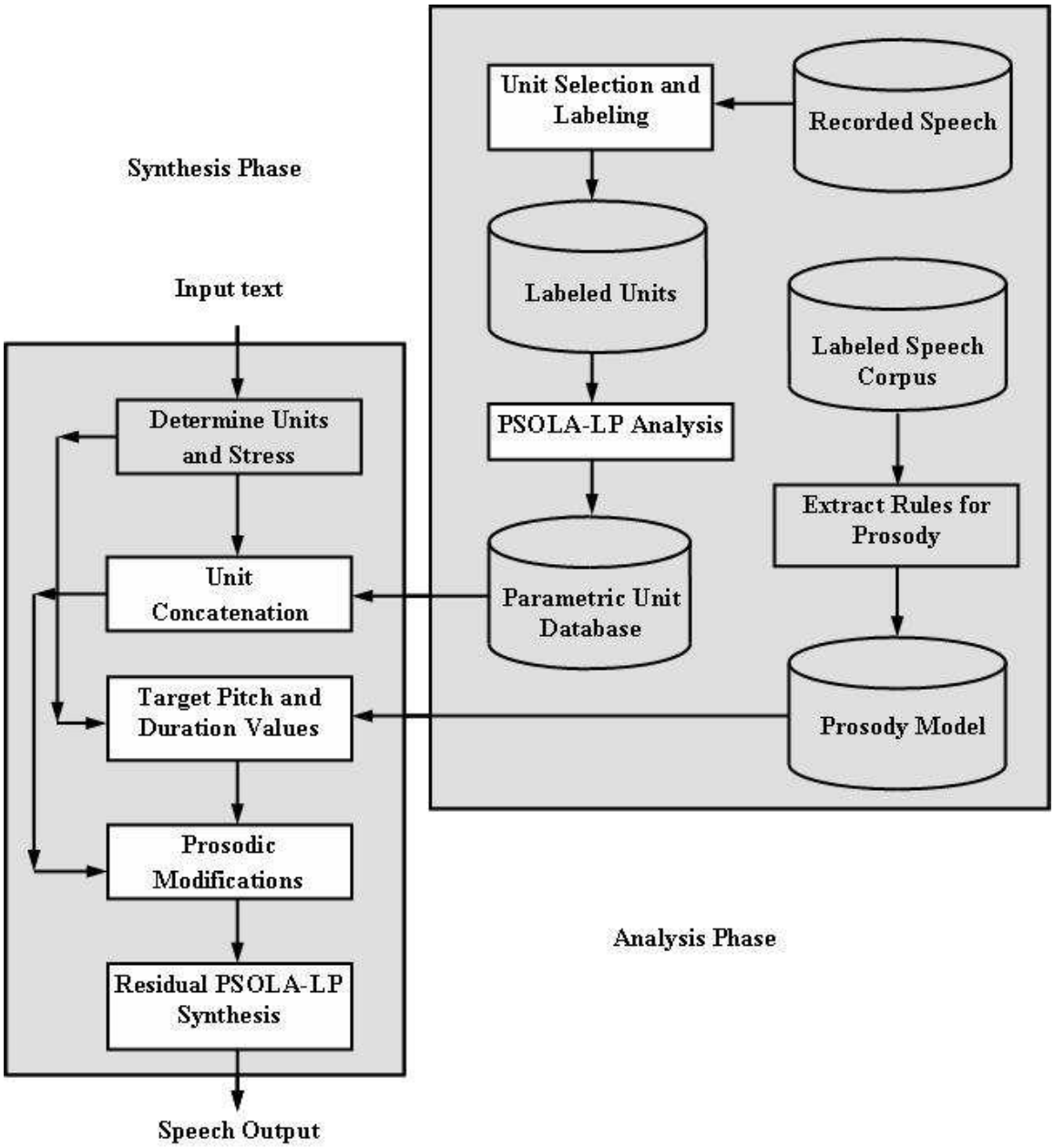

Figure 12: Block diagram of the implementation of the signal processing phase of TTS

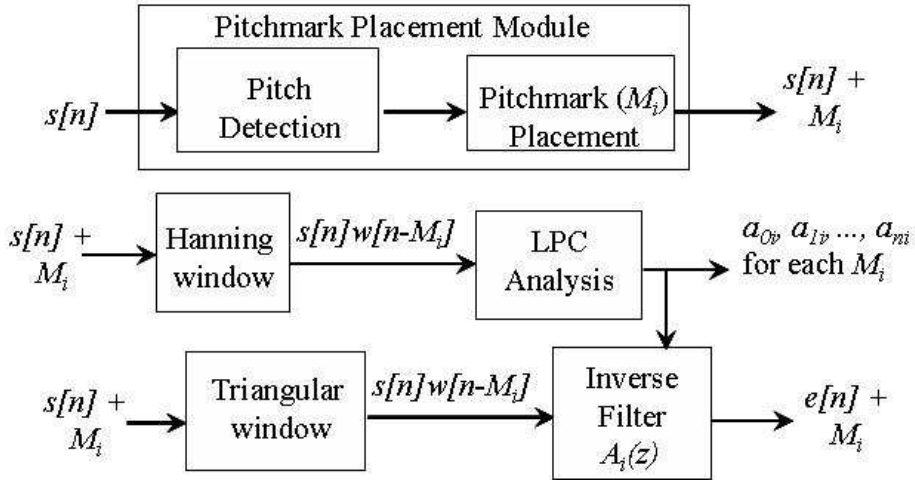

Figure 13: Block diagram of pitch-synchronous LP analysis

Festival's diphone-based synthesizers. Some of the methods implemented within Festival are based on, or are similar to, the NLP methods described in previous sections. Additonally, to analyze the voice quality of the RELP-PSOLA synthesis method, the prosody parameters of NLP module outputs were modified to match prosody parameters extracted from naturally spoken recordings of reference utterances. This way the synthesized utterances can be judged purely on the voice quality without being affected by the unnatural synthesized prosodics. At the back-end, the OGIresLPC diphone synthesizer was modified to allow for larger units for experimentation. The following sections describe the analysis and synthesis phases for implementation of the RELP-PSOLA synthesizer.

2.5.1 Analysis Phase

The high-level block diagram of the analysis and synthesis phases for the RELP -based TTS implementation is shown in Figure 12. The Boston University FM radio speech corpus, which consists of prosodically labeled speech recorded from news broadcasts for four different speakers [66], was used for the recorded speech corpus.

The parameter database was created by, first, extracting the desired units from

the recorded speech. A limited number of units, consisting of words and diphones necessary to synthesize a number of test utterances, have been selected from the male speaker, "m2b". The boundaries of the units can be determined automatically [53] [66]. However, since accuracy is a key concern, it was necessary to make minor hand corrections. The units were stored, in the database, with 20-30 milliseconds of speech preceding and following the boundaries to provide for windowing and overlap-add unit concatenation. Next, pitch-synchronous LP analysis was performed on all of the units to create the residual signal, $e[n]$, and the pitchmarks and LP parameters of each frame. As shown in Figure 13, the first step for the analysis is to determine the pitchmark locations, M_i, (instances of significant glottal excitation) for each of the units. The most reliable method to do this is to use a laryngograph when recording the database. However, if the laryngograph data is not available, a method for determining pitchmarks from just the recorded speech, based on the LP residual and group delay function, is given in [86]. Kleijn presented efficient methods for determining pitch intervals in the speech domain and in the residual domain [48]. The method implemented in this research are based on these techniques. It should be noted, however, that the location of the pitchmarks is highly critical reduce artifacts caused by prosody modifications. Hence, it may be necessary to hand correct the automatically determined locations. Pitchmarks must be placed on or very close to the instant of significant glottal excitation.

Based on the pitchmark locations, the LP analysis is performed pitch-synchronously on the speech files. That is, for voiced speech, the analysis frame increment is equal to the local pitch period, and the frame length is a factor of the local pitch period (refer to equation (18 in section 2.4.3) and centered at the pitchmark. In this implementation, the analysis frame length was twice the pitch periods ($2 * T_0$). For unvoiced speech, the local pitch period was set to a default value. The LP order, p, is 16 and the speech waveforms are sampled at 16kHz. Also, a 1st order preemphasis

filter is applied to the waveforms prior to analysis to boost the high frequencies. A Hanning window is applied to each frame, and analysis is performed as discussed in section 2.3, equations (5-12). The autocorrelation function of the windowed frame is the same as in equation (13), guaranteeing a stable prediction filter. From the LP coefficients, the residual signal is calculated by filtering the original signal with the inverse LP filter, as shown in Figure 13. Note that the calculation of the residual is also performed pitch synchrounously. The OGI residual LP analysis method uses a pitch-synchronous triangular window for residual calculation. Other windows may also be sufficient, however this research did not investigate this matter. The residual signal is synthesized using the overlap-add method discussed in section 2.4.3. An OLA factor, F_R, of 1 was used for this implementation, which means the frame lengths were twice the pitch periods. Note that the triangular window implemented in this method was asymmetric, centered at the pitchmarks, M_i. The window began at M_{i-1}, increasing linearly to 1 at M_i, and then decreasing back down to 0 at M_{i+1}.

For each unit, the database consists of the pitchmarks M_i, the LP coefficients $(a_0, a_1, ..., a_p)$, and the residual signal $e(n)$ for each pitch period, and the residual signal. In addition, a record of the locations of all the phoneme boundaries within the unit is also stored. Phonemes are most stable, containing minimal coarticulation effects, at their midpoint. Thus, the parameters for the units are stored from the midpoint of the first phoneme to the midpoint of the last phoneme of the diphone or word.

2.5.2 Synthesis Phase

The output speech waveform is produced by concatenation of the residual signal for each unit, modification of the residual for the desired prosody, and synthesis of the speech waveform using pitch-synchronous residual-excited LP synthesis. This method is similar to the one implemented by the CSLU at the Oregon Graduate Institute [55].

2.5.2.1 Unit Concatenation

The residual signal of each speech unit is concatenated by linear interpolation to create a residual for the desired output. The residual signal for each unit is stored with an excess of 20 milliseconds before and after the unit. This is the region where the linear interpolation is performed. For crossfading two units of speech, $e_k(n)$ and $e_{k+1}(n)$ over a period of N samples:

$$e(n) = (1 - \alpha)e_k(n) + \alpha e_{k+1}(n) \tag{26}$$

where $\alpha = n/N$.

For unvoiced speech, the crossfading is done at the boundaries over a default number of samples. For voiced speech, an optimal join point is calculated before crossfading. The optimal join point is found by cross-correlation over a region at the boundaries of the two units. A new pitchmark is placed in the crossfaded region, if necessary based on the local pitch period. The LP coefficients for the new pitchmark are calculated by linear interpolation of the neighboring line spectral frequencies.

While concatenating the residual, it is also necessary to update the phoneme boundary locations. Based on the optimal join point, the endpoint of the concatenated phoneme can be calculated. All other phoneme boundaries in the unit are adjusted accordingly.

2.5.2.2 Prosody Matching and Synthesis

The automatic prosody generation module outputs target pitch (F_0) values for each syllable. Based on these desired F_0 values, a pitch contour is interpolated for the entire phrase, and F_0 values are assigned to each phoneme. Since duration values are calculated at the phonemic level, there is no additional processing necessary. Hence, the inputs to the prosody matching module are the residual signal, pitchmarks, LP coefficients, and the phonemic pitch and duration values. The pitch and duration changes are made in the time-domain on the residual signal. The time-domain modification

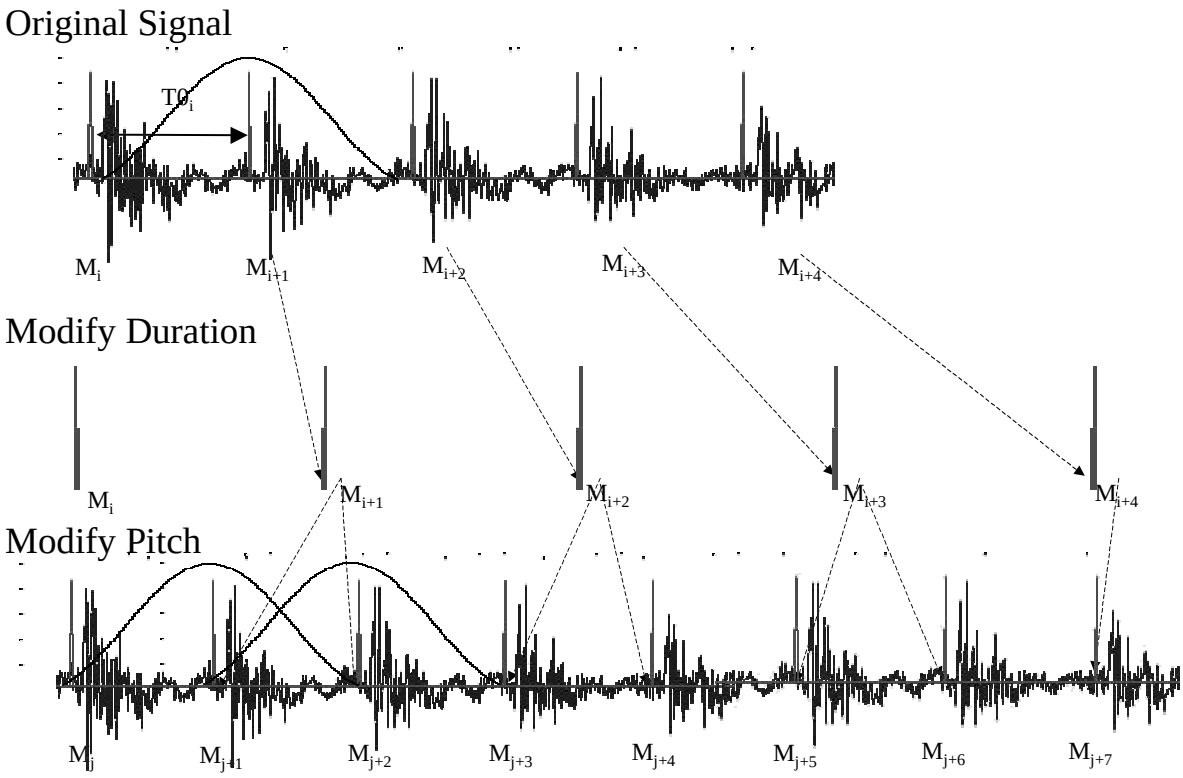

Figure 14: Time-domain prosody modification. (a) Original signal with pitchmarks (M_i). (b) Interpolated pitchmark locations for new duration. (c) New pitchmarks (M_j) are placed on the new time axis to match the desired pitch. The original signal is windowed and copied to the corresponding pitchmarks.

of pitch and duration, discussed earlier in section(2.2), is illustrated in Figure 14 and detailed below [38] [60].

Figure 14a shows the residual signal of a part of a phoneme. The pitchmarks, $(M_i, M_{i+1}, M_{i+2}, ...)$, are indicated by the grey vertical line in front of each pitch epoch. For the desired phoneme duration, a new axis of the corresponding number of samples is created. The pitchmark locations are interpolated linearly to the new axis. For the desired pitch, new pitchmarks $(M_j, M_{j+1}, M_{j+2}, ...)$ are placed on the new axis, at increments of the desired pitch period (Figure 14c). Windowed portions of the residual signal, which map the original pitchmarks (M_i) to the closest new pitchmarks (M_j), are copied to the pitchmark locations on the new axis. For example, Figure 14 illustrates an increase in duration; therefore a portion of the original residual

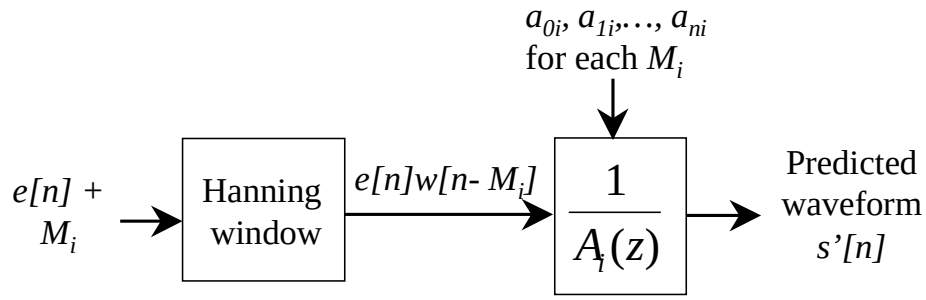

Figure 15: Block diagram of pitch-synchronous residual-excited LP synthesis

signal can be copied multiple times. In other cases where the duration is decreased or the pitch is increased, certain pitch periods of the original residual may be entirely omitted. The new residual is constructed using the overlap-add method by choosing a window length of twice the pitch period.

The output waveform, $s'(n)$, is synthesized by simply filtering the residual by the LP filter consisting of the corresponding coefficients, as shown in Figure 15. A Hanning window, spanning two pitch periods and centered at each pitchmark, is placed on the residual signal. The windowed signal is then passed through the corresponding filter to produce the waveform. The resulting waveforms from the LP filtering are overlap-added to produce the entire phrase. It is critical to maintain a mapping between the pitchmarks and the LP coefficients during the prosody modification.

2.5.3 Discussion

2.5.3.1 Synthetic Speech Signals

A useful method for analyzing the accuracy of a signal model is to use synthetic speech signals. Though for speech signals the accuracy of the model can not give complete information about the resulting subjective quality of the model, the method provides a deterministic approach to understanding the model advantages and limitations. One of the key problems with using real speech signals to analyze a model is the undeterministic nature of the pitch. This is even more important for a model that is pitch synchronous. When using synthetic speech signals, however, the periodicity

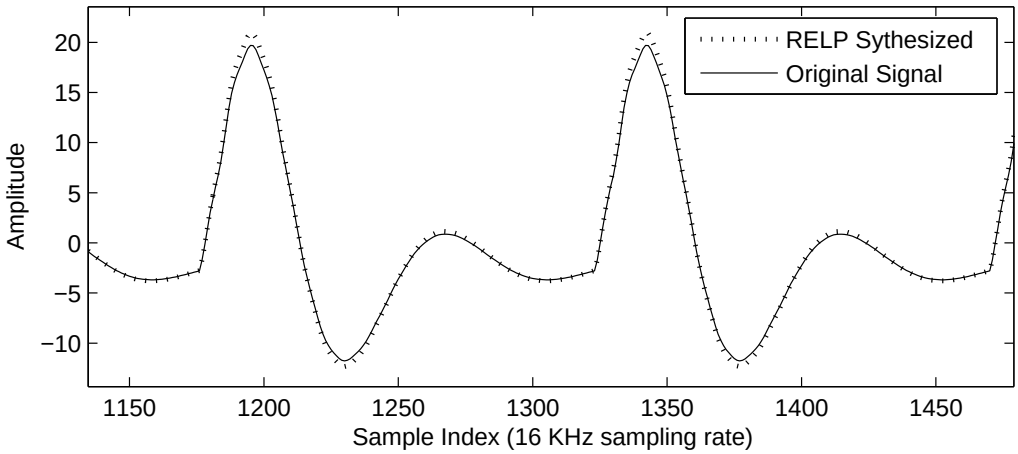

Figure 16: Comparison of the original (dashed) and RELP synthesized (dotted) waveforms of the nasal phoneme /n/

is fixed and spectra is known, allowing for meaningful comparisons of the modeled signal to the original signal.

In this study, synthetic speech signals of different voicing modes and known pitch periods were created to determine the inherent limitations of the model with respect to reconstruction SNR and pitch modifications. The reconstruction SNR is defined here as the ration between the original signal energy and the energy in the error between the original and RELP resynthesized signal. The synthetic speech signal were generated by performing traditional LP analysis on a 20 millisecond frame of voiced speech. The LP coefficients were used to synthesize a number of periods of the the speech using a fixed period impulse train as excitation. Figure 16 shows the time domain comparison of the original synthetic signal and RELP modeled version of the nasal phoneme /n/. The two waveforms are almost identical with differences only visible in the high energy regions. However, the reconstruction SNR for most voiced phonemes ranged from 25-35 dB. Even with this SNR, the model performs well subjectively when the pitch is unchanged. However, when the pitch is modified, the RELP synthesized waveform is unpredictable and can result in errors that are audible.

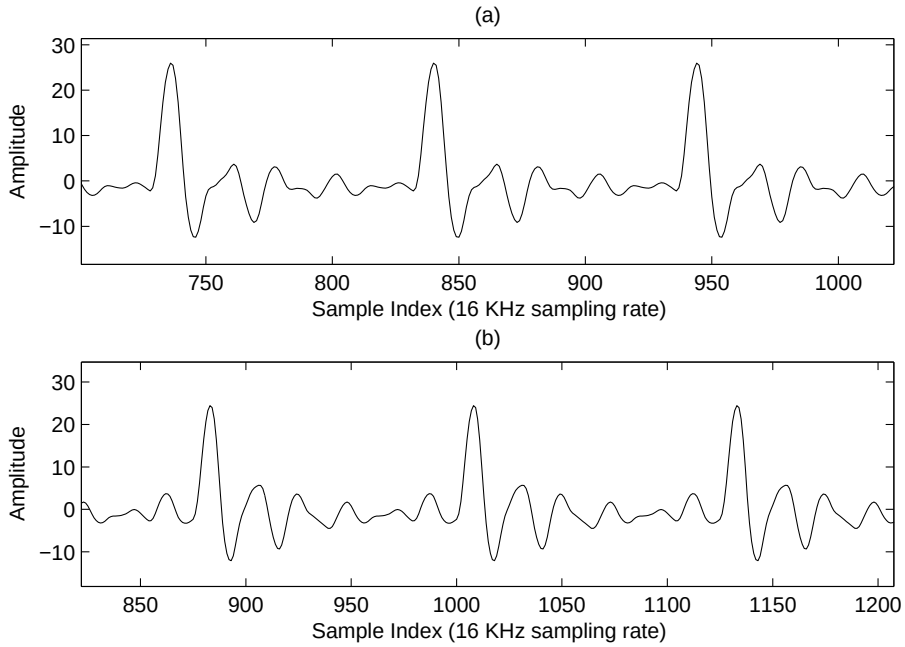

Figure 17: (a) The original synthetic waveform of the vowel phonme /aw/, and (b) the RELP synthesized waveform with the pitch increased by 20%

Figures 17 and 18 show the original and synthesized waveform of two different voicing modes (/aw/ and /n/) where the pitch is increased by 20% during synthesis. It can be seen that for the vowel, /aw/, the pitch increase may not be a problem. However, for the nasal, /n/, audible artifacts may be introduced. Another problem area for prosody modifications are transitions from voiced to unvoiced speech. Figure 19 demonstrates the artifacts introduced when increasing the pitch at the transition of the vowel and fricative phonemes /e/ and /s/.

2.5.3.2 Real Speech Signals

Real speech signals synthesized by the RELP concatenative TTS system were used for subjective analysis of the method. The synthesis was implemented with both naturally extracted prosodics and automatically generated prosodics. For the first case, the pitch and duration values for each segment were extracted from a recorded sentence. The same sentence was then synthesized from the segment database and

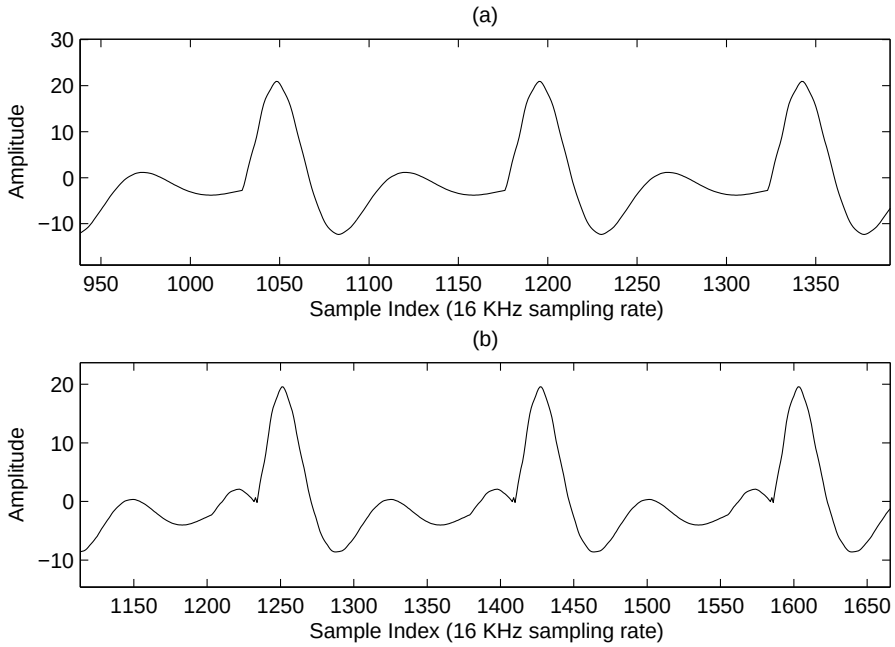

Figure 18: (a) The original synthetic waveform of the vowel phonme /n/, and (b) the RELP synthesized waveform with the pitch increased by 20%

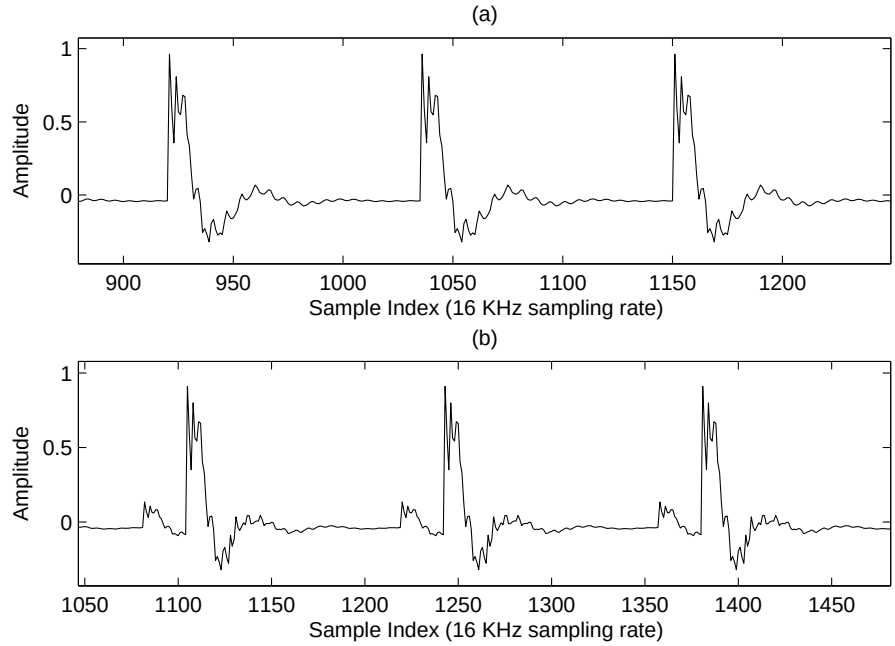

Figure 19: (a) The original synthetic waveform of the voiced to unvoiced transition /e-s/, and (b) the RELP synthesized waveform with the pitch increased by 20%

compared to the recorded version. Since, the goal of the research is to improve the voice quality (not the prosodics), this is a sufficient method to measure the quality of the synthesis technique. The automatically generated prosody was also implemented to compare the overall TTS intelligibility with other synthesis methods.

The output waveforms of the system generated using the natural prosodics were of good quality, and similar in sound to the original speech waveforms. The same sentences, synthesized with artificial prosody, were compared to Festival's diphone based system. The quality of the voice and the intelligibility are significantly better for our word/diphone system. For the case of artificially generated prosody, the speech does not sound as natural. However, voice quality and intelligibility is still significantly better than that of the diphone based system. This confirms that though larger units result in significantly better voice quality and intelligibility, good prosody generation is a major issue for natural sounding TTS.

Even though residual-excited LP results in good voice quality, the synthesized speech resulted in some junctural artifacts, due to spectral mismatches. These artifacts can be noticeably reduced using the spectral normalization and gain normalization techniques discussed in section . Additionally, junctural artifacts for certain types of boundary phonemes are audible. Initially, this research explored methods for resolving these junctural artifacts, by implementing linguistic rules with speech processing techniques that are dependent on phoneme types. However, the more robust ciruclar linear prediction synthesis method, presented later in the thesis, was found to resolve much of the artifacts, resulting in significantly higher quality synthesis.

CHAPTER III

NEW METHOD FOR UNIT SIZE DEFINITION

3.1 Background

Concatenative text-to-speech (TTS) systems are designed to produce speech by concatenating small, prerecorded units of speech, such as phonemes, diphones, and triphones. The choice of unit size is a key element in TTS systems for improving the voice quality and meeting storage requirements. Obviously, with larger units, there is a potential for higher quality since the number of concatenation junctures within an utterance decreases. However, the number of units necessary to cover the language grows exponentially. Existing TTS systems have implemented various unit sizes of fixed and varying lengths ranging from half-phones to disyllables. Howerver, the entire range of possible unit sizes has not been fully investigated, and the optimal unit size is still considered an open question [50].

This chapter discusses the advantages and disadvantages of concatenative synthesis using various unit sizes used in existing TTS systems. An experiment is conducted to study the spectral mismatches between numerous occurrences of a given phoneme within different corpora. Based on the experimental observations, this thesis proposes a new variable length unit definition, which can decrease the degree of concatenation mismatches.

3.1.1 Phonemes, Diphones, and Variants

Various unit sizes have been implemented in current concatenative TTS systems ranging from the sub-phonetic level (i.e. half-phones) to di-syllables (two adjoining syllables) to phrases (limited-domain TTS). The choice of unit size is motivated by storage requirements, synthesis quality, language, and vocabulary. For unlimited vocabulary

systems di-syllables are the largest size that have been documented. Phonemes, generally defined as the smallest unit of sound in a language, can not represents all of the sounds in most languages because they do not include the sounds produced by the interaction of phonmes at junctures. These interactions, called coarticulation, sometimes can span an entire phoneme or even multiple phonemes. Diphones, two adjacent phonemes, have been the preferred unit for many TTS systems due to the ability to produce unrestricted speech from a reasonably limited database of prerecorded segments (approximately 1200 diphones in the English language) [47]. Additionally, since diphones are usually cut at the center of each phoneme, the problem of resolving coarticulation is somewhat addressed. Though these systems are successful in producing speech that is intelligible, the quality of their voices is highly artificial. Discontinuities are often audible due to the large number of segment boundaries within each synthesized phrase. For certain phoneme classes, coarticulation effects from a previous or following phoneme often extend beyond the duration the current phoneme. In addition, because vowels are much higher in energy than the consonants, artifacts are often introduced at vowel-vowel boundaries and are audible. Hence, diphones alone are not considered the optimal set of units for synthesis [33].

The use of larger concatenation units significantly reduces the number of segment boundaries leading to more natural voice quality [83]. Sagisaka and Sato introduced the use of consonant-vowel-consonant (CVC) triphones along with diphones for a Japanese speech synthesis system [78] [80]. Uniquely, the German HADIFIX research TTS system, presented in 1990, utilizes a combination of demi-syllables (half syllables), diphones, and suffixes [71]. Another polyphone approach, implemented for French, uses 1047 triphones and quadraphones in addition to 1290 diphones. The units were chosen based on statistical observations of the most frequent occurrences. The addition of the triphones and quadraphones was motivated by the experimental

observations of intelligibility problems for certain phonemes. Specifically, the problem phonemes were semi-vowels (i.e. /w/ and /y/) and liquids (i.e. /l/ and /r/), which are often affected severely by coarticulaticulation when occurring in consonant clusters [10]. It was reported that the addition of the larger segments resulted in a 20% reduction in the intelligibility error rate.

On the other hand, improvements in unit-selection methods have lead to successful implementation of sub-phonetic units such as half-phones and even smaller. The HMM-based trainable TTS system introduced by Donovan uses units that are 1/2 and 1/3 the size of phonemes [28]. The AT&T Next-Gen TTS system is a high quality TTS system based on half-phones [9]. The advantage of units smaller than phonemes is that it becomes feasible to have a very large number of instances of each unit within a reasonably sized database, increasing the prosodic and contextual richness of the database. Prosodics can be realized through the unit-selection process alone. On the other hand scaling down the database can have a dramatic effect on quality.

Recently, Kishore and Black conducted experiments to determine the optimal unit size for the Indian Hindi language [46]. The candidate unit sizes were half-phone, phone, diphone, and syllable. The experiments consisted of subjective A-B comparisons of a number of synthesized sentences using the candidate unit sizes. The A-B comparison test also included a "No Preference" choice. The comparisons were made between all possible combinations of the candidate unit sizes (i.e. syllable vs. phone, syllable vs. diphone, phone vs. diphone, etc.). Synthesis was conducted by unit concatenation using PSOLA methods as described in section 2.4.3. The results of this test showed that syllables were preferred heavily over all other units. Since half-phones have been implemented successfully in existing high quality systems the preference of syllables over half-phones is most noteworthy in this experiment. This suggests that the natural prosodics in syllables are preferred over prosodics realized via concatenation of numerous units. Though half-phones were preferred

over diphones and phones, the statistical significance of the preference in those cases was questionable. This is due to the large majority of "No Preference" choices made by the subjects. It is important to note that the syllabic nature of the Hindi language could have influenced the outcome of the experiment. Hence, the results presented should probably be generalized only to languages that have significant coverage of all words with a reasonable number of syllable units.

3.1.2 Words

Initially, this research implemented words as units, since they resolve all intra-word discontinuities and coarticulation effects. Though the memory requirements for such a system may seem unreasonable, it is feasible for limited-domain applications, resulting in very high quality TTS. For example, a database of 10,000 words, averaging 0.3 seconds per word, sampled at 8KHz, stored at 16Kbps (ADPCM encoded), would only require approximately 5 megabytes of memory. Words alone, however, cannot resolve the inter-word discontinuities. Hence, "splicing units" such as diphones are necessary at the boundaries of words. The prerecorded words are cut at the midpoint of the first and last phonemes, and concatenated to diphones containing the last phoneme of the current word and the first phoneme of the next word. For example, the phrase "the larger issues" is produced by concatenating the following sequence of units: /pau/-/D/, the, /&/-/l/, larger, /3r/-/I/, issues, /z/-/pau/. The phoneme /pau/ refers to a pause. The diphones, which are usually less than 50 milliseconds in length would only require about 50-100 kilobytes more memory. The initial residual-excited linear prediction synthesis system was implemented using the word plus diphone model. Though, the naturalness of the speech improved when compared to using just diphones, the coarticulation problems associated with diphone concatenation of certain phonemes persisted in this model.

The limited-domain unit-selection synthesis system introduced by Black and Lenzo

Table 3: Construction of the phrase "the major issues" using three different unit classes: diphones, words+diphones, disyllables

Diphones	Words+Diphones	Disyllables
/pau/-/D/	/pau/-/D/	/pau/-/D/-/&/
/D/-/&/	the	/&/-/m/-/ei/
/&/-/m/	/&/-/m/	/ei/-/dZ/-/3r/-/I/
/m/-/ei/	major	/I/-/S/-/u/
/ei/-/dZ/	/3r/-/I/	/u/-/z/-/pau/
/dZ/-/3r/	issues	
/3r/-/I/	/z/-/pau/	
/I/-/S/		
/S/-/u/		
/u/-/z/		
/z/-/pau/		

also implemented words as concatenative units in a different manner[14]. This system is essentially a diphone based synthesizer. However, each phoneme is also tagged to the word that it comes from. In this method, if only one instance of each word exists in the database, utterances are synthesized in the same manner as the words plus "splicing" diphones method described above. However, if multiple instances of words exist in the database, the unit-selection algorithm can select phonemes from different instances of the desired words. Hence, the synthesized words can be better matched to the target prosody.

3.1.3 Disyllables

Coarticulation effects, which last longer than the duration of one phoneme, are most common in certain classes of phonemes. Additionally, vowels and dipthongs, in general, are the most stable phonemes. These observations can be exploited by using units called disyllables [46]. Disyllables are consonant clusters surrounded by vowels. They are defined as $V - C_m - V$, where V represents vowels or dipthongs and C_m represents a string of consonants of length m ($m \geq 0$). This guarantees that all concatenation points are periodic and stable. Though, in theory, these units may be ideal,

their numbers for complete coverage of a language may be unreasonably large for implementation. In order to determine the feasibility of implementing $V-C_m-V$ units, a search algorithm was written to parse large corpora to find all unique occurrences of disyllables. The text was first converted to phonemes using the Oregon Graduate Institute defined phoneme set. Pauses occurring at phrase breaks were included in the set of vowels and dipthongs. When applied to approximately 4 hours of news, the search algorithm resulted in 21,229 unique disyllable occurrences. The algorithm was them applied to 10 large novels resulting in approximately 47,000 unique $V-C_m-V$ units. In both cases the largest number of units were triphones and quadraphones. There were also a significant number of pentaphones and diphones. Even larger units do occur in the English language, but are very rare.

For limited-domain applications, it may be advantageous to use disyllables in combination with demi-words. A demi-word can be defined as a word that is spliced from its first vowel occurrence to its last vowel. The disyllables would then be used to interconnect the demi-words to adjacent demi-words as in the words and diphones model. Whether or not this model is significantly advantageous, in terms of memory savings, is dependent on the application and the number of unique words that exist. However, since quality is the dirving force for this research, both models in this section are good candidates. Table 3 illustrates how the phrase "the major issues" would be constructed using the three different unit classes discussed.

3.2 Variable-Sized Units Based on Junctural Phonemes

As stated earlier, certain classes of consonants are more likely to be affected by coarticulation than others. Also, some vowels, such as transitionals (/iU/), can have a higher probability of mismatches than quasi-stationary consonants. Disyllables, which force the join points to be only at vowels can be sub-optimal units in many cases. Additionally, the number of disyllables to cover a language is extremely large

compared to other units. The idea of disyllables can be extended to a set of uniquely defined multiple-phone units that include vowels and consonants that have better matching characteristics and a lower tendency to be affected by coarticulation. It would seem that certain consonant classes, such as low-energy fricatives and nasals, which are relatively stationary, would be good candidates for junctural phonemes.

3.2.1 Perceptual Measures for Spectral Discontinuities

In this research, a method was developed to define a new variable-length unit, that consists only of juntural (boundary) phonemes with lower perceptual spectral discontinuities. The method also includes a feasibility analysis of the unit definition with respect to the total number of units within a language. Also, included in the method, is the ability to scale the definition based on the feasibility study. The basics steps for this analysis method is given in Figure 20. In order to determine which vowels and consonants are less affected by coarticulation and better suited for concatenation, this analysis method first determines the spectral differences between different occurrences of the same phoneme within different speech corpora. The differences are then analyzed to define a variable size unit that is characterized by junctural phonmes with generally lower spectral differences. Finally, the feasability of the newly defined unit is considered by parsing large text corpora to find the total number of unique units that can exist within a given language.

The determination of spectral differences is based on the cepstral distance, which is commonly used in concatenation cost functions for unit-selection synthesis systems [11]. The cepstral distance measure has been implemented in numerous ways using different methods for representing the cepstra as well as different distance calculation techniques. The definition of the cepstrum of a frame of speech is the inverse FFT of the log of the FFT magnitude spectrum of the frame, as shown in equation (27). For

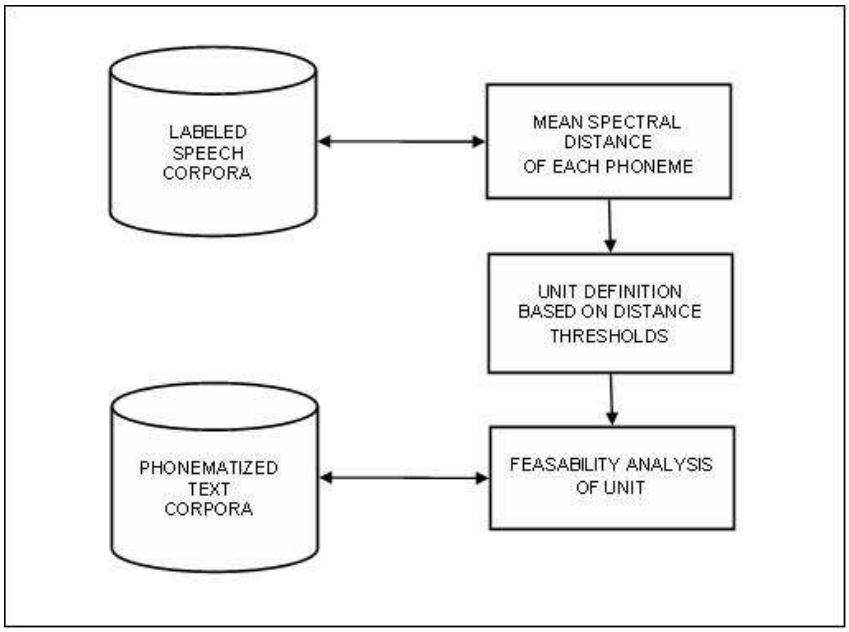

Figure 20: Block diagram of anlaysis method for defining variable size units based on reducing perceptual junctural mismatches

a speech signal, $s(n)$ the cepstrum is defined as:

$$c(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log \left| \sum_{n=0}^{N-1} s(n) e^{-j\omega n} \right| e^{j\omega m} d\omega \tag{27}$$

where $c(m)$ is referred to as the mth cepstral coefficient. This can be rewritten in terms of, $S(z)$, the log of the FFT of $s(n)$.

$$\log S(z) = \sum_{m=-\infty}^{\infty} c(m) z^{-m} \tag{28}$$

Generally, for many applications including speech, only a moderate number of coefficients in the order of 10-20 are necessary. Equation 27 is often referred to as the linear FFT-based cepstra.

For speech applications, cepstral distances have been implemented by various methods including the FFT-based cepstra, LPC-based cepstra, and LSF coefficients. The cepstral distance analysis performed by this research has been motivated by two significant research efforts conducted previously. The work by Macon and Wouters

evaluated the effectiveness of objective distance measures by determining their correlation to perceptual discontinuities in concatenative synthesis [57]. This work examined the spectral differences at unit junctures using the FFT-based cepstra, LPC-based cepstra, LSFs and other unorthodox methods: the Itakura distance and log area ratios. The spectral representations were calculated using linear, perceptual linear prediction (PLP) [41], and mel frequency scaling of the speech frames. The use of mel and PLP frequency scaling is motivated by its effective implementations in speech recognition systems [69]. The distances of these spectral measures were calculated at unit junctures of a number of synthesized monosyllabic words and compared to subjective evaluation of discontinuities. The correlation between the spectral distance measures and subjective results was then determined. Notably, this experiment showed that the FFT cepstra and LPC cepstra had the highest correlation to subjective results. Additionally, the cepstra computed using the mel scale had significantly better correlation than the linear cepstra. The correlation coefficient, which was identical for both methods, was reported to be 0.64. For comparing the correlation of the subjective results to different spectral distance measures, the study used two objective distances: the Euclidean distance and Mahalanobis distance [102]. Interestingly, for cepstral and LSF parameters computed using the mel frequency scaling, both distance measures had identical correlations to the subjective results.

Stylianou and Syrdal conducted experiments that are very similar, notably differing in the performance metric [91]. Instead of the correlation between the subjective and objective measures, this study reports the detection rate, defined by the number of discontinuities predicted by the various distances with respect to the number of perceptual discontinuities detected. In addition to some of the spectral representations and distances used in the work by Macon and Wouters, this study included the Euclidean distance between the log power spectra and the Kullback-Leibler distances between power spectra and LSFs. This study reported the highest detection of

Table 4: Comparison of the correlation of spectral distance measures to subjective measures based on correlation coefficients reported by [57] and detection rates reported by [90].

Spectral Distance Method	Correlation Coefficient	Detection Rate
mel-frequency FFT cepstra	0.64	35.811
mel-frequency LPC cepstra	0.64	N/A
PLP FFT cepstra	0.62	25.570
PLP LPC cepstra	0.61	N/A
PLP LSF	0.57	21.139
linear FFT cepstra	0.49	28.764
linear LPC cepstra	0.48	23.263
linear LSF	0.34	9.749

perceptual discontinuities by the Kullback-Leibler distance between the FFT-based power spectra, the Euclidean distance between the mel-frequency cepstral coefficients, and the Euclidean distance between the FFT-based log power spectra.

Some of the results of the two studies are shown in table 4. Only the Euclidean distance based results are shown in the table. Though the Kullback-Leibler distance based measures resulted in a high detection rate when applied to the linear FFT power spectra, the results are not included here since the distance metric was not used in both studies. Note that the Euclidean distances for the correlation coefficients in the first study were calculated by weighting the cepstral coefficients whereas only the PLP and LPC cepstral distances were weighted in the second study. The table shows that the the mel-frequency based FFT cepstral distance calculated from the mel-frequency cepstral (MFC) coefficients produced the highest correlations to subjective results in both studies. Note that only the results containing the highest correlations are given for relevance to this research. The Itakura Saito distance resulted in high linear, PLP, and mel correlation coefficients, which were equivalent to the FFT-based and LPC-based cepstra in one study. However, in the second study, the results of this distance were not significant enough to report.

3.2.2 Analysis Method

The studies described above form the basis for developing the method to determine "optimal" junctural phonemes and defining the new unit. The following summarizes the key results of the studies that were applied to this research:

- The mel-frequency cepstra (FFT or LPC based) is determined to have the best correlation to subjective measurements of spectral discontinuity
- The linear frequency FFT and LPC based cepstra are also relatively good perceptual measures for spectral discontinuity.
- The Euclidean distance is a sufficient distance measure since it produced equivalent or better correlations compared to the Mahalanobis and Kullback-Leibler distances.

For this research a combination of these metrics were utilized to analyze the speech corpora and determine good candidates for junctural phonemes. Specifically, the Euclidean distance was applied to LPC based MFC coefficients and the linear frequency FFT cepstra.

The FFT cepstra was implemented by calculating the distances between the discrete real cepstra. For every occurrence, j, of a given phoneme in the corpus, the M point real cepstrum, $c(m)$, is calculated on one frame after applying a window, $w(n)$, of length N. For the discrete real cepstra, equation (27) then becomes:

$$c(m) = \Re\left[\frac{1}{M}\sum_{m=0}^{M-1}\left[\log\left|\sum_{n=0}^{N-1} s(n)w(n)e^{\frac{-j2\pi mn}{N}}\right|\right]e^{\frac{j2\pi nm}{M}}\right] \tag{29}$$

where $s(n)$ represents the frame of the phoneme. In this research, the speech corpora used was sampled at 16kHz. The analysis frame length, N, was set to 320 samples (20 milliseconds), and the number of coefficients, M, was set to 16. Note that for each phoneme the frame for calculating the coefficients, $c(m)$, was selected at the

midpoint of the phoneme. This method has two inherent advantages. First, phonemes are usually in the most stable state at their midpoints resulting in good conditions for spectral comparisons. Second, for concatenative synthesis, units are generally concatenated at the midpoints of phonemes. Hence, comparing the midpoints of phonemes is a very close approximation of the spectral discontinuity sub-cost in the concatenation cost function calculated by unit-selection systems.

Similar to the FFT cepstral coefficients defined in equations (27) and (28), the LPC based cepstral coefficients, $c(m)$, are the FIR filter coefficients representing the log of the LP spectra. Replacing the FFT in equation (28) with the all-pole LP filter given in equation (5) results in:

$$\log \left| \frac{\sigma}{A_p(z)} \right| = \sum_{m=0}^{\infty} c(m) z^{-m}. \tag{30}$$

For the MFC coefficients, however, the LP coefficients and gain, σ, must be transformed to the mel frequency scale, $\tilde{z}^{-m}$, given by equation (31).

$$\tilde{z}^{-m} = \frac{z^{-m} - \alpha}{1 - \alpha z^{-m}}, \qquad |a| < 1 \tag{31}$$

where for the mel frequency scale, the factor, α, is set to 0.46. Then, equation $\tilde{c}(m)$ is modified to represent the relationship between the MFC coefficients and the LP filter, as shown below:

$$\log \left| \frac{\tilde{\sigma}}{1 + \sum_{k=1}^{P} \tilde{a}(k) \tilde{z}^{-k}} \right| = \sum_{m=0}^{M} \tilde{c}(m) \tilde{z}^{-m} \tag{32}$$

where P is the order of the LP analysis performed on each frame of the phoneme and M is the number of cepstral coefficients. By applying the constraint, $P >= M$, the MFC coefficients can be calculated efficiently from the LP coefficients via a recursive algorithm [97]. The recursion first transforms the LP coefficients from the linear frequency scale to the mel-scale. The intermediate coefficients of the recursion are

determined by:

$$\tilde{a}^i(0) = a(-i) + \alpha\tilde{a}^{(i-1)}(0), \qquad \tilde{a}^i(1) = (1-\alpha^2)\tilde{a}^{(i-1)}(0) + \alpha\tilde{a}^{(i-1)}(1), \tag{33}$$

$$\tilde{a}^i(k) = \tilde{a}^{(i-1)}(k-1) + \alpha(\tilde{a}^{(i-1)}(k) - \tilde{a}^{(i)}(k-1)), \tag{34}$$

for $k = 2, 3, ..., P$, where $i = -K, ..., -1, 0$. The transformed LP parameters are, then, calculated as shown below:

$$\tilde{K} = K/\tilde{a}^{(0)}(0), \quad \tilde{a}^{(0)}(m)/\tilde{a}^{(i)}(0), \quad 1 \leq m \leq P \tag{35}$$

Finally, the mel frequency cepstral coefficients are calculated recursively from the mel-frequency LP coefficients according to:

$$\tilde{c}(0) = log(\tilde{K}), \tag{36}$$

$$\tilde{c}(m) = \tilde{a}(m) + \sum_{k=1}^{m-1} \frac{k}{m}\tilde{c}(k)\tilde{a}(m-k), \tag{37}$$

where $1 \leq m \leq M$.

After obtaining the cepstral coefficients of all of the phonemes, the cepstral distances, $D_{j,j+k}$, between all unique pairs of each phoneme, j and $j+k$, are computed and averaged using the Euclidean distance method. The experiments by Macon and Wouters showed that placing exponential weights on the cepstral coefficients can improve the correlation of the objective distance measures to subjective results. The weighting of the coefficients, known as cepstral liftering, is performed by multiplying each cepstral coefficient, $c_j(m)$, by a factor, m^s as shown in (38).

$$D_{j,j+k} = \sqrt{\sum_{m=0}^{M-1} [m^s(c_j(m) - c_{j+k}(m))]^2} \tag{38}$$

where $c_j(m)$ and $c_{j+k}(m)$ are the MFCC or linear FFT-based, M-point cepstra coefficient values for two unique occurrences, $(j, j+k)$, of the same phoneme. The optimal value of the parameter, s, of the lifter weights, m^s, is determined experimentally.

Macon and Wouters demonstrated that the value of $s = 0.6$ produced the best correlation. The mean Euclidean cepstral distance for each phoneme is, then, computed by summing all the distances and dividing by the total number as given in (39) below.

$$\bar{D} = \frac{\sum_{j=1}^{N_p-1} \frac{\sum_{k=1}^{N_p-j} D_{j,j+k}}{N_p - j}}{N_p - 1} \tag{39}$$

where N_p is the total number of occurences of a given phoneme.

In order to understand the relative statistical significance of the mean cepstral distances of each phoneme, it is important to consider the relative standard deviation of $\bar{D}$ across all of the phonemes. The Z-score is a useful measure for determining the "normalized" relative performance of each element within a population. Recently, they have been used extensively for comparing features during the unit-selection process of TTS systems [12]. Therefore, the Z-scores of the mean cepstral distances were calculated to provide a more meaningful comparison of one distance to another. First, for all the mean cepstral distances, $\bar{D}_P$, of the phonemes, P, their average, $\bar{D}_{mean}$, and their standard deviation, $STD(\bar{D})$, are are determined. The mean cepstral distance Z-score for each phoneme, Z_P is given by equation 40.

$$Z_P = \frac{\bar{D}_P - \bar{D}_{mean}}{STD(\bar{D})} \tag{40}$$

Since, in this case good junctural phonemes correspond to lower $\bar{D}_P$, the Z-scores should also be negative or close to zeros.

This anlysis was conducted on the Boston University FM labeled speech corpus. This is a very large speech database consisting of many hours of news read speech. This corpus consists of automatically labeled phoneme boundaries. Since the analysis is conducted at phoneme midpoints, slight errors in the boundary location should not affect the analysis. The z-scores of the mean cepstral distances were calculated using three methods for representing the cepstra: the linear FFT based cepstra, the LP

Table 5: Phonemes with Z-scores < 0.5 for all cepstral distance methods, calculated using the linear frequency FFT, mel-frequency cepstral coefficients with liftering ($s = 0.6$), and mel-frequency cepstral coefficients without liftering

Phonemes	Z-score Z_P Linear FFT $\bar{D}$	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0.6	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0
/aU/	-1.891	-1.899	-1.565
/aI/	-1.825	-1.048	-1.405
/E/	-1.690	-0.694	-1.541
/@/	-1.355	-1.033	-0.902
/i:/	-0.809	-1.073	-1.042
/3r/	-0.802	-0.346	-0.634
/ei/	-0.792	-1.479	-1.357
/S/	-0.703	-1.471	-0.533
/A/	-0.643	0.125	-0.423
/v/	-0.557	0.237	-0.483
/N/	-0.533	-1.635	-0.675
/j/	-0.512	-1.321	-1.518
/l/	-0.397	0.158	-0.186
/w/	-0.394	0.153	-0.207
/h/	-0.348	-0.536	0.274
/s/	-0.279	-0.583	-0.730
/m/	-0.138	0.053	-0.516
/n/	0.1293	-0.294	-0.093
/f/	0.3558	-0.433	-0.214

based MFCC using liftering, and the LP based MFCC without liftering. Based on the results, a z-score threshold of 0.5 was chosen to classify the phonemes as good, marginal, and poor junctures. Tables 5, 6, and 7 show the z-scores of the mean cepstral distances of the phonemes, computed from the Boston University corpus "m1b" speaker. Table 5 shows the phonemes that had z-scores of ≤ 5 for all three cepstral distance measures. Table 6 consists of phonemes that performed well for at least 1 of the measures, and table 7 consists of the phonemes with z-scores that were consistently greater than 0.5.

Table 6: Phonemes with Z-scores > 0.5 for at least 1 cepstral distance method and < 0.5 for at least 1 cepstral distance method. The distances are calculated using the linear frequency FFT, mel-frequency cepstral coefficients with liftering ($s = 0.6$), and mel-frequency cepstral coefficients without liftering

Phonemes	Z-score Z_P Linear FFT $\bar{D}$	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0.6	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0
/$\hat{}$/	-0.1512	1.017	0.481
/&/	-0.018	0.979	0.326
/oU/	-0.005	1.068	0.162
/I/	0.011	0.893	0.360
/U/	0.153	0.521	0.735
/9r/	0.322	1.238	0.661
/b/	0.559	0.484	0.803
/p/	0.623	0.437	1.098
/dZ/	0.679	0.351	1.040
/tS/	0.685	-1.189	-0.887
/T/	0.872	0.489	0.535
/z/	2.600	-0.468	-0.002

Table 7: Phonemes with Z-scores > 0.5 for all cepstral distance methods, calculated using the linear frequency FFT, mel-frequency cepstral coefficients with liftering ($s = 0.6$), and mel-frequency cepstral coefficients without liftering

Phonemes	Z-score Z_P Linear FFT $\bar{D}$	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0.6	Z-score Z_P LPC MFCC $\bar{D}$ Lifter Exp = 0
/d/	1.209	1.293	0.795
/k/	1.284	2.074	2.318
/g/	1.295	1.421	1.865
/D/	1.671	1.369	1.633
/t/	1.743	1.293	1.721

These results support the hypothesis stated earlier in regards to nasals and unvoiced fricatives, which have relatively lower mean cepstral distance measures compared to other consonants. Interestingly, the mean cepstral distances for some unvoiced fricatives (i.e. /s/, /S/, /f/, etc.) are even lower than some vowels. Note that since this analysis was not conducted pitch-synchronously, it is possible that the cepstral distance of voiced phonemes, especially vowels, can be higher than consonant phonmemes. Certain stop consonants, plosives, and fricative (i.e. /d/, /g/, /k/, /t/, and /D/) had the greatest distances (z-scores > 1.0) suggesting that they would make poor junctures.

3.3 Unit Definition and Feasability

Based on the results of the analysis method, a new variable size multiple phoneme unit can be defined. An average of the three z-scores for each phoneme was calculated and a threshold was placed on the mean z-scores to define a set of "junctural phonemes". The new set of phonemes would then consist of the format, $J - C_m - J$, where J represents a phonemes from the set of junctural phonemes, and C_m represents a string of consonants that are outside of the threshold for good junctural characteristics. For example, if the threshold is set to 0.5 the set of endpoint phonemes consist largely of vowels and certain consonants. In turn, C_m consists of all of the phonemes with a mean cepstral distance z-score ≥ 0.5 (/p/, /T/, /D/, /t/, /dZ/, /k/, /b/, /d/, and /g/).

Following definition of the unit, its feasability needs to be determined in terms of practical storage. The feasability analysis is conducted by parsing large text corpora to determine the number units that exist within a language for the definition (based on the threshold). The text is first converted to a string of phonemes and each unique occurrence of units matching the definition above, is added to a list. The text corpora used in this analysis consisted of a number of free electronic books, courtesy

of the Project Gutenberg [39]. The Project Gutenberg, created in 1971, consists of a large collection of out-of-copyright books in text format, free of charge. A set of 10 novels in the English language were selected and converted to phonemes. From all of these novels, only 10,289 unique multiphones were found for the definition above. In this set of units, triphones and quadraphones had the most significant numbers being 4,259 and 4,198, respectively. Compared to disyllables, this unit definition has a significantly smaller number of units, and is much more practical for unlimited and limited vocabulary applications.

Note that in this method, the unit definition is scalable. In other words, by modifying the threshold, the total number of units can be increased or decreased. For example, if the threshold was changed from 0.5 to 0.1, the unit definition would include a larger number of endpoint phonemes. This, in turn would reduce the total number of units within a language. However, since the endpoint phonemes can have larger spectral variations, the concatenation quality may be compromised.

CHAPTER IV

THE CIRCULAR LINEAR PREDICTION MODEL AND THE CONSTANT PITCH TRANSFORM

The overall TTS process consists of a number of steps that can be classified under two general categories: natural language processing (NLP) and signal processing. The NLP phase consists of word and syllable boundary detection, syntactic prosodic parsing, and determination of target prosody. However, in achieving high quality TTS, the voice quality is purely dependent on the signal processing phase. This includes database preparation (unit selection and analysis) and synthesis (prosody matching, segment concatenation, and synthesis). The method presented in this paper focuses on improvements to the signal processing techniques of TTS.

Traditional linear prediction methods used in TTS (i.e. RELP) are based on analysis and synthesis of pitch-synchronous speech frames using the autocorrelation method to calculate the model parameters. This means that the frame rate is equal to the local pitch and the frame size is an overlapping factor of the local pitch period. The analysis frames are obtained by applying a Hanning window to each of the frames. Though efficient, such methods lead to signal distortion due to windowing and incorrect assumptions, made by the autocorrelation method, of the signal outside of the frame boundaries. In addition, the excitation signal is created using the pitch-synchronous overlap-add (PSOLA) method resulting in further errors in the overlapping regions. Specifically for TTS, segmental boundaries require parameter and excitation interpolation resulting in potentially unexpected behavior. In the case of stop consonants, voiced fricatives, and certain vocalic sounds, these errors often result in audible artifacts. The artifacts are further intensified when applying

prosodic modifications to the segments. On the other hand, the covariance method, which does not require windowing, would provide more accurate signal modeling at the cost of high computational complexity and potential instability [72]. For TTS, there would still be errors attributed to the parameter and excitation interpolation at segment boundaries.

An ideal synthesis method should allow for prosodic modification and segment concatenation without affecting the relationships between the linear prediction parameters. Circular linear prediction (CLP) was first introduced in the 1970s by [7] [8] as a windowless alternative of the autocorrelation and covariance analysis. In this model, each pitch epoch is represented by an infinite periodic signal of identical pitch epochs, and the autocorrelation coefficients are computed circularly. For this special case, the autocorrelation method and the covariance method are identical, resulting in a Toeplitz correlation matrix allowing for efficient computation of the LP coefficients. Ertan's Ph.D. thesis presents the proof of equivalence of the CLP analysis method to a number LP parameter estimation methods when the signal is either infinitely periodic or a single pitch period [34]. In addition to the advantages of efficient parameter estimation and guaranteed stability, tests conducted using known synthetic speech signals proved that this method results in predicted spectra containing fewer errors than the traditional autocorrelation method (section 4.1.2.1).

The Constant Pitch Trasform (CPT) has be recently revisited by Shukla, Ertan, and Barnwell [85] as a method for interpolating each pitch epoch of a periodic signal to a fixed length, resulting in a monotone signal. CLP combined with CPT results in a signal representation consisting of a set of circular LPC coefficients and a fixed (constant) length circular residual that can be used to create a reconstructed speech signal that is always perceptually indistinguishable from the original. With regard to TTS, this representation allows for non-overlapping segment concatenation and simplified pitch modification using the inverse CPT.

4.1 Circular Linear Prediction Modeling

The traditional linear prediction equation for representing a signal, $s(n)$, is well-known as

$$s(n) = \sum_{i=1}^{p} -a_i s(n-i) + e(n) \tag{41}$$

where $e(n)$ is the prediction error. The coefficients, a_i, can be determined efficiently, with guaranteed stability, using the autocorrelation method. This method, however, makes the assumption that the initial values outside of the modeling region are zero. If no assumptions are made for the signal outside of the modeling region, as in the covariance method, a more accurate model can be realized. For the latter, minimizing the squared error results in

$$\sum_{i=1}^{p} a_i r(i,j) = -r(0,j) \tag{42}$$

where $j = 1, 2, ..., p$ and the expression $r(i,j)$ is defined as:

$$r(i,j) = \sum_{n=0}^{T_0-1} s(n-i)s(n-j) = \sum_{n=0}^{T_0-1} s(n+i)s(n+j) \tag{43}$$

for $i \geq 0$ and $j \geq 0$. The assumption that circular linear prediction does require, however, is that every analysis frame is exactly periodic with period, T_0. Hence, $s(n)$ can be represented as an infinitely periodic signal and the index of $r(i,j)$ can be simplified to the difference between j and i, and Eq. (42) becomes:

$$\sum_{i=1}^{p} a_i r(j-i) = -r(j) \tag{44}$$

where $j = 1, 2, ..., p$ and the expression $r(k)$ is defined as:

$$r(k) = \sum_{n=0}^{T_0-1} s(n)s((n+k))_{T_0} \tag{45}$$

for $k \geq 0$, where $((.))_N$ is the modulo N operation. In matrix form, Eq. (44) can be represented as:

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(p) \end{bmatrix} \tag{46}$$

Eq. (44),(45), and (46) illustrate that the periodicity assumption made by the CLP model results in a Toeplitz structured autocorrelation matrix, which can be efficiently inverted using the Levinson recursion to solve for the coefficients $(a_1, ..., a_p)$ [40]. In addition, the Levinson recursion guarantees a stable prediction filter. Therefore, circular LPC modeling incorporates the advantages of the autocorrelation method without the windowing distortions.

4.1.1 CLP Analysis for Fractional Pitch

Since the circular linear prediction technique requires that the pitch periods be exact, the amount of precision plays an important role. If the pitch period is not exact, it was observed that each frame of the residual signal will contain a short high-energy burst at the beginning of the frame. If such a residual signal is modified (i.e. pitch transformations), the spikes at the beginning of each frame are amplified resulting in highly audible artifacts in the synthesized waveform. Though it is unclear why the artifact is amplified during pitch modifications, it can be attributed to the propagation of an error introduced in the model. Since the the pitch periods of real speech signals are not generally integers, modeling for fractional pitch becomes important. Experiments conducted on synthetic speech signals with known fractional pitch periods show that at least one decimal point of precision does improve the LP analysis results. Additionally, increasing the precision beyond this does not have a significant improvement to the performance [34]. Hence a method has been developed

for determining the fractional pitch period while simultaneously performing the CLP analysis on the units. This fractional pitch estimation is an exhaustive detection through analysis algorithm that is based on maximizing the prediction gain.

Initially, approximate integer pitchmark locations for a speech segment need to be determined using standard integer pitch estimation methods [48][86]. Recently [34] has implemented a very high quality pitch prediction algorithm reporting 99.5% accuracy. The speech segment is then upsampled by a factor, P, determined by the desired fractional precision. For each frame, i, of length, T_{0i}, CLP analysis and circular inverse filtering is performed incrementally on a number of fractional pitch periods in the range of $(T_{0i} \pm \alpha)P$ to find the period that results in a residual that is maximally smooth at the boundaries. The prediction gain is used as a measure for minimizing discontinuities at the frame boundaries. The range of fractional increments, f, for determining the correct fractional pitch should be selected based on the confidence of the integer pitch prediction. For example, if it can be assumed that the pitch has been determined accurately to the closest α integers, the range of f would be $-\alpha P < f < \alpha P$.

The CLP analysis for an upsampled frame, $s'(n)$, with fractional pitch periods requires special attention. In order to maintain the original sampling rate, the autocorrelation function is calculated by incrementing the upsampled signal index, n, by P samples at a time. The autocorrelation function in Eq. (45) becomes:

$$r(k) = \sum_{n=0}^{T_0-1} s'(nP)s'((nP + kP))_{T_0P} \tag{47}$$

However, the autocorrelation for each fractional pitch period $T_{0i} + f$ needs to be performed in a modulo fashion, until the index, n, of $s'(n)$ is equal to zero. In other words, $r^f(k)$ must be calculated over m number of periods ($m(T_{0i} + f)$ times), until the index nP returns back to zero. The value of m to satisfy this condition is:

$$m = \frac{P}{GCM(P, Pf)} \tag{48}$$

where the term $GCM(\cdot)$ refers to the greatest common multiple. For example, if $P = 10$ and $Pf = \pm 2$ (the fractional pitch period is $T_0 \pm 0.2$), the autocorrelation function is calculated over 5 periods of $s'(n)$, where each period is a fractionally offset version of the original signal, $s(n)$. In order to assure that $r^f(k)$ is calculated over the entire range of the fractional pitch period of $s'(n)$, the range of values for n must include the fraction. The extra values for the range of n to include the fraction is given by mf. Since $s'(n)$ is in the upsampled domain, Eq. (47) then becomes:

$$r^f(k) = \sum_{n=0}^{m(T_0+f)-1} \left[s'((nP))_{(T_0+f)P}\right] \left[s'((nP+kP))_{(T_0+f)P}\right] \tag{49}$$

where the symbol $((\cdot))_N$ represents the modulo function. For each frame i of speech, the autocorrelation function is determined circularly. With the assumption of exact periodicity, Eq. (49) can be evaluated at the original sampling rate in the following manner:

- repeat each upsampled frame, i, of length $(T_{0i} + f)P$, m number of times
- downsample the repeated frame of length $m(T_{0i} + f)P$ by the factor P
- perform the autocorrelation in the original sampling domain according to Eq. (47).

To calculate the residual signal, $e'_i(n)$, the circular LP inverse filter is also implemented using the modulo function:

$$e'_i((nP))_{(T_{0i}+f)P} = \begin{cases} \sum_{k=0}^{p} a_k s'_i((nP-kP))_{(T_{0i}+f)P} & \text{for } n \geq p \\ \sum_{k=0}^{n} a_k s'_i(nP-kP) + & \\ \sum_{k=n+1}^{p} a_k s'_i((T_{0i}+f)P-kP) & \text{for } n < p \end{cases} \tag{50}$$

Each frame of the residual signal must be calculated over a number of periods plus the fraction such that the index returns to zero. Thus, the range of n is $0 \leq n < m(T_{0i} + f)$.

Eq. (50) can be implemented by ciruclar inverse filtering of the upsampled fractional pitch frame $s'_i(nP)$, repeated m times. Downsampling the output by P results in a $m(T_{0i} + f)$ length residual, representing m fractional pitch periods. Only one pitch period is to be concatenated to build the residual, $e(n)$, of the entire segment. Since each of the cycles of $e_i(n)$ is of fractional length, the constant pitch transform (section 4.2) is applied so that all cycles are of integer length. This requires that the constant pitch period, T_C, be chosen carefully so that at least m integer pitch periods exist. Finally, one of the cylces of $e_{iC}(n)$ is chosen and concatenated to $e_C(n)$.

This exhaustive algorithm for determining the fractional pitch and transforming it to an integer pitch period is extremely computationally expensive, considering that fractional precision is generally at least 1 decimal place. For example, if it is assumed that the integer pitch periods are correct to the nearest ± 2 samples and $P = 10$, each frame needs to be analyzed $2(2P - 1)$ or 39 times. Informal listening results demonstrated that a fractional precision of 1 decimal place is indeed sufficient with less noticeable improvements at higher degrees of precision. Additionally, the interpolation filters for the constant pitch transform need to be very large since the sampling rate has been upsampled by P. However, since this step is only necessary during the database preparation stage, the computation time is not of primary concern. Note that synthesis using fractional pitch analysis algorithm does not affect the unit concatenation, prosody matching, and synthesis stages discussed in sections 5.3, 5.3.3, and 5.3.1.

4.1.2 CLP Synthesis

The output waveform is synthesized by filtering the residual by the all-pole synthesis filter, $1/A(z)$, as shown:

$$S'(z) = \frac{E'(z)}{A(z)} \tag{51}$$

Traditionally, when applying the inverse filter, the initial values of $s'(n)$ for $n < 0$ are set to 0. Since, during CLP analysis, the values of $s(n)$ for $n < 0$ are not 0, this would result in slight audible distortions in the output signal. In order to reduce this distortion, the filtering has been implemented using three unique methods, of differing complexity, all producing satisfactory preliminary results. In two of the methods the filter is implemented in IIR form with variations in the initial conditions. In the third method, the filter is implemented as a zero-phase FIR circular filter.

Method 1 The circular IIR filtering of the all-pole CLP filter can be described by:

$$s'(n) = e'(n) - \sum_{i=1}^{p} a_i s'(n-i) \tag{52}$$

for $n \geq p$, and

$$s'(n) = e'(n) - \sum_{i=1}^{n} a_i s'(n-i) - \sum_{i=n+1}^{p} a_i s'(T_0 + n - i) \tag{53}$$

for $n < p$, where the pitch period, T_0, is the frame length. In the first method, the initial values of $s'(n)$ $(n < 0)$ are allowed to evolve to the correct values by synthesizing each frame circularly N number of times. Initially, $s'(n)$ is set to 0. For each successive iteration, however, the previous $s'(n)$ is used as input. The frame corresponding to the final iteration is taken as the output frame. The circular filtering allows for the initial values to converge. Informal listening tests were conducted for this method for $N = 1$, $N = 2$, $N = 3$, and $N = 4$, with audible improvement in each case. It was determined that for $N = 4$, the synthesized speech had no audible distortion.

Method 2 The second method is very similar to the first method, except that the initial values of $s'(n)$ for each frame are the output values of $s'(n)$ for the previous frame. If the pitch periods are accurately predicted, for voiced speech this assumption for the initial values of $s'(n)$ will always be better than 0. Though, the resulting

waveform was of higher quality than in the first method, slight distortion was still audible. Hence, the iterative circular filtering was also implemented for this case. It was only necessary to iterate up to $N = 2$ to get the same perceptual quality as for $N = 4$ in the first case. The requirement of fewer iterations makes this method less computationally expensive than method 1.

Method 3 In the third method, an FIR filter is created from the original IIR LPC filter. The FIR filter coefficients are the impulse response of the LPC filter. The synthesis quality for this method is clearly dependent on the length of the impulse response. However, the length can be constrained to $T0$ by modulo wrapping and adding as shown in equation 55. Even though this method has a higher computational cost than the methods 1 and 2, it has been implemented for investigative purposes. This technique is implemented for each frame in the following manner:

- First, an impulse response, of length NT_0, is calculated from the synthesis filter.

$$h(n) = \delta(n) - \sum_{i=1}^{p} a_i h(n-i) \tag{54}$$

- Next, $h(n)$ is converted to a length, T_0, impulse response, $h'(n)$, by modulo wrapping of $h(n)$ by $T0$ and adding to itself, as follows:

$$h'(n) = \sum_{k=0}^{N-1} h(kT_0 + n) \tag{55}$$

- If the length of $h(n)$ is not an integer multiple of T_0, it is zero-padded before performing the modulo wrapping.

The waveform is synthesized by circular convolution of $e'(n)$ with the FIR filter $h'(n)$. Similar to the number of iterations in the previous two methods, in method 3, N represents the length of the impulse response in terms of multiples of the frame length (T_0). The informal listening tests showed that the performance of this method is similar to that of method 2 and satisfactory quality speech is achievable for $N = 3$.

4.1.2.1 Objective Analysis

To further validate the subjective quality analysis, the reconstruction signal-to-noise ratios (SNR) of the three methods were computed using synthetic signals. Synthetic speech signals were used to guarantee perfect periodicity of the speech. These signals were generated by performing traditional LPC analysis on windowed stationary voiced speech to obtain the coefficients, and exciting the all-pole filter with an impulse train of a given pitch period, T_0. To maintain consistency with the TTS implementation and speech database used for this research, the synthetic speech signals were attributed a 16kHz sample rate and the LPC filter order was 16. For this sample rate, synthetic speech signals were generated for a nominal pitch period of $T_0 = 110$ samples ($\sim 145Hz$). Additionally, since short pitch periods are a know problem area, a high frequency speech signal of $\sim 267Hz$ ($T_0 = 55$ samples) was also generated.

Table 8 compares the reconstruction SNR for the three synthesis methods for voiced speech synthesized at these pitch periods. The table shows that for a given number of iterations method 1 and method 3 are relatively similar in quality and method 2 is superior by nearly twice. Also, with only 1 iteration, the FIR filter designed in method 3 performs very poorly. Method 2 is clearly the best option with 114 dB SNR at $N = 2$ for nominal T_0. However, even though the SNR of method 3 at $N = 3$ was 91 dB, the audible quality of real speech signals was identical to method 2 at $N = 2$ and method 1 at $N = 4$ (114 dB SNR). Further analysis conducted for method 3 by Ertan[34] demonstrated more uniform performance in the case of speech that has short pitch periods combined with grouped formants. For this special case, the modeling error for CLP and traditional LP is high, and method 3 may be the better choice for synthesis. Though, the second method has the lowest complexity, and best reconstruction SNR per iteration for nominal pitch periods at 16 kHz, the third method may be a more robust approach, especially for critically sampled speech. However, since the signals used for this research are generally oversampled (16 KHz),

Table 8: Comparison of signal-to-noise ratios of the three different CLP synthesis techniques for N ranging from 1 to 4

	Reconstruction SNR (dB)					
	Method 1		Method 2		Method 3	
N	T_0 ***= 55***	T_0 ***= 110***	T_0 ***= 55***	T_0 ***= 110***	T_0 ***= 55***	T_0 ***= 110***
1	19.0	28.8	36.7	60.5	-2.95	-2.95
2	36.7	60.5	64.7	114	36.4	65.2
3	51.8	87.5	92.4	166	49.5	91.2
4	64.7	114	120	215	62.8	117

method 2 is the preferred choice.

Based on these objective observations one criteria for choosing N can be the number of iterations necessary to achieve a desired reconstruction SNR. The thesis by Ertan [34] metions a reconstruction SNR of 72 dB for the synthesized speech to be indistinguishable from the original. In this case, selecting $N = 2$ for method 2 and $N = 3$ for method 3 may be sufficient when computational complexity is an issue. Even though the reconstruction SNR for the short pitch period, $T_0 = 55$ samples, is not greater than 72 dB, periods of this length are not common at 16 kHz sample rate.

The analysis conducted in this section leads to another interesting observation. Recall from section 2.5.3.1 that the reconstruction SNR achieved for the vowel /aw/ was 30.8 dB. Further analysis showed that the reconstruction SNR for RELP using synthetic signals was rarely greater than 45 dB. This is significantly lower than the SNR achievable by CLP given in table 8.

4.2 Constant Pitch Transform

4.2.1 Theory

The CPT basically interpolates every frame of the input signal to a fixed length. Since, for CLP every frame is exactly one pitch period, T_C, of the residual signal as shown in Figure 21, applying the CPT to the residual signal, e_C, transforms the entire signal to one constant integer pitch. This results in a standardized database

that can be prosodically modified with ease during synthesis. Additionally, it removes the need to store the pitch epoch locations as required by RELP based TTS. The pitch warping is achieved by upsampling each frame of the residual signal by the constant pitch period, T_C, lowpass filtering for interpolation and anti-aliasing, and downsampling by the original pitch period, T_0 (see Figure 21). For a given input speech frame, $s(n)$, the upsampling results in $\tilde{s}(\tilde{n})$ defined by:

$$\tilde{s}(nT_C) = s(n) \tag{56}$$

and 0 otherwise. The Type I, FIR, zero-phase lowpass filter, $h(k)$ is then applied to $\tilde{s}(n)$ to interpolate the upsampled signal,

$$y(n) = \sum_{k=\lfloor -L/2 \rfloor}^{\lfloor L/2 \rfloor} h(k)\tilde{s}(n-k) = \sum_{k=n-\lfloor L/2 \rfloor}^{n+\lfloor L/2 \rfloor} \tilde{s}(k)h(n-k) \tag{57}$$

where the symbol $\lfloor \cdot \rfloor$ refers to the floor of that expression. Finally, the output (constant pitch) signal, $s_C(\eta)$, is calculated by downsampling $y(n)$ by T_0:

$$s_C(\eta) = y(\eta T_0) = \sum_{k=\eta T_0 - \lfloor L/2 \rfloor}^{\eta T_0 + \lfloor L/2 \rfloor} \tilde{s}(k)h(\eta T_0 - k). \tag{58}$$

In order to avoid aliasing, it is necessary to satisfy the condition that $T_C \geq T_0$. Note that the lowpass filtering of a signal upsampled by an entire pitch period makes this technique extremely computationally expensive. The length, L, of the filter, $h(k)$, needs to be very large because the upsampling leads to a very narrow bandwidth. This method can be implemented much more efficiently by deriving a multi-rate filter that only processes those values required to produce the output signal, $s_C(\eta)$. The filter in equation (58), can be implemented such that only the non-zero values of the upsampled signal $\tilde{s}$ are processed by substituting nT_C for the index, k,

$$s_C(\eta) = y(\eta T_0) = \sum_{nT_C = \eta T_0 - \lfloor L/2 \rfloor}^{\eta T_0 + \lfloor L/2 \rfloor} \tilde{s}(nT_C)h(\eta T_0 - nT_C). \tag{59}$$

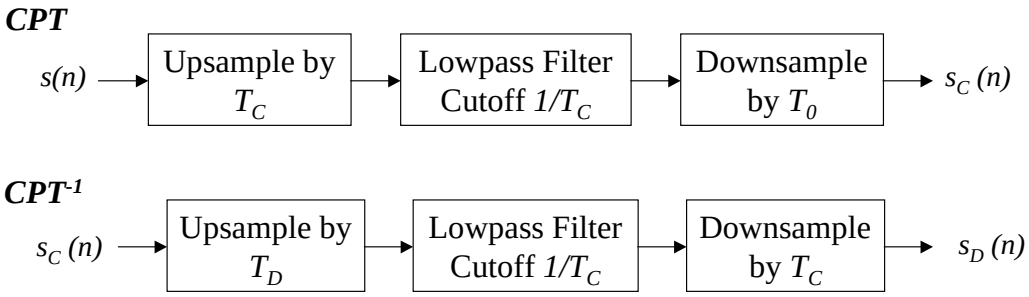

Figure 21: Block diagram of the constant pitch transform and the inverse constant pitch transform.

From equation (56), $s(n)$ can be substituted back into equation (59) resulting in a single equation that can efficiently compute the entire CPT operation,

$$s_C(\eta) = \sum_{n=\lfloor(\eta T_0 - \lfloor L/2 \rfloor)/T_C \rfloor}^{\lfloor(\eta T_0 + \lfloor L/2 \rfloor)/T_C \rfloor} s(n) h(\eta T_0 - n T_C). \tag{60}$$

The inverse constant pitch transform is performed in the same manner as the CPT, except that the upsampling and downsampling pitch periods are different, as shown in Figure 21. The constant-pitch frames are now upsampled by the desired pitch period, T_D, and interpolated by the low-pass filter. Then, the frame is down-sampled by T_C, resulting in a frame of speech at the desired frequency. For the inverse constant pitch transform, the desired pitch is not constant. Therefore, it would appear that a new low pass interpolation filter with a cutoff at the desired frequency, F_D, needs to be designed for each frame. However, this can be avoided by ensuring that the desired pitch period, T_D is always less than the T_C. Then using the same filter as in the CPT, with cutoff frequency, $1/T_C$, is sufficient.

4.2.2 Pitch Modifications Using CPT

Though the CPT and inverse CPT present a theoretically capable method for modifying the pitch of exactly periodic signals, the method must be analyzed with speech signals to understand its performance. As in section 2.5.3.1, the analysis was conducted using synthetic speech signals to guarantee exact periodicity. The synthetic

signals were generated using traditional LP analysis on real speech signals in the same manner as in section 2.5.3.1. The pitch was modified by applying the CLP/CPT analysis method to the signals followed by the inverse CPT for pitch modification of the residual. The signals were re-synthesized using method 2 of CLP synthesis (see section 4.1.2). Figure 22 shows the resulting waveform for the vowel phoneme /aw/ after modifying the pitch period by a factor of 1.20 (20% increase). Note that even though the frames are concatenated from end to end with no interpolation or windowed overlap-add smoothing, the transition from one frame to the next appears relatively smooth. This can be attributed to CLP/CPT analysis/synthesis of an exactly periodic signal. More significantly, Figure 23b shows that the pitch modified, CLP/CPT synthesized waveform for the nasal phoneme, /n/, was relatively smooth. When compared to the RELP-PSOLA modified waveform of the same nasal phoneme (Figure 18b), this waveform appears to have much smoother frame transitions. Pitch modification of voiced to unvoiced speech transition, represented by the phonemes /e-s/, is shown in Figure 24. For this case, the CLP/CPT representation also appears to have more robust frame transitions than for RELP-PSOLA shown in Figure 19. Figure 25 shows the effect of pitch increase at the voicing transition of /b-a/ in the word "balloon". This transition of a plosive to a vowel appears to be difficult to model. This problem is addressed by modifying the line spectral pairs, discussed later in the next chapter.

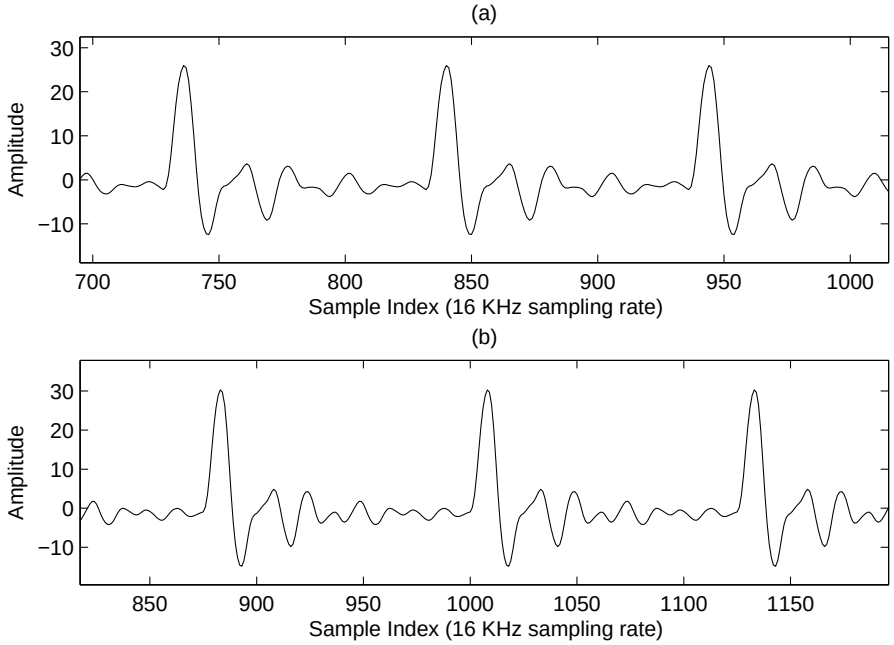

Figure 22: (a) The original synthetic waveform of the vowel phoneme /aw/, and (b) the CLP/CPT synthesized waveform with the pitch increased by 20%

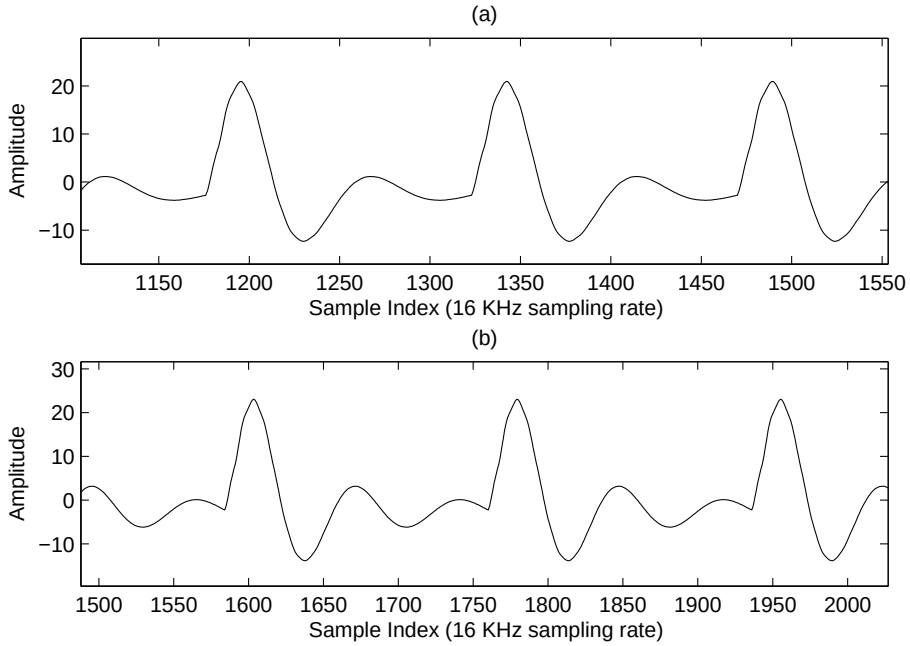

Figure 23: (a) The original synthetic waveform of the nasal phoneme /n/, and (b) the CLP/CPT synthesized waveform with the pitch increased by 20%

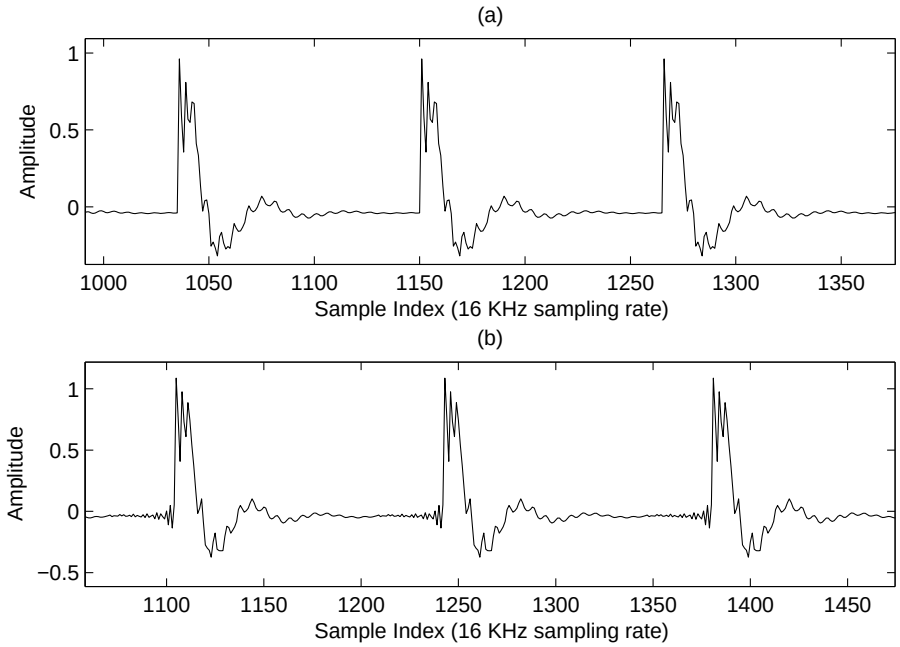

Figure 24: (a) The original synthetic waveform of the voiced-unvoiced transition phonemes /e-s/, and (b) the CLP/CPT synthesized waveform with the pitch increased by 20%

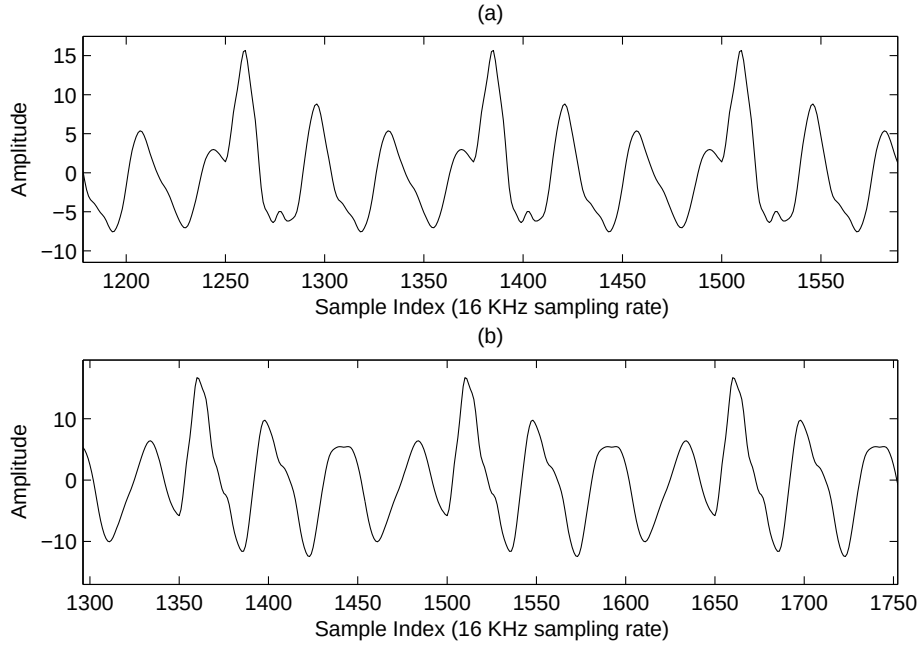

Figure 25: (a) The original synthetic waveform of the voicing transition phonemes /b-a/, and (b) the CLP/CPT synthesized waveform with the pitch increased by 20%

CHAPTER V

TTS IMPLEMENTATION USING CLP/CPT

The implementation of TTS for this research was conducted using two existing, phonetically labeled speech databases that are available for research: the Boston University FM radio speech corpus [66], also used for the RELP-PSOLA implementation described in section 2.5, and the CMU Communicator speech database [76]. Since the phoneme boundaries for both databases have been determined automatically, the labels were hand corrected to avoid synthesis errors. Hand correction is a time consuming but necessary step for preparing the synthesis database to minimize artifacts. Since the experimental synthesis database for this research is a very small subset of the corpora, hand corrections was not an overwhelming task. Additionally, boundary locations for voiced and unvoiced speech were determined automatically and hand corrected. The other parameters consistng of the pitch, pitch epoch locations, residual signal, and LP coefficients were derived using the methods described in this chapter.

One of the key implementation problems with synthesizing artifact-free, "natural quality" speech is the variations in spectral characteristics of the segments. Many databases, including the ones used for this research, are recorded over a relatively long period of time leading to slight changes in the acoustical environment. This results in audible spectral and dynamic variations in the synthesis units. This section, initially, introduces a design for LPC-based spectral correction to equalize the database. Other implementation issues for CLP/CPT analysis and synthesis including unit concatenation and prosody matching are also detailed.

5.1 Database Equalization

Often the recording environment of the speech corpus used for creating the database does not remain constant throughout its acquisition. Even in a controlled environment, when recording a large corpus, it may not be possible to maintain consistent spectral characteristics throughout the recordings. In discussing the pros and cons of unit selection based TTS, Breen indicates that for large TTS databases, ensuring consistent voice quality during the recording stage is a key concern [17]. During TTS implementation using RELP and later the CLP/CPT method in this research, it was discovered that this was the case for the CMU Communicator speech database, and particularly prominent for the news reports in the Boston University FM radio speech corpus. There were noticeable differences in the acoustical characteristics of the different units of speech in the database. These acoustical differences are characterized by differences in the general shape of the spectra of the units.

Stylianou presented a method to compensate for the spectral differences by modeling the acoustical space of the speech recordings using a Gaussian Mixture Model (GMM) and deriving autoregressive correction filters [89]. He attributed the acoustical "inter-session" and "intra-session" variabilities during recordings to a number of factors including the emotional state and health of the speaker, differences in recording equipment, and fatigue during lengthy sessions. In this method, the GMM model is combined with the log likelihood function to detect whether the in voice quality between a reference and test segment is the same or different. For the test segments with different voice qualities, an autoregressive correction filter is derived based on the power spectral densities. This method, though effective, is dependent on the accuracy of the detection algorithm. A similar method of deriving autoregressive filters for channel equalization was presented by Shi and Chang et al [82]. The recording channels are modeled by 30 point IIR filters derived from 256 point PSDs. This method

demonstrated effective equalization without the use of a detection algorithm for determining whether or not to apply equalization. The performance of both techniques were measured subjectively by synthesizing utterance before and after equalization with results significantly favoring the equalized database.

In this research a linear prediction based equalization method for spectral normalization of the units was implemented, prior to the CLP analysis, to resolve the acoustic/spectral mismatches. This method was independently developed at an early stage of the thesis, and included in a poster presentation at the Acoustical Society of America meeting [84]. In this method, equalization of the database is achieved by generating a unique spectral correction filter, for each database unit. The filters are derived from the LP coefficients of a reference utterance and the LP coefficients of each unit. The LP analysis is performed with a relatively low order and large frame size, to ensure that the prediction coefficients do not model the speech itself, but rather the general spectral characteristics. The method is outlined below:

- For the reference spectrum, select a speech unit that has the desired acoustical characteristics, from the same database (same speaker) to be equalized.
- Apply a 1024 sample Hanning window to the reference segment that is centered at the midpoint of the segments.
- Determine the reference prediction coefficients $(a_{R1}, ..., a_{R5})$ from the analysis frame of the reference unit, using 5th order autocorrelation LP analysis.
- For every other unit within the database, obtain 1 analysis "test" frame by applying a 1024 sample Hanning window to each unit, centered at the midpoint of the units.
- For each unit, find the prediction coefficients $(a_{T1}, ..., a_{T5})$ of the spectrum of the "test" frames, using 5th order autocorrelation LP analysis.

- Filter each of the database units by the normalization filter given below:

$$H(z) = \frac{G_d}{G_u}\left[\frac{1 - \sum_{i=1}^{p} a_{Ti} z^{-i}}{1 - \sum_{i=1}^{p} a_{Ri} z^{-i}}\right] \tag{61}$$

  where the order p is set to 5.

This spectral equalization method is computationally simpler than the methods presented above. Instead of calculating FFTs and deriving correction filters from the PSDs, this method uses only 5th order LP coefficients. Perceptually, the resulting speech units had a very similar acoustical quality. The synthesized utterance had an improved voice quality since audible spectral mismatches were removed. However, since the normalization filters have non-linear phase, they can introduce a phase mismatch between units resulting in junctural artifacts. This problem can be resolved by implementing the filter in equation (61) as a zero-phase FIR filter. This method is outlined below:

- Determine the filter $H(z)$ in Eq. (61).
- Calculate a 64-point impulse response, $h(n)$, in the range of $-\pi$ to π.
- Filter each unit by the zero-phase FIR filter, $h(n)$.

To analyze the results of this equalization technique, the power spectral densities (PSD) of a number of randomly selected units were observed before and after applying equalization. Figure 26 shows the results of the spectral normalization algorithm for two units. Figure 26(a) shows the power spectral density of the unit with the desired spectral characteristics, chosen from the Boston University FM corpus. The unit with the target spectral/acoustic characteristics was chosen subjectively by listening to various units within the database. Figures 26(b) and (c) show the before and after PSDs of a randomly selected unit. Similarly, Figures 26(d) and (e) show the before

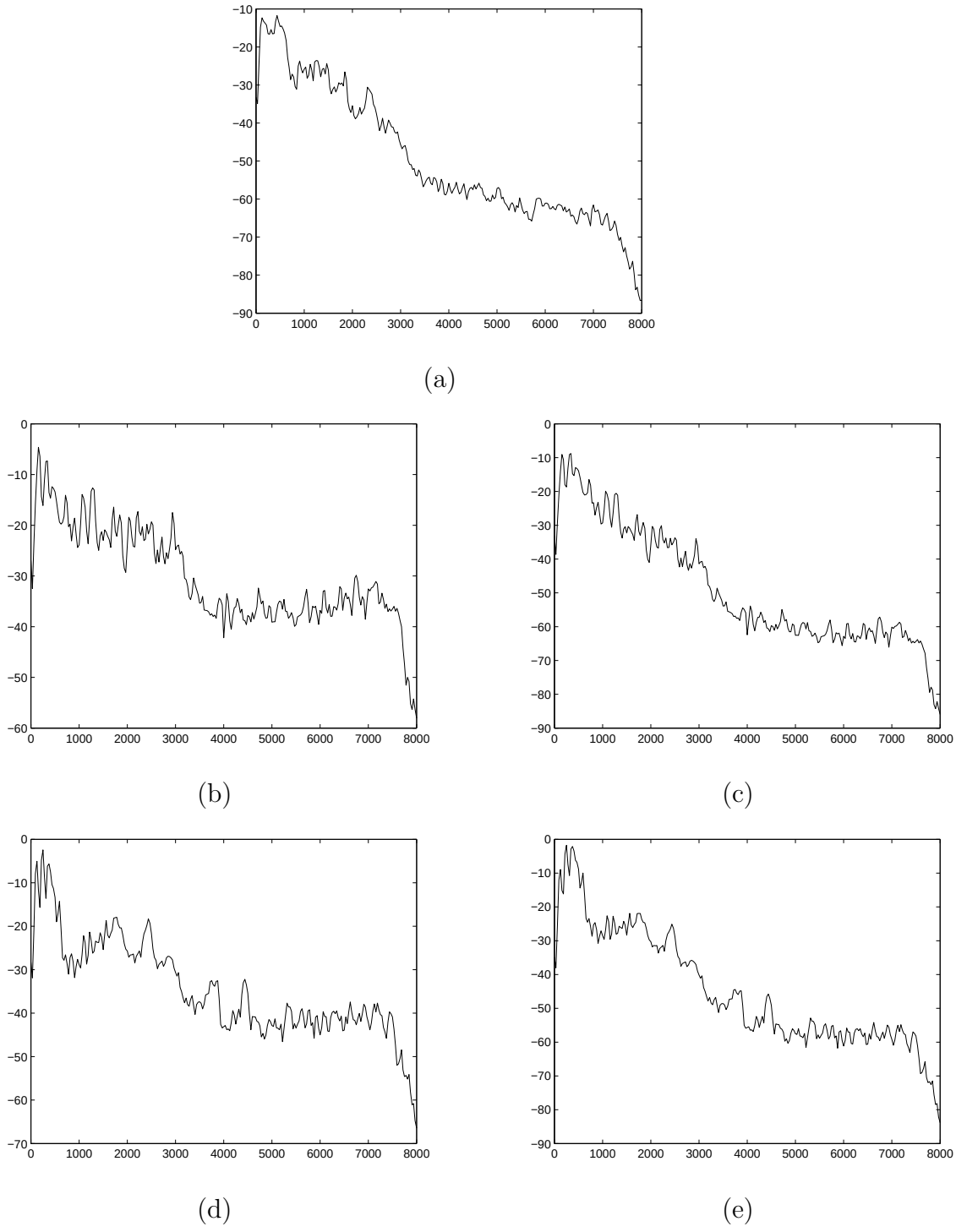

Figure 26: Results of spectral normalization on a randomly selected unit;(a) PSD of unit with desired spectral characteristics; (b) and (d) PSDs of two different database units before spectral normalization; (c) and (e) PSDs of both units after spectral normalization

and after PSDs of a different unit from the database. Note that the general shape of the spectrum of the normalized unit is relatively similar to the spectral shape of the unit with the desired acoustic characteristics. The location of the formants remain the same as the original database unit by this method.

In addition to mismatches in acoustical characteristics, it was found by initial TTS implementations that audible artifacts were present at boundaries if the gains of two adjacent units were drastically different. Even though the correction filter in equation (61) accounts for gain normalization based on the linear prediction gain, it affects the overall gain of the entire segment. The gain at the unit boundaries can still differ significantly. Hence, an additional gain normalization technique, which resulted in smoother transitions across unit boundaries was implemented. The normalization is performed on the database units following the acoustic equalization and prior to the LP or CLP analysis. In this method a gain vector is calculated for each unit that only normalizes the gain of the phonemes at the unit boundaries. Note that the normalization is only applied to unit boundaries with voiced phonemes. This method is outlined below:

- For each database unit, the average energy of a number of 20 millisecond overlapping frames at the beginning, E_{Ti}, and end, E_{Tf}, of the unit are calculated. The energy of each frame, $s(n)$, is calculated according to equation (62) below:

$$E = \frac{1}{N}\sum_{n=1}^{N}[s(n)w(n)]^2 \tag{62}$$

  where $w(n)$ is a Hanning window of length N.

- Based on a predetermined constant reference energy level, E_R, a new gain vector for the entire unit, $GV(n)$, is calculated that only affects the beginning and

ending phonemes of the unit as shown in equation (63) below:

$$GV(n) = \begin{cases} \frac{n}{N_i}(1 - \sqrt{E_R/E_{Ti}}) + \sqrt{E_R/E_{Ti}} & \text{for } n \leq N_i \\ 1 & \text{for } N_i < n < N_f \\ \frac{n}{N_f}(\sqrt{E_R/E_{Tf}} - 1) + 1 & \text{for } n \geq N_f \end{cases} \quad (63)$$

where N_i and N_f are the lengths of the initial and final phoneme in the unit, respectively.

- Each unit is multiplied by its corresponding gain vector.

Note that the initial and final energy values, E_{Ti} and E_{Ri}, are calculated only for voiced and partially voiced boundary phonemes. For unvoiced unit boundaries, the gain factors in equation 63 are set to 1. Since phonemes at unit boundaries are cut at the midpoints, voicing transitions are not an issue.

5.2 Analysis

5.2.1 Pitchmark Placement

It is well-known that real speech signals are not perfectly periodic and even at the fractional resolution there will be slight errors in the pitch period. Though, not audible in synthesized speech (see section 4.1.2) these errors are magnified during prosodic modifications, sometimes resulting in audible artifacts. To minimize the effect of these errors, the pitch cycles begin and end, not exactly on the pitch epochs, but at the low instantaneous energy region at the onset of the pitch epochs. This is determined by locating the first zero-crossing prior to the pitch epochs. The pitchmark locations, M_i, are determined in the residual domain according to the method by [48]. For this method a residual signal is initially calculated for each unit using the standard windowed LPC analysis and inverse filtering. Since accurate pitchmark placement is key to the success of CLP analysis, hand correction of placement errors may be necessary. Following the pitchmark placement procedure, unit boundaries need to

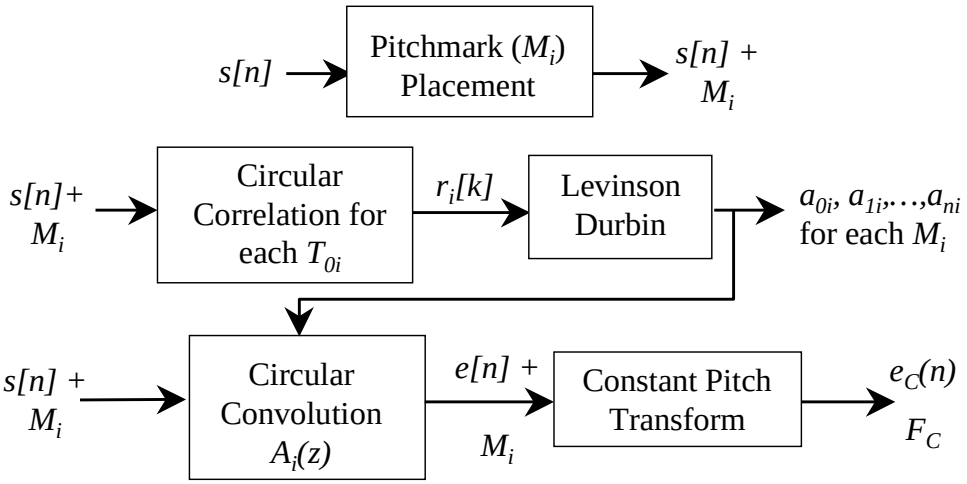

Figure 27: Block diagram of the circular linear prediction analysis phase.

be truncated so that each database unit begins exactly at the beginning of a pitch cycle, and ends exactly at the ending of a pitch cycle. This assures that units are concatenated only at pitchmark locations.

5.2.2 Constant-Pitch Segment Database

Once the unit and pitchmark database have been created, CLP analysis is performed on each unit and the residual signal is transformed to a constant pitch by the CPT to create a uniform parametric database, as shown in the block diagram in Figure 27. The analysis is performed on each non-overlapping frame of speech at a pitch-synchronous frame rate. The pitch periods are fine tuned to a fractional length with one decimal point of precision. The fractional length residual from CLP analysis is converted to integer length by the CPT. Generation of the uniform parametric database based on the CLP/CPT described in sections 4.1.1 and 4.2, is conducted on a frame-by-frame basis on every speech segment as summarized below:

- Upsample the speech segment by $P = 10$.
- For each upsampled frame, i, of length $T_{0i}P = (M_{i+1} - M_i)P$ perform fractional CLP analysis on all pitch periods in the range of $((T_{0i} - \alpha)P, (T_{0i} + \alpha)P)$, where

α is the assumed accuracy (in samples) of the integer pitch periods. For the current implementation α was set to 2 samples.

- Select the parameters, $A_i(z)$ that correspond to the pitch period that results in the maximum prediction gain.

- Calculate the residual signal $e_i[n]$ for the frame through circular inverse filtering and apply CPT to it to obtain the constant pitch residual, $e_{Ci}[n]$. The constant pitch T_C is fixed to be greater than the largest pitch period in the entire database. Furthermore, since the $s[n]$ has been upsampled by P, the condition for T_C becomes:

$$T_C \geq MAX(T_0) \times P \tag{64}$$

- Finally, the residual signal for the current frame is appended to the end of the previous frame.

For each unit, the CLP coefficients, $(a_0, a_1, ..., a_p)$ of each frame, the residual signal, e_C, and the original pitchmark locations, M, are stored in the parametric database. Even though the residual signal has been transformed to a constant pitch, the original pitchmark locations are necessary for constraining the prosodic modifications (pitch scaling) during synthesis (section 5.3.1). Additionally, it is necessary to maintain phoneme boundary locations within the unit. Since the phoneme boundaries occur at pitchmark locations, they are stored in terms of the pitchmark number. Their actual location in time may change depending on the target synthesis pitch track of the phoneme.

5.2.3 Constraints on the LSPs

It was observed by Ansari [5] that, with regards to residual-excited linear prediction, the peakiness and poor bandwidth estimates inherent in the LP spectra affects the

quality of speech when the pitch is modified. This observation resulted in an improvement to RELP-based TTS where the LP model was modified to produce a less peaky magnitude response. As opposed to modifying the model itself, this research has implemented a method for widening the bandwidths of extremely narrow-bandwidth formants by constraining the movement of LSP coefficient tracks in a way that reduces undesirable artifacts due to pitch modifications. This technique is based on the observation, presented in the research by Crosmer and Barnwell [22], that the speech formants are marked by two corresponding line spectral pairs (LSP) that are close together, and the formant bandwidth is related to the distance between the LSP coefficients. Hence, the line spectral frequencies (LSF) can be used to indicate the occurrence of narrow-bandwidth formants. Figures 28(c) and 29(c) show the LSF tracks for two different types of transition regions. It can be seen that the LSF tracks for the first two coefficient pairs, (P_0, Q_0) and (P_1, Q_1), are very close together, and the first two formants are poorly predicted by LP analysis.

For the CLP model, as observed by [34] and stated in section 4.1.2.1, "grouped" formants result in relatively higher spectral mismatch. This is because formants with very narrow bandwidths are generally not modeled well by LP analysis. In this case the synthesized speech may have an unpleasant "click" or "chirp". These artifacts can become even more audible when frames consisting of narrow-bandwidth formants are modified for prosody matching, the artifacts become intensified. This problem often occurs when transition regions of voicing modes (vowels, fricatives, nasals, etc.) are combined with short pitch periods. Figures 28(b) and 29(b) illustrate this problem for the two different types of transition regions. The dotted lines in the plots give the spectra obtained using autocorrelation LP analysis. Widening these predicted formant bandwidths by a small amount results in a smoother LP spectra, and minimizes the artifacts.

Fixed bandwidth expansion, presented by [103], is a simple and popular method

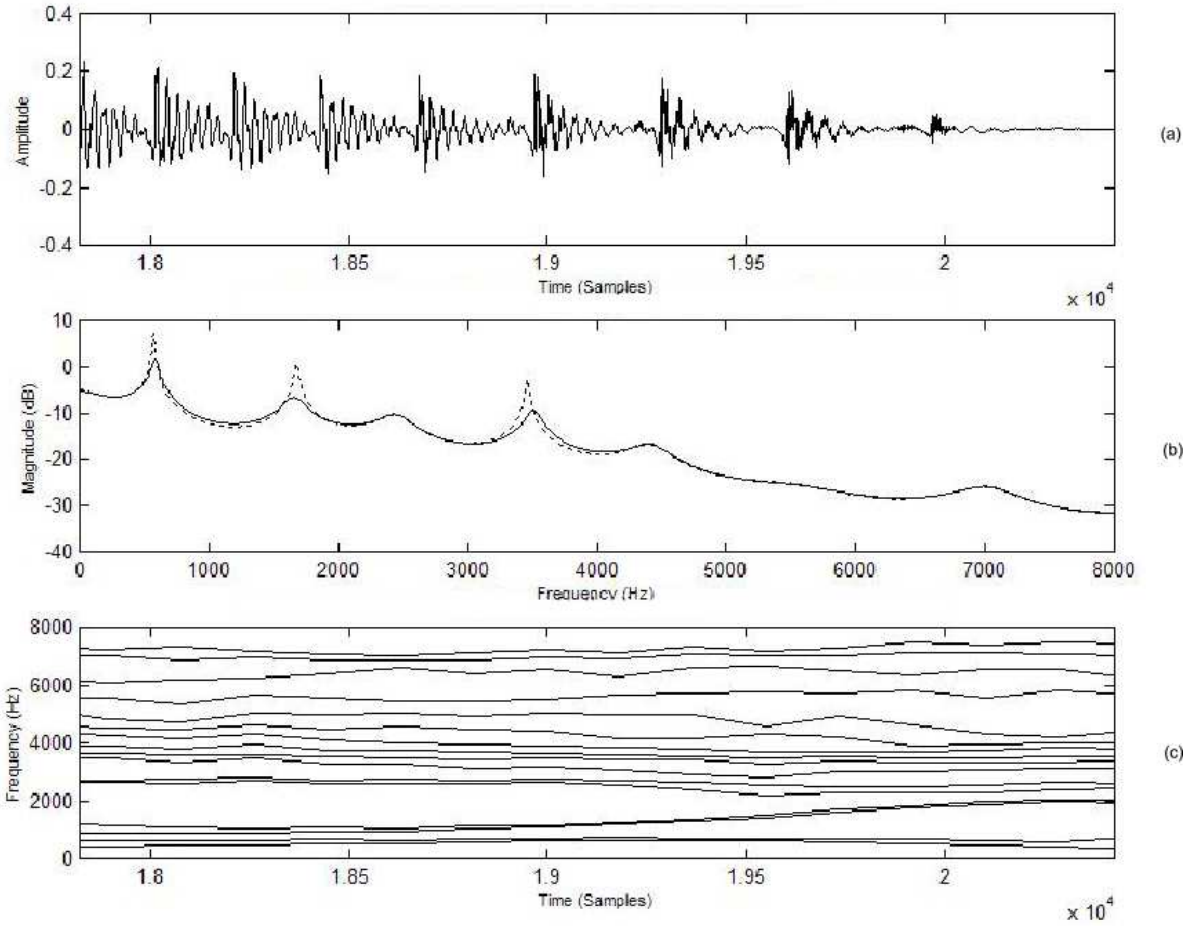

Figure 28: (a) Speech waveform for the transition "l-i" in "Flight", (b) the formants for the frame at $\approx$ 18700 samples generated using both methods, and (c) the LSF track for the coefficients obtained from CLP analysis.

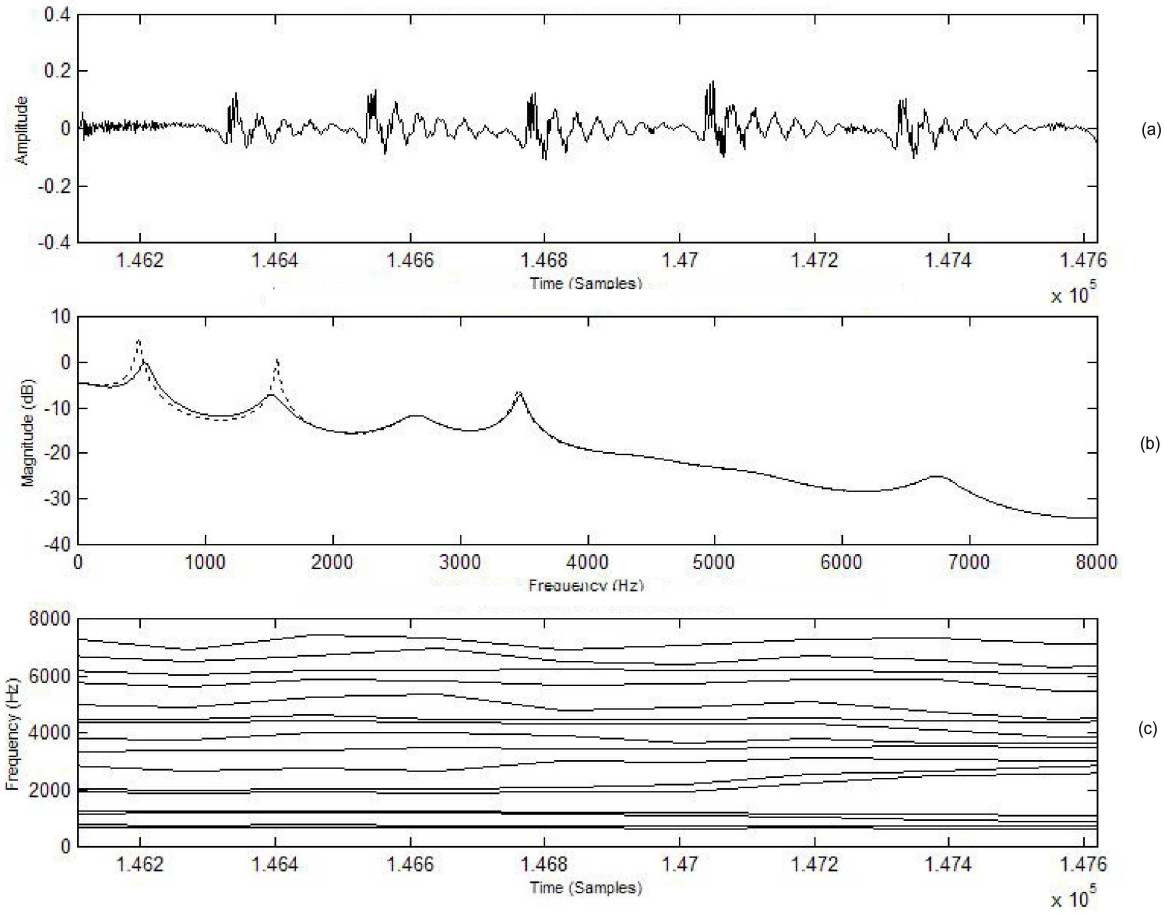

Figure 29: (a) Speech waveform for the transition "t-o" in "Charleston", (b) the formants for the frame at $\approx$ 14600 samples generated using both methods, and (c) the LSP track for the coefficients obtained from CLP analysis.

for reducing "peakiness" in formants. However, since this method affects the bandwidth of all of the formants equally, the accuracy of the LP model can be compromised significantly. In implementation it was found that even though this method reduces artifacts, it leads to an overall change in spectral characteristics of the voice when the pitch is modified. In this research, a method has been implemented for widening only the bandwidths of extremely narrow-bandwidth formants by adaptively modifying the LSP coefficients. This can reduce the undesirable artifacts due to prosodic modifications without affecting the overall speech characteristics. For this method, thresholds are applied to the LSP coefficients to maintain a minimum distance between each pair. Though, this minimum spectral distance threshold, F_m, is a constant value, it is not implemented in a strict sense. While maintaining a certain distance between each pair of coefficients, an acceptable distance between adjacent pairs (i.e. Q_i, P_{i+1}) also needs to be maintained. This procedure is implemented iteratively, first applying the threshold between each pair of LSP coefficients (i.e. P_i and Q_i), and then between adjacent pairs (i.e. P_i and Q_{i-1}). Before applying the threshold, the midpoints between each pair of coefficients, $C_{i,i} = (P_i + Q_i)/2$, and the midpoints between adjacent pairs, $C_{i,i+1} = (Q_i + P_{i+1})/2$, are determined. Then, the threshold is applied between each pair as shown in Eqs. (65) and (66).

$$P_i = MAX(MIN(P_i, C_{i,i} - F_m), C_{i-1,i}) \tag{65}$$

$$Q_i = MIN(MAX(Q_i, C_{i,i} + F_m), C_{i,i+1}) \tag{66}$$

After applying the threshold to all the (P_i,Q_i) pairs, a second pass is made on the LSP coefficients to apply the threshold to the adjacent pairs as shown in Eqs. (67) and (68).

$$Q_i = MAX(MIN(Q_i, C_{i,i+1} - F_m), C_{i,i}) \tag{67}$$

$$P_{i+1} = MIN(MAX(P_{i+1}, C_{i,i+1} + F_m), C_{i+1,i+1}) \tag{68}$$

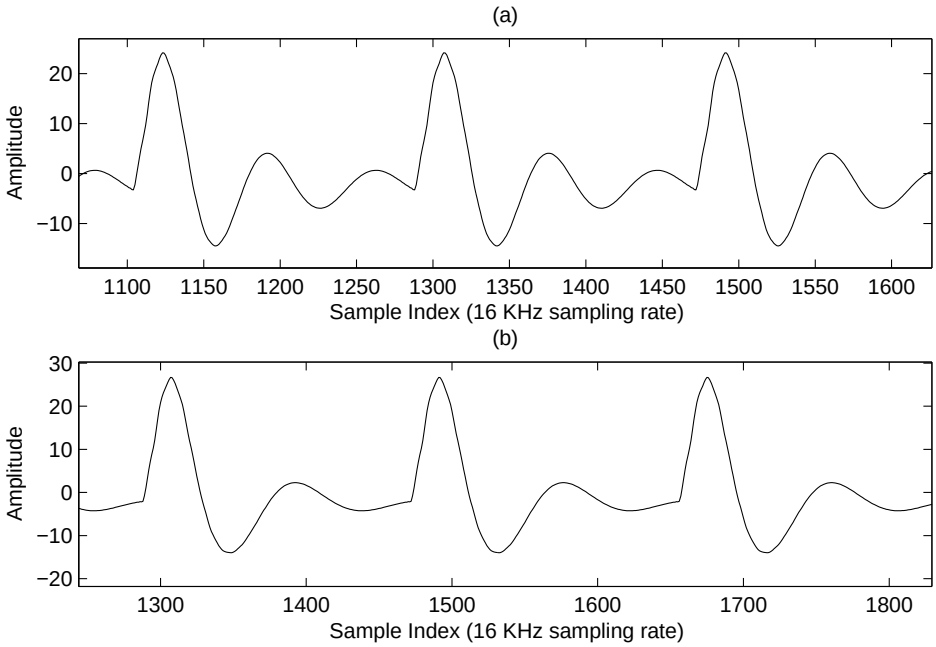

Figure 30: The CLP/CPT synthesized waveforms of the synthetic speech nasal phoneme /n/ with the pitch increased by 20% (a) before applying LSF thresholds and (b) after applying LSF thresholds.

Note that for N order CLP analysis, $i = 0...(N/2-1)$. For $i = 0$, $C_{-1,0}$ is set to $P_i/2$, and for $i = N/2-1$, $C_{N/2-1,N/2}$ is set to $(P_{N/2-1}+\pi)/2$. From Eqs. (65), (66), (67), and (68), it can be determined that the coefficients will not be modified unless the distance between any of them is greater than $2F_m$. The solid lines in the plots of Figures 28(c) and 29(c) show the CLP spectra obtained after applying the constraints on the LSP coefficients. It can be observed that applying the thresholds has an effect of spreading the bandwidth and reducing the peakiness of the LP spectra. It can also be seen that when the LP formants are relatively smooth, the spectra is not affected.

When pitch modifications are applied, the relatively smooth predicted spectra resulting from this method, provide for synthesis with fewer to no audible artifacts. The LSP constraints were applied before modifying the pitch of the same synthetic signals demonstrated earlier for RELP PSOLA synthesis (Figures 17, 18, and 19) and CLP/CPT synthesis (Figures 22, 23, 24, and 25). The results showed (slight)

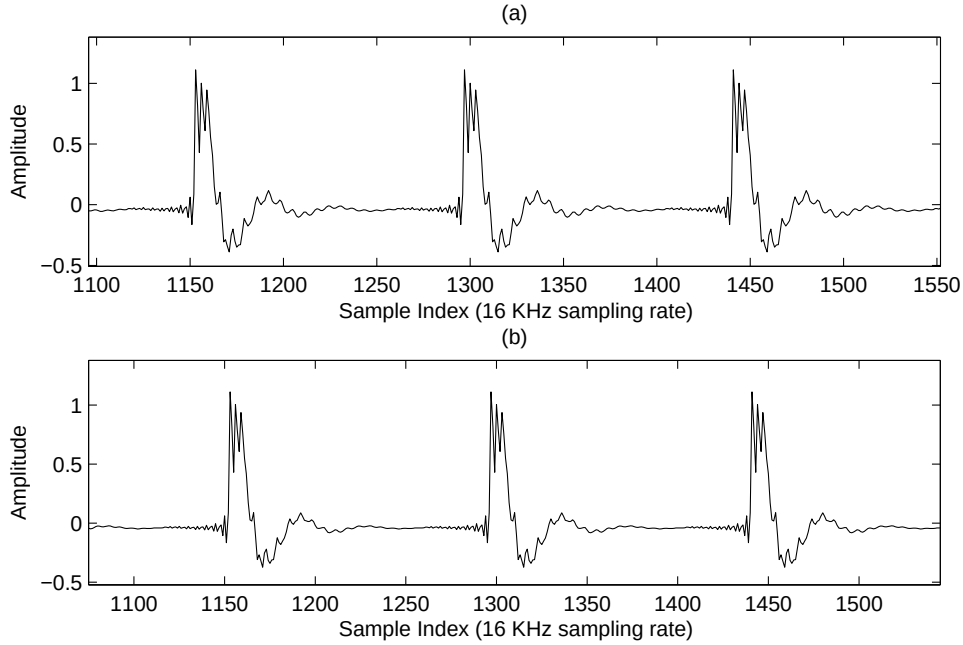

Figure 31: (a) The CLP/CPT synthesized waveform of the synthetic speech transtion /e-s/ with the pitch increased by 20% (a) before applying LSF thresholds and (b) after applying LSF thresholds.

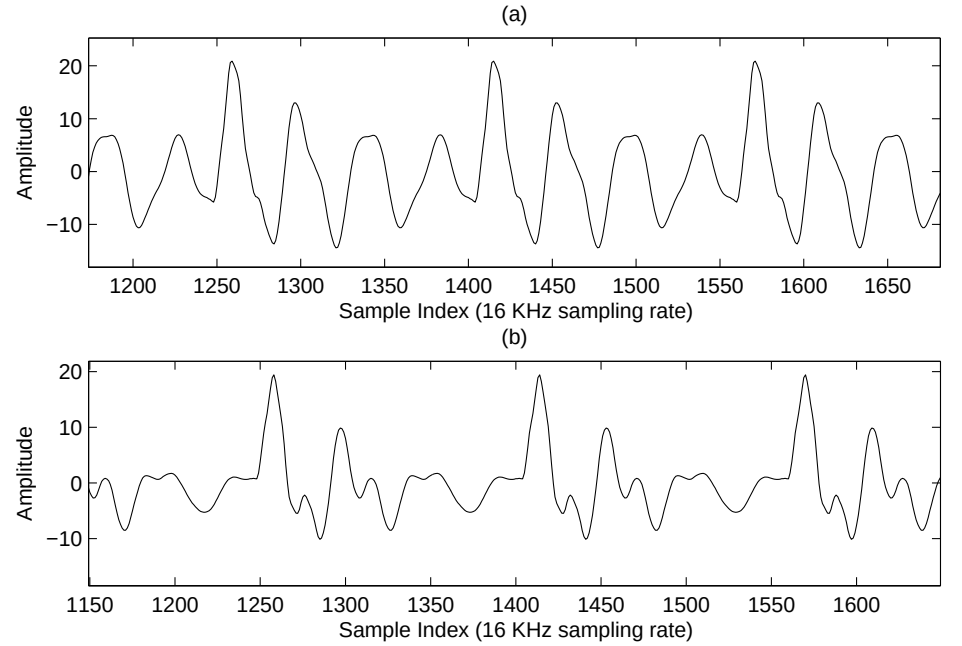

Figure 32: (a) The CLP/CPT synthesized waveform of the synthetic speech transition /b-a/ with the pitch increased by 20% (a) before applying LSF thresholds and (b) after applying LSF thresholds.

improvements in the smoothness of transitions between pitch modified frames. This is shown in Figure 30 for the phoneme /n/, Figure 31 for the vowel-consonant transition /e-s/, and Figure 32 for the consonant-vowel transition /b-a/. The Figures compare the resulting waveform after pitch modifications for the CLP/CPT synthesis without (a) and with (b) the application of the LSF thresholds.

A more dramatic example of the improvement made by this technique can be seen in the real speech signals in Figures 33 and 34. The Figures show the pitch-modified, real speech signals for the transitions /pau-b/ (Figure 33) and /n-d/ (Figure 34) realized using the CLP/CPT method without (a) and with (b) the LSF constraint technique. The pitch modifications are made without applying the LSF thresholds in Figures 33a and 34a, and with the thresholds to limit the minimum formant bandwidths in Figures 33b and 34b. Without the application of thresholds on the LSF tracks during analysis, the synthesized waveforms in both transitions contain loud "pop" artifacts. When the threshold technique is applied, the artifacts are significantly reduced and become inaudible.

5.3 Synthesis

The signal processing stage of synthesis in concatenative TTS consists of unit concatenation, prosody matching, and waveform synthesis. This sections details the implementation of these stages for CLP based TTS. The text parsing, unit selection, and target prosody determination are outside the scope of this research.

5.3.1 Prosody Matching

The target prosody for TTS is generally determined by the Natural Language Processing module in a TTS system using various methods, some of which are described in section 2.2. For the CLP/CPT based TTS implementation, however, the target pitch and duration were extracted from real speech recordings of the utterances to be synthesized. This was done to guarrantee that the target prosody was natural. Since

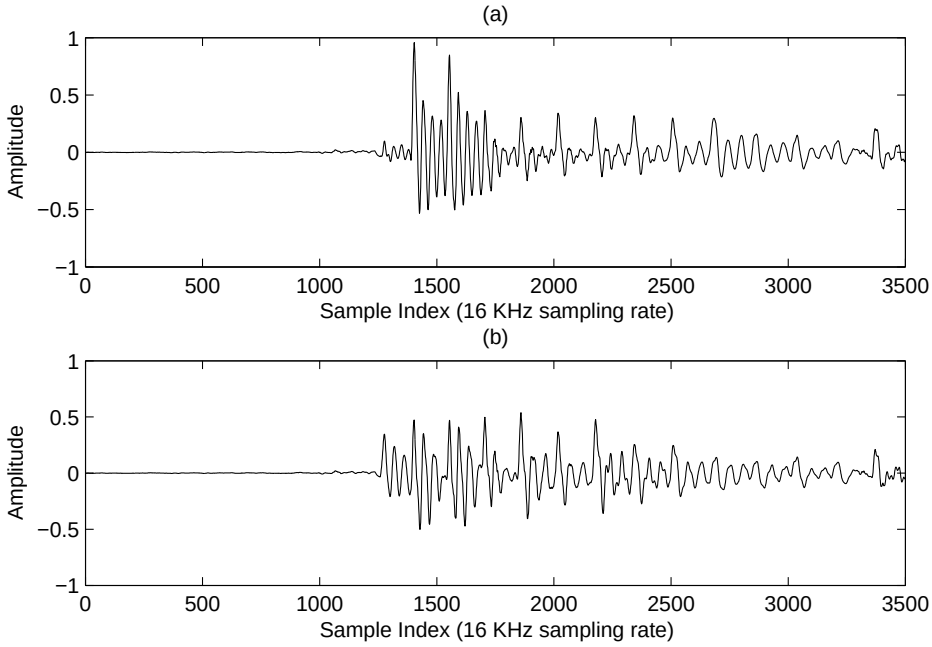

Figure 33: The CLP/CPT synthesized waveform of the real speech transition /pau-b/ with the pitch increased by 20%, (a) without applying thresholds to the LSF tracks and (b) with the application of the LSF thresholds to expand the formant bandwidths.

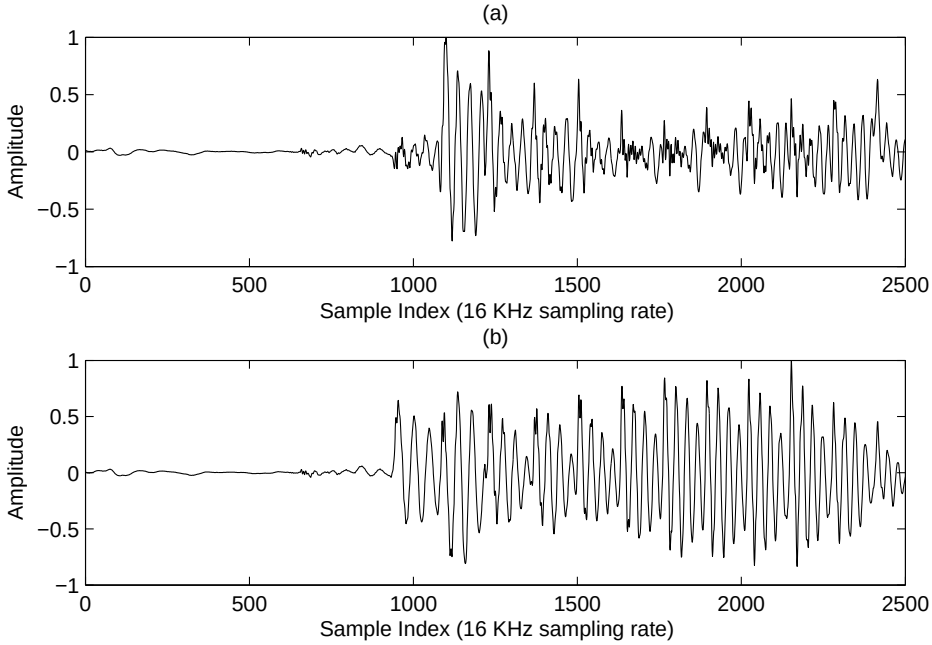

Figure 34: The CLP/CPT synthesized waveform of the real speech transition /n-d/ with the pitch increased by 20%, (a) without applying thresholds to the LSF tracks and (b) with the application of the LSF thresholds to expand the formant bandwidths.

the number of utterances synthesized for the purposes of this research was limited, this was feasible and helpful in studying the synthesis quality without the negative bias of unnatural prosodics.

At the time of unit concatenation, as the residual signal of the entire utterance is being constructed, the prosody modifications are implemented on each unit. The duration modifications are performed in the time domain in a manner similar to TD-PSOLA, and the pitch changes are implemented spectrally using the inverse constant pitch transform (section 4.2). The pitch modifications are performed prior to duration modifications, because the resulting residual signal will also change in duration. The target pitch track for a given segment is mapped to every frame so that there is a target pitch T'_{0i} for every pitch period. For each database unit, given the target T'_{0i} values, the constant pitch, F_C, and the unit length, N_e, a new set of pitchmarks, M'_i, is created, as follows:

$$M'_i = M'_{i-1} + T'_{0i} \quad \text{for } 1 < i < N_e F_C \tag{69}$$

where $M'_0 = 1$ and F_C is in units of (1/samples). The new residual signal for the target pitch, $e'_C(n)$, is calculated from the target pitchmarks, M'_i, by implementing the inverse CPT on every pitch period as shown in Figure 35. The target durations are achieved by either repeating or deleting entire frames of the residual. The duration factor for each phoneme is calculated from the target duration and the current phoneme duration. Based on this duration factor, frames of the phoneme are either repeated or deleted to increase or decrease the duration, respectively. As stated in 5.2.2, the phoneme boundaries are stored in terms of pitchmark (frame) indices, i. A mapping function is derived that maps the original frame indices, i, to the indices for the duration modified frames, j. The mapping function is applied to the pitch periods (frames), T'_{0i}, of each segment, to create a new set of pitch periods, $T'_{0MAP(j)}$,

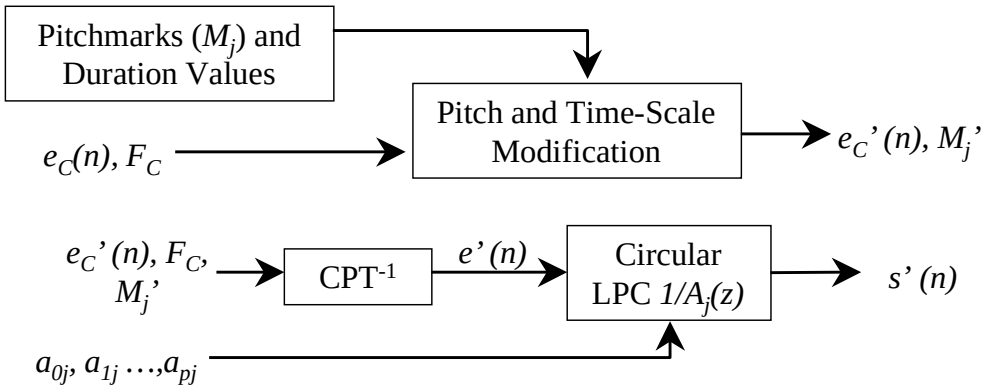

Figure 35: Block diagram of prosody matching and synthesis stages of CLP based TTS

and a third set of pitchmark locations, $\bar{M}'_j$, are calculated as follows:

$$\bar{M}'_j = \bar{M}'_{j-1} + T'_{0MAP(j)} \tag{70}$$

where $\bar{M}'_0 = 1$. The residual modified to the desired target pitch, $e'(n)$, is also modified in a similar manner by applying the mapping function, MAP, to implement the duration changes. The resulting residual, $\bar{e}'(n)$, consists of repeated and deleted pitch periods of $e'(n)$, as shown in equation 71.

$$\bar{e}'(n) = e'_{MAP(j)}(n) \tag{71}$$

Figure 35 gives a block diagram of the prosody matching and synthesis stages.

5.3.2 Prosody Modification Constraints

Though the CLP/CPT speech model allows for prosody modifications with minimal artifacts in the synthesized speech, as expected, there is a limit to the extent of the modifications before artifacts become audible. The limitations exist due to inherent errors in the modeling. For example, since speech is inherently quasi-periodic, an exact pitch cannot always be accurately determined, even with fractional resolution. Another reason is that the pitch at transition regions that contain stop consonants cannot be modified significantly without causing artifacts. This, in turn, limits the

amount that the adjacent voiced speech can be modified to prevent sudden inflections of the pitch.

The limits for the modifications are determined subjectively and can vary based on the type of phonemes. In this research, thresholds have been derived to limit the pitch-scale factor and duration factor. Based on informal subjective testing, different thresholds were derived for the various phoneme types. For example, vowels were limited to duration fluctuations of ±30%, fricatives were limited to ±18%, and stop consonants were limited ±10%. Additionally, to prevent sudden inflections in pitch, the pitch-scale factors are smoothed by a 6-tap moving average filter. Note that the thresholds derived in this research are considered specific to the CMU Communicator database. The best value for the prosody modification thresholds will depend on the database. Hence they should be adjusted through informal listening of synthesized utterances for the given database.

5.3.3 Unit Concatenation and Synthesis

With the assumption of exact periodicity, CLP/CPT parametric units can be concatenated simply by connecting the residual signals from end-to-end. For unit boundaries of voiced and partially voiced speech, it is important to maintain consistency of the pitch epoch location between units. The analysis method resolves this issue, since for voiced frames, the frame boundaries are from one pitch epoch to the next. For unvoiced frames the periodicity assumption does not apply and the concatenation points are not critical. It is good practice, however, to assure that the join points are near a zero crossing. Unlike RELP or PS-RELP, neither the residual signal nor the LSF parameters are interpolated at unit boundaries. Figure 36 shows the concatenation juncture for two different voiced phonemes. The small dotted lines on the waveforms mark the pitch epoch locations (frame boundaries) and the large dashed line shows the juncture of two units. It can be seen that since the units begin and

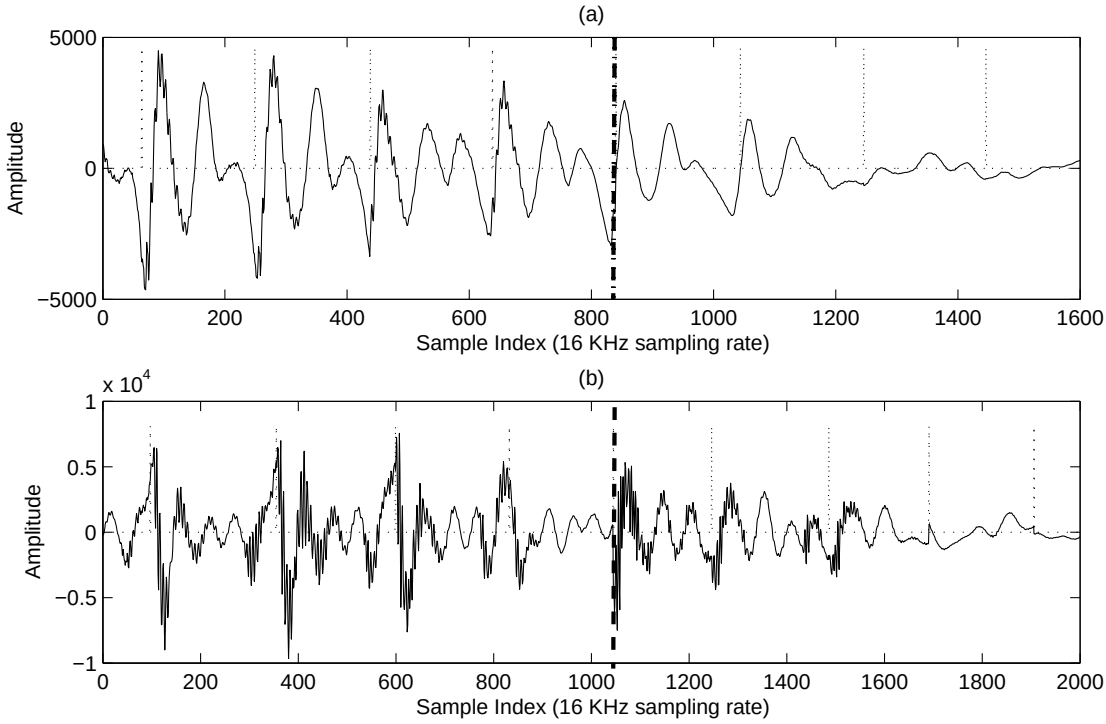

Figure 36: Examples of unit concatenation for CLP/CPT synthesis for (a) the nasal /n/ joining the units "in" and "n-Ch", and (b) the vowel phoneme /i:/ joining the units "seventy" and "y-two". The small dotted lines indicate the pitch mark locations (frame boundaries) and the large dashed line marks the boundaries of two units.

end at generally the same location within a pitch period, the concatenation does not lead to significant artifacts.

After the new pitchmarks, $\bar{M}'_j$, representing the target pitch periods and durations, and the new residual $\bar{e}_C$ for the target durations have been created, the inverse constant-pitch transform is applied to implement the pitch modification in the residual signal. Synthesis of the waveform is achieved by the CLP synthesis methods described in 4.1.2. All three methods were tested with similar results in terms of audible artifacts. Hence, method 2 was selected in the implementation for the reasons of lower computational complexity and superior reconstruction SNR.

CHAPTER VI

SUBJECTIVE TESTING OF SYNTHESIS QUALITY

Objective experiments conducted on synthetic speech signals in the previous chapters demonstrated that the CLP model is a more accurate linear prediction representation, resulting in a lower reconstruction SNR. However, as is the case for most speech processing studies, objective measures alone cannot present a true indication of the performance of the system. Subjective comparisons to an existing "reference" method must be conducted to accurately measure the improvement in quality of the new "test" method. Specifically for text-to-speech synthesis, as mentioned in the introduction and background chapters, one of the largest challenges is to synthesize speech with prosodic inflections without compromising the natural speech quality and intelligibility. Hence, for the CLP/CPT based TTS method, the subjective tests must be designed to focus on its ability to implement pitch and duration modifications without adding noitceable artifacts.

Synthesizing speech with increased emphasis is one method to highlight the pitch and duration modification capabilities of a TTS system. Though, the majority of existing TTS systems are designed to synthesize speech with "neutral" prosody, generating prosody with increased emphasis is an area that has recently gained attention [73]. The purpose of adding emphasis to the prosody is to give importance to certain words within an utterance and/or to improve the intelligibility in a harsh environment. This can be highly useful and effective in a spoken dialog application such as the CMU Communicator. For example, if the spoken dialog system gave information to a user that was not fully understood, the system can repeat the information with increased emphasis on the key words. A subjective listening test that compares the

quality of utterances synthesized with emphasized prosody to the original utterances allows the subject to focus on the prosody modifications while assessing the quality of the synthesized utterances. Additionally, the results of the test provide two kinds of information:

- The overall quality of the synthesis method when the prosody is modified.
- The capability of the synthesis method to generate speech with emphasized prosody.

A second consideration that is necessary for subjective testing is the listening environment. Because a large number of existing TTS applications are in mobile communications and for hands-free operation of vehicles, it is highly likely that the environment can have large amounts of varying background noise. For telecommunication services and telemarketing applications, the synthesized speech can undergo additional degradation due to transmission loss in the networks. For example, a listening test taken in a controlled, noise-free environment using good quality, closed headphones may have a different result than the same test taken in an acoustic environment with varying background noise. However, the results in the latter case could be a more useful measure of performance because it is representative of "real use" conditions. Therefore, it would be advantageous to conduct the subjective listening tests in conditions that are representative of a "real use" environment. The environment can be simulated, to some degree, by adding recorded background noise, such as a road noise from inside a moving vehicle.

When conducting a comparative subjective study, it is important to select an appropriate "reference" system. Since the CLP/CPT method is a linear prediction based TTS method, it would suggest that comparison to previous LP based TTS methods would be adequate. However, in recent times, limited-domain TTS

systems using unit-selection time-domain (TD-PSOLA) synthesis methods are becoming widely used because of their high quality. To evaluate the contribution of the CLP/CPT based TTS method to currently exisitng systems, it would be more appropriate to use a unit-selection synthesis system as the reference.

Based on the above considerations, two different subjective listening tests were developed to verify the advantages of applying the CLP/CPT representation to existing unit-selection based TTS systems. These tests are:

- Comparison of the synthesis quality of utterances synthesized by an existing unit-selection TTS system and by the CLP/CPT method, with matched prosodics.

- Comparison of utterances synthesized by an existing unit-selection TTS system and by the CLP/CPT method with over-emphasized prosodics.

Both tests were developed using the CMU Communicator travel reservations dialog system [14][75] and its limited-domain TTS database. The control for the tests were utterances synthesized by Communicator that are in the context of a trip (flight, hotel, rental car) reservation. The test utterances were synthesized by the same database, however the units were not necessarily the same and the phonemes had slightly different prosodics from the same phonemes of the control utterances. In order to simulate the environment of a real application (i.e. hands-free telecommunications), roadnoise recorded on the highway at normal highway speeds (55 to 70 mph) were added to the utterances.

6.1 Comparison to Unit Selection Synthesis

As detailed in section 5.3.1, the CLP/CPT model allows for limited prosody modifications with very little perceivable speech degradation. For current concatenative TTS systems based on unit-selection synthesis, this method provides the advantages of

achieving the same or similar quality TTS with a smaller database and/or improving the richness in prosody and quality of synthesis. When acheiving similar quality TTS, with this model the database itself need not be as prosodically rich because fewer instances of each unit need to be stored. On the other hand if it is not necessary to reduce the database, this method can improve the "naturalness" of the synthesized speech by providing for more refined prosodic movements to the best selected units from the synthesis database.

The amount of database reduction and improvement in quality that is acheivable is difficult to generalize as it varies based on two key factors: the extent of pitch and duration modifications that can be achieved without noticeable artifacts, and the contents of the database. The first is the prosody thresholds that limit the degree of pitch and time scale modifications, detailed in section 5.3.2, that are determined by subjective testing. The second, which is dependent on the TTS application, can have a significant impact on both the amount of database reduction and quality. For example, for limited domain TTS applications that strive for "natural quality" TTS, the database often has numerous instances of the same segments with very slight changes in prosody. With the CLP/CPT model, such a database can be reduced greatly and still achieve "natural quality". On the other hand databases designed for unrestricted TTS have fewer instances or just one instance of a given speech segment. In this case the database may not be reduced, but the synthesis quality can be improved since a larger number of prosody inflections are acheivable than exist in the database.

This test evaluates the quality of utterances synthesized using the CLP/CPT method proposed in this thesis by comparing the utterances to the same utterances synthesized by the CMU Communicator limited-domain unit-selection TTS system. The units for the CLP/CPT method, ranging from diphones to phrases, were randomly selected from the same synthesis database used by the CMU Communicator.

To generate the test utterances, they were equalized, prosodically modified to match the control utterances, concatenated, and synthesized. The goal of this test is to demonstrate that the utterances synthesized by this method are at least equal in quality to the utterances synthesized by the CMU Communicator system. If this is indeed the case, then the CLP/CPT method would be considered a contribution in the fact that applying this method to current unit-selection TTS systems would allow for a reduction of the number of instances of each unit within a database.

6.1.1 Test Method

The purpose of this test is to determine the quality of utterances synthesized by the CLP/CPT method with prosodic modifications. The utterances are compared to the same utterances produced by unit-selection synthesis with no prosodic modifications. The test utterances were created using the method given below:

- Obtain the control utterances from the CMU Communicator limited-domain TTS system and label phoneme boundaries. The phoneme boundaries will be used to generate target duration of the test utterances. The labeling was conducted by hand using the OGI CSLU Toolkit [55].

- Extract the target pitch and duration values for each of the phonemes in the control utterance. The target pitch contour is determined using a pitch detector and duration values are determined directly from the phoneme boundary labels.

- Create a synthesis speech database that is a subset of the CMU Communicator database, such that each of the control utterances can be synthesized and only one instance of each unit exists. Since, these units are not the same as the ones used to create the control utterances, the prosodics will naturally differ.

- Label the segments in the synthesis database with pitchmark locations (section 5.2.1 and fine tune segment and phoneme boundaries locations.

Table 9: Modified Comparison Category Rating scale used for comparing emphasized speech to unmodified speech.

Description	**Rating**
Strong Prefernce for the Control	-2
Weak Prefernce for the Control	-1
No Preference	0
Weak Prefernce for CLP/CPT Emphasis	1
Strong Prefernce for CLP/CPT Emphasis	2

- Normalize the spectral characteristics of the segments (section 5.1) and perform CLP/CPT analysis with fractional precision (section 5.2) on each of the units.
- Synthesize the test utterances by segment concatenation of the CLP/CPT analyzed database, while applying the target pitch and duration values.
- Add road noise to both the test and control utterances.

The subjective testing was conducted by having the subjects listen to a set of control and matching test utterances and selecting a preference. There were a total of 12 subjects and 6 control and test utterances resulting in a total of 72 responses. The subjects were allowed to listen to the utterances repeatedly as desired. For each set of utterances, the subjects could select whether they had a strong or weak preference for the control utterance, a strong or weak preference for the CLP/CPT synthesized test utterance, or no prefernce. This is a 5-point comparison scale based on the Comparison Category Rating (CCR) scale, which has a 7-point scale (-3 to +3). The modified scale is given in table 9. To prevent bias, the choices were presented to the subjects in random order.

6.1.2 Results and Analysis

The results of the subjective test demonstrated that the preferences of the subjects slightly favored the utterances synthesized by the CLP/CPT method with prosodic modifications. These results are shown in Figure 37, in which the "Control" refers to

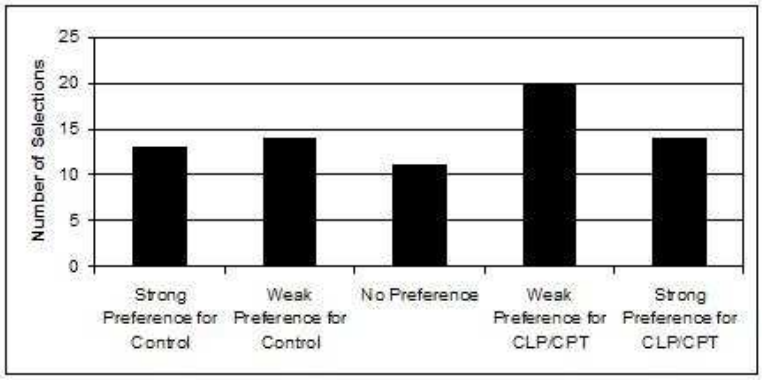

Figure 37: Results of subjective listening test showing the preference of utterances synthesized by the CMU Communicator and the CLP/CPT method with prosodic modifications with road noise.

the utterances synthesized by the CMU Communicator. Of the 72 total preference selections made by the subjects, 34 were made in favor of CLP/CPT based TTS with prosody modifications and 27 were made in favor of the CMU Communicator. The remaining 11 selections were "No Preference". Since the result indicates that the CLP/CPT based TTS method is at least equal in voice quality to the CMU Communicator, the result is favorable for this thesis. As stated above, the advantage of this method is a reduction in size of the synthesis database when applied to existing systems. Though it was expected that the distribution of preferences would be more even distribution, the slight favor for the CLP/CPT method can be explained by various factors including:

- Smoother prosodic inflections across unit boundaries due to the smoothing of target pitch and duration.
- More accurate labeling of segement boundaries due to hand correction.

In order to determine the statistical significance of this slight preference, a one way analysis of variance (ANOVA) was performed for the distribution of preferences. For the ANOVA, the preference groups were modified to 3 groups: preference for control, no preference, and preference for CLP/CPT. This was done by combining the "strong"

Table 10: One-way ANOVA on the distribution of preferences for limited-domain unit-selection TTS without prosody modifications and with prosody modifications using the CLP/CPT method.

Group	**Count**	**Sum**	**Mean**	**STD**
Preference for Control	6	27	4.50	2.59
No Preference	6	11	1.83	3.06
Preference for CLP/CPT	6	34	5.67	2.88
Source of Variation	**Sum of Squares**	**Degrees of Freedom**	**Mean Square**	**F Value**
Between Groups	46.33	2	23.17	2.856
Within Groups	121.7	15	8.111	
Total	168.0	17		

and "weak" preference groups into one group. The results of the ANOVA, reported in table 10, show a relatively low probability for assumming the null hypothesis ($P = 0.089$). For the $F - Value$ shown in table 10, the confidence interval was found to be above 0.91. This indicates that the preference shown by the data in Figure 37 has some statistical significance albeit not very strong. Once again, very high significance of the preference for the CLP/CPT method is not extremely important for this test since the goal was to demonstrate that the preferences were at least equal.

These results suggest that at the least there is little difference in subjective quality between the two synthesis methods when compared in a practical environment (road noise). The prosodic modifications applied to the units did not affect their quality in an adverse manner. Therefore, in the CMU Communicator and similar limited-domain unit-selection TTS systems, the redundancy in the database can be reduced significantly.

6.2 TTS with Emphasis

Many real world applications that use TTS interface with the user over a telecommunications link. Often the user is using a wireless device in a vehicle or other environments are prone to significant background noise. Venkatagiri conducted a

study for intelligility of TTS systems in noisy environments using the Festival TTS system, IBM Via Voice, and AT&T Next-Gen systems [101]. This study revealed that the better performing systems were, on the average, 22% less intelligible than human voice under comparable noisy conditions. For such applications, a system that is capable of adding emphasis to key words/syllables in the utterances can be desirable for improving intelligibility. For example, if the user does not obtain the necessary information from the TTS utterance, she may request to repeat it. In this case, repeating the utterance with emphasis added to certain syllables could be more desirable than just repeating the utterance with the original prosodics. For standard unit-selection systems, this would require an even richer database that contains emphasized units. The CLP/CPT model allows for adding emphasis easily to any existing TTS database by making slight prosodic variations to phonemes of syllables during synthesis. A test was designed to measure the ability of the CLP/CPT model to apply emphasis in a real world application.

6.2.1 Test Method

The subjective test for TTS with emphasis was done by creating a simulated conversation between the CMU Communicator system and a user in a harsh environment that would necessitate the need for emphasis. The user is attempting to make flight and car rental reservations for a trip and having severe difficulty in undertanding the TTS system. The user's inability to understand the system was exaggerated for this test so that emphasis would seem necessary in many of the words. Once again, roadnoise at highway speeds was added to the reference and test utterances to simulate a real use case (making reservations while driving a car). The roadnoise was a significant factor in this test to create the necessity for emphasis of important words.

For this test a total of 19 test utterances were synthesized. Once again the control utterances were obtained from the CMU Communicator system and labeled for

phoneme boundary locations. Additioanlly the pitch epoch locations and voicing modes were labeled. The test utterances were created by modifying the prosodics of the control utterances to create over-emphasized speech in the following manner:

- Create the target pitch and duration values for the phonemes in the words to be emphasized by modifying the control pitch and duration values. This was conducted manually in this test by simply scaling the original values to realize emphasis.

- Perform CLP/CPT analysis with fractional resolution (section 5.2), while applying the LSF constraints, on all of the control utterances.

- Synthesize the test utterances by realizing the emphasized target pitch (inverse CPT) and duration values and performing CLP synthesis.

- Add road noise to both the test and control utterances.

The target pitch and duration values for the emphasized words in the test utterances are calculated by modifying the control utterances values by 10% to 15% for certain syllables. Note that in some cases, adjacent syllables may have to be deemphasized to some degree.

The subjective test was designed such that the subject is a third person observer of a dialog between the user and the TTS trip reservation system. The following details the manner in which the test was conducted:

- Subjects first listen to a response given by the CMU Communicator to an initial request by the user.

- Subjects then read a request made by the user to repeat a certain part of the response that was not understood.

- Subjects can then select between the reference response (same prosody as original) and the test response (emphasized prosodics).

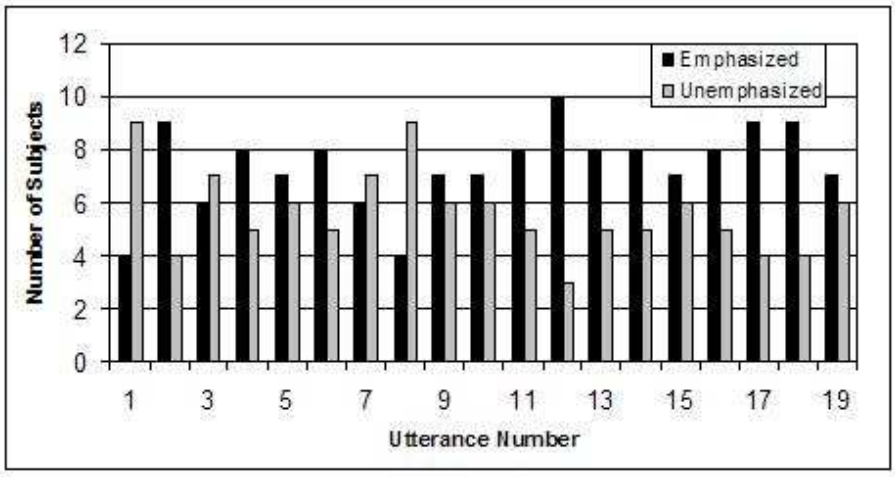

Figure 38: Distribution of preference for the subjective listening test that compared unemphasized utterances to emphasized utterances synthesized by the CLP/CPT method.

For the success of this test, the subject should select the response that would seem appropriate for the user, and that would be agreeable in terms of quality. Even though the intial responses made by the TTS system may be intelligible, the user of the system is having trouble understanding many words. Often the intial response is adequate and an ordinary user may select that response over an emphasized response. For this reason, the added roadnoise is important and gives perspective to the subjects to aid in selecting the appropriate response.

6.2.2 Results and Analysis

The results of this test demonstrated that, on the whole, the subjects preferred the emphasized responses synthesized by the CLP/CPT model over the unemphasized responses. All together, out of the 228 total selections by all of the subjects, they preferred the emphasized responses 57% of the time and the unemphasized responses 43% of the time. To get better insight on these results, Figure 38 shows the distribution of the preferences for each set of potential synthesized responses. A closer look at the results, analyzed with respect ot each utterance pair, reveals that the

unemphasized utterances were preferred by a majority of subjects for only 4 of the 19 pairs.

Though for the majority of the responses the emphasized utterances were preferred by a majority of the subjects, they were not always the clear winner. Preference of the unemphasized utterance can be explained by the following reasons:

- The subject perceived the prosodics of the original utterance to have sufficient emphasis on the key words to sound natural. In this case added emphasis would sound unnatural to the subject.

- The subject felt that the addition of emphasis to syllables was not necessary to increase intelligibility. Addition of more emphasis would sound undesirable to the subject.

- Applying emaphasis to the prosody using the CLP/CPT method caused undesirable artifacts and/or degraded the overal voice quality.

The first two reasons are actually a negative bias for this test since the purpose of this test is only to determine the ability of the CLP/CPT model to apply emphasis without noitceable degradation to the quality of the synthesized speech. Since, withstanding these biases, the subjects still preferred a majority of the emphasized utterances, the results clearly verify the advantages of the CLP/CPT method for applying emphasis to an existing unit-selection based synthesis system. Again a one-way ANOVA was conducted on the data for this test to determine the statistical significance of the preference for the CLP/CPT method. Since this test only has two distributions (A-B test), this is similar to the student's t-test. The results of the ANOVA for the emphasis realization test is given in table 11.

The results indicate a very high statistical significance ($P = 0.0016$) for the preference of the emphasized speech using the CLP/CPT method for prosody modifications and synthesis. The value for the F-distribution of this data is 11.613, which is much

Table 11: One-way ANOVA on the distribution of preferences for unit-selection TTS without emphasis and with emphasis using the CLP/CPT method.

Group	**Count**	**Sum**	**Mean**	**STD**
Control (no emphasis)	19	107	5.63	1.57
CLP/CPT with emphasis	19	140	7.37	1.57
Source of Variation	**Sum of Squares**	**Degrees of Freedom**	**Mean Square**	**F Value**
Between Groups	28.66	1	28.658	11.613
Within Groups	88.8421	36	2.468	
Total	117.5	37		

larger than the F table value for $\alpha = 0.01$: $F_{0.99}(1, 36) = 5.2$. Therefore the confidence interval of the preference is greater than 0.995.

CHAPTER VII

CONCLUSIONS

The main goal of this research is to develop methods to improve currently existing unit-selection based concatenative TTS system. Recent advances in text-to-speech has lead to increased usage of systems in the main stream for limited-domain applications, such as travel information, travel reservations, telemarketing, etc. Though current systems are capable of high quality TTS, they are limited by memory constraints and computing power. For smaller footprint applications, the systems need to be scaled down, compromising the quality. This thesis addresses these limitations by introducing the application of a robust linear prediction model for synthesis and a pitch transformation method that allow for prosodic movements without compromising speech quality.

The thesis begins with a background on the entire TTS process, identifying the NLP (front-end) and DSP (back-end) stages. The focus of this research is only to improve the DSP stage (synthesis and prosody realization) of TTS. Previous and existing methods for TTS synthesis are presented to form basis for the research. Various models implemented in the past are discussed with their advantages and disadvantages. Of these models, linear prediction provides the advantages of the ability to modify prosodics and lower complexity. On the other hand the speech quality consists of noticeable artifacts and is generally lower than that of the highly complex hybrid/sinusoidal model. This research re-introduces circular linear prediction as a robust speech model for representing a TTS speech database and synthesizing speech. This method is a windowless approach to LP modeling that provides the accuracy of the covariance method for analysis with the efficiency of the autocorrelation method.

Specifically, for TTS this method reduces the potential for distortion caused by windowing combined with interpolation of segmental boundaries and prosody modifications. Objective analysis was performed on various synthesis methods for this model, using synthetic known signals. The results of the analysis demonstrate significantly greater accuracy of the re-synthesized signals using this model compared to traditional autocorrelation LP. However, the improved accuracy relies on the analysis frames to be exactly periodic.

The constant pitch transform is presented here as an effective method for realization of pitch inflections for improving prosodics. It also serves the purpose of transforming fractional pitch periods to integer length. Additionally, to further improve the performance for prosody modifications, a variation to the CLP model is presented in which the LP coefficients are modified by placing dynamic thresholds on the LSF tracks that are generated from the LP coefficients. Combined with the constant pitch transform, the CLP/CPT model allows for creating a uniform pitch database of speech that is well suited for limited prosodic variations and concatenation resulting in minimal artifacts.

This thesis introduces the implementation of the CLP/CPT representation to TTS synthesis. Notable advantages and key issues for this implementation, presented in the thesis, are listed below:

- Unit concatenation can be acheived efficiently by placing units end-to-end with little to no smoothing required.

- Time-scaling is acheived by deriving a map of pitch periods to be repeated or deleted within each phoneme, based on a scale factor. Pitch periods containing voicing transitions are not scaled.

- The pitch periods are repeated or deleted with no smoothing required because of the accuracy in pitch period estimation.

- Pitch-scaling is implemented during synthesis by the inverse CPT, using the target pitch pitch periods for each unit.

- An initial database equalization step is introduced before the CLP analysis of the unit database to match the acoustical spectral shape of all recorded units.

As a related study, this thesis investigates the problem of optimal unit size for TTS systems. A method for defining new variable-size units that would lower the amount of spectral variation at concatenation junctures is introduced. The junctural phonemes (phonemes at the endpoints) of the variable-sized units must be a member of a set of phonemes that have the lowest spectral variation within the database. Phonemes with higher spectral variations would always exist in the middle of units.

Subjective tests were developed using the available limited-domain TTS CMU Communicator database to:

- Test the quality of prosodically modified utterances synthesized by the CLP/CPT model.

- Test the ability of the model to produce emphasized speech.

The tests demonstrated that prosody modifications could be performed on the units to a limited degree without causing noticeable degradation in speech quality. Additionally, in noisy environments (road noise, wireless communications, etc.), the CLP/CPT method can emphasize the speech to improve intelligibility. Statistical significance tests were conducted on the data gathered from the two tests to substantiate the results.

7.1 Future Work

The goal of current unit-selection based TTS systems is to synthesize very high quality, artifact-free speech by using a large, prosodically rich speech database. Historically, applying prosodic variations with a speech model resulted in artifacts and

unnatural speech quality. This research demonstrates that by applying the CLP/CPT representation to existing systems, a fewer number of prosodically varied instances of each unit are necessary to achieve similar synthesis quality. Alternatively, this method can be utilized to further improve the prosodics and add emphasis. However, there are a number of ways in which the work presented in this thesis can be improved upon.

The labeling of speech during the synthesis database preparation stage, before CLP analysis is performed, required hand corrections. This was due to the accuracy necessary for the CLP/CPT based TTS method to be effective. Since TTS database preparation is conducted "offline", without major time constraints, this can be considered adequate. However, if the database is very large (e.g. unilimited vocabulary applications), this can become expensive. Greater accuracy in automated pitch estimation can resolve this problem. The other labels (i.e. phoneme boundaries, voiced/unvoiced speech markers) need not have the same accuracy as they can be tied to the nearest pitch period boundaries.

In the area of pitch modifications dynamic thresholds applied to the LSP tracks of the CLP coefficients can be investigated further. Though, the thresholds implemented in this research were derived through subjective analysis of pitch-modified speech, it cannot be said that these were the optimal thresholds to reduce artifacts during pitch modifications. Implementing different thresholds for different voiced phonemes was not investigated. Though this investigation can be highly time intensive, it can result in allowing for a greater degree of pitch modifications without audible artifacts. Alternatively, developing an objective method for acheiving this, though a difficult task in itself, can solve this problem in a highly efficient manner.

Determining the exact extent of artifact-free prosody modifications for every voiced phoneme can be investigated to improve the performance of the CLP/CPT based TTS method. Though, different pitch-scale thresholds were used for different

voicing modes, a unique threshold for every phoneme is still open for research. The thresholds for pitch and duration scale factors presented in this thesis are based on informal listening tests. A formal test conducted to determine the exact thresholds for every phoneme can be useful for characterizing the full capabilities of this synthesis method.

Finally, integration of the CLP/CPT based TTS method with a high quality, low bitrate vocoder can further reduce the required footprint of an existing TTS system and provide a complete solution for mobile communication applications. The main limitation for this is that it would require the vocoder to be pitch-synchronous. This is an area that is still under research.

APPENDIX A

DETAILS OF THE CLP/CPT BASED TTS SUBJECTIVE TESTS

The complete details of the setup and execution of the subjective tests for evaluating the synthesis quality and capabilities of the CLP/CPT based TTS method are presented in this appendix. The first section discusses the comparison of this method to the limited-domain unit-selection TTS system in the CMU Communicator. The second section details the comparison of TTS with emphasized prosody using the CLP/CPT method to TTS with "normal" prosody.

A.1 Comparison to Unit-Selection TTS

The comparison of the CLP/CPT based TTS to the CMU Communicator was conducted by selecting a few utterances synthesized by the CMU Communicator system and using the CLP/CPT based method to synthesize the same utterances. The concatenation units for the CLP/CPT based synthesis were selected randomly from the same segment database used by the CMU Communicator. Note that all of the signal processing algorithms (i.e. the CLP analysis, CLP synthesis, CPT, inverse CPT, database equaliztion, etc.) were implemented by writing MATLAB scripts. The details of test preparation and execution are given below.

A.1.1 Test Setup

For this test, 6 utterances were selected from a set of synthesized utterances available on the CMU Communicator website as demonstration examples of the TTS system. These utterances are the "control" utterances for this test. The text of the utterances is given below:

Table 12: Distribution of different unit types for synthesis of test utterances.

Unit Type	**Count**
Large Phrases (more than 2 words)	8
Two Word Phrases	10
Single Words	6
Diphones	6

1- "Your first flight is a US Airways flight 4072."

2- "I've made a request for a car with Avis in Charleston."

3- "Leaving Charleston at 11:05 AM, on Saturday March 18th, arriving in Pittsburgh at 1:13 PM."

4- "Do you want a summary of your trip?"

5- "Leaving Pittsburgh at 2:10 PM on Wednesday, March 15th, arriving in Charleston at 4:10 PM."

6- "Then the next flight is a US Airways flight 4120."

The next step was to determine the target pitch and duration tracks for each utterance. The utterance wave files were passed through a pitch detector to detemine the target pitch for each phoneme. For the target durations, the phoneme boundaries were determined using tools in the Festival TTS system and hand corrected using the Speech Viewer in the OGI Speech Toolkit.

The units for the CLP/CPT synthesis database were chosen randomly from the CMU Communicator unit database. There were a total of 30 units for synthesizing the utterances, consisting of phrases, words, and diphones. The distribution of the units is given in table 12. Note that the words and phrases in this definition include adjacent phonemes cut at their midpoint.

The selected units were equalized to match the acoustic spectral characteristics and pitch period boundaries were estimated and hand corrected using the OGI Speech

Viewer. Additionally, voiced/unvoiced speech labels and phoneme boundary labels for each unit were placed using the Speech Viewer. The units were analyzed, using the CLP analysis method with fractional resolution (section 4.1.1), and transformed to a constant integer pitch by the CPT. For the 16 kHz sampled units, the analysis order was set to 16 as indicated in section 4.1.2.1. The LP coefficients were modified by converting them to LSP parameters, applying the dynamic thresholds, and converting back to the LP coefficients. The result of the analysis of each unit was:

- a set of modified LP coefficients for every pitch period,
- a residual signal of constant pitch,
- the vector of original pitch values of every pitch period,
- the phoneme boundary locations,
- the voiced/unvoiced labels.

The 6 test utterances were then synthesized by concatenating the units and matching the prosody of "target" reference utterances. The prosody modification and synthesis techniques are well detailed in sections 4.1.2 and 5.3. Finally, wave files containing varying road and traffic noise, recorded inside a vehicle moving at highway speeds, were additively mixed with both the test and control utterance wave files.

A.1.2 Test Execution

The subjective testing was executed using PowerPoint as the interactive interface for the subjects. Each set of test and reference wave files were placed on a separate slide with check boxes for the five choices indicated in table 9. The subjects were given the ability to listen to the utterances as many times as necessary and mark one of the check boxes indicating their preference. Note that the two wave files on each slide were always marked "A" and "B", and the order of the test and reference waves was

random for each set. In other words, in one set the reference file would be choice "A", while in another set it would be choice "B". The results for each subject were saved as individual PowerPoint files and tabulated using Excel.

A.2 TTS with Emphasis

The subjective test for TTS with emphasized prosody was conducted by comparing a set of CMU Communicator synthesized utterances with the same utterances re-synthesized with emphasized prosody using the CLP/CPT method. A mock conversation between the CMU Communicator and a user making a trip reservation was designed to artificially create a necessity for emphasis in the synthesized speech. The test subjects were observers to this conversation. The subjects would listen to the prompts given to the user by the Communicator and read the questions asked by the user. The subject would, then, select between two potential responses: unemaphasized and emphasized. The details of the setup and execution of this test are given in this section of the appendix.

A.2.1 Test Setup

The generation of the test utterances for this test is very similar to the first test in many ways. However, since the test utterances are generated by resynthesizing the control utterances, the unit database prepartion during the analysis phase, and unit concatenation during the synthesis phase are omitted. The labeling and analysis algorithms were performed on the entire set of control utterances. The test utterances were, then, synthesized with the target prosody.

What was unique in the setup of this test was that the target pitch and duration values were emphasized versions of the original. The pitch and duration of certain words and/or syllables were increased by 10% to 15% to emphasize key words. Also, the pitch and duration values of adjacent syllables were slightly decreased (deemphasized) to acheive noticeable emphasis.

To design the mock conversation, some creativity was necessary to necessitate emphasis in the responses. In the conversation that was designed, the user of the Communicator system was artificially made to have serious difficulty understanding the responses the first time and inevitably requested the system to repeat every response. As an example, a small transcript of the conversation is given below:

Communicator: "The next flight is a US Airways flight 4120, leaving Charleston at 11:05 AM, on Saturday March 18th."

User: "Thats US Airways flight 4170, right?"

- Subject selects from an emphasized and unemphasized version of the following responses given by Communicator: "US Airways flight 41**20**."

  User: "Got it. Flight 4120 leaves Charleston at 11:09 AM ?"

- Subject selects from an emphasized and unemphasized version of the following responses given by Communicator: "At eleven oh **five** AM."

  User: "I need to leave on Saturday. Did you say the flight departs on Sunday, March 19th?"

- Subject selects from an emphasized and unemphasized version of the following responses given by Communicator: "On **Sa**turday, March **eigh**teenth."

This transcript consists of only 3 of the 19 sets of utterances presented to the subjects. Note that the bold letters indicate the syllables that were emphasized using the CLP/CPT method.

A.2.2 Test Execution

This test was also implemented using PowerPoint as the interactive interface for the subjects. This time however, the subjects were forced to choose a preference for

one of the two responses in each test set. Again, the emphasized and unemphasized responses choices were presented in random order.

REFERENCES

[1] Abrantes, A. and Marques, J., "Hybrid harmonic coding of speech," in *Eusipco 92*, pp. 487–491, 1992.

[2] Allen, J., "An overview of text to speech systems," in *Advances in Speech and Signal Processing* (Furui, S. and Sondhi, M., eds.), pp. 741–790, London: Dekker, 1992.

[3] Allen, J., Hunnicut, S., and Klatt, D., *From Text to Speech, The MITalk System*. Cambridge University Press, 1987.

[4] Anderson, M. and Pierrehumbert, J., "Synthesis by rule of English intonation patterns," in *International Conference on Acoustics Speech and Signal Processing*, pp. 2.8.1–2.8.4, IEEE, 1984.

[5] Ansari, R., "Inverse filter approach to pitch modification: Application to concatenative synthesis of female speech," in *International Conference on Acoustics Speech and Signal Processing*, pp. 1623–1626, IEEE, 1997.

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

[7] Barnwell, T., "Circular correlation and the LPC," in *International Conference on Communications*, pp. 31–5–31–10, IEEE, 1976.

[8] Barnwell, T., "Windowless techniques for LPC analysis," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 421–427, August 1980.

[9] Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A., "The AT&T Next-Gen TTS System," in *137th meeting of the Acoustical Society of America*, 1999.

[10] Bigorne, D., Boefard, O., Cherbonnel, B., Emerard, F., Larreur, D., Saintmillon, J. L., Metayer, I., Sorin, C., and White, S., "Multilingual PSOLA text-to-speech system," in *International Conference on Acoustics Speech and Signal Processing*, pp. 187–190, IEEE, 1993.

[11] Black, A. and Campbell, N., "Optimising selection of units from speech databases for concatenative synthesis," in *Proceedings of Eurospeech*, pp. 581–584, IEEE, 1995.

[12] Black, A. and Hunt, A., "Generating F0 contours from ToBI labels using linear regression," in *International Conference on Spoken Language Processing*, pp. 1385–1388, IEEE, 1996.

[13] Black, A. and Hunt, A., "Unit selection in a concatenative speech synthesis system using a large speech database," in *International Conference on Acoustics Speech and Signal Processing*, pp. 373–376, IEEE, 1996.

[14] Black, A. and Lenzo, K., "Limited domain synthesis," in *International Conference on Acoustics Speech and Signal Processing*, IEEE, 2000.

[15] Black, A. and Taylor, P., "Automatically clustering similar units for unit selection in speech synthesis," in *Proceedings of Eurospeech*, pp. 601–604, 1997.

[16] Black, A. and Taylor, P., "Festival Speech Synthesis System: system documentation (1.1.1)," University of Edinburgh: Human Communication Research Centre, 1997. Technical Report HCRC/TR-83.

[17] Breen, A., "Issues in the development of the next generation of concatenative. speech synthesis systems," in *IEE Seminar on the State of the art in speech synthesis*, pp. 1–4, 2000.

[18] Campbell, N., "CHATR: A high-definition speech re-sequencing system," in *Proceedings of ASA/ASJ Joint Meeting*, pp. 1223–1228, 1996.

[19] Campbell, N. and Isard, S., "Segment durations in a syllable frame," *Journal of Phonetics*, vol. 19, no. 1, pp. 37–47, 1991.

[20] Charpentier, F. and Stella, M., "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *International Conference on Acoustics Speech and Signal Processing*, pp. 38.5.1–38.5.4, IEEE, 1986.

[21] Courbon, J. and Emerard, F., "SPARTE: A text-to-speech machine using synthesis by diphones," in *International Conference on Acoustics Speech and Signal Processing*, pp. 1597–1600, IEEE, 1982.

[22] Crosmer, J. and Barnwell, T., "A low bitrate segment vocoder based on line spectrum pairs," in *International Conference on Acoustics Speech and Signal Processing*, pp. 240–243, IEEE, 1985.

[23] Crystal, T. and House, A., "Segmental durations in connected-speech signals: Current results," *Journal of the Acoustical Society of America*, vol. 83, pp. 1553–1573, 1988.

[24] Crystal, T. and House, A., "Segmental durations in connected-speech signals: Syllabic stress," *Journal of the Acoustical Society of America*, vol. 83, pp. 1574–1585, 1988.

[25] de Campos, G. L., Gouvea, E., Bunnell, H., and Idsardi, W., "Speech synthesis using the CELP algorithm," in *International Conference on Spoken Language Processing*, pp. 1417–1420, ASA, 1996.

[26] Ding, W., Fujisawa, K., and Campbell, N., "Improving speech synthesis of CHATR using a perceptual discontinuity function and constratints of prosodic modification," in *International Conference on Spoken Language Processing*, 1998.

[27] Donovan, R., Franz, M., and Roukos, S., "Phrase splicing and variable substitution using the IBM Trainable Speech Synthesis System," in *International Conference on Acoustics Speech and Signal Processing*, pp. 373–376, IEEE, 1999.

[28] Donovan, R. and Woodland, P., "Improvements in an HMM-based speech synthesiser," in *Eurospeech95*, pp. 573–576, 1995.

[29] Donovan, R., "Segment pre-selection in decision-tree based speech synthesis systems," in *International Conference on Acoustics Speech and Signal Processing*, pp. 937–940, IEEE, 2000.

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

[31] Dutoit, T. and Gosselin, B., "On the use of a hybrid harmonic/stochastic model for TTS synthesis-by-concatenation," *Speech Communication*, vol. 19, pp. 119–143, 1996.

[32] Dutoit, T. and Leich, H., "MBR-PSOLA: Text-to-speech synthesis based on an MBE resynthesis of the segments database," *Speech Communication*, no. 13, pp. 435–440, 1993.

[33] Dutoit, T., *An Introduction to Text-to-Speech Sythesis*. Kluwer Academic Publishers, 1997.

[34] Ertan, A. E., "Pitch-synchronous processing of speech," Georigia Institute of Technology: School of Electrical and Computer Engineering, 2004. Ph.D. Thesis.

[35] Fant, G., *Acoustic Theory of Speech Production*. The Hague, Netherlands: Mouton, 1960.

[36] Fujisaki, H. and Ljungqvist, M., "Analysis of voice fundamental frequency contours for declarative sentences of japanese," *Journal of the Acoustical Society of Japan*, vol. 5, no. 4, pp. 233–242, 1984.

[37] Fujisaki, H. and Ljungqvist, M., "Generating intonation for swedish text-to-speech conversion using a quantitative model for the F0 contour," in *Proceedings of Eurospeech*, pp. 873–876, 1993.

[38] Hamon, C., Moulines, E., and Charpentier, F., "A diphone system based on time domain prosodic modifications of speech," in *International Conference on Acoustics Speech and Signal Processing*, pp. 238–241, IEEE, 1989.

[39] Hart, M., "Gutenberg: The history and philosophy of project," *Project Gutenberg*, 1992. http://www.gutenberg.org last accessed in September 2006.

[40] Hayes, M., *Statistical Digital Signal Processing and Modeling*. New York: John Wiley and Sons, 1987.

[41] Hermansky, H., "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, April 1990.

[42] Hermes, D. and Rump, H., "Perception of prominence in speech intonation induced by rising and falling pitch movements," *Journal of the Acoustical Society of America*, vol. 96, no. 1, pp. 83–92, 1994.

[43] Huang, X., Acero, A., Adcock, J., Hon, H., Goldsmith, J., Liu, J., and Plumpe, M., "Whistler: A trainable text-to-speech system," in *International Conference on Spoken Language Processing*, pp. 2387–2390, 1996.

[44] Itakura, F., "Line spectrum representation of linear predictive coefficients," *Journal of the Acoustical Society of America*, vol. 57, no. 1, p. S35, 1975.

[45] Iwahashi, N. and Sagisaka, Y., "Statistical modelling of speech segment duration by constrained tree regression," *IEICE Transactions on Information and Systems*, vol. E83-D, pp. 1550–1559, July 2000.

[46] Kishore, S. and Black, A., "Unit size in unit selection speech synthesis," in *Proceedings of Eurospeech2003*, 2003.

[47] Klatt, D. H., "Review of text to speech conversion for English," *Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[48] Kleijn, W., "Encoding speech using prototype waveforms," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 386–399, 1993.

[49] Kleijn, W., Backstrom, T., and Alku, P., "On line spectral frequencies," *IEEE Signal Processing Letters*, vol. 10, pp. 75–77, March 2003.

[50] Kominek, J., Bennet, C., Langner, B., and Toth, A., "The Blizzard Challenge 2005 CMU Entry a method for improving speech synthesis systems," in *Interspeech05*, 2005.

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

[52] Liberman, M. and Church, K., "Text ananlysis and word pronunciation in text-to-speech synthesis," in *Advances in Speech and Signal Processing* (S. Furui, M. S., ed.), pp. 791–830, London: Dekker, 1992.

[53] Ljolje, A., Hirschberg, J., and van Santen, J., "Automatic segmentation and labeling of speech," in *Proceedings of the ESCA/IEEE Workshop on Speech Synthesis*, pp. 93–96, IEEE, 1994.

[54] Macchi, M., Altom, M., Kahn, D., Singhal, S., and Spiegel, M., "Intelligibility as a function of speech coding method for template-based speech synthesis," in *Proceedings of Eurospeech*, pp. 893–896, 1993.

[55] Macon, M., Cronk, A., Wouters, J., and Kain, A., "OGIresLPC: Diphone synthesiser using residual-excited linear prediction," Oregon Graduate Institute of Science and Technology: Department of Computer Science, 1997. Technical Report CSE-97-007.

[56] Macon, M. and Clements, M., "Speech concatenation and synthesis uing an overlap-add sinusoidal model," in *International Conference on Acoustics Speech and Signal Processing*, pp. 361–364, IEEE, 1996.

[57] Macon, M., Cronk, A., and Wouters, J., "Generalization and discrimination in tree-structured unit selection," in *International Conference on Spoken Language Processing*, 1998.

[58] McAulay, R. and Quatieri, T., "Speech analysis/synthesis based on a sinusoidal representation of speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1961.

[59] McAulay, R. and Quatieri, T., "Magnitude only reconstruction using a sinusoidal speech model," in *International Conference on Acoustics Speech and Signal Processing*, pp. 27.6.1–27.6.4, IEEE, 1984.

[60] McAulay, R. and Quatieri, T., "Shape invariant time-scale and pitch modification of speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 40, no. 3, pp. 497–510, 1992.

[61] McCree, A. and Barnwell, T., "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 242–250, July 1995.

[62] Moulines, E. and Charpentier, F., "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, vol. 9, no. 5, pp. 453–467, 1990.

[63] Nakajima, S., "Automatic synthesis unit generation for english speech synthesis based on context oriented clustering," *Speech Communication*, vol. 7, no. 14, pp. 313–324, 1994.

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

[65] O'Shaughnessy, D., "Specifying intonation in a text-to-speech system using only a small dictionary," in *International Conference on Acoustics Speech and Signal Processing*, pp. 34.5.1–34.5.4, IEEE, 1987.

[66] Ostendorf, M., Price, P., and Shattuck-Hufnagel, S., "The Boston University Radio News Corpus," Boston University: Electrical Computer and Systems Engineering Department, 1995. Technical Report ECS-95-001.

[67] Ostendorf, M. and Veilleux, N., "A heirarchical stochastic model for automatic prediction of prosodic boundary location," *Computational Linguistics*, vol. 20, no. 1, pp. 27–55, 1994.

[68] Page, J. and Breen, A., "The Laureate text-to-speech system - architecture and applications," in *Speech Technology for Telecommunications*, Chapman and Hall, 1998.

[69] Picone, J., "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, pp. 1215–1247, 1993.

[70] Pierrehumbert, J., "Synthesizing intonation," *Journal of the Acoustical Society of America*, vol. 70, no. 4, pp. 985–995, 1981.

[71] Portele, T., Steffan, B., Preuss, R., Sendlmeier, W. F., and Hess, W., "HADIFIX - a speech synthesis system for german," in *International Conference on Spoken Language Processing*, pp. 1227–1230, 1992.

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

[73] Raux, A. and Black, A., "A unit selection approach to f0 modeling and its application to emphasis," in *ASRU*, IEEE, 2003.

[74] Rodet, X. and Depalle, P., "Synthesis by rule: LPC diphones and calculation of formant trajectories," in *International Conference on Acoustics Speech and Signal Processing*, pp. 736–739, IEEE, 1985.

[75] Rudnicky, A., Bennet, C., Black, A., Chotomongcol, A., Lenzo, K., Oh, A., and Singh, R., "Task and domain specific modelling in the Carnegie Mellon Communicator system.," in *International Conference on Spoken Language Processing*, 2000.

[76] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W., and Oh, A., "Creating natural dialogs in the carnegie mellon communicator system," in *Proceedings of Eurospeech*, vol. 4, pp. 1531–1534, 1999.

[77] Sagisaka, Y., Kaiki, N., Iwahashi, N., and Mimura, K., "ATR – ν-TALK speech synthesis system," in *International Conference on Spoken Language Processing*, pp. 483–486, 1992.

[78] Sagisaka, Y. and Sato, H., "Composite phoneme units for the synthesis of speech," *Speech Communication*, vol. 5, no. 2, pp. 217–223, 1986.

[79] Sanderman, A. and Collier, R., "Prosodic rules for the implementation of phrase boundaries in synthetic speech," *Journal of the Acoustical Society of America*, vol. 100, pp. 3390–33397, November 1996.

[80] Sato, H., "Speech synthesis for text-to-speech systems," in *Advances in Speech and Signal Processing* (Furui, S. and Sondhi, M., eds.), pp. 833–850, London: Dekker, 1992.

[81] Schroeter, J., "The fundamentals of text-to-speech synthesis," *VoiceXML Forum*, vol. 1, March 2001. http://www.voicexml.org last accessed on November 2006.

[82] Shi, Y., Chang, E., Peng, H., and Chu, M., "Power spectral density based channel equalization of large speech database for concatenative TTS system," in *International Conference on Spoken Language Processing*, pp. 2369–2372, 2002.

[83] Shukla, S., "Methods for speech synthesis and the generation of prosodic features," Georgia Institue of Technology: Center for Signal and Image Processing, 1998. Qualifying Examination Report.

[84] Shukla, S. and Barnwell, T., "Implementation of high quality text-to-speech synthesis for limited domain applications," in *The 138th meeting of the Acoustical Society of America*, ASA, 2000.

[85] Shukla, S., Ertan, A., and Barnwell, T., "Circular LPC analysis and Constant Pitch Transform for accurate speech analysis and high quality speech synthesis," in *International Conference on Acoustics Speech and Signal Processing*, pp. 269–272, IEEE, 2002.

[86] Smits, R. and Yegnanarayana, B., "Determination of instants of significant excitation in speech using group delay function," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 325–333, 1995.

[87] Sproat, R., Hirshberg, J., and Yarowsky, D., "A corpus-based synthesizer," in *Internation Conference for Spoken Language Processing*, pp. 563–566, 1992.

[88] Stylianou, Y., "Concatenative speech synthesis using a harmonic plus noise model," in *Third ESCA Speech Synthesis Workshop*, 1998.

[89] Stylianou, Y., "Assessment and correction of voice quality variabilities in large speech databases for concatenative speech synthesis," in *International Conference on Acoustics Speech and Signal Processing*, IEEE, 1999.

[90] Stylianou, Y., "Applying the harmonic plus noise model in concatenative speech synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 1, pp. 21–29, 2001.

[91] Stylianou, Y. and Syrdal, A., "Perceptual and objective detection of discontinuities in concatenative speech synthesis," in *International Conference on Acoustics Speech and Signal Processing*, 2001.

[92] Syrdal, A., Wightman, C., Conkie, A., Stylianou, Y., Beutnagel, M., Schroeter, J., Strom, V., Lee, K., and Makashay, M., "Corpus-based techniques in the AT&T NextGen synthesis system," in *International Conference on Spoken Language Processing*, Invited Paper, 2000.

[93] Taylor, P. and Black, A., "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, vol. 12, pp. 99–117, April 1998.

[94] Terken, J., "Fundamental frequency and perceived prominence of accented syllables II. non final accents," *Journal of the Acoustical Society of America*, vol. 95, pp. 3662–3665, 1994.

[95] Terken, J. and Collier, R., "The generation of prosodic structure and intonation in speech synthesis," in *Speech Coding and Synthesis* (Kleijn, W. and Paliwal, K., eds.), pp. 635–662, New York: Elsevier, 1995.

[96] t'Hart, J., "F0 stylization in speech: Straight lines versus parabolas," *Journal of the Acoustical Society of America*, vol. 90, no. 6, pp. 3368–3370, 1991.

[97] Tokuda, K., Kobayashi, T., Masuko, T., and Imai, S., "Mel-generalized cepstral analysis —a unified approach to speech spectral estimation," in *International Conference on Spoken Language Processing*, pp. 1043–1046, 1994.

[98] van Santen, J. and Mobius, B., "Modeling pitch accent curves," in *Intonation: Theory, Models and Applications - Proceedings of an ESCA Workshop*, pp. 321–324, 1997.

[99] van Santen, J., "Computation of timing in text-to-speech synthesis," in *Speech Coding and Synthesis* (W.B. Kleijn, K. P., ed.), pp. 663–684, New York: Elsevier, 1995.

[100] Varga, A. and Fallside, F., "A technique for using multipulse linear predictive speech synthesis in text-to-speech type systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 4, pp. 586–587, 1987.

[101] Venkatagiri, H., "Segmental intelligibility of four currently used text-to-speech methods," *Journal of the Acoustical Society of America*, vol. 113, pp. 2095–2104, April 2003.

[102] Vepa, J., King, S., and Taylor, P., "Objective distance. measures for spectral discontinuities in concatenative. speech synthesis," in *International Conference on Spoken Language Processing*, 2002.

[103] Vishwanathan, R. and Makhoul, J., "Quantization properties of transmission parameters in linear predictive systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 309–321, June 1975.

[104] Willems, N., Collier, R., and t'Hart, J., "A synthesis scheme for British English intonation," *Journal of the Acoustical Society of America*, vol. 84, pp. 1250–1261, 1988.

[105] Wouters, J. and Macon, M. W., "Control of spectral dynamics in concatenative speech synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 9, pp. 30–38, January 2001.

VITA

Sunil Ravindra Shukla was born in Stamford, Connecticut on March 14, 1972, the son of Dr. Ravindra S. Shukla and Mrs. Charugita R. Shukla. In 1989, he graduated from Cooper City High School, in Cooper City, Florida and went on to the University of Florida in Gainesville to begin his higher education studies. In December of 1993, he graduated from the University of Florida with a Bachelor of Science degree with High Honors in Electrical Engineering.

In September of 1994, he began his Master of Science degree in Electrical Engineering at the Georgia Institute of Technology as Robert Shackelford Fellow. Simultaneously, from 1993 to 1996, he worked at Motorola and Texas Instruments as a an applied research intern for speech recognition and synthesis applications. He completed the Ph.D. Comprehensive Examination on "Methods for Speech Synthesis and the Generation of Prosodic Features" in 1998. He began his dissertation research, funded by Texas Instruments, in improving the quality of current text-to-speech systems with his adviser Dr. Thomas P. Barnwell III. Also in 1998, he got married in Ahmedabad, India to his current wife, Minu Shukla, a highly talented artist.

In October 2001, he began working full-time with the automotive supplier, Visteon Corporation in Dearborn, Michigan, as a DSP Design Engineer for the Audio Electronics Group. He worked on numerous projects in the area of audio signal processing and was awarded 3 patents for his work on improving the performance of AM signals in the presence of adjacent channel interference. While working full-time, he continued to conduct the Ph.D. research and plans to receive his Ph.D. degree in Electrical and Computer Engineering from the Georgia Institute of Technology, Atlanta, Georgia, in the fall of 2006.