Enhancement and Recognition of Whispered Speech

A Thesis
Presented to
The Academic Faculty
by
Robert W. Morris

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2003

Copyright © 2003 by Robert W. Morris
Enhancement and Recognition of Whispered Speech

Approved by:

Mark A. Clements, Committee Chair

Thomas P. Barnwell III

Bihg-Hwang Jeang

Date Approved 8/30/2003
To my parents: Bill and Carolyn.
ACKNOWLEDGEMENTS

Oak is strong and also gives shade.

—Anonymous

During my senior year of college, inspiration came to me in the form of a cartoon show. I was taking a study break to watch *South Park* when one of the characters with a total laryngectomy spoke with an electrolarynx. The next day, my friends Allen Leftwich, Ted Rieger, and David Markle convinced me that I should design something better. As a hobby, I tried to develop an algorithm to convert whispers into normal speech.

I eventually showed this method to Dr. Mark Clements during my first year at Georgia Tech. Before I knew it, my research was funded and my Ph.D. research was on its way. The choice of advisor is always critical, and I feel that I was lucky to work with Mark Clements. He has always allowed me the freedom to pursue my own ideas, while giving sound advice at the key points along the way. The members of my committee, especially Dr. Thomas Barnwell and Dr. Bling-Hwang Juang, have also given many useful comments during the proposal process.

I would like to thank Paul Hong, who has always been a great roommate during most of my graduate career. His conversations have always been interesting and helpful, and this thesis would not be possible without his zealous red pen.

I would like to thank all of the students at the Center for Signal and Image Processing for making it a great place to study. I am glad to have spent time with everyone on the "Student Action Council" and I will always miss Donuts, Coffee and Tea (DCT) time. I've been lucky to be friends with so many great people: Amer Abuafadel, Jon Arrowood, Peter Cardillo, Adriane Durey, Erdem Ertan, Jon Glotzbach, Matt Lee, Sam Li, Elliot Moore, Raviv Raich, Gadi Rosen, Greg Slabaugh, Spencer Whitehead, and Xin Zhong. I would also like to thank the administrative staff: Christy Ellis, Kay Gilstrap, and Charlotte Doughty,
who have always come to my rescue when I ran into problems. I can only hope that my future work environments will be as positive.

Of course, there have been many of people outside of CSIP that have provided moral support and just made life wonderful the last five years. Special thanks to Michael Deisher, Tom Drewes, Dawnmarie Hill, Martin Izenburg, Joonhun Sung, and everyone over at Outdoor Recreation Georgia Tech.

Above all, I am thankful for my supportive family, whom I have always been able to turn to when things got tough. Without their encouragement, I would never have finished this thesis.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xii</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>xv</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation of Research</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Scope of Thesis</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Organization of Thesis</td>
<td>2</td>
</tr>
<tr>
<td>2 BACKGROUND</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Whispered Speech</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Acoustic Differences</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2 Perceptual Differences</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Speech Enhancement</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Speech Recognition</td>
<td>11</td>
</tr>
<tr>
<td>2.4 MELP Model</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Quality Measures</td>
<td>15</td>
</tr>
<tr>
<td>2.5.1 Subjective Measures</td>
<td>15</td>
</tr>
<tr>
<td>2.5.2 Objective Measures</td>
<td>16</td>
</tr>
<tr>
<td>3 CORPUS AND STANDARD CODECS</td>
<td>17</td>
</tr>
<tr>
<td>3.1 Corpus Description</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Intelligibility Tests</td>
<td>18</td>
</tr>
<tr>
<td>3.3 Subjective Quality Tests</td>
<td>20</td>
</tr>
<tr>
<td>4 JUMP MARKOV LINEAR SYSTEMS</td>
<td>24</td>
</tr>
<tr>
<td>4.1 State Estimation using JMLSs</td>
<td>25</td>
</tr>
<tr>
<td>4.1.1 Filtering</td>
<td>25</td>
</tr>
<tr>
<td>4.1.2 Smoothing</td>
<td>27</td>
</tr>
</tbody>
</table>
5 ENHANCEMENT I: CREATING NORMAL SPEECH

5.1 Estimation Framework
5.1.1 An "Ideal" Estimator
5.1.2 Model Overview
5.2 Theoretical and Empirical Spectral Differences
5.2.1 Theoretical Results
5.2.2 Empirical Models
5.3 Formant Modification
5.3.1 Linear Prediction Relationships
5.3.2 Modification Algorithm
5.3.3 Algorithm Results
5.4 Prior Distribution
5.4.1 Spectral Evolution
5.4.2 Pitch, Energy, and Voicing
5.5 Breath Noise
5.6 Results
5.6.1 Objective Results
5.6.2 Subjective Results

6 ENHANCEMENT II: NOISE REMOVAL

6.1 Parameter Estimation
6.1.1 Single Block Estimators
6.1.2 Maximum-Likelihood Approach (ML-xE)
6.1.3 Gibbs Sampler (MMSE-xE)
6.1.4 Effect of Assumptions
6.1.5 Filtering Spectral Parameters (r:S)
6.2 Waveform Enhancement
LIST OF TABLES

2.1 DRT word pairs listed by distinctive feature. ........................................... 16
3.1 Empirical signal to noise ratios calculated from DAM data. ...................... 18
3.2 Raw scores for DRT results in the quiet environment. ............................. 19
3.3 Raw scores for DRT results in noisy environments. .................................. 20
3.4 Raw scores for DAM tests in the quiet environment. ................................ 21
3.5 Raw scores for DAM results in noisy environments. ............................... 22
3.6 Results of noise preprocessing in normal speech. ................................... 22
5.1 Average absolute errors of formants and bandwidths ................................ 51
5.2 Computational requirements of formant shifting algorithm ....................... 52
5.3 Pitch-gain models derived from DAM lists for different speakers ............... 54
5.4 Effect of the number of iterations on the median Itakura distance between whispered and normal speech. .................................................. 60
5.5 Effect of different spectral modification algorithms on the median Itakura distance between whispered and normal speech. ......................... 60
5.6 Mean square errors for various pitch estimators ..................................... 61
5.7 Equal error points for various voice detectors. ..................................... 63
5.8 Whisper to voice DMOS test results ....................................................... 65
6.1 Effect of different settings for the EM algorithm on mean and median Itakura distances for white noise. .................................................... 96
6.2 Effect of number of iterations in the Gibbs sampler on mean and median Itakura distances for white noise. .................................................... 96
6.3 Median Itakura distance and the Gibbs sampler with 6000 iterations .......... 97
6.4 Median Itakura distance and the EM algorithm when the noise spectrum is known and K = 32 ................................................................. 98
6.5 Raw scores for DAM results in noisy environments. ............................... 105
6.6 Raw scores for DRT results in noisy environments. ................................ 105
6.7 Noise Removal DMOS test results. ....................................................... 106
6.8 Results of AB tests for noise removal. ................................................... 107
7.1 Average weighted Euclidean distances between normal speech features and enhanced whispered features. ..................................................... 113
7.2 Results of the DRT test using ASR. ....................................................... 113
7.3 Percent correct and accuracy of the recognizer using /CVC/ grammar.       113
7.4 Accuracy of vowel recognition using /CVC/ grammar.                   114
7.5 Vowel confusion matrix for normal speech with using MFCC features.  115
7.6 Vowel confusion matrix for whispered speech using MFCC features and no adaptation.  115
7.7 Vowel confusion matrix for whispered speech using warped MFCC features and no adaptation.  115
7.8 Vowel confusion matrix for whispered speech using MFCC features and adaptation.                  115
7.9 Vowel confusion matrix for whispered speech using warped MFCC features and adaptation.                 116
7.10 Results of the AR and NP enhancement algorithms using as ideal noise estimator.  122
7.11 AR enhancement results using nonspeech frames for noise PSD estimation.  124
7.12 Comparison between standard HMM decoding, UO decoding using variances from the AR sampler, and UO decoding using ideal variances.  124
A.1 IPA symbols used in this thesis.                                      134
B.1 DAM scores - Full Voice, NULL, Quiet.                                 135
B.2 DAM scores - Full Voice, MELP, Quiet.                                136
B.3 DAM scores - Whispered, NULL, Quiet.                                 136
B.4 DAM scores - Whispered, MELP, Quiet.                                136
B.5 DAM scores - Whispered, NULL, Street Cafe.                           137
B.6 DAM scores - Whispered, MELP, Street Cafe.                           137
B.7 DAM scores - Whispered, EM, Street Cafe.                             137
B.8 DAM scores - Whispered, Gibbs, Street Cafe.                          138
B.9 DRT scores - Full Voice, Quiet, NULL codec.                          139
B.10 DRT scores - Full Voice, Quiet, MELP 2400 codec.                   140
B.11 DRT scores - Whispered, Quiet, NULL codec.                          141
B.12 DRT scores - Whispered, Quiet, MELP 2400 codec.                    142
B.13 DRT scores - Whispered, Street cafe, MELP 2400 codec.              143
B.14 DRT scores - Whispered, Street cafe, EM estimation.                144
B.15 DRT scores - Whispered, Street cafe, Gibbs sample.                 145
B.16 ASR DRT scores - Full voice, LPCC Features.                         146
B.17 ASR DRT scores - Full voice, MFCC features. ................. 147
B.18 ASR DRT scores - Whispered, LPCC Features, No adaptation. .... 148
B.19 ASR DRT scores - Whispered, MFCC Features, No adaptation. .... 149
B.20 ASR DRT scores - Whispered, LPCC Features, Adapted. .......... 150
B.21 ASR DRT scores - Whispered, MFCC Features, Adapted. .......... 151
LIST OF FIGURES

2.1 Noise sources in larynx during whispering ........................................ 5
2.2 Modified source-filter model for whispers ......................................... 5
2.3 Formant location differences ............................................................ 6
2.4 Example of a three-state HMM generating state and observation sequences. 12
2.5 Example of a word network for the connected digit task .......................... 13
2.6 Simplified block diagram of MELP model ............................................ 14
2.7 Enhancement scheme of MELP codec .................................................. 14
3.1 Example average power spectrum densities from four samples of street cafe noise ............................................................... 18
3.2 Example spectrogram from the street cafe condition ................................ 19
3.3 Diagnostic Rhyme Test results in quiet environment for full voiced and whispered speech ....................................................... 20
3.4 Diagnostic Rhyme Test results different noise environments .................. 21
3.5 DAM results in the quiet environment for full voiced and whispered speech. 22
3.6 DAM results for different noisy environments ....................................... 23
4.1 Description of the indexing scheme for the JMLS filter ........................... 26
4.2 Example sequence generated by λ ....................................................... 35
4.3 Log-likelihood of sampled state sequence ........................................... 35
5.1 Dynamic time-warping alignment of whispered and normal speech ........... 38
5.2 Comparison of whispered and voiced sample "Shall we play a game?" .......... 39
5.3 Overall model of whispered speech .................................................... 39
5.4 Definition of terms for chain matrix vocal tract simulator ....................... 41
5.5 Effect of varying the glottal opening on the neutral vowel in the simulated vocal tract transfer function ...................................................... 41
5.6 Effect of distributing the excitation source on the neutral vowel in the simulated vocal tract transfer function .............................................. 42
5.7 Effect of excitation difference in the neutral vowel on the simulated output power spectrum ......................................................... 43
5.8 Simulated power spectrum densities for whispered and phonated vowels .... 43
5.9 Examples of function for shifting formants from whispered speech to normal speech ................................................................. 44

xii
5.10 Scheme for aligning the whispered and voiced MELP data sets. 45
5.11 An example of the proposed algorithm on a difficult spectrum. 46
5.12 Block diagram of the formant modification algorithm. 50
5.13 Modeling the LSF residual with Gaussian mixtures. 53
5.14 Example spectrum of whispers with breath noise. 56
5.15 Power spectrum densities of whispered speech and whispered speech with
breath noise. 57
5.16 Amplitude response of the Wiener filter used to remove breath noise from
whispered speech. 58
5.17 Example of breath noise removal in whispered speech. 58
5.18 Example phrase “Some diphthongs have no sound” with four pitch estimators 62
5.19 Receiver operating characteristic curves for the JMLS-based voicing detector. 63
6.1 Comparison of parametric and non-parametric spectrum estimators - voiced
sample “Why were you away a year Roy” 0 dB SNR (female) 67
6.2 Comparison of parametric and non-parametric spectrum estimators - whis-
pered sample “Shall we play a game?” 0 dB SNR(male) 68
6.3 Block diagram of the proposed algorithm. 70
6.4 Example of a EM algorithm operating on a noisy spectrum. 73
6.5 Examples of the inverted gamma distribution with $\sigma^2 = 1$. 75
6.6 Example of multiple local maxima found by the EM algorithm. 77
6.7 Example of multiple local maxima in likelihood surface. 78
6.8 EM algorithm estimated spectrum during a period of silence. 79
6.9 Example posterior distribution of a vowel segment in white noise at 10 dB
SNR 84
6.10 Correlation of spectral components in an example autoregressive process. 85
6.11 Spectral samples of three spectral estimation algorithms. 86
6.12 Real and theoretical distributions of estimated LSFs. 87
6.13 Example of discontinuity in the LSF parameters. 89
6.14 Examples of spectral amplitude weighting enhancement methods on normal
speech. 91
6.15 Examples of spectral amplitude weighting enhancement methods on normal
speech. 93
6.16 Prior model for AR-HMM enhancement schemes 94
6.17 Framework for noise removal enhancement of the MELP system 95
6.18 Clean LPC spectrum of whispered sample of male speaking "Shall we play a game?" 99
6.19 Effect of smoothing on the MMSE-E spectrum estimator 101
6.20 Comparison of spectrum estimators 102
6.21 Comparison of smoothed spectrum estimators 103
7.1 Block diagram of the baseline model training algorithm 110
7.2 ASR test for whispered speech 110
7.3 Block diagram of different feature extraction schemes 111
7.4 Weighting functions for standard and frequency warped feature extraction 112
7.5 Block diagram of the proposed feature extraction algorithm under the AR assumption 118
7.6 Comparison of AR and NP Gibbs samplers, averaged over Sets A and B, across SNRs 123
7.7 Example processing using the AR assumption on a block of noisy speech from Set A, noise condition 3, block SNR = -3 dB 125
7.8 Speaker recognition accuracy over six speakers using a GMM with 10 seconds of training data 127
7.9 Speaker recognition accuracy over six speakers using a GMM with 0.75 second test queries 128
7.10 Speaker recognition accuracy over six speakers using a JMLS with 10 seconds of training data 130
7.11 Speaker recognition accuracy comparison between JMLS and GMM modeling 130
The goal of this thesis is to study whispering from a signal processing perspective. Although it is a common mode of speech, there is little research on the subject and even less in the traditional processing fields. To fill this gap, this thesis focuses on three areas: whisper to voice speech conversion, noise mitigation for whispered speech coding, and whispered speech recognition.

In the speech conversion problem, the relationships between normal and whispered speech are cast as a statistical estimation problem. To model whispered speech, new statistical models of speech based on jump Markov linear systems (JMLS) are developed. These models are helpful to determine interframe relationships in the mixed excitation linear prediction (MELP) model that is used in the experiments to model and code speech. In addition, new methods for modifying linear prediction spectra are developed and used to explore the acoustic differences between phonated and whispered speech. These algorithms are combined to create estimates of the MELP parameters of normal speech, which are in turn synthesized using a MELP decoder to create normal speech. Surprisingly, prosodic features are easier to estimate than the true linear prediction spectrum. In fact, the mismatched spectrum was found to affect perceptual quality more than synthesizing the pitch and voicing.

In order to remove noise from whispered speech, several algorithms for estimating spectral parameters from noisy environments are proposed. These schemes are based on direct estimation of the LPC parameters instead of the spectrum. To improve these estimates, the JMLS-based spectral models described above are applied to develop spectral smoothers. These methods are shown to achieve better performance than traditional speech enhancers on whispered speech when the noise is nearly stationary.

Finally, speech and speaker recognition tasks are conducted on whispered speech data. When speech recognition systems trained on normal speech were applied to whispered
speech, the performance was found to drop significantly due to the model mismatch. Methods that modified the spectrum were found to help slightly, but the greatest performance gain was made by using the maximum likelihood linear regression adaptation method.

Overall, most speech processing algorithms for normal speech are applicable to whispers. However, there is usually a performance drop from normal speech to whispered speech that can be improved by using methods that are designed for whispered speech. In addition, many of the methods developed for whispered speech are applicable to normal speech with some modification.
CHAPTER 1

INTRODUCTION

1.1 Motivation of Research

Whispering is the mode of speech defined as speaking softly with little or no vibration of the vocal cords to avoid being overheard. This speech modality is commonly used, even in tonal languages [47], and has been studied in a variety of disciplines for many reasons. For example, general speech scientists use whispers to determine perceptual constants in the speech process [98], while medical doctors want to know if whispering is safe for recovering larynx surgery patients [91]. Speech therapists want to learn more about this mode of speaking to help evaluate voice disorders in aphonic patients [50], and forensic scientists would like to be able to recognize speaker identities from whispered speech [49]. Finally, whispered speech has also been included in the evaluation databases for NATO’s low bitrate speech codecs [96]. Despite the many applications of whispered speech, study of this mode is nearly absent in the speech processing literature.

1.2 Scope of Thesis

Different algorithms for reconstruction of normal speech, noise removal, and automatic speech recognition are investigated in this thesis. The goal is to test standard techniques and then find methods that are better suited to whispered speech than normal speech. In order to accomplish these goals, several algorithms for estimating spectral parameters from noisy environments are proposed and used in coding and recognition. These estimates are improved using new statistical models of speech based on jump Markov linear systems (JMLs). Using these models, interframe relationships have been found in the mixed excitation linear prediction (MELP) model, which is used in the experiments to model and code speech. In addition, new methods for estimating and modifying linear prediction spectra are proposed. To improve recognition of whispered speech, different adaptation schemes for
automatic speech recognition systems are evaluated. Finally, because most of these methods developed in this thesis are applicable to any speech signal, there are several tests that involve normal speech as well.

1.3 Organization of Thesis

As stated in the title, the focus of this thesis is the enhancement and recognition of whispered speech. The development of algorithms for this purpose are presented in three chapters: two for enhancement and one for recognition. The enhancement chapters are split since their natures are very different.

Chapter 5 contains an analysis of the differences between normal speech and whispered speech. These differences are exploited to reconstruct normal speech from whispered utterances.

Chapter 6 is dedicated to algorithms for removing noise from whispered speech, with an emphasis on coding applications.

Chapter 7 determines the effect of whispering on automatic speech and speaker recognition. Model adaptation algorithms and feature compensation schemes are tested to improve speech recognition performance.

In addition to the primary chapters, there are several support chapters. These chapters include background information about previous work, as well as the development of algorithms that are used across the primary chapters.

Chapter 2 includes background information on various topics relevant to this thesis, including previous research on whispered speech, speech enhancement and recognition, low bit-rate speech coding, and quality measures of speech.

Chapter 3 describes the whispered speech test data used in this thesis. A discussion of standard listening tests performed on this data is included as well.

Chapter 4 contains the algorithms for training the parameters of a jump Markov linear system. In addition, methods for smoothing parameter estimates and performing recognition based on these models are presented.
Chapter 8 contains conclusions from this work as well as suggestions for further lines of inquiry.

A few additional pieces of information have also been placed in the appendices, including a chart containing the International Phonetic Alphabet (IPA) and ARPAbase for interpretation of the phonetic transcriptions, and detailed results from experiments described in this thesis.
CHAPTER 2

BACKGROUND

2.1 Whispered Speech

2.1.1 Acoustic Differences

In normally phonated speech, air from the lungs causes the vocal folds of the larynx to vibrate, exciting the resonances of the vocal tract. In whispered speech, the glottis is opened and turbulent flow created by exhaled air passing through this glottal constriction provides a source of sound. Typically, this sound source is distributed through the lower portion of the vocal tract [94] as shown in Figure 2.1, where the left figure is a mid-sagittal slice and the right figure is a longitudinal slice. The acoustic energy is created by turbulence at the constriction at the vocal folds and interaction of the flow with the ventricular folds and the epiglottis. The resulting speech is completely noise excited with 20 dB lower power than its equivalent phonated speech [49]. The spectrum of whispers also rolls off under 500 Hz [45] due to an introduced spectral zero [94] and is typically flatter than the voiced spectrum between 500 and 2000 Hz [88]. These differences result in a more complicated model of speech shown in Figure 2.2. Another significant change is caused by the increased coupling between the trachea and the vocal tract created by the open glottis. During aspiration, additional poles and zeros, as well as damping of the first formant, have been observed [53]. Formant shifts, especially increases in the first formant frequency, occur in whispered speech [30, 48] as well as in mechanically excited speech with an open glottis [79]. These differences are shown in Figure 2.3, where the different formant locations are shown using the ARPAbet, which is listed in Appendix A. The values on the abscissa represent the average formant frequencies of the appropriate whispered vowel while the values on the ordinate represent the average difference between the whispered vowel and the voiced vowel formant frequency. For example, the first formant “A” has an average frequency of 800 Hz when whispered but is located at 600 Hz when voiced.
2.1.2 Perceptual Differences

The human perception of whispered speech is of great interest because it provides bounds on what is achievable by computer systems. For example, in order to reconstruct normal speech from whispers, one needs to know if any perceivable pitch exists in whispers. For automatic speech recognition, one would like to predict the performance loss from whispering. Finally, before attempting automatic speaker recognition, one needs to know if this task is possible when the subjects whisper.

2.1.2.1 Pitch Perception

In whispered speech, the lack of voicing results in a signal with no fundamental frequency. However, many languages use pitch to differentiate phonemes. In a recognition study of
Norwegian, Swedish, Slovenian, and Mandarin tones were recognized by the majority of listeners [47]. In fact, pitch is not a measure of fundamental frequency, but is defined as “that attribute of auditory sensation in terms of which sounds may be ordered on a musical scale” [71]. From this definition, the study the pitch of whispered speech is valid.

The first person to study the pitch of whispers was Helmholtz. Using tuning forks and whistles, he determined that the perceived pitches corresponded to the first formant frequency, \( F_1 \), for the back vowels /u, o, a/, and the second formant frequency, \( F_2 \), for the front vowels, /i, y, ø, j/. To test these relationships, McGlone and Manning recorded the vowels /i, I, e, o, a/ spoken in /hV/ and /pVp/ contexts and found that test subjects rank ordered the vowels the same as the frequencies of \( F_2 \), even when the fundamental frequency, \( F_0 \), was available in the voiced speech. In another study, judges were presented the whispered vowels /I, e, ø, a, o, u/ and then asked to adjust the frequency of a sinusoidal oscillator until it matched the whisper pitch. For every vowel, the mean frequency was equal to \( F_2 \). However, if the subjects were allowed to choose multiple pitches, \( F_1 \) was also perceived for /a/ and /ø/ as secondary pitches [99].

While the previous studies attempted to find the pitches of different vowels, Meyer-Eppler searched for the origin of pitch within vowels by analyzing spectrograms of the whispered vowels /u, ø, i, o, a/ “sung” over the range of a musical fifth. As the pitch increased, the third formant of /u/ increased from 2500 to 3000 Hz while the first formant of /a/ increased from 600 to 700 Hz. The intensity increased and the formant bandwidths widened.
with the increased pitch for every vowel. The increased pitch was also accompanied by a physical rise in the larynx. These effects were found to occur in both isolated and continuous speech [66].

Higashikawa et al. recorded whispered /a/ in three pitches: low, high, and ordinary. The correct order was found by a majority of the listeners (nine of the twelve speakers). The location of F1 rose significantly from low- to high-pitched whisper, and F2 was significantly higher in both the high and ordinary pitches than in the low-pitched whispers. The second formant frequency increased as well, but this was not statistically significant [39]. In a follow-up study, male and female /a/ whispers were synthesized with the values of F1 and F2 shifted by ± 20, 40, and 60 Hz to simulate whisper pitch. Pitch perception was stronger when F1 and F2 were moved together, and shifts in F2 created more perceptible changes in pitch than F1. The percentage of pitch matches also increased with the magnitude of the formant shifts [38].

2.1.3.2 Intelligibility

Kallail and Emanuel had subjects whisper and phonate the vowels /i, ɛ, ɔ, ʊ/ in isolation. A listening panel matched them to the following vowels: /i, ɛ, ɔ, ʊ, ʌ, ə/. This test yielded an overall accuracy of 82% for phonated vowels and 65% for whispered vowels [51].

In Tartter’s intelligibility experiment, three male and three female speakers whispered and phonated the vowels /i, ɛ, ɔ, ʊ, ʌ, ə/ in /hVd/ context. The overall accuracy by the six listeners was 82% for whispers and 92% for voiced syllables. When using the vowels from Kallail and Emanuel’s experiment, the phonated and whispered accuracies were 88% and 81%, respectively. The large difference in accuracy between these studies has been attributed to the use of context from surrounding consonants [98].

There are several ways that consonants can be classified. Tartter tested the ability of listeners to judge the voicing, place, and manner of articulation of whispered consonants. Place of a consonant is defined as the location of maximum constriction and can be categorized as labial, labio-dental, dental, alveolar, or velar. The different manners of
articulation include stops, nasals, fricatives, and glides. For this experiment, a male and female whispered the nonsense consonant-vowel syllable /C a/ with the following consonants: /b,d,g,k,m,n,p,t,ʃ,t,ʁ,w,j,f,v,s,z,ʃ,ʒ/. Six listeners identified these consonants with an overall accuracy of 64%. The majority of these errors (68%) occurred in the voicing decision, where the accuracy was 72% with a bias towards choosing unvoiced consonants. Miller and Nicely observed 98% accuracy for voicing in a study that included the dental fricatives /θ,ð/ but excluded the glides /r,l,ʍ/w/. However, the accuracy for the place of articulation in whispered speech was 91%, which is similar to the 89% accuracy cited for normal speech. The manner of speech accuracy was 86%, with the majority of these errors made by recognizing glides as stops. By removing the glides, the accuracy on whispered speech for manner improves to 90%, which is much closer to the 97% cited for voiced speech. The majority of listeners made correct identifications the majority of the time on the whispers, with the exceptions of /v/, /l/, /b/, and /g/ [67].

Dannembring tested the voicing decision in isolation by studying the perception of the voiced-unvoiced cognates /d-t/, /g-k/, /b-p/, /z-s/, /v-f/, and /θ-ð/. The results revealed significant accuracy differences among the cognates. Most of the stop pairs were discriminated better than the fricatives; the performance in discriminating the /z-s/ and /θ-ð/ pairs were poor compared to that of /d-t/. In addition, there was a statistically significant interaction between the consonant pairs and the vowels [16].

The voicing property of stops is usually characterized by the voice onset time (VOT) that measures the time between release of the stop and the beginning of voicing. Other possible cues of unvoiced consonants include increased relative amplitude of aspiration noise to the following vowel [16], increased peak flow [101], and longer closure times for unvoiced stops [87]. The first formant transition has also been shown to play a role in the voicing of stop consonants [90]. Unvoiced fricatives also tend to be shorter than their voiced counterparts.
2.1.2.3 Speaker Identification

Schwartz and Reid tested the ability of ten listeners to determine speaker sex from sustained whispered /i/ and /a/. Their judgements were correct 80 out of 80 times for /a/ and 76 out of 80 times for /i/ [88]. Less et al. did not see this level of performance in their experiments on the identifiability of speaker sex from the isolated vowels /i,e,æ,a,o,u/. The accuracy was 96% for voiced vowels, but this dropped to 76% for whispered vowels [54].

In Tartter's vowel recognition experiment, listeners attempted to identify the 6 speakers. They achieved recognition rates ranging from 39.3% to 96.3% across vowels and speakers, which was better than chance for all speakers and listeners. An acoustic analysis revealed that the most consistent difference between speakers was the syllable durations [98].

2.2 Speech Enhancement

Speech enhancement for noise removal can be used for a wide variety of situations. Although improvement of intelligibility has been elusive, enhancement research has made progress for several niche applications. In particular, waveform enhancers have been used as preprocessors for speech coding and speech recognition systems and have been shown to reduce fatigue when used to listen to large quantities of data. Several different methods, many of which are summarized in [27, 58], have been proposed for the removal of noise from speech signals. For the purposes of this thesis, these algorithms are divided into two groups: algorithms based on a subspace assumption and algorithms based on an autoregressive signal assumption.

In the subspace methods, one assumes that for a block of noisy speech, the true speech lies in some lower dimensional subspace. This is an intuitive assumption because for a periodically excited signal, the speech energy will be concentrated in the harmonics of the fundamental pitch frequency. This method has been very successful and includes algorithms such as spectral subtraction and its variations [10, 28, 29], sinusoidal modeling [3, 46], and subspace modeling [41].

Probably the most successful algorithms of this class are the spectral weighting algorithms based on the Ephraim-Malah suppression rules [12, 28, 29]. These have been
successfully used in tandem with low-bit rate speech coders including MELPs [62], where it improved both quality and intelligibility of the codec [15]. In these methods, the spectral amplitudes are modified according to an MMSE estimator, while the phase of the noisy signal is maintained. The suppression rules most commonly used are Wiener filtering and MMSE log spectral amplitude (LSA) estimators.

The LSA estimates are given by

\[
|\hat{S}(\omega_k)| = \frac{\xi_k}{1 + \xi_k} \exp \left\{ \frac{1}{2} \int_{\omega_k}^{\infty} \frac{e^{-t}}{t} dt \right\} |Y(\omega_k)|, \tag{2.1}
\]

\[
\xi_k \triangleq \frac{P_0(\omega_k)}{P_0(\omega_k)}, \quad \gamma_k \triangleq \frac{|Y(\omega_k)|^2}{P_0(\omega_k)}, \quad \nu_k \triangleq \frac{\xi_k \gamma_k}{1 + \xi_k^2}, \tag{2.2}
\]

while the Wiener solution is given by

\[
|\hat{S}(\omega_k)| = \frac{\xi_k}{1 + \xi_k} |Y(\omega_k)|. \tag{2.3}
\]

One issue with these equations is that the a priori signal to noise ratio, denoted by \( \xi_k \), is unknown. For this reason, a scheme must be devised for estimating its value. In fact, this has been shown to be the most important step in the Ephraim-Malah noise suppressor [12], which uses a decision-directed method given by

\[
\hat{\xi}_k(t) = \alpha |\hat{S}(\omega_k; t-1)|^2 / P_0(\omega_k) + (1 - \alpha) u(\nu_k(t) - 1), \tag{2.4}
\]

where \( t \) is the block number, \( u(x) \) is the unit step function, and \( \alpha \) is a smoothing constant typically set close to one.

The other class of algorithms makes the assumption that speech is a quasi-stationary Gaussian autoregressive signal. Methods for estimating these parameters based on an approximate maximum a posteriori (MAP) solution [57] have been implemented. True maximum likelihood estimates have also been proposed [70] without any experimental results. Unfortunately, these techniques have been shown to produce spectral estimates that do not follow a realistic trajectory, although heuristic methods have been shown to improve these estimates [36]. Another set of algorithms assume that the speech block is generated from an autoregressive hidden Markov model (HMM) [25, 30].

Regardless of the type of enhancement scheme used, a model of the additive noise is always necessary. The most straightforward solution to this problem is to use a voice activity
detector (VAD) to find periods of silence, then average the spectrum across these regions. Since VADs produce many errors in practice, noise trackers based on minimum statistics have been proposed and implemented in coding applications [61]. These methods track the minima of a smoothed periodogram to create an unbiased estimate of the noise spectrum without using a VAD.

2.3 Speech Recognition

Although the field of speech recognition is quite large, ranging from acoustic modeling to language understanding, this thesis deals more closely with the acoustic models that relate a phoneme or word sequence to the acoustic waveform. This section outlines the relevant steps of the speech recognition process. The recognition process can be split into two parts: feature extraction and decoding.

The feature extraction phase converts the waveform data into a sequence of vectors in time. The feature vectors usually consist of mel-frequency cepstral coefficients (MFCC) or some similar construction. The jth element of the MFCC vector, \( \mathbf{F} \), is calculated from the spectrum of a short segment of the speech, \( S_\omega(\omega) \), by

\[
O_n = [F(S_\omega)]_j = \sum_{k=0}^{K} d_{j,k} \log \left( \sum_{l=0}^{L} w_{j,l} |S_\omega(\omega_l)| \right),
\]

where \( w_{j,l} \) are called the filter-bank coefficients. These form triangular windows that are spaced evenly along the mel-frequency axis to mimic the human hearing system. These coefficients are decorrelated by using the discrete cosine transform (DCT), which has coefficients given by \( d_{j,k} \). This creates a time series of observation vectors, \( O_n \).

The decoding step requires a model that relates the phoneme sequence to the observation vectors. This is almost always accomplished by modeling either words or phonemes using a hidden Markov model (HMM) [81]. In these systems, an example of which is shown in Figure 2.4, one assumes that there is an underlying discrete state sequence, \( r_n \), for each phonetic unit. This state sequence is modeled by a discrete Markov chain where \( \Pr(r_n = j | r_{n-1} = i) = a_{ji} \). Given the underlying state, the observation vector is given by some
probability distribution, which is usually expressed as a Gaussian mixture model (GMM)

\[ p(O_n|\gamma_n = j) \triangleq b_j(O_n) = \sum_{k=1}^{K} c_{jk}N(O_n; \mu_{jk}, \Sigma_{jk}). \]  

(2.6)

These parameters must be trained from a large database of transcribed speech. This is usually accomplished iteratively using the EM algorithm [81].

Figure 2.4: Example of a three-state HMM generating state and observation sequences.

With a set of acoustic models trained, the next step is to use them to recognize a phoneme or word string from a sequence of observation vectors. In order to do this, a score must be defined for any given phrase. If the base unit is a phoneme, then the spoken phrase must be broken into a phoneme string. For example, "whisper" becomes /w i s p w/, which can be modeled by concatenating the five component HMM’s into a larger HMM, which will be referred to as \( P \). For the purposes of this thesis, the word hypothesis is scored using the Viterbi algorithm. In this algorithm, one approximates the likelihood of each phrase by using the state sequence that maximizes the likelihood of the conditional probability

\[ S(P) \triangleq \tilde{p}(O_{1:N}|P) = \arg \max_{\gamma_{1:N}} p(O_{1:N}|\gamma_{1:N}) \]

\[ = \arg \max_{\gamma_{1:N}} \prod_{n=1}^{N} b_n(O_n|a_n, \gamma_{n+1}). \]

(2.7)

The recognizer chooses the phrase, \( \tilde{P} \), that generates the largest score. When recognizing speech, there is usually a large number of these hypotheses to compare. These can be described with a grammar that gets compiled into a word network. An example of the word
network for the digit recognition task is shown in Figure 2.5, where each string consists of silence which is followed by one or more digits, and ends with silence. Again, more complicated language models are commonly used in speech recognition, but these are beyond the scope of this thesis.

![Diagram](image)

**Figure 2.5:** Example of a word network for the connected digit task.

### 2.4 MELP Model

There are a variety of parametric models of speech production. The Mixed Excitation Linear Prediction (MELP) model, which has been used successfully for low bit-rate coding, uses a source-filter decomposition of the speech signal with a parametric excitation model [64] as shown in Figure 2.6. The short-time spectrum is represented using standard linear prediction techniques, while the excitation is created by mixing periodic and random signals in five frequency bands. The random signal is simply white noise, while the periodic excitation is a pulse train with fundamental frequency given by the pitch parameter. This signal can be modified by the aperiodic flag, which jitters the pulse train, and the Fourier magnitudes, which make adjustments to the first harmonics of the pulse train.

More recently, a modified version of MELP, called MELPs, was ratified as a NATO standard [100]. This codec is very similar to MELP, but includes a 1200 bps mode and also incorporates a noise preprocessor [62] that uses the methods described in Section 2.2. This preprocessor is placed before the MELP codec as shown in Figure 2.7 to create a more accurate description of the true speech.

Since the linear prediction model used by MELP is studied extensively in this thesis, it is important to be familiar with this spectral representation. In linear prediction, the
signal is assumed to be autoregressive, where each sample is given by the weighted sum of the previous values:

\[ s[n] = \sum_{k=1}^{p} a_k s[n-k] + e[n], \]  

(2.8)

where \( e[n] \) is assumed to be iid Gaussian with variance \( \sigma^2 \). The power spectrum density of this signal is then

\[ \mathbb{E}\{S(\omega)^2\} = \frac{\sigma^2}{1 + \sum_{k=1}^{p} a_k \exp(-j\omega k)} \Delta \frac{\sigma^2}{|A(\omega)|^2}. \]  

(2.9)

There are many different ways that these parameters can be represented. One of the most commonly used is the line spectrum frequencies (LSFs), which are used for quantization in MELP. These are given by the angles of the roots \( \alpha \) of the following polynomials:

\[ P(z) = A(z) + z^{-(p+1)}A(z^{-1}), \]  

(2.10)

\[ Q(z) = A(z) - z^{-(p+1)}A(z^{-1}). \]  

(2.11)
These yield a sorted vector of numbers from zero to one that have a one-to-one correspondence to the original linear prediction coefficients.

2.5 Quality Measures
2.5.1 Subjective Measures

In order to properly evaluate any speech coding or enhancement scheme, one needs to have some method of evaluating performance. Since the end user of these systems are human listeners, it is intuitive to have this criterion based on listening tests. Quality tests of a speech system usually fall into two groups: intelligibility and subjective quality. For an effective system, it is necessary to have good scores for both of these criteria.

Although there are many intelligibility tests, one of the most common tests, and the one used in this thesis, is the diagnostic rhyme test (DRT). In this test, the user is presented a sequence of single syllable words of the form /Consonant Vowel Consonant/, which is usually written as /CVC/. The test listener is then asked to choose between two different words that include the test word and an alternative word. These alternatives always rhyme and the first consonants only differ by a single distinctive feature. These features include: voicing, nasality, sustension, sibilation, graveness, and compactness. The word pairs used in this test are listed in Table 2.1. By isolating these effects, the DRT can be used to find places where the voice system can be improved.

Like the intelligibility tests, there are different standard tests for subjective quality. In this thesis, the diagnostic acceptability measure (DAM) and the degradation mean opinion score (DMOS) are used to evaluate the algorithms. In the DAM test, the listeners are presented a series of phonetically balanced sentences. They are then asked to rate the quality using 16 different criteria from three groups: signal quality, background quality, and total quality. By adding this detail, it is easier to compare algorithm results. Because of the difficulty in administering the DAM test, the DMOS is used for many of the tests in this thesis. In this method, the user is presented with a reference waveform and a sample waveform. The user then rates the degradation of the sample with respect to the reference on a five point scale. Although this test does not require trained listeners, it does not give
<table>
<thead>
<tr>
<th>Voicing</th>
<th>Nasality</th>
<th>Sustaination</th>
<th>Sibilation</th>
<th>Gravleness</th>
<th>Compactness</th>
</tr>
</thead>
<tbody>
<tr>
<td>vvee</td>
<td>vee</td>
<td>zee</td>
<td>thee</td>
<td>wood</td>
<td>yield</td>
</tr>
<tr>
<td>bean</td>
<td>meee</td>
<td>beat</td>
<td></td>
<td>wood</td>
<td>yield</td>
</tr>
<tr>
<td>gene</td>
<td>meee</td>
<td>beat</td>
<td></td>
<td>wood</td>
<td>yield</td>
</tr>
<tr>
<td>tin</td>
<td>nip</td>
<td>dip</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>zoo</td>
<td>num</td>
<td>boot</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>dune</td>
<td>news</td>
<td>dune</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>vole</td>
<td>moom</td>
<td>bone</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>goset</td>
<td>note</td>
<td>dote</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>red</td>
<td>mend</td>
<td>bend</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>bone</td>
<td>neck</td>
<td>deck</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>vast</td>
<td>mad</td>
<td>bad</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>gaff</td>
<td>nab</td>
<td>dash</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>vault</td>
<td>nose</td>
<td>bow</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>daut</td>
<td>tawnt</td>
<td>grow</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>jock</td>
<td>knock</td>
<td>dock</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>bond</td>
<td>pond</td>
<td>knock</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>sion</td>
<td>nip</td>
<td>dip</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>tin</td>
<td>nip</td>
<td>dip</td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>zee</td>
<td>thee</td>
<td></td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>thee</td>
<td></td>
<td></td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>thee</td>
<td></td>
<td></td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>thee</td>
<td></td>
<td></td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
<tr>
<td>thee</td>
<td></td>
<td></td>
<td></td>
<td>moon</td>
<td>coop</td>
</tr>
</tbody>
</table>

the same level of detail as the DAM test.

2.5.2 Objective Measures

Although only human listeners can make truly meaningful measures of speech quality, it is useful to have objective measures to help develop enhancement schemes. Many of these are based on spectral distances between original and processed data. One commonly used measure is the Itakura distance, which is based on the linear prediction spectrum. This distance is given by

$$d_j(a, b) = \log \frac{|R_{ab}|}{\alpha |R_a|^\frac{1}{2}}$$  \hspace{1cm} (2.12)

where \(a\) and \(b\) are the linear prediction polynomials of the blocks to be compared, and \(R_a\) is the Toeplitz autocorrelation matrix of the first block. Intuitively, this is a measure of how much better the predictor polynomial, \(a\), is able to model the speech block than the other polynomial, \(b\).
CHAPTER 3

CORPUS AND STANDARD CODECS

3.1 Corpus Description

In order to evaluate the algorithms described in this thesis, whispered speech data is required for training model parameters and testing enhancement schemes. For this purpose, a whispered speech data corpus was recorded by ARCON Corporation. This corpus consists of normal and whispered speech from three male and three female speakers in three conditions: quiet, office, and street cafe. The scripts include diagnostic rhyme tests (DRT) with 232 words and diagnostic acceptability measure (DAM) lists with 15 phrases. In addition, the normal speech equivalents for the DAM lists have also been provided in the quiet environment.

The DAM and DRT tests were conducted by ARCON using these data sets. The DRT tests were conducted with eight listeners, while the DAM tests used fourteen people. In addition, the data was coded using several standard codecs, as well as two algorithms proposed in this thesis. The standard codecs tested include the 16 kbps military standard continuous variable slope delta (CVSD) algorithm and the MELP algorithm. For the MELP tests, both the 1200 and 2400 kbps modes were tested with and without a noise pre-processor. In this chapter, the results from the raw waveforms and the standard codecs are summarized.

To help interpret the results from these tests, it is useful to look at the noise characteristics in each condition. Table 3.1 contains the signal to noise ratios from the different DAM lists. Each row contains the SNRs over the waveforms for the six test speakers. Interestingly, the office environment is often less noisy than the quiet data set, where there is a steady background hum. There is occasional babble noise in the office environment, but the overall noise level is quite low. The street cafe environment offers the only truly noisy speech in this corpus, with the average SNR in the 0 dB range. This noise consists of two parts: fairly stationary primary low frequency background noise along with occasional
Table 3.1: Empirical signal to noise ratios calculated from DAM data.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Quiet (dB)</th>
<th>Office (dB)</th>
<th>Street Cafe (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>db</td>
<td>18.0</td>
<td>18.9</td>
<td>1.0</td>
</tr>
<tr>
<td>jd</td>
<td>17.1</td>
<td>20.2</td>
<td>-0.9</td>
</tr>
<tr>
<td>ps</td>
<td>29.2</td>
<td>35.7</td>
<td>0.6</td>
</tr>
<tr>
<td>dn</td>
<td>19.4</td>
<td>24.4</td>
<td>0.1</td>
</tr>
<tr>
<td>js</td>
<td>28.1</td>
<td>27.5</td>
<td>-2.1</td>
</tr>
<tr>
<td>ps</td>
<td>24.6</td>
<td>23.7</td>
<td>2.2</td>
</tr>
</tbody>
</table>

non-stationary babble. The stationary portion is fairly constant in each file but can vary from waveform to waveform. Figure 3.1 contains four different power spectrum densities and is representative of the types of noise seen in the corpus. A spectrogram of a typical portion of the noise is shown in Figure 3.2.

Figure 3.1: Example average power spectrum densities from four samples of street cafe noise.

3.2 Intelligibility Tests

To test the intelligibility of the different codecs, DRT tests were conducted on listeners with the different codecs in the different conditions. The intelligibility results, along with standard errors, for the quiet data set are listed in Table 5.2 and plotted in Figure 3.3. The NULL codec is the original waveform presented without any modification. The mos
Table 3.2: Raw scores for DRT results in the quiet environment.

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>CVSD</th>
<th>MELP 2400</th>
<th>MELP 1200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Voice</td>
<td>36.79</td>
<td>0.235</td>
<td>91.91</td>
<td>51.82</td>
</tr>
<tr>
<td>Whisper</td>
<td>82.66</td>
<td>0.363</td>
<td>75.72</td>
<td>77.52</td>
</tr>
<tr>
<td>Whisper NV</td>
<td>94.40</td>
<td>–</td>
<td>86.59</td>
<td>–</td>
</tr>
</tbody>
</table>

obvious difference is that the whispered speech performs poorly in all conditions. However, the primary source of the lower scores is the voicing decisions, which one would expect to be poor in a mode of speech with no voice. For this reason, a third row was added to the results. This score, Whisper NV, is the result of retesting the raw data and ignoring the voicing decisions. Even without the voicing decisions, Whisper NV scores are consistently lower than the Full Voice counterparts, including the NULL codec. When the codecs are applied to the speech, the Full Voice scores are all lowered but are fairly independent of the codec used. This is not the case for whispered speech, where the performance of the MELP 1200 algorithm is far worse. This is most likely due to the fact that only 9 bits (instead of 25) are allocated to the line spectrum [64, 100], which is where all of the information is carried in whispered speech. More bits could be allocated, but the bits are primarily used to code the voiced frames to higher precision.

It is important for speech codecs to perform well in noisy environments. The results from all three noise environments are included in Table 3.3 and plotted in Figure 3.4. The codecs chosen for this table include CVSD, MELP 2400, and MELP 2400 NPP. The last codec is the same as MELP 2400 with the speech enhancement front end removed. Because the SNRs are very similar for the quiet and office environments, the test characteristics are
Figure 3.3: Diagnostic Rhyme Test results in the quiet environment for full voiced and whispered speech. *Whisper* NV represents whispered speech with voicing decisions removed.

### Table 3.3: Raw scores for DRT results in noisy environments.

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>CVSD</th>
<th>MELP 2400</th>
<th>MELP 1200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td>quiet</td>
<td>84.46</td>
<td>0.41</td>
<td>77.26</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.74</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.56</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75.52</td>
<td>0.49</td>
</tr>
<tr>
<td>street cafe</td>
<td>82.9</td>
<td>0.41</td>
<td>70.77</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>65.21</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>67.65</td>
<td>0.65</td>
</tr>
</tbody>
</table>

very similar. Before coding, the street cafe noise is not strong enough to cause a significant degradation in intelligibility. However, when the noise is combined with the codecs, the distortion lowers the scores significantly. This effect is more pronounced in the MELP codec than in the CVSD codec, which performs the same in quiet conditions.

### 3.3 Subjective Quality Tests

The other test conducted on the waveforms is the DAM test. Starting with the quiet scores in Table 3.4 and Figure 3.5, one can see that the DAM scores are much lower for whispered speech before coding. The reason for this can be seen by inspecting the detailed results in Tables B.1 and B.3. Most of the individual scores remain the same, but the signal quality scores relating to thinness and harshness (SH, SD, ST) are all lower for the whispered speech. In addition, the hissing noise characteristic (BH) is also lower. Although the baseline scores are lower, the effect of coding is much smaller in whispered speech, with MELP incurring a 4.0 point drop compared to the 11.2 points seen in the full
Figure 3.4: Diagnostic Rhyme Test results for different noise environments.

Table 3.4: Raw scores for DAM tests in the quiet environment.

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>CVSD</th>
<th>MELP 2400</th>
<th>MELP 1200</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td>Full Voice</td>
<td>90.4</td>
<td>1.58</td>
<td>59.5</td>
<td>1.33</td>
</tr>
<tr>
<td>Whisper</td>
<td>95.5</td>
<td>1.50</td>
<td>47.9</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Voice data. Details for the results can be seen in Tables B.2 and B.4. The largest effects that the MELP algorithm has on normal speech is that the signal is more harsh, muffled, and nasal (SD, SL, SN). While each of these scores goes down in whispered speech, the effect is much smaller in each area, especially in signal muffling (SL).

As expected, background noise degrades the DAM scores as well. The raw scores for the noisy environments are shown in Table 3.5 and Figure 3.6. The street cafe environment lowers the DAM score of the raw waveform by 5.0 points, although there is a high degree of variability in these scores. More importantly, the MELP codec drops this down another 11.0 points, although the noise removal preprocessor mitigates this effect by 3.9 points. This combination gives similar results to the CVSD data.

Ideally, the noisy whispered speech results would be compared with the noisy full voice results. Unfortunately, the same noise conditions were not created with normal speech. For this purpose, it is necessary to compare with other experiments using the MELPv2 algorithm. In [15], the preprocessor was tested on normal speech with HMMVW troop transport test
Table 3.5: Raw scores for DAM results in noisy environments.

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>CVSD</th>
<th>MELP 2400</th>
<th>MELP NPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>quiet</td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>55.5</td>
<td>1.50</td>
<td>41.9</td>
<td>0.87</td>
</tr>
<tr>
<td>office</td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>53.9</td>
<td>1.43</td>
<td>50.0</td>
<td>1.16</td>
</tr>
<tr>
<td>street cafe</td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td></td>
<td>47.5</td>
<td>1.94</td>
<td>40.4</td>
<td>1.35</td>
</tr>
</tbody>
</table>

Figure 3.5: DAM results in the quiet environment for full voiced and whispered speech.

noise. From the results given in Table 3.6, one can see that the acoustic noise has a minor effect on the DRT scores of unprocessed waveforms but a major effect on the DAM score. When MELP coding is applied, the effect on the DRT score is far larger. By adding the noise preprocessor, the intelligibility is improved by a statistically significant amount. However, this improvement only small compared to the degradation caused by the codec. On the other hand, the DAM score is improved greatly, with the score of the processed speech rising above the score of the raw noisy waveform. These improvements are not seen in the whispered speech, where the score increases are very small and of questionable statistical significance. For this reason, it is worthwhile to pursue different methods for improving the codec's performance in noise.

Table 3.6: Results of noise preprocessing in normal speech from [15]

<table>
<thead>
<tr>
<th></th>
<th>Quiet DRT</th>
<th>HMMNw DRT</th>
<th>Quiet DAM</th>
<th>HMMNw DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SE</td>
<td>Mean</td>
<td>SE</td>
</tr>
<tr>
<td>NULL</td>
<td>97.8</td>
<td>0.23</td>
<td>91.0</td>
<td>0.35</td>
</tr>
<tr>
<td>MELP</td>
<td>93.0</td>
<td>0.30</td>
<td>67.3</td>
<td>0.80</td>
</tr>
<tr>
<td>MELPe</td>
<td>93.6</td>
<td>0.88</td>
<td>72.0</td>
<td>0.64</td>
</tr>
</tbody>
</table>

22
Figure 3.6: DAM results for different noisy environments.
CHAPTER 4

JUMP MARKOV LINEAR SYSTEMS

One of the central themes of this thesis is the application of the Jump Markov Linear System (JMLS) for speech parameter estimation. For this reason, several algorithms for signal smoothing and model estimation based on JMLSs are derived here. The JMLS is the combination of two of the most successful models for engineering applications: the dynamic linear model (DLM) and the hidden Markov model (HMM). These systems are formulated by

\[ x_{k+1} = A(r_k)x_k + B(r_k)w_{k+1} + F(r_{k+1})z_{k+1}, \quad (4.1) \]

\[ z_k = C(r_k)x_k + D(r_k)u_k + G(r_k)u_k, \quad (4.2) \]

where

- \( r_k \) unknown linear system class;
- \( x_k \) state of the jump linear system;
- \( u_k \) independent Gaussian white process noise with covariance \( Q \);
- \( v_k \) independent Gaussian white observation noise with covariance \( R \);
- \( z_k \) noisy observation process;
- \( u_k \) endogenous/exogenous variable (known).

The system class \( r_k \) is drawn from a finite Markov chain with \( S \) states and transition probabilities given by \( p_{ij} \triangleq \Pr[r_1 = i|r_{-1} = j] \). These models have been applied to a variety of situations, including seismic, communications, and target-tracking applications. In most of these cases, the system parameters are derived from the problem statement. However, if one wants to estimate the parameters of the system from training data, there are few techniques available in the literature. For notation, sequences of values in time will be denoted by \( x_{1d} \triangleq [x_1, \ldots, x_d]^T \).
4.1 State Estimation using JMLSs

The problem of JMLS state estimation has been studied extensively, especially in the target-tracking field. Although the exact state estimation problem is intractable, several different approximations have been proposed. Non-iterative techniques include the interacting multiple model (IMM) algorithm and the generalized pseudo-Bayesian (GPBz) algorithm [9]. Iterative methods [22, 59] as well as stochastic algorithms [23, 24] have been recently proposed for JMLS state estimation. Since the systems used in this thesis are less difficult to work with, the less complicated algorithms will be discussed.

4.1.1 Filtering

In order to generate the optimal filter for the JMLS, one needs to have the posterior distribution of the state vector given the observations. This is given by

\[ p(x_{1:t} | z_{1:t}) = \sum_{r_{1:t}} p(x_{1:t} | r_{1:t}, z_{1:t}) Pr(r_{1:t} | z_{1:t}). \] (4.3)

The first term in the summation is simply the output of the Kalman filter under the state sequence, \( r_{1:t} \). The main problem with performing this algorithm is that it requires a separate Kalman filter for every state hypothesis \( r_{1:t} \) which results in \( S^t \) filters at time \( t \). In addition, the conditional probability of each state path, \( Pr(r_{1:t} | z_{1:t}) \), must also be calculated.

When one performs Kalman filtering, the posterior distributions are recursively updated. The IMM and GPBz algorithms consist of the following five recursive updates:

\[ Pr(r_{t-1} | z_{t-1}) \xrightarrow{\text{mixing}} Pr(r_{t} | z_{t}), \] (4.4)

\[ p(x_{t-1} | r_{t-1}, z_{t-1}) \xrightarrow{\text{mixing}} p(x_{t-1} | r_{t-1}, z_{t-1}), \] (4.5)

\[ p(x_{t-1} | r_{t-1}, z_{t-1}) \xrightarrow{\text{evolution}} p(x_{t-1} | r_{t-1}, z_{t-1}), \] (4.6)

\[ Pr(r_{t-1} | z_{t-1}) \xrightarrow{\text{Bayes}} Pr(r_{t} | z_{1:t}), \] (4.7)

\[ p(x_{t} | r_{t}, z_{t}) \xrightarrow{\text{Bayes}} p(x_{t} | r_{t}, z_{1:t}). \] (4.8)

Equations (4.6) and (4.8) are the time and measurement updates of a Kalman filter, while the other three equations can be derived in a straightforward manner. In order to create
a tractable algorithm, a hypothesis reduction scheme must be employed at every iteration. In this step, filter densities with similar state histories are merged together. The IMM is the most computationally efficient scheme since it performs the merging after the second mixing step (4.5), keeping a state history of length one, resulting in only 2 Kalman filters. The GPBr algorithm keeps a state history of length \( x \) by merging similar hypotheses after the last step (4.8), but requires \( S^x \) Kalman filters. However, this algorithm is more easily extended to arbitrary precision, so it is used in the experiments discussed in this thesis.

![Figure 4.1](image)

**Figure 4.1**: Description of the indexing scheme for the JMLS filter.

During the time update, the number of mixtures is multiplied by \( S \). For \( i = 0, \ldots, (S^T - 1) \) and \( m = 0, \ldots, (S - 1) \),

\[
\begin{align*}
\mathbf{z}^{(i+mS^T)}_{h_{m-1}} & = A(m)\mathbf{z}^{(i)}_{h_{m-1}} + F(m)\mathbf{u}_n, \\
p^{(i+mS^T)}_{h_{m-1}} & = A(m)p^{(i)}_{h_{m-1}}A(m)^T + B(m)QB^T(m), \\
p_{h_{m-1}}^{(iS^T-1)} & = p_{(iS^T-1)h_{m-1}-1}^{(iS^T-1)}. 
\end{align*}
\]
Next, the equations are updated with the observations by

\[ \varphi_n^{(e + m S T)} = \varphi_n^{(e + m S T)} + K_n^{(e + m S T)}z_n^{(e + m S T)}, \]  
\[ \hat{x}_{n|n}^{(e + m S T)} = \hat{x}_{n|n-1}^{(e + m S T)} + K_n^{(e + m S T)}z_n^{(e + m S T)}, \]  
\[ P_{n|n}^{(e + m S T)} = P_{n|n-1}^{(e + m S T)} - K_n^{(e + m S T)}P_{n|n-1}^{(e + m S T)}K_n^{(e + m S T)^T}, \]  
\[ \varphi_n^{(e + m S T)} \propto P_{n|n-1}^{(e + m S T)}N(\hat{x}_{n|n}^{(e + m S T)}, P_{n|n}^{(e + m S T)}), \]  
\[ x_n^{(e + m S T)} = x_n - C(m)u_n - C(m)\hat{x}_{n|n-1}^{(e + m S T)}, \]  
\[ P_{n|n}^{(e + m S T)} = C(m)P_{n|n-1}^{(e + m S T)}C(m)^T + D(m)RD(m), \]  
\[ K_n^{(e + m S T)} = P_{n|n}^{(e + m S T)}(P_{n|n}^{(e + m S T)})^{-1}. \]  

To stop the exponential growth of mixtures, filters with similar histories are combined into a single Gaussian with the same second-order statistics. For \( i = 0, \ldots, (S^T - 1), \)

\[ P_{0|n}^{(i)} = \sum_{k=0}^{S^T - 1} P_{n|n}^{(k(i + k))}, \]  
\[ \hat{x}_{0|n}^{(i)} = \sum_{k=0}^{S^T - 1} \hat{x}_{n|n}^{(k(i + k))} P_{n|n}^{(k(i + k))}/P_{n|n}^{(i)}, \]  
\[ P_{0|n}^{(i)} = \sum_{k=0}^{S^T - 1} \left[ P_{n|n}^{(k(i + k))} \left( \hat{x}_{0|n}^{(i)} - \hat{x}_{0|n}^{(i)} \right) \left( \hat{x}_{0|n}^{(i)} - \hat{x}_{0|n}^{(i)} \right)^T \right]^{1/2} P_{n|n}^{(k(i + k))}/P_{n|n}^{(i)}. \]  

With the Gaussian mixture model of the posterior distribution, it is trivial to get the conditional expectation of the state variable

\[ E[x_n|x_{1:n}] \approx \sum_{i=0}^{S^T - 1} P_{0|n}^{(i)} \hat{x}_{0|n}^{(i)}. \]  

4.1.2 Smoothing

In addition to filtering, fixed-lag smoothing can also be accomplished in a fairly straightforward manner. The smoothing posterior is approximated by

\[ p(z_{n-k}\ldots z_0) \approx \sum_{i=0}^{S^T - 1} P_{0|n}^{(i)}N(x_n|\hat{x}_{0|n-l|n}^{(i)}; P_{0|n-l|n}^{(i)}). \]  

The following new parameters can be found through backward Kalman filtering:

\[ \hat{x}_{n|n-k}^{(i)} = \hat{x}_{n|n-k}^{(i)} + H_n^{(i)}(\hat{x}_{n-k+1|n}^{(i)} - \hat{x}_{n-k+1|n-k}^{(i)}), \]  
\[ H_n^{(i)} = \frac{P_{n-k|n-l|n}^{(i)}}{(S^T - 1)} \left( \frac{P_{n|n-k|n}^{(i)}}{P_{n|n-k}^{(i)}} \right)^{-1}. \]  

27
4.2 Training JMLSs

For simplicity, training is restricted to the special case where the continuous state variable, \( z_n \), is directly observed in the training data. In this case, the expectation maximization (EM) algorithm [18] can be derived to find a local maximum in the likelihood surface. This method has been used to derive the Baum-Welch algorithm [7, 81] for HMMs and to estimate the system matrices for DLMs [21]. The EM algorithm can be formulated if \( z_n \) is directly observed during training sequences.

The derivation proceeds in a similar manner to the derivation of the forward-backward algorithm in [81]. The forward and backward functions are defined as follows:

\[
p(x_1:T, r_1) = p(x_{1:T}=j)p(x_T|x_{1:T-1}, r_T) \Delta \alpha_j(t) \beta_j(t).
\]

(4.25)

These functions are useful for calculating the following probabilities:

\[
\xi_{j}(t, k) \triangleq \Pr(r_{t-1} = j, r_t = k | x_{1:T}) = p_{jk} \alpha_{j}(t-1) \beta_{k}(t) \mathcal{N}(z_{t-1} - A(j)z_{t-2} - F(j)u_{t-2}, B(j)Q^{Bj}(j))/p(x_1:T).
\]

(4.26)

\[
\gamma_j(t) \triangleq \Pr(r_t = j | x_{1:T}) = \sum_{k=1}^{S} \xi_{j+1}(t, k).
\]

(4.27)

where \( \mathcal{N}(\mu, \Sigma) \) is the Gaussian probability density function. These can be manipulated into the following recursion:

\[
\alpha_j(t) = p(x_{1:T-1}, r_{1:T}=j) = \sum_{i=1}^{S} p(x_{1:T-1}, r_{1:T-1}=i) \Delta \sum_{i=1}^{S} p(x_{1:T-1}, r_{1:T-1}=i)
\]

\[= \sum_{i=1}^{S} p(x_{1:T-1}, r_{1:T-1}=i)p(r_t = j | x_{1:T-2}, r_{1:T-1} = i)p(x_{1:T-2}, r_{1:T-1} = i)
\]

\[= \sum_{i=1}^{S} \mathcal{N}(r_{t-1} - A(i)x_{t-2} - F(i)u_{t-2}, B(i)Q^{Bj}(i))p_{ij} \alpha_i(t-1).
\]

(4.28)
Likewise, the backward variable can be expressed as

$$
\beta_t(j) = p(x_t, x_{t+1:T} | x_{t-1}, r_t = j) = p(x_{t+1:T} | x_t, x_{t-1}, r_t = j) p(x_t | x_{t-1}, r_t = j)
$$

$$
= p(x_t | x_{t-1}, r_t = j) \sum_{i=1}^S p(r_{t+1} = i | x_t, r_t = j) p(x_{t+1:T} | x_t, x_{t-1}, r_t = i, r_{t+1} = i)
$$

$$
= N \left( x_t - A_j (j) x_{t-1} - F(j) w_t, B(j) Q B(j)^T (i) \right) \sum_{i=1}^S p_{j_i} B_{i_{i+1}} (i).
$$

(4.29)

With these functions defined, the actual derivation of the iteration can begin. First, the conditional probability of the current state given the previous is given by

$$
p(x_t, r_t | x_{t-1}, r_{t-1})
= p(x_t | x_{t-1}, r_t, r_{t-1}) p(r_t | x_{t-1}, r_{t-1})
$$

$$
= p(x_t | x_{t-1}, r_t) p(r_t | r_{t-1})
$$

$$
= N \left( x_t - A(r_t) x_{t-1} - F(r_t) w_t, B(r_t) Q B(r_t)^T \right) p_{r_t | r_{t-1}}
$$

$$
= p_{r_t | r_{t-1}} \sum_{i=1}^S \left[ N \left( x_t - A(i) x_{t-1} - F(i) w_t, B(i) Q B(i)^T \right) \right]^{p_{r_{t-1} = i}}.
$$

Assembling these together and including initial conditions yields

$$
p(x_{1:T}, r_{1:T})
= p_0 \prod_{t=1}^T \left[ N \left( x_t - x_{0:t}, P_{0:t} \right) \right]^{p_{r_{t-1} = i}}
$$

$$
\times \prod_{t=2}^T p_{r_t | r_{t-1}} \sum_{i=1}^S \left[ N \left( x_t - A(i) x_{t-1} - F(i) w_t, B(i) Q B(i)^T \right) \right]^{p_{r_{t-1} = i}}.
$$

(4.31)

Next, the logarithm is taken and constant terms are dropped to get

$$
l(x_{1:T}, r_{1:T})
= \log p_0 - \frac{1}{2} \sum_{t=1}^T \log P_{0:t} + \log P_{x_t} (i) + (x_t - x_{0:t})^T P_{x_t}^{-1} (i) (x_t - x_{0:t})
$$

$$
+ \frac{3}{4} \sum_{t=2}^T \left[ \frac{1}{2} \sum_{t=1}^T \log \left| B(i) Q B(i)^T \right| + (x_t - A(i) x_{t-1} - F(i) w_t)^T (B(i) Q B(i)^T)^{-1} (x_t - A(i) x_{t-1} - F(i) w_t) \right].
$$

Taking the conditional expectation and substituting $\xi_t(j, k)$ and $\gamma_t(i)$ where appropriate yields

$$
l(x_{1:T}, r_{1:T})
= \frac{1}{2} \sum_{t=1}^T \log p_t - \frac{1}{2} \log \left| P_{x_t} (i) \right| + (x_t - x_{0:t})^T P_{x_t}^{-1} (i) (x_t - x_{0:t})
$$

$$
+ \frac{3}{4} \sum_{t=2}^T \sum_{j=1}^S \xi_t(j, k) \log p_{j_t} - \frac{1}{2} \sum_{t=1}^T \log \left| B(i) Q B(i)^T \right|
$$

$$
+ (x_t - A(i) x_{t-1} - F(i) w_t)^T (B(i) Q B(i)^T)^{-1} (x_t - A(i) x_{t-1} - F(i) w_t) \right].
$$

29
Taking derivatives with respect to $p_{jk}$ and $p_j$ and using Lagrange multipliers to ensure proper structure, this yields the same result as for HMMs:

\[ p_j = \gamma_j(j) \]  
\[ p_{jk} = \frac{\sum_{i=1}^{T} \delta(i,k)}{\sum_{i=1}^{T} \gamma(i,j)} \]  
\[ \dot{p}_{jk} = \frac{\sum_{i=1}^{T} \delta(i,k)}{\sum_{i=1}^{T} \gamma(i,j)} \gamma(j,i) \gamma(i,j) \]  

The system matrix estimation starts by gathering the following statistics:

\[ \Gamma_{xz1}(t) \triangleq \sum_{i=1}^{T} \gamma(i)(x_{z1}^{T} x_{z1}) \]  
\[ \Gamma_{xz1}(t) \triangleq \sum_{i=1}^{T} \gamma(i) x_{z1} x_{z1}^{T} \]  
\[ \Gamma_{xu}(t) \triangleq \sum_{i=1}^{T} \gamma(i) x_{u} x_{u}^{T} \]  
\[ \Gamma_{xu}(t) \triangleq \sum_{i=1}^{T} \gamma(i) x_{u} x_{u}^{T} \]  
\[ \Gamma_{wu}(t) \triangleq \sum_{i=1}^{T} \gamma(i) w_{u} w_{u}^{T} \]  
\[ \Gamma_{wu}(t) \triangleq \sum_{i=1}^{T} \gamma(i) w_{u} w_{u}^{T} \]  

Taking derivatives with respect to the different system matrices yields the following relationships:

\[ A(t) F(t) = \left[ \begin{array}{c} \Gamma_{xz1}(t) \\ \Gamma_{xu}(t) \end{array} \right]^{-1} \left[ \begin{array}{c} \Gamma_{xz1}(t) \\ \Gamma_{xu}(t) \end{array} \right] \]  
\[ B(i) Q B^T(i) = \left[ \begin{array}{c} \Gamma_{xz1}(t) \\ \Gamma_{xu}(t) \end{array} \right] - \left[ \begin{array}{c} A(t) F(t) \end{array} \right] \left[ \begin{array}{c} \Gamma_{xz1}(t) \\ \Gamma_{xu}(t) \end{array} \right] \]  
\[ \sum_{i=1}^{T} \gamma(i) \]  

The product term $B(i) Q B^T(i)$ can be decomposed in any manner, such as the Cholesky decomposition. This is useful since only the product is relevant to filtering operations.

If one does not have enough training data to accurately estimate the full transition matrices, $A(i)$, one can constrain them to be diagonal by solving:

\[ \left[ \begin{array}{c} A(i) F(i) \end{array} \right] = \left[ \begin{array}{c} \Gamma_{xz1}(i) \\ \Gamma_{xu}(i) \end{array} \right] \circ \left[ \begin{array}{c} A(i) F(i) \end{array} \right] \]  
\[ \sum_{i=1}^{T} \gamma(i) \]  

where $\circ$ represents the element-wise Hadamard product. This yields the system of equations:

\[ \left[ \begin{array}{c} \Gamma_{xz1}(i) \\ \Gamma_{xu}(i) \end{array} \right] A(i) F(i) = \left[ \begin{array}{c} \Gamma_{xz1}(i) \\ \Gamma_{xu}(i) \end{array} \right] \circ \left[ \begin{array}{c} A(i) F(i) \end{array} \right] \]  
\[ \sum_{i=1}^{T} \gamma(i) \]  

which can be expressed by

\[ \text{diag}^{-1}(A(i)) F(i) = \text{diag}^{-1}(A(i)) F(i) \]  
\[ \text{diag}^{-1}(A(i)) F(i) \]  
\[ \text{diag}^{-1}(A(i)) F(i) \]  
\[ \left[ \begin{array}{c} \Gamma_{xz1}(i) \\ \Gamma_{xu}(i) \end{array} \right] \circ \left[ \begin{array}{c} A(i) F(i) \end{array} \right] \]  
\[ \sum_{i=1}^{T} \gamma(i) \]  

where $\text{diag}^{-1}(A(i))$ is the inverse of the diagonal elements of $A(i)$. 

\[ \text{diag}^{-1}(A(i)) F(i) \]  
\[ \left[ \begin{array}{c} \Gamma_{xz1}(i) \\ \Gamma_{xu}(i) \end{array} \right] \circ \left[ \begin{array}{c} A(i) F(i) \end{array} \right] \]  
\[ \sum_{i=1}^{T} \gamma(i) \]  

30
where "diag" constructs a diagonal matrix from a vector and "diag\(^{-1}\)" constructs a vector from a matrix's diagonal. If the transition matrix is constrained to be diagonal, the noise matrix update can no longer be simplified and must be written explicitly as

\[
B(i)QB^T(i) = \begin{bmatrix}
A(i) & F(i) & I
\end{bmatrix}
\begin{bmatrix}
\Gamma_{z1z1} & \Gamma_{z1u} & -\Gamma_{u1z1}^T
\Gamma_{z2u} & \Gamma_{uu} & -\Gamma_{u2u}^T
-\Gamma_{u1z2} & -\Gamma_{u2u} & \Gamma_{zz}
\end{bmatrix}
\begin{bmatrix}
A^T(i)

F^T(i)

I
\end{bmatrix}
\left/ \sum_{i=1}^{T} \gamma(i) \right.,
\]

which can, in turn, be constrained to be diagonal by simply setting the off-diagonal elements to zero.

If the state and observation variables are both available, and one wants to estimate every parameter in the JMLS, then the equations for the forward and backward variables must be modified to include the observation equations

\[
\alpha_i(j) = \sum_{i=1}^{S} \mathcal{N}(x_{i-1} - A(i)x_{i-2} - F(i)u_{i-1}, B(i)QB^T(i)) \times \mathcal{N}(z_{i-1} - C(i)z_{i-1} - G(i)u_{i-1}, D(i)RD^T(i)) \prod_p \beta_{p,i}(i),
\]

\[
\beta_i(j) = \mathcal{N}(z_{i-1} - A(j)x_{i-1} - F(j)u_{i-1}, B(j)QB^T(j)) \times \mathcal{N}(z_{i-1} - C(j)z_{i-1} - G(j)u_{i-1}, D(j)RD^T(j)) \sum_p \beta_{p,i}(i).
\]

The conditional log-likelihood with the new terms is equal to

\[
\ell(x_{1:T}, r_{1:T}, z_{1:T}) = \ell(x_{1:T}, r_{1:T}) + \sum_{i=1}^{T} \left\{ \frac{1}{2} \sum_{i=1}^{S} \gamma(i) \left[ \log |D(i)RD^T(i)| \right] + (z_i - C(i)x_i - G(i)u_i)(D(i)RD^T(i))^{-1} (z_i - C(i)x_i - G(i)u_i) \right\}.
\]

Maximizing the conditional likelihood requires a new set of conditional statistics:

\[
\Gamma_{zz}(i) \triangleq \sum_{i=1}^{T} \gamma(i)z_i z_i^T, \quad \Gamma_{zu}(i) \triangleq \sum_{i=1}^{T} \gamma(i)z_i u_i^T, \quad \Gamma_{uz}(i) \triangleq \sum_{i=1}^{T} \gamma(i)u_i z_i^T,
\]

\[
\Gamma_{uu}(i) \triangleq \sum_{i=1}^{T} \gamma(i)u_i u_i^T, \quad \Gamma_{zz}(i) \triangleq \sum_{i=1}^{T} \gamma(i)z_i z_i^T.
\]

(4.42)
This yields the following updates:

\[
\begin{bmatrix}
C(i) & G(i)
\end{bmatrix} = \begin{bmatrix}
\Gamma_{zz}(i) & \Gamma_{zn}(i) \\
\Gamma_{zn}(i)^\top & \Gamma_{nn}(i)
\end{bmatrix}^{-1},
\]

\[
D(i)RD^T(i) = \left\{ \begin{array}{c}
\Gamma_{zz}(i) - C(i)G(i) \\
\end{array} \right\} \begin{bmatrix}
\Gamma_{zz}(i) & 1
\end{bmatrix}^{-1} \sum_{t=1}^{T} \gamma(i). \tag{4.44}
\]

Again, the matrices can be constrained to be diagonal, which yields the updates

\[
\begin{bmatrix}
\text{diag}^{-1}(C(i)) & G(i)
\end{bmatrix} = \begin{bmatrix}
\text{diag}^{-1}(\Gamma_{zz}(i)) & \Gamma_{zn}(i) \\
\text{diag}(\Gamma_{nn}(i)) & 1
\end{bmatrix}^{-1},
\]

\[
D(i)RD^T(i) = \begin{bmatrix}
C(i) & G(i) & I
\end{bmatrix} \begin{bmatrix}
\Gamma_{zz} & \Gamma_{zn} & -\Gamma_{zn} \\
\Gamma_{zn}^\top & \Gamma_{nn} & -\Gamma_{nn} \\
-\Gamma_{zn} & -\Gamma_{nn} & \Gamma_{zz}
\end{bmatrix}^{-1} \sum_{t=1}^{T} \gamma(i). \tag{4.46}
\]

### 4.2.1 Implementation Issues

Many of the implementation issues that arise in HMMs also arise in this training algorithm. For example, there is the issue of scaling the forward and backward variables. Although the equations given above would work using infinite precision arithmetic, \( \alpha_t \) and \( \beta_t \) both go to zero exponentially fast and quickly generate underflows. The scaling procedure used in [81] is applicable to this training algorithm as well.

It is also well known that the EM algorithm does not necessarily converge to the global maximum of the likelihood surface. For this reason, initial estimates of the parameters are critical. However, if the number of states, \( S \), is large, it is not obvious how to initialize the model. For this reason, it is useful to initialize the model with one state, then add additional states using a "model splitting" procedure. This procedure starts by finding the most probable state, \( r_{\text{max}} \), by looking at the steady state distribution of the transition matrix, \( p \). The parameter matrices for this model are duplicated into a new state with index \( S + 1 \), but the transition noise matrix is multiplied by a scale factor, \( 1 + \epsilon \). The transition matrix is updating by increasing the size to \( S + 1 \) and copying row \( r_{\text{max}} \) into row \( S + 1 \),
then dividing column \( v_{\text{reset}} \) by two and copying it into column \( S + 1 \). The EM algorithm is applied to this new model until it converges.

In this thesis, the continuous state is assumed to be known. However, if only \( z_n \) is available, the problem becomes more difficult. The expectation step requires solving for the conditional means and covariance matrices of \( z_n \), which is known to be intractable. For this reason, approximations would be required for any scheme to estimate the parameters in this situation.

### 4.3 Recognition

Another task that could be useful for JMLs is the determination of the probability that an observation is generated from a particular model. In this case, one would like to find \( p(\pi_1:T|\lambda) \), where \( \lambda \) is the model of interest. In order to accomplish this, the forward variable, \( \alpha_i(t) \), in Equation 4.28 must be modified to include the continuous hidden state, \( z_n \), to yield

\[
\alpha_i(j,y) \triangleq p(z_{1:T}, \pi_1 = j, \pi_T = y). \tag{4.47}
\]

Given state \( i \), the function is given by a Gaussian mixture. For this purpose, the notation from the filtering section is used:

\[
\alpha_i(j,y) = \sum_{k=1}^{K^{(i)}} \pi_{y^i}^{(h)} N \left( y; \mu_{y^i}^{(h)} , \sigma_{y^i}^{(h)} \right). \tag{4.48}
\]

This can be written recursively using the methods described in the training method, and yields the same updates for the parameters as in the smoothing method. The recursions are different in one regard: the value for \( p_i^{(t)} \) in Equation 4.14 is not normalized. This is critical, because these are the probabilities that are used in the final probability calculation.

This yields a score given by

\[
p(z_{1:T} | \lambda) = \sum_{i=0}^{S-1} \int cy(i,y)dy \tag{4.49}
\]

\[
= \sum_{k=0}^{S-1} \int p^{(K)}_{\pi^i_{T}} N \left( y; \mu_{y^i}^{(K)} , \sigma_{y^i}^{(K)} \right) dy \tag{4.50}
\]

\[
= \sum_{k=0}^{S-1} p^{(K)}_{\pi^i_{T}}. \tag{4.51}
\]

33.
4.4 Test Example

To show the training process, an artificial example JMLS, $\lambda$, is constructed. The system consists of two states with transition probabilities given by $p_{11} = p_{22} = 0.8$. For this example, only the state variables are available, and the $F$ matrix is zero for both states.

The other matrices are given by

$$A(1) = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.9 \end{pmatrix}, \quad A(2) = \begin{pmatrix} 0.1 & 0.8 \\ 0.8 & 0.1 \end{pmatrix},$$

$$B(1) = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}, \quad B(2) = \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}.$$ (4.52)

A 500 sample state sequence, which is displayed in Figure 4.2, was randomly sampled using $\lambda$. A single state model was estimated with this sequence to get the estimates

$$\hat{A}_0(1) = \begin{pmatrix} 0.7612 & 0.1961 \\ 0.1510 & 0.7552 \end{pmatrix}, \quad \hat{B}_0(1) = \begin{pmatrix} 0.9522 & 0.00 \\ -0.4716 & 0.8737 \end{pmatrix}. \quad (4.53)$$

This was then split into two models with equal probability. The EM algorithm was then performed with 20 iterations to get the new estimate, $\hat{\lambda}$. The auxiliary function, $Q$, converged rapidly as shown in Figure 4.3. The final estimate is

$$\hat{A}_1 = \begin{pmatrix} 0.9119 & 0.0001 \\ 0.0073 & 0.9301 \end{pmatrix}, \quad \hat{A}_2 = \begin{pmatrix} 0.0977 & 0.7885 \\ 0.8033 & 0.9964 \end{pmatrix},$$

$$\hat{B}_1 = \begin{pmatrix} 0.9792 & 0.00 \\ -0.0187 & 1.0356 \end{pmatrix}, \quad \hat{B}_2 = \begin{pmatrix} 0.1045 & 0.00 \\ 0.6018 & 0.1025 \end{pmatrix},$$ (4.54)

$$\hat{p} = \begin{pmatrix} 0.8419 & 0.1581 \\ 0.1623 & 0.8377 \end{pmatrix}.$$

One can see that the initial single state estimate of the JMLS does not adequately describe the true system. For example, the state noise is found to be negatively correlated, which is, in fact, not the case. However, when two states are used, the two different modes
Figure 4.2: Example sequence generated by $\lambda$.

Figure 4.3: Log-likelihood of sampled state sequence.

of the system are automatically identified. Of course, the actual values are different than the true values, but the overall two-model nature of the signal has been determined.
CHAPTER 5

ENHANCEMENT I: CREATING NORMAL SPEECH

5.1 Estimation Framework

In order to find algorithms to enhance whispered speech, it is useful to first look at the problem in a decision theoretic framework. The random parametric representation of the voiced utterance is referred to as \( V \), and the potentially noisy whispered utterance is called \( W \). In the MELP model, these utterances are expressed as a sequence of fixed-length frames sampled every 22.5 ms:

\[
V \triangleq \{ V_1, \ldots, V_N \}, \quad W \triangleq \{ W_1, \ldots, W_N \}, \quad (5.1)
\]

where the individual frames are expressed by

\[
V_n \triangleq \{ f_n, g_n, p_n, v_n \}, \quad W_n \triangleq \{ \tilde{f}_n, \tilde{g}_n \}, \quad (5.2)
\]

where \( \tilde{f}_n \) is the vector of line spectrum frequencies (LSF), \( g_n \) is the gain, \( p_n \) is the pitch, and \( v_n \) is the vector of five bandpass voicing values and the aperiodic flag. The whispered LSF and gain parameters are given by \( \tilde{f}_n \) and \( \tilde{g}_n \), respectively, while pitch and voicing are not represented because they do not exist in whispered speech. Of course, these parameters are not directly observed. Instead, the acoustic signal, \( y \), is gathered from a microphone and sampled at 8 kHz. For notation, this signal is also expressed as a sequence of frames:

\[
y \triangleq \{ y_1, \ldots, y_N \}, \quad y_n \triangleq [ p_n[i], \ldots, y_n[t] ]^T, \quad (5.3)
\]

\[
y_n[i] = s_n[i] + v_n[i], \quad (5.4)
\]

\[
es_n[i] = \sum_{k=1}^{P} s_n[k] y_n[t-k] + e_n[i], \quad (5.5)
\]

\[
e_n \sim \mathcal{N}(0, \sigma_n^2 I), \quad (5.6)
\]

where \( v_n[i] \) is a random process representing the background noise, \( s_n[i] \) is the actual whisper waveform that is assumed to be autoregressive with residual variance, \( \sigma_n^2 \), and prediction
polynomial, \( a_n[k] \). This variance is set such that \( 10 \log_{10} [E(x_i^2[k])] = \gamma_n \). The prediction polynomial is calculated from the line spectrum pairs, \( f_n \). One solution to the problem of estimating the normal speech from whispers is to find the best function \( \delta \) to minimize the expected value of some loss function \( \mathcal{L} \):

\[
\hat{\delta} = \arg \min_{\delta} \mathbb{E}[\mathcal{L}(\mathcal{V}, \delta(y))].
\] (5.7)

It is also desirable for real-time implementations to have causal and fixed-lag estimators of the \( n \)th block:

\[
\hat{V}_n = \arg \min_{\delta} \mathbb{E}[\mathcal{L}(\mathcal{V}, \delta(y_{1:n+L}))] = \mathcal{L}(V_n, \delta(y_{1:n+L})),
\] (5.8)

where \( L \) is the delay. One might also want to estimate the whispered speech parameters:

\[
\hat{W}_n = \arg \min_{\delta} \mathbb{E}[\mathcal{L}(\mathcal{W}, \delta(y_{1:n+L}))].
\] (5.9)

In order to create the estimator, one needs the joint distribution \( p(V, W, y) \) and a loss function \( \mathcal{L} \).

5.1.1 An "Ideal" Estimator

Given a whispered utterance, the naive approach to synthesizing the true speech is to estimate the excitation parameters and use the spectral parameters from the whispered utterance. This is accomplished by using the parameters from two waveforms: the first is whispered and the second is spoken normally. Ideally, one would directly copy the pitch and voicing information from the normal speech. However, the two waveforms are never aligned in time. In order to align the waveforms, the optimal mapping between the two is found using a dynamic time warping algorithm as shown in Figure 5.1. This estimator is given by

\[
(\delta_n, \delta_n) = \arg \min_{(\delta_n, \delta_n)} \sum_n \mathcal{L}(\mathcal{V}_n, \mathcal{W}_n) + D(\delta_n, \delta_n),
\] (5.10)

\[
\hat{V}_n(\mathcal{V}_n) = \left\{ \delta_n(\mathcal{V}_n), \delta_n(\mathcal{V}_n), \mathcal{W}_n(\mathcal{V}_n) \right\}.
\] (5.11)

where \( \mathcal{L} \) is the weighted sum of the Itakura distance between the two sets of prediction coefficients and the gain parameter, and the pair \( (\delta_n, \delta_n) \) represents the time warping. The
function $D$ weights different paths to encourage realistic time warps. As an example, the phrase “Shall we play a game?” was recorded twice by the same speaker. This was processed using the method described above, and the linear prediction spectrograms of $\tilde{V}$ and $V$ are shown in Figure 5.2. The waveform synthesized from $\tilde{V}$ has poor quality for two reasons: the spectral estimates, $\tilde{F}_n$, are more erratic because the signal is noise-excited and the resonance structure is altered during whispering as described in Section 2.1.1. To compensate for these issues, the spectrum must first be smoothed, then the formants must be shifted to their proper locations.

![Figure 5.1: Dynamic time-warping alignment of whispered and normal speech.](image)

5.1.2 Model Overview

At this point, the overall model for whispered speech can be described as shown in Figure 5.3. This model consists of three parts: the prior model of normal speech $p(V)$, the conditional distribution of the whispered parameters, $p(W|V)$; and the acoustic waveform model, $p(y|W)$.

The acoustic model has been defined in Equations 5.3-5.5, while the other two elements are described in the following sections. In particular, the distribution of $p(W|V)$ is modeled as a deterministic modification of the formants and a complete loss of voicing and pitch information. Finally, an underlying model, $p(V)$, that describes how both the spectral and prosodic parameters evolve through time is derived using jump Markov linear systems.
Figure 5.2: Comparison of whispered and voiced sample “Shall we play a game?”

\[ p(\mathbf{V}, \mathbf{W}, \mathbf{y}) = p(\mathbf{y} | \mathbf{W}) p(\mathbf{W} | \mathbf{V}) p(\mathbf{V}) \]

\[ p(\mathbf{y} | \mathbf{W}) \]

\[ e_n[r] \xrightarrow{} \frac{\sigma_n^2}{A_n(z)} \xrightarrow{} y_n[r] \]

\[ p(\mathbf{W} | \mathbf{V}) \]

\[ f_n \xrightarrow{} \text{Formant Shift} \xrightarrow{} \tilde{f}_n \]

\[ g_n \]

\[ p(\mathbf{V}) \]

\[ r_1 \ldots r_n \]

\[ X_n = A(r_n)X_{n-1} + B(r_n)w_n + F(r_n) \]

\[ p(w_n) = N(0, Q) \]

Figure 5.3: Overall model of whispered speech.
5.2 Theoretical and Empirical Spectral Differences

Empirical and theoretical studies have been conducted to determine the effects of whispering on the speech spectrum. In this section, theoretical models are explored to determine which physical differences affect the speech signal most significantly. The data corpus is used to create empirical models that relate whispered spectra with their normally phonated equivalents.

5.2.1 Theoretical Results

As previously stated, there are three primary reasons that whispered vowels can vary from phonated vowels. These include differences in the excitation spectrum, coupling with sub-glottal structures, and acoustic source location. In this section, several tests are constructed that test the effects of aspiration on different vowels.

In these models, the vocal tract is modeled with chain matrices to generate a frequency-domain vocal tract transfer function. In this method, the pressure and volume velocity at the ends of a uniform cylindrical tube are given by

\[
\begin{pmatrix}
    P_{\text{out}}(\omega) \\
    U_{\text{out}}(\omega)
\end{pmatrix} =
\begin{pmatrix}
    A(\omega) & B(\omega) \\
    C(\omega) & D(\omega)
\end{pmatrix}
\begin{pmatrix}
    P_{\text{in}}(\omega) \\
    U_{\text{in}}(\omega)
\end{pmatrix},
\]

where the elements of the chain matrix are described in [92]. To find the response of a series of concatenated tubes, one needs to simply multiply the individual matrices. This method is very useful for simulating the modified model in Figure 2.2 since sources can be placed between any two sections of the vocal tract. For the purposes of these tests, the synthesizer is configured as shown in Figure 5.4. In this configuration, there is a single source at some point in the vocal tract. The source is always a velocity source, as this has been shown to be more effective than pressure sources in modeling fricatives [92]. Given the two chain matrices, \( K_1 \) and \( K_2 \), the glottal and lip impedances, \( Z_G \) and \( Z_L \), and the source spectrum, \( U_s(\omega) \), it is possible to synthesize vowels, which are represented by \( P_{\text{out}} \).

The transfer function is then given by

\[
H_{\text{out}}(\omega) = \frac{P_{\text{out}}(\omega)}{U_s(\omega)} = \frac{Z_G(\omega)}{A_1(\omega) - C_1(\omega)Z_G(\omega)Z_L(\omega) + Z_L(\omega)}.
\]

40
where the first fraction is the transfer function of the system without the subglottal coupling. The second term can be viewed as a current divider between the two branches. The two impedances are given by

\[
Z_1(\omega) = \frac{A_1(\omega)Z_2(\omega) - B_1(\omega)}{D_1(\omega) - C_1(\omega)Z_2(\omega)} \quad Z_2(\omega) = \frac{D_2(\omega)Z_1(\omega) - B_2(\omega)}{A_2(\omega) - C_2(\omega)Z_1(\omega)}
\]

(5.14)

where the lip impedance is modeled by a pulsating sphere with radius equal to that of the last tube in \(K_3\) [92]. The glottal impedance is given by an acoustic mass in series with a resistance as described in [94].

The first test is to determine the effect of glottal opening on the vocal tract frequency responses. This test was performed on a neutral vowel with transfer function, \(H_{\text{out}}\), shown in Figure 5.5. The glottal opening was simulated with a single tube with a length of 3 mm and the desired opening area. One can see that as the opening is widened, the formants rise in frequency. However, this effect is more pronounced in the lower frequencies, especially the first formant, which is also attenuated.

![Figure 5.5: Effect of glottal opening on vocal tract](image)

Next, the effect of having a distributed source is computed by moving the source along
the vocal tract as shown in Figure 2.2. The resulting power spectra are averaged to get the output transfer function in Figure 5.6. In general, the formants are widened, and the higher formants are raised in frequency. The widening is caused by averaging over different formant locations due to the source locations. Unfortunately, this is not consistent with real spectra seen in the whispered speech corpus.

![Effect of distributing source excitation](image)

**Figure 5.6:** Effect of distributing the excitation source on the neutral vowel in the simulated vocal tract transfer function.

The last experiment is designed to determine the effect of the spectrum on altered excitations. In Figure 5.7, the neutral vowel is displayed with two different types of excitation spectra. The normal spectrum uses the Liljescrants-Fant model of the glottal pulse, while the whispered speech uses the dipole spectrum described in [94]. As expected, the whispered speech is flatter than the normal speech. However, the normal speech tends to have slightly more low frequency energy than the model shows. Finally, the simulated spectra for four different vowels are shown in Figure 5.8. These simulations are based on the opening of the glottis and the modified excitation. In general, these simulated spectra have formants that are higher in frequency.

### 5.2.2 Empirical Models

Although there are theoretical relationships between normal and whispered speech, these are many consonant phonemes where the relationship differs from the steady vowels. Another issue with the results in the previous section is that they do not account for changes in the articulation that may exist between normal and whispered speech. For this reason, the focus
Figure 5.7: Effect of excitation difference in the neutral vowel on the simulated output power spectrum.

Figure 5.8: Simulated power spectrum densities for whispered and phonated vowels.

now turns to actual speech data. Although it is impossible to create a true stereo database, the data described in Section 3.1 can be used by warping the waveforms onto the same time axis as described in Section 5.1.1. In order to improve the matches, the whispered speech is preprocessed using a breath removal algorithm described in Section 5.5. This provides a set of matching spectral pairs that can be used to train a spectral model.

In order to form estimates of the formant differences, it is necessary to parameterize this function. In this thesis, this function is described by modifying the formant locations
and altering the spectral tilt. This is denoted by

$$\hat{F}_n = \begin{cases} S(f_n; F_{max}, F_s, T), & \text{Voiced,} \\ f_n, & \text{Unvoiced,} \end{cases}$$  \ (5.15)$$

where $S$ is the spectrum modification function that has three parameters: $F_{max}$, $F_s$, and $T$. The formant modification is described by a piecewise linear function that determines how far a formant at a given frequency must be shifted. The maximum change in formant frequency is given by $F_{max}$ and occurs at $F_s$. In addition, the formant shift at both 0 Hz and 2500 Hz is constrained to be zero. Examples of this function are shown in Figure 5.9. The $T$ parameter describes the amount by which the spectral tilt is increased.

![Whisper to Voice Formant Shifting Rules](image)

**Figure 5.9:** Examples of functions for shifting formants from whispered speech to normal speech with $F_{max} = 200, 400, 600$ and $F_s = 700, 1000$.

Since there is no closed form method for estimating these parameters, a grid search of the different parameter values is performed to maximize the overall likelihood of collected whispers. This is accomplished by aligning the frames as shown in Figure 5.10. The parameters are aligned, and the three spectral parameters are modified until the overall spectral distance is minimized. In the next section, the method for modifying formants is described.

### 5.3 Formant Modification

Formant modification is useful for many applications. For example, speech produced in non-air atmospheres [8] and under high G-force and pressure [93] is characterized by modified formants. Formant locations and bandwidths also are changed during whispered speech due
to tracheal coupling [48, 79].

One method to compensate for these changes is to modify the pole locations of the linear prediction (LP) polynomial. However, this method has problems due to pole interaction [40, 68, 69]. In addition, accurate determination of the complex roots of a polynomial requires extensive floating point calculations. Enhancement schemes based on simple transformations of the line spectrum pair (LSP) frequencies have also been developed [65]. These rely on the well-known fact that narrowband formants are represented by two LSPs located near the formant frequency [41]. However, when formants have wider bandwidths, or are clustered together, this relationship is less obvious and applicable.

In Figure 5.11, a sample LPC spectrum for speech is displayed in three formats. The top plot contains the power spectral density (PSD). The second plot represents the LSP frequencies with vertical bars, while the markers in the bottom plot show the location of the roots of the prediction polynomial. In this example, the first two formants have five LSPs and three poles associated with them while the third formant has four LSPs and two poles. This ambiguous relationship among the formants, LSPs, and poles makes it difficult to modify these formants using a simple LSP or pole modification. In order to deal with these issues, we propose an algorithm that takes advantage of the nearly linear relationship between the LSPs and formants.

5.3.1 Linear Prediction Relationships

The Jacobian matrices relating the linear prediction coefficients, formant structure, and line spectrum pair frequencies are expressed in this section. These matrices are useful for a variety of purposes including finding statistics and modifying parameters.
Figure 5.11: An example of the proposed algorithm on a difficult spectrum: $\Delta F_1 = 150$ Hz, $\Delta F_2 = -100$ Hz.

5.3.1.1 Line Spectrum Pair Frequencies

The LSP frequencies are a representation of linear prediction coefficients that are commonly used for coding. The two LSP polynomials are given by

$$P(z) = A(z) + z^{-2p+1}A(z^{-1}), \quad (5.16)$$
$$Q(z) = A(z) - z^{-2p+1}A(z^{-1}), \quad (5.17)$$

where $A(z)$ is the linear prediction polynomial of order $p$. These polynomials are factored as

$$P(z) = (z^{-1} + 1) \prod_{i=1}^{p/2} (z^{-2} - 2z^{-1} \cos \phi_i + 1), \quad (5.18)$$
$$Q(z) = (z^{-1} - 1) \prod_{i=1}^{p/2} (z^{-2} - 2z^{-1} \cos \phi_i + 1), \quad (5.19)$$
where $\theta_i$ and $\phi_i$ are the interleaved line spectrum pair frequencies. The derivatives of the polynomials are

\[
\frac{\partial P(z)}{\partial \theta_k} = \frac{2z^{-1}\sin \theta_k}{(z^2 - 2z^{-1} \cos \theta_k + 1)}P(z),
\]

\[
\frac{\partial Q(z)}{\partial \phi_k} = \frac{2z^{-1}\sin \phi_k}{(z^2 - 2z^{-1} \cos \phi_k + 1)}Q(z),
\]

\[
\frac{\partial P(z)}{\partial \phi_k} = 0.
\]

For notation, the coefficients of $P(z)$ and $Q(z)$ are expressed as $p_i$ and $q_i$, respectively, and

\[
\frac{\partial P(z)}{\partial \theta_k} = \sum_{i=1}^{P} p_i^{(k)} z^{-i}, \quad \frac{\partial Q(z)}{\partial \phi_k} = \sum_{i=1}^{Q} q_i^{(k)} z^{-i}.
\]

Derivatives of the LP coefficients are expressed as

\[
\frac{\partial a_k}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \left( \frac{1}{2} (p_k + q_k) \right) = \frac{1}{2} \left( \frac{\partial p_k}{\partial \theta_k} + \frac{\partial q_k}{\partial \theta_k} \right) = \frac{1}{2} p_k^{(k)},
\]

\[
\frac{\partial a_k}{\partial \phi_k} = \frac{\partial}{\partial \phi_k} \left( \frac{1}{2} (p_k + q_k) \right) = \frac{1}{2} \left( \frac{\partial p_k}{\partial \phi_k} + \frac{\partial q_k}{\partial \phi_k} \right) = \frac{1}{2} q_k^{(k)}.
\]

5.2.1.2 Formant Frequencies

For the purposes of this paper, the formant frequencies are defined as the frequencies that correspond to the maxima of the LP spectrum. To provide better resolution of the formants, the spectrum can be evaluated along a circle of radius $\rho$ using

\[
A_\rho(f) = 1 - \sum_{k=1}^{P} \rho^{-2k} a_k e^{-j2\pi f_k}, \quad 0 < \rho \leq 1.
\]

The formant locations $F_k$ are the minima of the magnitude of the spectrum

\[
F_k = \arg \min |A_\rho(f)|^2.
\]

This is also equivalent to finding the roots of the derivative of $|A_\rho(f)|^2$. The relationship between the formant and the LP coefficients is

\[
\frac{\partial F_k}{\partial \theta_k} = \left( \frac{\partial |A_\rho(f)|^2}{\partial \theta_k} - \frac{\partial |A_\rho(f)|^2}{\partial f^2} \right)_{f=F_k}.
\]

Differentiating $|A_\rho(f)|^2$ with respect to frequency, $f$, yields

\[
\frac{\partial |A_\rho(f)|^2}{\partial f} = \Re \left\{ A_\rho^*(f) \frac{\partial}{\partial f} A_\rho(f) \right\},
\]

47
where the derivative of $A_p(f)$ can be expressed as

$$\frac{\partial A_p(f)}{\partial f} = j2\pi \sum_{k=1}^{P} k^2 - \frac{1}{\pi} a_k e^{-j2\pi f_k}.$$  

(5.30)

The second derivative with respect to frequency is

$$\frac{\partial^2 |A_p(f)|^2}{\partial f^2} = 2\left| \frac{\partial A_p(f)}{\partial f} \right|^2 + 2\text{Re}\left\{ A_p(f) \frac{\partial^2}{\partial f^2} A_p(f) \right\},$$  

(5.31)

where the second derivative of $A_p(f)$ is

$$\frac{\partial^2 A_p(f)}{\partial f^2} = 4\pi^2 \sum_{k=1}^{P} k^2 - \frac{1}{\pi} a_k e^{-j2\pi f_k}.$$  

(5.32)

The relationship with respect to the LP coefficients is

$$\frac{\partial^2 |A_p(f)|^2}{\partial f \partial a_k} = -2\text{Re}\left\{ e^{-j2\pi f_k} \left[ 2\pi i A_p(f) + \frac{\partial A_p(f)}{\partial f} \right] \right\}.$$  

(5.33)

### 5.5.1.3 Formant Bandwidths

The formant bandwidths can be approximated by using the relationship between the group delay and the bandwidths for single pole systems [77]. The bandwidths are approximated by

$$B_k \approx -\frac{1}{\pi} \left[ \frac{\partial \angle A_p(f)}{\partial f} \right] - \left[ 2\pi + \frac{\partial \angle A_p(f)}{\partial f} \right] \right\}_w.$$  

(5.34)

Taking the derivative with respect to the prediction polynomial coefficients yields

$$\frac{\partial B_k}{\partial a_k} \approx 2\frac{\partial^2 \angle A_p(f)}{\partial f \partial a_k} \left[ \frac{\partial \angle A_p(f)}{\partial f} \right] \left[ 2\pi + \frac{\partial \angle A_p(f)}{\partial f} \right]^{-1}.$$  

(5.35)

where

$$\frac{\partial^2 \angle A_p(f)}{\partial f^2} = -\text{Im}\left\{ \frac{e^{-j2\pi f}}{A_p(f)} \left[ 2\pi i A_p(f) + \frac{\partial A_p(f)}{\partial f} \right] \right\},$$  

(5.36)

and

$$\frac{\partial \angle A_p(f)}{\partial f} = -\text{Im}\left\{ \frac{\partial A_p(f)}{A_p(f)} \right\}.$$  

(5.37)

### 5.3.1.4 Spectral Tilt

Another quantity that one might like to modify is the spectral tilt. This tilt can be viewed as

$$T = \int_{-1/2}^{1/2} \log |A(f)| V(f) df,$$

(5.38)
where \( V(f) \) is some symmetric basis function. Again, we can take derivatives with respect to the coefficients. This expression simplifies to
\[
\frac{\partial T}{\partial \Delta_k} = \int_{-1/2}^{1/2} \frac{V(f)}{A(f)} e^{-j2\pi kf} df,
\] (5.39)

which is given by
\[
\frac{\partial T}{\partial \Delta_k} = 2^{T-1} \left[ \frac{V(f)}{A(f)} \right]^{[-k]},
\] (5.40)

which can be calculated by filtering the basis function in the time domain with the linear prediction filter. A linear tilt function can be approximated by
\[
u[n] = w[n] \sin \left( \frac{n}{2} \right) / 4,
\] (5.41)

where \( w[n] \) is some finite length and symmetric window function.

### 5.3.2 Modification Algorithm

In this section, the formant structure is denoted by \( S = [F^T, B^T]^T = [F_1, \ldots, F_N, B_1, \ldots, B_N]^T \), where \( N \) is the number of formants. The LSPs are represented by \( f = [\theta_1, \phi_1, \ldots, \theta_{N/2}, \phi_{N/2}]^T = [f_1, \ldots, f_N]^T \). For small LSP shifts, formant charges are approximated by
\[
\Delta S = \begin{bmatrix} \Delta F^T \\ \Delta B^T \end{bmatrix} \approx \begin{bmatrix} \frac{\partial S}{\partial \Delta f} \end{bmatrix} ^T \begin{bmatrix} \frac{\partial B}{\partial \Delta f} \end{bmatrix} \begin{bmatrix} \Delta F \end{bmatrix},
\] (5.42)

with the matrix \( \frac{\partial S}{\partial \Delta f} \) defined in Section 5.3.1.

The most straightforward approach to finding the new LSPs is to solve the underdetermined linear system \( \frac{\partial S}{\partial \Delta f} = \Delta S \) while minimizing \( \| \Delta f \|_2 \). Unfortunately, this can result in poor maintenance of formant bandwidths. Instead, the two-step procedure shown in Figure 5.12 is used.

The formant shift, \( \Delta f_p = [\Delta f_1, \ldots, \Delta f_N]^T \), is found from
\[
\min \sum_{k=1}^N \left( \frac{\Delta f_k - \Delta f_{k-1}}{\alpha + f_k - f_{k-1}} \right)^2, \quad \text{s.t.} \quad \frac{\partial f}{\partial \Delta f_p} = \Delta F,
\] (7.43)

where \( f_0 \) and \( \Delta f_0 \) are both defined as zero, and \( \alpha \) is an additional tuning factor. This expression minimizes the weighted change in the relative line spectrum pair locations. Since the frequencies of clustered LSPs have a greater effect on the formant bandwidths, these changes are weighted more heavily.
This is equivalent to the least squares problem

\[
\min ||D\Delta f||_2, \text{ s.t. } \frac{\partial F}{\partial f} \Delta f = \Delta F,
\]  

(5.44)

where \(D\) is the \(p \times p\) matrix defined as

\[
[D]_{ij} = \begin{cases} 
-(\alpha + f_{i+1} - f_i)^{-1}, & i = j, \\
(\alpha + f_{i+1} - f_i)^{-1}, & i = j + 1, \\
0, & \text{otherwise}.
\end{cases}
\]  

(5.45)

The bandwidth alteration, \(\Delta B\), is found from

\[
\min \|\Delta f\|_2, \text{ s.t. } \frac{\partial S}{\partial f} \Delta f = \begin{pmatrix} 0 \\ \Delta B \end{pmatrix}.
\]  

(5.46)

The modified LSPs, \(f_M\), are found by summing the shifts

\[
f_M = f + \Delta f + \Delta B,
\]  

(5.47)

followed by an enforcement of LSP ordering.

Figure 5.12: Block diagram of the formant modification algorithm.

5.3.3 Algorithm Results

To show that this algorithm performs competitively with existing algorithms, the proposed algorithm, with \(\alpha = 0\) and \(\rho = 1\), was compared with direct pole modification. The direct pole modification uses the assumption that the poles are related to the formants by

\[
p_i = r_i e^{j\phi_i}, \quad F_i = \phi_i / (2\pi), \quad B_i = -\ln(r_i) / \pi.
\]  

(5.48)

This assumption results in the following formant modification method:

\[
\phi_i^{(M)} = \phi_i + 2\pi \Delta F_i, \quad r_i^{(M)} = r_i e^{-\pi \Delta B_i}.
\]  

(5.49)
Table 5.1: Average absolute errors of formants and bandwidths with $\Delta F_1 = 150$ Hz and $\Delta B_3 = -50$ Hz.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pole Measure</th>
<th>FFT Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_{Err}$</td>
<td>$B_{Err}$</td>
</tr>
<tr>
<td>Pole Mod.</td>
<td>0.0 Hz</td>
<td>0.0 Hz</td>
</tr>
<tr>
<td>LSP Mod.</td>
<td>5.2 Hz</td>
<td>12.6 Hz</td>
</tr>
</tbody>
</table>

The two methods were tested on ten synthetic North American English vowels with $\Delta F_1 = 150$ Hz and $\Delta B_3 = -50$ Hz. The average absolute errors in the new formant locations and bandwidths are shown in Table 5.1. The pole-based measurement uses Equation 5.48 to determine the formants, while the FFT measurement uses a dense FFT calculated along the unit circle to find the formant locations and 3 dB bandwidths. The new algorithm produces pole location changes similar to the direct modification method. However, if the formant locations are defined as the frequencies of the maxima of the LP spectrum, then the proposed method actually performs better than pole modification.

An example of the algorithm operating on an actual spectrum can be seen in Figure 5.11. This shows the input and output parameters using $\Delta F_1 = 150$ Hz and $\Delta B_3 = -100$ Hz. The algorithm successfully shifts the first formant by moving the first two poles. The bandwidth alteration in the third formant is accomplished with a small change in the LSPs, while the poles are changed in a manner that would not be obvious.

Computationally, the proposed algorithm has many simple steps. First, the formant frequencies must be found. This can be accomplished in many ways, although it is only necessary to find the formants that require modification. The Jacobian terms are efficiently calculated when the expressions are simplified to real arithmetic. The rest of the algorithm consists of solving two least squares problems. The operation count, where a multiply and add is viewed as a single flop, is summarised in Table 5.2. In this analysis, $N$ is the number of formants to be modified, and $p$ is the order of the linear prediction model. The algorithm also requires the computation of $2p(N + 1)$ sinusoids to generate $\partial x/\partial \theta$ and $\partial S/\partial \theta$. 

51
### Table 5.2: Computational requirements

<table>
<thead>
<tr>
<th>Oper.</th>
<th>Flops</th>
<th>$N = 4, p = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta S/\delta a$</td>
<td>$18pN$</td>
<td>720</td>
</tr>
<tr>
<td>$\delta a/\delta f$</td>
<td>$p^2$</td>
<td>100</td>
</tr>
<tr>
<td>$\delta S/\delta f$</td>
<td>$2Np^2$</td>
<td>800</td>
</tr>
<tr>
<td>$\Delta \nu$</td>
<td>$p(N^2 + 6N)$</td>
<td>400</td>
</tr>
<tr>
<td>$\Delta \lambda$</td>
<td>$p(4N^2 + 4N)$</td>
<td>800</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>$p(5N^2 + 28N + p(2N + 1))$</td>
<td>2890</td>
</tr>
</tbody>
</table>

### 5.4 Prior Distribution

In order to complete the model of whispered speech, it is necessary to create an underlying distribution for the normal speech parameters. This can be used for developing algorithms to smooth spectral parameters and to synthesize non-existent prosodic information.

#### 5.4.1 Spectral Evolution

To reduce the estimation variance of the speech spectrum, one can define a joint distribution between the spectral parameters as they progress through time. For this purpose, the LSF trajectories are first assumed to be generated by a dynamic linear model (DLM). The LSF vector at time $n$ is $f_n$, which is modeled by

$$f_n = A f_{n-1} + w_n + F,$$  \hspace{1cm} (5.50)

$$z_n = f_n + \nu_n,$$  \hspace{1cm} (5.51)

$$w_n \sim \mathcal{N}(0, \Sigma), \quad \nu_n \sim \mathcal{N}(0, P_0^2),$$  \hspace{1cm} (5.52)

where $z_n$ is the observed spectral parameter vector that is corrupted with some random vector, $\nu_n$. This random disturbance is assumed to be a zero mean normal random variable with covariance equal to $P_n^2$, which is dependent on the estimation scheme used to calculate the spectrum from the acoustic waveform, $y_n$.

The parameters for this model were derived using a subset of sentences from the TIMIT database, which were assumed to be direct observations of the state process, $f_n$. This subset includes 700,000 sample vectors over the entire 462 speakers in the corpus. On inspection of the residual vectors, $w_n$, the data was found to be heavy-tailed and did not fit a Gaussian distribution well. However, this data is well modeled by Gaussian mixture-
Figure 5.13 contains a histogram of the first elements of $\mathbf{w}_n$. From this plot, one can see that a two-mixture Gaussian fits much better than with a single mixture.

![First LSF Statistics](image)

**Figure 5.13:** Modeling the LSF residual with Gaussian mixtures.

The dynamic linear model with Gaussian mixture plant noise can be further generalized by using a jump Markov linear system. In this case the model is given by

\[
\begin{align*}
\mathbf{f}_n &= A(\mathbf{r}_n)\mathbf{f}_{n-1} + B(\mathbf{r}_n)\mathbf{w}_n + F(\mathbf{r}_n), \\
\mathbf{z}_n &= \mathbf{f}_n + \mathbf{v}_n, \\
\mathbf{w}_n &\sim \mathcal{N}(0, I), \\
\mathbf{v}_n &\sim \mathcal{N}(0, \mathbf{P})
\end{align*}
\] (5.53) (5.54) (5.55)

where $\mathbf{r}_n$ is the state of an $M$-state discrete Markov chain with $\Pr(\mathbf{r}_n = i | \mathbf{r}_{n-1} = j) = p_{ij}$, and $A$, $B$, and $F$ are the dynamic system parameters that depend on $\mathbf{r}_n$. In this formulation, the formant differences can have dependence on the state, $\mathbf{r}_n$. Note that by constraining $p_{ij}$ and removing the dependence of $A$ and $F$ on the Markov chain, this becomes a dynamic linear model driven by Gaussian mixture noise.

Again, the parameters for this model were derived using sentences from the TIMIT database. This data was used to determine the system parameters $A$, $B$, $F$, and $p_{ij}$. In this case, finding ML estimates of the system can be calculated by a relatively straightforward extension of the Baum-Welch re-estimation algorithm described in Chapter 4.
Table 5.3: Pitch-gain models derived from DAM lists for different speakers.

<table>
<thead>
<tr>
<th>Gender</th>
<th>Speaker</th>
<th>( \tilde{p} )</th>
<th>( \alpha_p )</th>
<th>( \alpha_g )</th>
<th>( \alpha_r )</th>
<th>( \sigma_p )</th>
<th>( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>db</td>
<td>169.9 Hz</td>
<td>0.621</td>
<td>0.706</td>
<td>6.18 dB</td>
<td>10.37 Hz</td>
<td>0.579</td>
</tr>
<tr>
<td></td>
<td>jd</td>
<td>202.0 Hz</td>
<td>0.641</td>
<td>0.736</td>
<td>9.35 dB</td>
<td>9.31 Hz</td>
<td>0.819</td>
</tr>
<tr>
<td></td>
<td>ps</td>
<td>165.9 Hz</td>
<td>1.566</td>
<td>0.759</td>
<td>6.10 dB</td>
<td>9.31 Hz</td>
<td>0.421</td>
</tr>
<tr>
<td>m</td>
<td>dn</td>
<td>114.9 Hz</td>
<td>0.903</td>
<td>0.677</td>
<td>6.65 dB</td>
<td>7.53 Hz</td>
<td>0.744</td>
</tr>
<tr>
<td></td>
<td>js</td>
<td>115.4 Hz</td>
<td>0.611</td>
<td>0.842</td>
<td>6.51 dB</td>
<td>7.53 Hz</td>
<td>0.523</td>
</tr>
<tr>
<td></td>
<td>sa</td>
<td>103.1 Hz</td>
<td>0.613</td>
<td>0.743</td>
<td>5.42 dB</td>
<td>8.84 Hz</td>
<td>0.282</td>
</tr>
</tbody>
</table>

5.4.2 Pitch, Energy, and Voicing

Dynamic systems have been used for modeling fundamental frequency and energy contours in the past for intonation labeling [78] and speech synthesis [84]. In the first development of the voiced-speech synthesizer, the prior distribution of \( V \) was described by a simple DLM:

\[
\begin{bmatrix}
\nu_n \\
\rho_n \\
\end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} w_{p1} , \tag{5.56}
\]

where \( \nu_n = [n_{n-1} n_{n-2} n_{n-3} n_{n-4}] \). The transition matrix is given by

\[
\begin{bmatrix}
\rho_n \\
\rho_{n-1} \\
\end{bmatrix} = \begin{bmatrix} \alpha_p & 0 \\
0 & \alpha_g \\
\end{bmatrix} \begin{bmatrix} n_{n-1} - \tilde{g} \\
\rho_{n-1} - \tilde{\rho} \\
\end{bmatrix} + w_\nu \begin{bmatrix} \tilde{g} \\
\tilde{\rho} \\
\end{bmatrix}, \tag{5.57}
\]

and the noise covariance matrix is

\[
w_\nu = \mathcal{N}_2 \begin{pmatrix} \sigma^2_n & \rho \sigma_n \sigma_g \\
\rho \sigma_n \sigma_g & \sigma^2_g \\
\end{pmatrix} . \tag{5.58}
\]

When the Kalman filter is constructed, the pitch is essentially a lowpass filtered version of the gain. In order to maintain reasonable quality, it is necessary to make the parameters for this system speaker dependent. To train these models, gain and pitch data was taken from the full voice DAM lists for each speaker and used to create maximum likelihood estimates. The actual parameters are included in Table 5.3. As one would expect, the average pitch is much higher for the female speakers. In general, most of the other parameters are similar across speakers, with the exception of correlation, \( \rho \), which ranged from very strong in speaker \( \beta \) to very weak in \( \alpha \). The value of \( \rho \) is a good indicator of how well the pitch can be predicted from the gain.

Since this approach produced reasonable results, the next step is to train jump Markov linear systems to explore the relationships between the pitch, gain, and voicing parameters.
Again, the usual JMLS formulation is made:

\[
\begin{align*}
\mathbf{x}_n &= A(r_n)\mathbf{x}_{n-1} + B(r_n)w_n + F(r_n), \\
\mathbf{z}_n &= G(r_n)\mathbf{x}_n + D(r_n)u_n + G(r_n), \\
\mathbf{v}_n &= \begin{cases} 
[0 0 0 0 0]^T, & r_n = 1 \\
[1 1 1 1 0 0]^T, & r_n > 1
\end{cases}, \\
\mathbf{x}_n &\triangleq [a_n, g_n]^T,
\end{align*}
\]

(5.59) (5.60) (5.61) (5.62)

where the state vector, \(\mathbf{x}_n\), is again defined as a vector that contains both the pitch and gain of the voiced speech. In this case the observation, \(\mathbf{z}_n\), is a vector composed of the whispered gain and the first \(K\) LPC-based mel-cepstral coefficients derived from the whispered speech.

The discrete state, \(r_n\), is arranged so that \(r_n = 1\) represents unvoiced speech and all of the other states represent voiced segments. In order to train this model, aligned whispers and voiced speech are created again using the method described in Figure 5.10. Training this model involves some special considerations. First, unvoiced frames are all assigned to have \(r_n = 1\), while the voiced frames are allowed to jump between the other remaining states. In addition, the pitch is undefined in the unvoiced states, so the state matrices are constrained so that there is no coupling between pitch and gain in these periods. This configuration allows for some flexibility in the number of states used and the number of cepstral observations used from the whispered speech. Due to the variation between speakers, these model parameters are speaker dependent and are trained again on the DAM list pairs. In order to reduce the number of parameters, the matrix \(D(r_n)\) is constrained to be diagonal.

### 5.5 Breath Noise

In whispered speech, noise sources are not confined to background sources. Due to the low signal levels and increased air flow during whispering, breath noise becomes a significant source of noise. This noise is created when the airflow from the speaker's mouth directly impacts the microphone. This noise is intermittent and is highly dependent on the microphone and speaker. In general, the noise occurs in regions of greater aspiration such as stop consonants. In Figure 5.14, one can see an example of this breath noise in the phrase "he
was placed too far back." This noise is present in the word "he" and the stop consonants /p/ and /t/. The bulk of the energy in these noise bursts occurs beneath 500 Hz, where very little signal energy exists in whispered speech.

Figure 5.14: Example spectrum of whispers with breath noise, "He was placed too far back"

The simplest solution for removing this noise is to create a simple highpass filter with a roll-off frequency around 500 Hz. However, it is preferable to find something that has a better theoretical basis. To facilitate this, the locations of whispered speech and breath noise-corrupted speech in the DAM lists were hand labeled. The averaged power spectra for whispered speech and noisy whispers are shown in Figure 5.15. These spectra were divided to construct the Wiener filter response shown in Figure 5.16. As expected, this filter primarily attenuates the signal at frequencies under 1000 Hz. This filter was applied directly to the whispered speech, but it yielded audio that sounded too thin. Ideally, the Wiener filter should only be applied to regions with breath noise. By using the knowledge that breath noise occurs during some portions of the waveform, yields the spectral estimator

\[ \hat{S}(\omega; t) = (1 + (W(\omega) - 1) \Pr[\text{Breath}|Y(\omega; t), \ldots, Y(\omega; 0)]) Y(\omega; t), \]  

(5.63)

where \( W(\omega) \) is the Wiener filter response. In order to calculate \( \Pr[\text{Breath}|Y(\omega; t), \ldots, Y(\omega; 0)] \), a simple two-state HMM was trained using MFCCs from the labeled data set. Calculating the probability of breath noise presence is a straightforward implementation of the HMM forward recursion.

As an example of the application of this algorithm, the waveform shown in Figures 5.14
is processed by the proposed algorithm. The results are included in Figure 5.17, where the original spectrum is in the top plot. The probability of breath noise is shown in the second plot. This value is accurate for the most part, although there is a false alarm at 0.9 seconds during the release of the /d/ in "placed." Finally, the output spectrogram is shown in the bottom plot, where the severity of the breath noise bursts has been reduced.
Figure 5.16: Amplitude response of the Wiener filter used to remove breath noise from whispered speech.

Figure 5.17: Example of breath noise removal in whispered speech.
5.6 Results

5.6.1 Objective Results

5.6.1.1 Spectrum Estimation

In order to estimate the spectrum, the parameters for the shifting function, \( S(k_0, F_{\text{max}}, F_k, T) \), must be calculated. To do this, the alignment procedure described in Section 5.2.2 is followed by several configurations. In all of the formant shifting modeling methods, the whispered speech has the breath noise manually removed and the line spectrum pair frequencies are smoothed. In addition, the shift estimation algorithm is iterated three times so that after each iteration, the speech is realigned using the latest values of \( F_{\text{max}}, F_k, \) and \( T \). In Table 5.4, the results using several different algorithm configurations are listed. In None, all three parameters are set to zero, while in Tilt and Formant Shift, the \( F_{\text{max}} \) and \( T \) settings are forced to zero respectively. In the last two tests, all of the parameters are free for optimization. However, in the Combination Optimized test, the shifting algorithm is applied on all of the tie frames, while in Voice Optimizer, it is applied only to voiced frames.

In general, the effect of iterating the procedure to improve the alignment has little effect on the score. In fact, the alignments change very little from iteration to iteration. The effect of modifying the tilt and the formant shifting is similar, with even greater improvement when they are used together. Finally, the performance is also improved slightly by only modifying voiced segments of speech. The optimal parameters for voice dependent shifting is found to be \( T = 4.8 \text{ dB}, F_{\text{max}} = 300 \text{ Hz}, \) and \( F_k = 700 \text{ Hz} \) for the shift bend point. If voicing is not known, the optimal point is \( T = -3.2 \text{ dB}, F_{\text{max}} = 300 \text{ Hz}, \) and \( F_k = 700 \text{ Hz} \).

In the next experiment, the effects of smoothing and different breath removal algorithms are compared. These results are contained in Table 5.5. For breath removal, three different methods were used: taking no action, automatic breath removal, and manual breath removal where the breath noise is manually labeled. Strangely enough, the algorithms for breath noise removal actually increase the average spectral distance when no formant shifting is performed. The reason for this is that breath noise tends to increase the spectral tilt of whispered speech, which makes it slightly closer to normal speech. However, when different formant shifting algorithms are used, the breath removal algorithms improve the
Table 5.4: Effect of the number of iterations on the median Itakura distance between whispered and normal speech.

<table>
<thead>
<tr>
<th>Spectrum Enhancer</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>None</td>
<td>1.285</td>
</tr>
<tr>
<td>Tilt</td>
<td>1.073</td>
</tr>
<tr>
<td>Formant Shift</td>
<td>1.019</td>
</tr>
<tr>
<td>Combination Optimized</td>
<td>0.847</td>
</tr>
<tr>
<td>Voice Optimized</td>
<td>0.742</td>
</tr>
</tbody>
</table>

Table 5.5: Effect of different spectral modification algorithms on the median Itakura distance between whispered and normal speech.

<table>
<thead>
<tr>
<th>Original Spectral Estimator</th>
<th>Spectrum Modification Algorithm</th>
<th>No Shift</th>
<th>Simple Shift</th>
<th>V. Depend.</th>
<th>Auto. V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smoothed</td>
<td>Breath Noise</td>
<td>1.092</td>
<td>0.8659</td>
<td>0.7614</td>
<td>0.7806</td>
</tr>
<tr>
<td></td>
<td>Auto. Debreath</td>
<td>1.176</td>
<td>0.8350</td>
<td>0.7417</td>
<td>0.8011</td>
</tr>
<tr>
<td></td>
<td>Manual Debreath</td>
<td>1.286</td>
<td>0.8504</td>
<td>0.7460</td>
<td>0.7867</td>
</tr>
<tr>
<td>Not smoothed</td>
<td>Breath Noise</td>
<td>1.102</td>
<td>0.8767</td>
<td>0.7067</td>
<td>0.7925</td>
</tr>
<tr>
<td></td>
<td>Auto. Debreath</td>
<td>1.190</td>
<td>0.7522</td>
<td>0.7522</td>
<td>0.8188</td>
</tr>
<tr>
<td></td>
<td>Manual Debreath</td>
<td>1.297</td>
<td>0.7474</td>
<td>0.7474</td>
<td>0.7996</td>
</tr>
</tbody>
</table>

Performance slightly. The smoothing algorithms create the opposite effect. In the raw waveform, smoothing the LSFs decreases the spectral distance by 0.1, while for the processed waveforms, this difference is far smaller.

The formant shifting algorithms consistently decrease the spectral distance. This is further improved by taking advantage of voicing information. In reality, the voicing information is not directly available, so an automatic voicing detector must be used. In the last column of Table 5.5, the spectral distance from using this algorithm is tabulated. The results are not as good as the method that uses the actual voicing information, but it is still better than the voice-independent formant modification.

5.6.1.2 Pitch and Voicing

Like the spectral models, the pitch and voicing models have a variety of possible configurations. Since there are no accepted measures of pitch quality, the mean square error is used for the following tests. Also, since the pitch dynamics vary among the speakers, each speaker must be modeled independently. This yields three female speakers, who are listed

60
### Table 5.6: Mean square errors for various pitch estimators.

<table>
<thead>
<tr>
<th>Gnd</th>
<th>Spkr</th>
<th>Mono</th>
<th>Orig</th>
<th>$g$</th>
<th>$g_{l0}$</th>
<th>$g_{l0,2s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>db</td>
<td>db</td>
<td>240.5</td>
<td>226.3</td>
<td>273.0</td>
<td>285.4</td>
<td>297.3</td>
</tr>
<tr>
<td></td>
<td>jd</td>
<td>316.1</td>
<td>270.6</td>
<td>237.7</td>
<td>257.5</td>
<td>236.0</td>
</tr>
<tr>
<td></td>
<td>ps</td>
<td>229.8</td>
<td>210.9</td>
<td>260.6</td>
<td>223.9</td>
<td>226.2</td>
</tr>
<tr>
<td>da</td>
<td>da</td>
<td>216.5</td>
<td>199.4</td>
<td>176.2</td>
<td>190.5</td>
<td>192.4</td>
</tr>
<tr>
<td></td>
<td>js</td>
<td>241.7</td>
<td>263.5</td>
<td>304.5</td>
<td>330.6</td>
<td>414.5</td>
</tr>
<tr>
<td></td>
<td>sa</td>
<td>180.8</td>
<td>180.3</td>
<td>167.7</td>
<td>171.3</td>
<td>247.3</td>
</tr>
<tr>
<td>Average</td>
<td>237.5</td>
<td>228.2</td>
<td>235.9</td>
<td>243.2</td>
<td>285.7</td>
<td>257.5</td>
</tr>
</tbody>
</table>

as db, jd, and ps, and three male speakers, who are called da, js, and sa. Due to the limited data, the pitch and voicing models are evaluated using a jackknife procedure. For each DAM sentence tested, the other 14 sentences are used to train the model. In this way, all of the data can be tested while still keeping the training and testing data separate.

The mean squared errors for the pitch estimators are shown in Table 5.6. The Mono entry simply uses the mean pitch, $\bar{p}$, as the pitch estimate, while the Orig column uses the estimator based on the method described in Equations 5.66-5.58. The other estimators are based on the the JMLS description. In these, the name refers to the nature of the observation vector: $g$ signifies the gain parameter, while the observation vector in $g_{l0}$ consists of the gain and the first $N$ LPCC coefficients. Finally, adding $2s$ to the description means that two states are used to describe the voiced speech. Interestingly, except for the monotone speech, the mean square increases with the complexity of the model. Several examples of these pitch estimates are shown in Figure 5.18. One can see that none of these methods provide exact estimates of the pitch. However, the estimates do have similar characteristics to the true contour, and in many cases, follow the same trend.

The evaluation of the voicing detector is a little more straightforward, since it is a detection problem and performance can be interpreted with tools such as the receiver operating characteristic (ROC) curve. Various methods are tested again by listing the probability of voicing detection at the operating point where $\text{isis}$ is equal to the probability of false alarm. The naming convention is the same as the pitch estimators, with the exception of HMM, which uses a simple two-state HMM with $g_{l0}$ features to differentiate between voiced and unvoiced speech.

61
Figure 5.18: Example phrase “Some diphthongs have no sound” with four pitch estimators. The Mono Pitch estimator uses the speaker mean, while the Orig Method and the other estimators use the JMS scheme described in Section 5.4.2.

unvoiced speech. One can see from the results in Table 5.7 that the performance is very speaker dependent, with all of the algorithms working better on the male speakers than the female speakers. In contrast to the pitch estimators, the voice detection performance improves with the model complexity.

The ROC curves provide more information about how an estimator performs. As an example, the ROC curve for g13.s is given for each speaker, and for the composite performance in Figure 5.19. One can see that for this estimator, the performance for most of the speakers is good, with the exception of speaker d8, who performs very poorly for all of the proposed methods. Most importantly, the ROC curves are asymmetric; it is easier to drive the probability of voice detection close to one than it is to eliminate false alarms. In
Table 5.7: Equal error points for various voice detectors.

<table>
<thead>
<tr>
<th>Gender</th>
<th>Speaker</th>
<th>g13</th>
<th>g110</th>
<th>g18.26</th>
<th>g180.26</th>
<th>hmm</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>db</td>
<td>0.64</td>
<td>0.64</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>jd</td>
<td>0.67</td>
<td>0.71</td>
<td>0.63</td>
<td>0.69</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>ps</td>
<td>0.70</td>
<td>0.72</td>
<td>0.69</td>
<td>0.72</td>
<td>0.71</td>
</tr>
<tr>
<td>m</td>
<td>dn</td>
<td>0.68</td>
<td>0.76</td>
<td>0.73</td>
<td>0.76</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>js</td>
<td>0.69</td>
<td>0.75</td>
<td>0.71</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>sa</td>
<td>0.72</td>
<td>0.74</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.667</td>
<td>0.728</td>
<td>0.688</td>
<td>0.726</td>
<td>0.732</td>
</tr>
</tbody>
</table>

In addition, it typically sounds worse to have voiced speech classified as unvoiced. For this reason, the voice detection threshold is lowered so that the probability of detection is close to 0.95, even though this increases the false alarm rate above 0.5.

Figure 5.19: Receiver operating characteristic curves for the JMLS-based voicing detector.
5.6.2 Subjective Results

Before starting formal listening tests, informal tests were conducted to determine which pitch tracker to use for the final evaluation. This test consisted of six listeners, who were presented two sentences from the following pitch trackers: Orig/hmm, g, gl5, gl10, and gl5,2s. The listeners all felt that gl3, gl10, and gl5,2s were of better quality than the Orig/hmm and g pitch and voicing estimators. The majority of listeners found that gl5,2s sounded slightly better than gl3. In general, the listeners had difficulty finding any differences between gl3 and gl10. For this reason, the gl5,2s pitch estimator was chosen for the final experiments.

In order to test the results of these algorithms, subjective listening DMOS tests were conducted. In these tests, listeners were asked to compare two speech samples that contain the same phrase spoken by the same speaker. These phrases were constructed using pairs of whispered and phonated DAM sentences that were aligned according to the method described in Figure 5.10. The reference phrase consisted of the aligned voiced speech, synthesized using the MEELP codec. The sample waveform was synthesized from a hybrid of parameters derived from the whispered speech, and true parameters from the normal speech.

The parameters altered in the DMOS included the pitch, voicing, and LPC spectrum. For the pitch and voicing, these included the true voiced parameters and parameters synthesized from the whispered waveform. For the spectral parameters, the true voiced spectrum, the whispered spectrum, and the enhanced spectrum were used for testing. The results of this experiment are listed in Table 5.8.

The results of this test were unexpected. First, the effect of the spectrum, pitch, and voicing synthesis were very different. Surprisingly, the effect on the DMOS for synthesizing the pitch was only 0.5 points, with the resulting speech having an average score of 4.27, which means that most people could not tell the difference, or found the change to net be annoying. The automatic voicing synthesizer did not fare quite as well with a 0.87 drop in performance. In general, these differences are more perceivable, but still not annoying. On the other hand, switching the spectral parameters to their whispered counterparts caused a major drop of 2.0 points. Unfortunately, the spectrum enhancer had a negligible effect on
the subjective listening tests, despite the fact that it decreased the average spectral distance to the true voiced speech. From this test, it is apparent that the primary focus of further research should be on improving the spectral modeling between whispered and normal speech. This does show promise for algorithms that can synthesize normal speech, since the seemingly difficult problem of synthesizing prosodic features from whispered speech is feasible.
CHAPTER 6

ENHANCEMENT II: NOISE REMOVAL

While noise robustness is very important for any speech application, its importance is increased by the simple fact that the signal-to-noise ratio is lowered by 20 dB when one whispers. For this reason, several noise robustness schemes are investigated. As described in the background chapter, most speech enhancement systems are based on either the subspace assumption or the autoregressive assumption. Intuitively, one would assume that the autoregressive assumption class of algorithms would work better for whispered speech, since the signal is excited by broad-band noise that fits the iid Gaussian assumption more closely, and is not confined to any linear subspace. To justify the focus on these methods in this chapter, an illustrative example is given next.

Two different clean speech waveforms are given in this example. The first is a primarily voiced phrase from a female speaker, while the second is a whispered sample. The spectrograms of these phrases are shown at the top of Figures 6.1 and 6.2, respectively. Next, Gaussian white noise was added with 0 dB average SNR, yielding the waveforms shown in the second plots of each figure. The spectral amplitudes of the two methods were estimated using two different algorithms: a reference implementation of a log-spectrum amplitude estimator [2], and a Gibbs sampler-based method that uses the autoregressive assumption. More details of the latter method is provided in this chapter. In the voiced waveform, one can see that the Ephraim-Malah estimator provides clearer resolution of the spectral peaks, although the overall spectral shape of the estimated waveforms is similar. However, one can see in the whispered speech that the Gibbs sampler estimator is more effective at finding the correct spectrum.
Figure 6.1: Comparison of parametric and non-parametric spectrum estimators - voiced sample “Why were you away a year Roy” 0 dB SNR (female)
Figure 6.2: Comparison of parametric and non-parametric spectrum estimators - whispered sample "Shall we play a game?" 0 dB SNR (male)
It is important to question the validity of any statistical model. The autoregressive model of speech fits whispered vowels quite well; the excitation energy is well modeled by Gaussian noise, and the vocal tract transfer functions can be modeled by acoustic tubes. However, it is important to note that even during steady vowels, zeros do exist in whispered vowels and the turbulent flow exciting the tract is not white. Other problems shared with normal speech include nasalization, fricatives, and stops that are definitely not from Gaussian autoregressive processes. Despite these problems, the LPC model has been used successfully in speech applications, especially low bit-rate coding. Since this model fits whispered speech more closely in many situations, it provides a tractable starting point for investigating new algorithms for speech enhancement.

6.1 Parameter Estimation

Recent publications have proposed modeling the trajectories of the line spectrum frequencies (LSF) with Gaussian mixture models (GMMs) for coding [37] and estimation of missing parameters [63]. By combining this model of the LSFs with the autoregressive enhancement methodology, the new framework for parameter estimation and noise reduction algorithms outlined in Figure 6.3 is created.

This system consists of two steps: a block estimator and a spectral smoother. The block estimator generates the posterior distribution of the spectral parameters for each speech frame. This estimator assumes that the waveform frames, \( y_n \), are conditionally independent given the spectral parameters. There are different algorithms that can be used for each phase. For example, the different block estimation algorithms are all referred to as \( x \E \), where \( x \) is the algorithm choice. Similarly, the assumptions about the noise are given by \( x \N \), and the type of LSF smoothing used is given by \( x \S \).

6.1.1 Single Block Estimators

From Section 5.1, the observed signal blocks, \( y_n \), are assumed to be the sum of a Gaussian autoregressive speech, \( s_n[t] \), and noise, \( u_n[t] \), signals. For some schemes, it useful to model
the noise as autoregressive as well:

\[ v_n[t] = \sum_{k=1}^{a} \theta_k v_n[t-k] + w_n[t] \quad (6.1) \]

\[ w_n \sim \mathcal{N}(0, \sigma^2_n) \quad (6.2) \]

In this section, different methods for determining the parameter set, \( \theta \triangleq (a, \sigma^2) \), are proposed. The parameter set is dependent on the structure of the noise PSD, yielding \( \theta \triangleq (a, \sigma^2, b, \sigma^2_b) \) when the noise is assumed to be autoregressive, and \( \theta \triangleq (a, \sigma^2, P_s(\omega)) \) for nonparametric noise models. For further processing by the smoother, it is desirable to know the full distribution of the posterior, \( \theta_n | y_n \). In particular, the line spectrum pairs, \( f_n \), are assumed to be normally distributed by \( f_n | y_n \sim \mathcal{N}(\theta_n, P_f^2) \).

6.1.2 Maximum-Likelihood Approach (ML-E)\( E \)

One method for determining the parameter, \( \theta \), from the observed waveform is maximum likelihood (ML) estimation. The form of these estimators varies depending on the assumptions made. In this section, the speech is assumed to be equal to zero outside of the frame. For this reason, the data must be windowed first. If there is no noise present, the solution that maximizes the likelihood of the data is the autocorrelation method. Estimation
of these parameters can also be accomplished by processing the data in the frequency domain. When this is done, the frequency components are assumed to be decorrelated by the Fourier transform [32]. This is usually justified by the fact that the correlation of the Fourier coefficients approach zero as the analysis window length approaches infinity [28]. In the frequency domain, the likelihood of the speech frame becomes

\[
p(y|\theta) = \prod_{\omega} p(Y(\omega)|\theta) = \prod_{\omega} \sqrt{Y(\omega) + \frac{\sigma^2}{1 + \sum_{k=1}^{\infty} \text{exp}(-j\omega k)}} + P_\nu(\omega),
\]

If \( P_\nu(\omega) \) is equal to zero, then this can be maximized by taking the inverse Fourier transform of \(|Y(\omega)|^2\) to get the autocorrelation sequence and solving the Yule-Walker equations [32].

Unfortunately, there is no closed form solution for the maximum likelihood estimate of autoregressive signals in additive noise. However, the iterative expectation-maximization (EM) algorithm generates a sequence of parameters, \( \theta^{[n]} \), that converges to a maximum on the likelihood surface [70]. This method maximizes the expected likelihood of the complete data, \( x = \{x, v\} \), at each iteration of the algorithm. The iteration is given by

\[
\theta^{[n+1]} = \arg \max_{\theta} Q(\theta|\theta^{[n]}),
\]

where \( Q(\theta|\theta^{[n]}) \) is often referred to as the auxiliary function. The algorithm consists of two steps: the expectation (E) step that uses the previous iterations parameter estimates to estimate new values for the complete data, and the maximization (M) step that calculates new parameters to maximize the likelihood of the updated complete data.

The E step of this iteration can be accomplished both in the time domain and in the frequency domain. In the time domain, a fixed interval Kalman smoother [83] can find the posterior distribution, \( s, v(y, \theta^{[n]}) \), if \( v_0 \) is autoregressive. However, this smoother requires computations on the order of \( T^2 \), and places an autoregressive constraint on the noise spectrum.

To derive the frequency domain E step, one needs the distribution for each speech spectral component,

\[
S(\omega)Y(\omega), \theta^{[n]} \sim N\left(\frac{P_B(\omega)}{P_B(\omega) + P_\nu(\omega)}, \frac{P_B^R(\omega)}{P_B(\omega) + P_\nu(\omega)} \right),
\]

71
where the speech power spectrum is
\[ P_N^\text{SNR}(\omega) = \frac{\sigma_2^2|\theta|^2}{1 + \sum_{l=1}^L \sigma_l^2|\theta|^2 e^{-j\omega l}}. \]  
(6.7)

The squared spectral amplitude is estimated by
\[ \mathbb{E} \left[ |S(\omega)|^2 |Y(\omega), \theta| \right] = \mathbb{E} \left[ |S(\omega)|^2 Y(\omega), \theta| \right]^2 + \text{Var} \left[ S(\omega)y(\omega), \theta| \right] \]
\[ = \frac{\sigma_1^2 (Y^2(\omega) + P_N^\text{SNR}(\omega)) + \sigma_2^2 (A(\omega) P_N^\text{SNR}(\omega))}{(\sigma_1^2 + P_N^\text{SNR}(\omega))^2}. \]  
(6.8)

Similarly, the noise can be estimated by
\[ \mathbb{E} \left[ |V(\omega)|^2 |Y(\omega), \theta| \right] = \frac{|A(\omega)|^2 P_N^\text{SNR}(\omega) Y(\omega)|^2 + \sigma_2^2 (A(\omega) P_N^\text{SNR}(\omega)) + \sigma_2^2 (A(\omega) P_N^\text{SNR}(\omega))}{(\sigma_1^2 + P_N^\text{SNR}(\omega))^2}. \]  
(6.9)

With the conditional expectations calculated, the conditional likelihood must be maximized. This is accomplished by calculating the conditional autocorrelation function
\[ r_{nm}^{[k+1]} = \frac{1}{L} \sum_{\omega} \mathbb{E} \left[ |S(\omega)|^2 |Y(\omega), \theta| \right] e^{j\omega m}. \]  
(6.10)

The auxiliary function is then maximized by solving the Yule-Walker equations to get \( r_{nm}^{[k+1]} \) and \( \sigma_{nm}^{[k+1]} \) from \( r_{m}^{[k+1]} \). These iterations continue until some stopping criterion is met.

An example of the algorithm operating on a block of noisy whispered speech is included in Figure 6.4. In this example, the spectrum is flattened by the added white noise. However, the EM algorithm is able to tighten the formants and increase the depth of the spectral valleys.

6.1.2.1 Covariance-Based Maximum Likelihood Approach (ML-COV:E)

In many communications systems, the computational resources available may be limited. For example, if one is using a mobile transmitter, one may not be able to perform these enhancement algorithms. If the noise-corrupted spectrum is transmitted, then this is equivalent to transmitting the sample covariance of the noisy speech. If the receiver has more resources, it can still attempt to enhance the spectrum. However, unlike in the estimation of clean speech, the first \( p+1 \) elements of the sample covariance function are not a sufficient statistic, so estimates based only this information will be suboptimal. However, one can still
create an ML estimate of the spectrum by viewing the sample covariance as incomplete data and finding the conditional expectation of the speech and noise covariances. This algorithm is referred to as ML-COY-E. The definition of the sample covariance is given by

\[ [c_{yi}] = \frac{1}{N-1} \sum_{t=1}^{N} y(t)y^*(t-l) \]

\[ = \frac{1}{N-1} \sum_{t=1}^{N} (s(t)s^*(t-l) + s(t)v^*(t-l) + v(t)s^*(t-l) + v(t)v^*(t-l)) \]

\[ = [c_s]_l + 2[c_{sv}]_l + [c_v]_l. \]  

(6.11)

By utilizing the asymptotic distribution of these covariance estimates [4],

\[ c_s \sim \mathcal{N}(c_s^{[0]}, P_s^{[0]}), \quad c_v \sim \mathcal{N}(c_v^{[0]}, P_v^{[0]}), \quad c_{sv} \sim \mathcal{N}(0, P_{sv}^{[0]}), \]  

(6.12)

\[ [c_s^2]_l = \int_{-\pi}^{\pi} \cos(\omega)P_s^{[0]}(\omega)d\omega, \]  

(6.13)

\[ [P_{sv}]_l = \frac{4\pi}{N} \int_{-\pi}^{\pi} \cos(\omega)\cos(\omega)L_s^{[0]}(\omega)d\omega. \]  

(6.14)

The multiple integrations required to calculate the covariance matrix can be efficiently approximated using a single FFT by

\[ [P_s^{[0]}]_l = \frac{4\pi}{N} \int_{-\pi}^{\pi} \cos(\omega(t-l))P_s^{[0,2]}(\omega)d\omega + \int_{-\pi}^{\pi} \cos(\omega(t-l))P_s^{[0,2]}(\omega)d\omega \]

\[ = \frac{4\pi}{N} (p_s[t-l] + p_s[t+l]), \]  

(6.15)
where $g_t[\theta]$ is the inverse Fourier transform of $P^{[	heta]}(\omega)$. Using these distributions, the posterior can be calculated as

\[
\begin{align*}
    f(c_t, c_{t-1}, \theta|\theta) &\propto f(c_t, c_{t-1}, \theta|\theta) \\
    &\propto \mathcal{N} \left( c_t | c_{t-1} + c_{t-1}^{[\theta]}, P^{[\theta]} + 2P^{[\theta]} \right) \mathcal{N} \left( c_{t-1} | c_{t-1}^{[\theta]}, P^{[\theta]} \right) \\
    &\propto \mathcal{N} \left( \left( P^{[\theta]} + 2P^{[\theta]} + P^{[\theta]} \right)^{-1} \left( P^{[\theta]} + P^{[\theta]} \right) c_t - c_{t-1}^{[\theta]} \right) \left( P^{[\theta]} + 2P^{[\theta]} + P^{[\theta]} \right)^{-1} \left( P^{[\theta]} + P^{[\theta]} \right) c_{t-1}^{[\theta]} \\
    &= \mathcal{N} \left( c_t | c_{t-1} + c_{t-1}^{[\theta]}, P^{[\theta]} + 2P^{[\theta]} + P^{[\theta]} \right). \tag{6.16}
\end{align*}
\]

Since the quantities of interest are the conditional autocorrelations, this yields the expectation step

\[
\begin{align*}
    r_t^{[k+1]} &= \left( P^{[\theta]} + 2P^{[\theta]} + P^{[\theta]} \right)^{-1} \left( P^{[\theta]} + P^{[\theta]} \right) c_t - c_{t-1}^{[\theta]} \tag{6.17} \\
    c_t^{[k+1]} &= \left( P^{[\theta]} + 2P^{[\theta]} + P^{[\theta]} \right)^{-1} \left( P^{[\theta]} + P^{[\theta]} \right) c_{t-1}^{[\theta]} \tag{6.18}.
\end{align*}
\]

In the maximization step, the auxiliary function, given by

\[
Q(\theta|\theta^{[k]}) = -\frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \left[ r_t^{[k]} - 2r_t^{[k]} e_t^{[k]} + a^2 R_t^{[k]} e_t^{[k]} \right] \tag{6.19}
\]

is maximized with respect to $\theta$. The new parameters are given by

\[
\begin{align*}
    a^{[k+1]} &= \left( R_t^{[k]} \right)^{-1} r_t^{[k]} \tag{6.20} \\
    (c_t^2)^{[k+1]} &= \frac{1}{N} \left[ r_t^{[k]} - \left( a^{[k+1]} \right)^T r_t^{[k]} \right]. \tag{6.21}
\end{align*}
\]

These parameters are used for the next iteration.
6.1.2.2 Noise Modeling

The EM algorithm can also be used to form MAP estimates when the prior has a convenient form [18]. This can be especially useful when one has incomplete knowledge of the noise statistics. For example, if the noise floor is estimated by averaging over \( K \) independent frames, then the prior is given by

\[
\sigma_n^2 \sim IG(K/2-1, K\tilde{\sigma}_n^2/2),
\]

(6.22)

where \( IG \) is the inverted gamma distribution, which is given by

\[
p_{IG}(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-(\alpha+1)} \exp(-\beta x).
\]

(6.23)

By varying the value of \( K \) and setting the variance, \( \tilde{\sigma}_n^2 = 1 \), a family of pdfs are plotted in Figure 6.5. As expected, the pdf decreases in width as the number of samples, \( K \), increases.

With this prior distribution, the noise variance update becomes

\[
(\sigma_n^{(k+1)}) = \frac{1}{N+K} \left( K\tilde{\sigma}_n^2 + \sum_{i=0}^{N} \hat{y}^i \hat{y}^i - (y^{(k+1)})^T y^{(k+1)} \right).
\]

(6.24)

![Figure 6.5: Examples of the inverted gamma distribution with \( \tilde{\sigma}_n^2 = 1 \).](image)

Similarly, one can use a nonparametric model of \( R_s(\omega) \), which is estimated using a smoothed periodogram with \( K \) frames. This yields the distribution

\[
R_s(\omega) \sim IG \left( \frac{K}{2} - 1, \frac{K}{2} \tilde{R}_s(\omega) \right).
\]

(6.25)
where $P_\nu(\omega)$ is the estimated spectrum. The noise update then becomes

$$P^{k+1}_\nu(\omega) = \frac{1}{K+1} \left( KP_\nu(\omega) + E \left[ |Y(\omega)|^2 \left| Y(\omega), d\theta_0 \right| \right] \right).$$

(6.26)

In addition, one can also simply assume that any of the parameters is known. If no assumptions are made about the noise, this is denoted by NONE:N. If a smoothed periodogram is available, the estimation algorithm is called P-Y. If the spectral shape of the noise is known in the form of $b$, but only partial knowledge of the variance is known, as in (6.22), this is called B-N. If the noise statistics are known, the method is called SIG-B-N.

### 6.1.2.3 Estimating LSF Variance

One problem with ML estimation is that the variance of the estimator is unknown. However, the variance can be approximated by appealing to the asymptotic efficiency of ML estimators. If the number of samples is large enough, the variance of the estimator approaches the Cramér-Rao lower bound. To find this bound, one starts with the spectral density of the block,

$$P_\nu(\omega) = P_0(\omega) + P_\nu(\omega) = \frac{\sigma^2}{|A(\omega)|^2} + P_\nu(\omega) = \frac{\sigma^2}{|A(\omega)|^2} + \frac{|A(\omega)|^2 P_\nu(\omega)}{|A(\omega)|^2}.$$  

(6.27)

Using the equation for asymptotic bounds of stationary Gaussian processes [52], the Fisher information matrix, $I(\alpha)$, is given by

$$I(\alpha) = \left. \frac{\partial \ln P_\nu(\omega)}{\partial \alpha_k} \frac{\partial \ln P_\nu(\omega)}{\partial \alpha_l} \right|_{\omega} \, d\omega$$

$$= N \int_{-\pi}^{\pi} \left( \frac{P_\nu(\omega)}{P_\nu(\omega)} \right)^2 \frac{P_\nu(\omega)}{\sigma^2} e^{j\omega t} d\omega$$

$$+ N \int_{-\pi}^{\pi} \left( \frac{P_\nu(\omega)}{\sigma^2} \right)^2 \frac{1}{|A(\omega)|^2} e^{-j\omega t} dt.$$  

(6.28)

In the absence of noise, $P_\nu(\omega) = P_\nu(\omega)$, and the expression simplifies to the well known bound

$$I(\alpha) = \frac{N}{\sigma^2} R_\nu.$$  

(6.29)

However, when noise is present, the integrations in Equation 6.28 can be approximated by two IFPTs. This can be used to determine a lower bound on the LSF covariance, $C(\Gamma)$, by

$$C(\Gamma) = \left( \frac{\partial \Gamma}{\partial \Gamma} I(\alpha) \frac{\partial \Gamma}{\partial \Gamma} \right)^{-1} \geq 0,$$  

(6.30)

76
where $\partial a / \partial t$ can be calculated as in Section 5.3.1.

6.1.2.4 Local Maxima in Likelihood Surface

Another issue with maximum likelihood estimation in noise is that, unlike ML estimation without noise, the surface can have multiple local maxima. In addition, the EM algorithm does not guarantee convergence to the global maximum, so care should be taken when choosing the initial estimates. An example of data with three local maxima is shown in Figure 6.6. The observed data is sampled from an AR(4) process with two strong resonances that has been corrupted by additive white noise. From this data, the EM algorithm finds the maximum likelihood AR(2) process. With three different sets of initial conditions, there are three distinctly different solutions: two with the estimated resonance on one of the true resonances, and one with a relatively flat spectrum. For further analysis, the likelihood surface is plotted in Figure 6.7. In order to simplify the surface to two dimensions, the log-likelihood is maximized over $\sigma^2$:

$$L(l_1, l_2) = \max_{\sigma^2} \log p(y | \sigma^2, l_1, l_2). \quad (6.31)$$
In the left plot, the background noise assumption is made, which results in the three distinct maxima found in Figure 6.6. In the right plot where no noise is assumed, the solution is unique as expected.

Figure 6.7: Example of multiple local maxima in likelihood surface.

6.1.2.5 Performance in Silence

One major point of interest is how the algorithms perform in silence. As shown above, the EM algorithm can converge to different local maxima. This is one of the weaknesses of the EM algorithm, since it tends to place a single narrowband formant at a peak in the observed spectrum as shown in Figure 6.8.

Recently, a voice activity detector (VAD) based on a generalized likelihood ratio test (GLRT) has been proposed [89, 90]. In this method, two hypotheses are given at each block:

\[ H_0: \text{Speech absent: } y_n = n_n, \]
\[ H_1: \text{Speech present: } y_n = s_n + n_n. \]

Since this is a composite test, there are many different options for testing. In the GLRT, the likelihood is maximized under each hypothesis. The test is then given by

\[
\Lambda_n = \frac{\max p(y_n|H_1)}{\max p(y_n|H_0)} \geq \eta
\]

(6.32)

In the notation of the a priori and a posteriori signal to noise ratios, the test statistic is
Figure 6.8: EM algorithm estimated spectrum during a period of silence.

given by

$$\log(\Lambda_n) = \sum_{\omega} \frac{1}{1 + \xi(\omega)} \exp \left( \frac{\xi(\omega) \gamma(\omega)}{1 + \xi(\omega)} \right),$$

(6.33)

where the a priori SNR, $\xi$, and the a posteriori SNR, $\gamma$, are given by

$$\xi(\omega) \triangleq \frac{\rho_0(\omega)}{\rho_0(\omega)/\rho_0(\omega)/\rho_0(\omega)^p},$$

$$\gamma(\omega) \triangleq \frac{|Y(\omega)|^2}{\rho_0(\omega)}.$$

(6.34)

The test statistic, $\Lambda$, is summed over all frequencies, $\omega$, except for the DC term. This method could be used with the ML estimation to reduce the erratic noise during silence.

6.1.3 Gibbs Sampler ($\text{MMSE}_{\iota}=E$)

In the previous sections, many issues have been shown with the maximum likelihood estimates. These algorithms suffer from multiple solutions, approximations of estimation variance, and erratic behavior during silence. Although these problems can be mitigated by careful starting point selection and silence detection, there is still room for improvement.

As a more robust alternative to the ML estimator, a minimum mean square error (MMSE) estimator of the parameters of autoregressive speech in known noise (SIG-B:N, PV:N) is proposed.
Like the ML solution, there is no closed-form solution to this formulation. Instead, this is accomplished by using Monte-Carlo integration, which approximates the conditional expectation,

$$
E[l_n | y_n] = \int l_n p(l_n | y_n) \, dl_n.
$$

(6.25)

Suppose that one was able to draw samples of LSF’s from the posterior distribution $p(l_n | y_n)$. The pdf of the distribution could be approximated by

$$
P_N(l_n | y_n) = \frac{1}{N} \sum_{i=1}^{N} \delta(l_n - l_i^0),
$$

(6.36)

where $l_i^0$ is the $i$th sample. The conditional expectation of this distribution acts as the estimator

$$
l_n = \frac{1}{N} \sum_{i=1}^{N} l_i^0.
$$

(6.37)

This estimator converges almost surely to the true expectation under mild conditions.

Unfortunately, it is difficult to express analytically the desired posterior distributions, $a(y_n, \sigma^2 | y_n)$ and $s(y_n)$. However, the distributions for $a(\sigma^2, s, y_n)$, $s(y_n)$, and $a(\sigma^2, \sigma, y_n)$ are fairly straightforward. These can be arranged as a Markov chain and if sampled sequentially, will converge in distribution to the desired posteriors. For completeness, a fourth random element, $p(\omega)$, is added to the problem. The quantity $p(\omega)$ is equal to one when there is any speech signal present at frequency $\omega$. This is useful for voiced speech where the signal is not present at all frequencies due to its periodic nature. For whispered speech, however, one should assume that the signal is always present, or $p(\omega) \equiv 1$.

6.1.3.1 Derivation

Like the EM algorithm, this algorithm can be performed in both the time and frequency domains. The iterations have been derived for the time domain case and are given in [35].

The computations required for the Gibbs sampler can be reduced when it is expressed in the frequency domain.

When the signal is present, or $p_s^H(\omega) = 1$, the signal distribution is given by

$$
S^H(\omega) | a^H, (\sigma^2)^H, Y(\omega) \sim \mathcal{N} \left( \frac{P_{S}^H(\omega) + P_{S}(\omega)}{P_{S}^H(\omega) + P_{S}(\omega)} Y(\omega), \frac{P_{S}^H(\omega) P_{S}(\omega)}{P_{S}^H(\omega) + P_{S}(\omega)} \right).
$$

(6.38)
This follows directly from the assumption that the sequence length is long enough to assume that the spectral components are uncorrelated. However, if the signal is not present, then \( S^0(\omega) \) is equal to zero with probability 1.

The other two distributions are obtained by looking at the following conditional distribution under the assumption that speech is present at \( \omega_0 \):

\[
\begin{align*}
 f(S(\omega)|a, \sigma^2) &= \frac{|A(\omega)|}{\sqrt{2\pi} \sigma} \exp \left\{ \frac{-1}{2\sigma^2} |A(\omega)|^2 |S(\omega)|^2 \right\} \\
 &= \frac{|A(\omega)|}{\sqrt{2\pi} \sigma} \exp \left\{ \frac{-1}{2\sigma^2} [S(\omega)]^2 + 2\text{Re} \left( \sum_{k=1}^p a_k |S(\omega)|^2 \exp(-j\omega k) \right) \right\} \\
 &\quad + \frac{p}{\sqrt{2\pi} \sigma} \sum_{k=1}^p \sum_{m=1}^p a_k a_m |S(\omega)|^2 \exp(-j\omega(k-m)) \right\}. \quad (6.39)
\end{align*}
\]

At this point, it is useful to define the set where speech is present as \( P = \{ l : p(\omega_l) = 1 \} \).

By again assuming the independence of the spectral components,

\[
\begin{align*}
 f(S|a, \sigma^2, P) &= \left( \frac{1}{2\pi \sigma^2} \right)^{|P|/2} \exp \left\{ \sum_{l \in P} \log |A(\omega_l)| - \frac{1}{2\sigma^2} \sum_{l \in P} |S(\omega_l)|^2 \right\} \\
 &\quad + 2\text{Re} \left( \sum_{k=1}^p a_k \sum_{l \in P} |S(\omega_l)|^2 \exp(-j\omega_k) \sum_{m=1}^p a_m |S(\omega_m)|^2 \exp(-j\omega(k-m)) \right) \prod_{l \in P} \delta(S(\omega_l)). \quad (6.40)
\end{align*}
\]

At this stage, \( p(\omega) \) is assumed to sample the spectrum closely enough to approximate the first sum by its integral. By expressing \( R(r) \) as the inverse Fourier transform of \( |S(\omega)|^2 p(\omega) \)

\[
\begin{align*}
 f(S|a, \sigma^2) &\approx \left( \frac{1}{2\pi \sigma^2} \right)^{|P|/2} \exp \left\{ \frac{L}{2\pi} \int_{-\pi}^{\pi} \log |A(\omega)| \, d\omega - \frac{L}{2\sigma^2} |R(0)| \right\} \\
 &\quad + 2\text{Re} \left( \sum_{k=1}^p a_k R(k) \right) + \sum_{k=1}^p \sum_{m=1}^p a_k a_m R(k-m) \right\}. \quad (6.41)
\end{align*}
\]

Since the integral is equal to zero for all stable polynomials, so the term with the integral is dropped. For notation, \( r_{00} \triangleq R(0), r_s \triangleq \{ R(1), \ldots, R(p) \}^T \), and the matrix \( |R_s|_{km} \triangleq R(k-m) \). This yields

\[
\begin{align*}
 f(S|a, \sigma^2) &= \left( \frac{1}{2\pi \sigma^2} \right)^{|P|/2} \exp \left\{ -\frac{L}{2\sigma^2} \left[ r_{00} + 2a^T r_s + a^T R_s a \right] \right\}. \quad (6.42)
\end{align*}
\]

The distributions for \( a|S, \sigma^2 \) and \( \sigma^2|S, a \) are the same as \( a|S, \sigma^2 \) and \( \sigma^2|S, a \), since no additional information is conveyed by \( y \) about the speech parameters if the speech waveform
is known. If uninformative priors are assumed, the distribution for the prediction polynomial is given by

$$f(a|\sigma^2, S) \propto \exp \left\{ \frac{-L}{2\sigma^2} \left[ a^T R_a a + 2a^T r_s + a^T R_s a \right] \right\}$$

$$\propto \exp \left\{ \frac{-L}{2\sigma^2} \left[ a^T R_a a + a^T R_a a + r_s^T R_s^{-1} r_s \right] \right\}$$

$$= \exp \left\{ \frac{-L}{2\sigma^2} \left[ a + R_s^{-1} r_s \right]^T R_s \left[ a + R_s^{-1} r_s \right] \right\}$$

$$\propto \mathcal{N} \left( a; -R_s^{-1} r_s, \sigma^2 (I_{R_s})^{-1} \right).$$

(6.43)

However, a stability constraint is placed on $a$, so the actual distribution is a truncated Gaussian. This is simulated by drawing samples from Equation 6.43 until a stable polynomial is drawn. The posterior for $\sigma^2$ is given by

$$f(\sigma^2|a, S) \propto (\sigma^2)^{-P/2} \exp \left\{ - (\sigma^2)^{-1} \left( \frac{L}{2} [r_s + 2a^T R_a a + a^T R_s a] \right) \right\}$$

$$\propto \mathcal{IG} \left( \frac{P}{2} + 1, \frac{L}{2} [r_s + 2a^T R_a a + a^T R_s a] \right).$$

(6.44)

The conditional signal presence is a Bernoulli random variable that is dependent on the prior probability $q(\omega) \triangleq \Pr(p(\omega) = 1)$, with its parameter calculated using Bayes law:

$$p^{[b]}(\omega)|a^{[b]}, (\sigma^2)^{[b]}, S^{[b]}|Y_n(\omega) \sim \mathcal{B} \left( \begin{array}{c} \mathcal{N}(Y_n(\omega); 0, P_{y_0}^{[b]}(\omega)) \end{array} \right) \cdot \frac{\mathcal{N}(Y_n(\omega); 0, P_{y_0}^{[b]}(\omega)) q(\omega) + \mathcal{N}(Y_n(\omega); 0, P_{y_0}^{[b]}(\omega))(1 - q(\omega))}{q(\omega) + \mathcal{N}(Y_n(\omega); 0, P_{y_0}^{[b]}(\omega))(1 - q(\omega))}.$$

To summarise for whispered speech:

$$a^{[b]}(\omega)|a^{[b]}, (\sigma^2)^{[b]}|Y_n(\omega) \sim \mathcal{N} \left( \begin{array}{c} (RL)^{-1} \sigma^{[b]} T_{[b]}^{[b]} \left( \sigma^2 \right)^{[b]} T^{[b]} \left( LRL^{-1} \right)^{-1} \end{array} \right),$$

$$\sigma^{[b]}(\omega)|a^{[b]}, (\sigma^2)^{[b]}|Y_n(\omega) \sim \mathcal{IG} \left( \frac{L}{2} - 1, \frac{L}{2} \left[ r_s^{[b]} + 2(a^{[b]} T R_a^{[b]} a^{[b]})^{[b]} \right] \right),$$

$$S^{[b]}(\omega)|a^{[b]}, (\sigma^2)^{[b]}|Y_n(\omega) \sim \mathcal{N} \left( \begin{array}{c} P_{y_0}^{[b]}(\omega) \end{array} \frac{P_{y_0}^{[b]}(\omega)}{P_{y_0}^{[b]}(\omega) + P_{y_0}^{[b]}(\omega)} \end{array} \right).$$

(6.45)

where $L$ is the number of frequency samples. The values $R_{y_0}^{[b]}$, $r_s^{[b]}$, and $s_{y_0}^{[b]}$ are constructed from the autocorrelation sequence generated from the inverse Fourier transform of $|S^{[b]}(\omega)|^2$.

With the generated samples, one can easily approximate any conditional expectation.

For example, in order to take advantage of correlations between blocks, the second order
statistics of the estimated LSFs are required. These statistics are found by

\[
\begin{aligned}
\eta_n & \sim \mathcal{N}\left(\mu_n, \hat{P}_n^2\right), \\
z_n & = \frac{1}{K - K_b} \sum_{k=K_b+1}^{K} \mathcal{F}(a[k]), \\
P_n^2 &= \frac{1}{K - K_b} \sum_{k=K_b+1}^{K} \left(\mathcal{F}(a[k]) - z_n\right) \left(\mathcal{F}(a[k]) - z_n\right)^T,
\end{aligned}
\]

where \(\mathcal{F}(a)\) is the transformation from linear prediction polynomials to line spectrum pairs, \(K\) is the number of samples generated, and \(K_b\) is the number of "burn-in" samples required for the sampler to converge.

6.1.3.2 Posterior Distributions

In addition to being able to produce conditional expectations, the Gibbs sampler also gives an opportunity to explore the nature of the parameter posterior distributions. These can be created by generating histograms from the samples. An example of this analysis can be seen in Figure 6.9, where a single vowel segment in 10 dB SNR additive noise is processed to create a series of LSF vectors and prediction variances. The histograms of LSF vectors show that the distribution is reasonably well modeled by a single modal Gaussian random variable. One can also see that the variance is highly dependent on the vector element. Also, the Gaussian assumption starts to become less accurate as the variance rises, such as in the case of the last LSF value. On the other hand, the residual variance has a distribution that looks more like an inverted-gamma distribution. However, when this is transformed to get the gain parameter, the distribution looks far closer to being Gaussian. For this reason, it appears to be appropriate to model the distribution of the LSFs and gain parameter using Gaussian random variables.

6.1.4 Effect of Assumptions

In the previous sections, several assumptions were made. For example, the Fourier components are assumed to be independent. The actual correlation is given by

\[
E\left[S(\omega_k)S^*(\omega_l)\right] = \frac{1}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} E\left[\eta[n]\eta^*[\omega_l]+\eta[n]\eta^*[\omega_l]\right] e^{-j\omega_l(\omega_n-\omega_m)},
\]

(6.46)
where \( w[n] \) is the analysis window. For a white noise process, this simplifies to

\[
E[S(\omega_k)S^*(\omega_k)] = \frac{1}{N} \sum_{n=0}^{N-1} w^2[n] e^{-j(\omega_k - \omega)n}
\]

\[
= \frac{1}{N} (W^*W)(\omega_k - \omega).
\]  

(6.47)

For most processes without extreme resonances, the correlation is very close to the expression for white noise. An example of this is shown in Figure 6.10, where the top plot contains eleven lines, each showing the correlation coefficient of each spectral magnitude to its neighboring magnitude. The second plot shows the difference between this process and the values calculated in Equation 6.47. The question remains whether this has a significant effect on the EM iteration. The E-step becomes significantly more complicated since the observations are correlated:

\[
S[Y, \theta] \sim \mathcal{N}(KY, \text{diag}(p^R) \left[I - K^2\right]),
\]

\[
K = \text{diag}(p^R) \left[ \text{diag}(p^R) + W \text{diag}(p^R) + \text{diag}(p^R) W \right]^{-1}.
\]  

(6.48)

(6.49)

Again, the effect of this assumption is tested using an example autoregressive process shown in Figure 6.11. In this example, 30 samples are taken from the process with additive
white noise at 4 dB SNR. Four different estimators are then compared: standard LPC, MMSE estimation, ML estimation with the independence assumption, and ML estimation without the independence assumption. In general, the standard LPC produces very biased estimates. Both of the ML estimates produce nearly unbiased estimates with very high variances. However, the overall characteristics are very similar. The MMSE estimator outperforms all of these methods with estimates that are unbiased and with relatively small variance.

Since the parameter estimates from the single block algorithms will be smoothed, one needs to know how well the variance estimates perform. To test this, a sample segment of the same noisy speech process is simulated a large number of times to generate histograms of the LSF estimates. The results of this experiment are shown in Figure 6.12. In all four plots, the bell curves represent the distribution of the ideal minimum variance unbiased
Figure 6.11: Spectral samples of three spectral estimation algorithms. The true spectrum is represented by the wide line with SNR = 4 dB.

The estimator with Gaussian distribution. In the first plot, the standard LPC estimator is tested on signals with 10 dB SNR. From the histograms, it is apparent that the variances are lower than the lower bound. This is due to the fact that the estimator is biased. The plot underneath includes EM estimates of the same signal. It is apparent from this plot that the ML estimates come very close to satisfying asymptotic efficiency. However, when the SNR is lowered to 0 dB, this property no longer holds for the ML estimator. The distribution of this estimator can be seen in the third plot. Finally, the distribution of the minimum mean squared error estimates from the Gibbs sampler are shown in the last plot. These are much closer to, but not exactly equal to, the theoretically best distribution.
Figure 6.12: Real and theoretical distributions of estimated LSPs.
6.1.5 Filtering Spectral Parameters (ω:S)

One simple and robust method for smoothing the LSF trajectory is to median filter the individual LSF parameters in time [36] (MED-S). However, this method is not flexible, introduces a delay of one or two frames in causal systems, and is not optimal in any sense. For this reason, algorithms that utilize the JMLS description of the parameter trajectories are preferable.

One issue with this methodology is that the estimator is the conditional mean of the line spectrum frequencies, which imposes a quadratic loss function. Ideally, we would like to use a loss function proportional to an objective speech measure such as the Kakura distance. However, in the context of LSF vector quantization, the weighted quadratic distortion measure has been found to be a good approximation to the true spectral distortion [34]. This MMSE estimator is justified by the fact that the optimal estimator for any weighted quadratic loss function is the conditional mean.

Finally, the method does not take into account the possibility of discontinuities in the LSF domain. Although the trajectories for the speech tend to be fairly continuous, there are cases where two frames with similar spectra have drastically different LSF representations. This can be seen in Figure 6.13, where the formants are very similar, but a small change in the spectral tilt increases the spectral distance to the LSFs. For this reason, future work could investigate different parameterizations of the MELP codec that are more continuous across time.

6.2 Waveform Enhancement

To this point, the enhancement goal has been to estimate the parameters of the MELP model. One might also be interested in estimating the actual original waveform from the sampled data. There are two ways that the algorithms given in this chapter can be used for waveform enhancement. The first method uses the Gibbs sampler described in Section 7.2.2. The other more general method integrates the spectral envelope estimate with more traditional spectral subtraction methods.
6.2.1 Gibbs Sampling

One can generate estimates of the signal, the spectral amplitudes, and the log-spectral amplitudes by using the samples from the Gibbs sampler in Equation 6.45. For example, the log-spectrum of the speech can be estimated by

\[
\hat{S}(\omega) = \exp \left( E \left[ \log |S(\omega)| \right] \right) \left[ \hat{Y}(\omega), l = 1 \ldots L \right] \tag{6.50}
\]

\[
\hat{S}(\omega) = \exp \left[ \frac{1}{K - R_0} \sum_{k=K+1}^{K} \log |S(\omega)| \right] \tag{6.51}
\]

One problem with using this method is that none of the interframe information is used for the spectral estimation.

6.2.2 Spectral Amplitude Weighting

The spectral estimators described in this section can be adapted to create a spectrum amplitude weighting type enhancer. As described in the background section, the key element of this type of enhancer is determining the a priori SNR, \( \xi_k \). Using the AR estimation algorithms, the SNR is given by

\[
\xi_k \triangleq \frac{P_x(\omega_k)}{P_e(\omega_k)} = \frac{\delta^2}{P_x(\omega_k)|\hat{A}(\omega_k)|^2} \tag{6.55}
\]

89
In this formulation, the values for $\hat{A}(\omega)$ and $\delta^2$ are the output from the estimation scheme, and $P_k(\omega)$ is the noise spectrum. With this given, any noise suppression rule can be used, although the MMSE LSA estimator will be used for the experiments.

6.2.3 Normal Speech

Although this thesis is focused on processing whispered speech, several new enhancement schemes have been proposed that can be applied to normal speech. In general, the a priori estimates in Equation 6.32 result in enhanced speech that is not of very high quality. The biggest issue with voiced speech is that the methods described here do not take advantage of its harmonic nature. In fact, many modern spectrum domain enhancers explicitly model the signal presence in each signal bin to take advantage of this property of speech [14, 60].

This technique can be incorporated into both the Gibbs sampling and amplitude weighting enhancement schemes.

For the Gibbs sampler, the signal presence variable, $p(\omega)$, is introduced. By adding this to the model, the Gibbs sampler now includes an additional step to sample $p$. In addition, the three other samplers must be modified to reflect knowledge of the signal presence. The full sampler is given by

$$
\begin{align*}
\alpha^{[k]} &\sim \mathcal{N}(\sigma^2)^{[k-1]}, 1) \cdot \mathcal{N}(\mathbf{Y}_n, \mathbf{R}_n) \\
\sigma^{[k]} &\sim \mathcal{N}(0) \\
p^{[k]} &\sim \mathcal{N}(0, 1) \\
p^{[k]} &\sim \mathcal{N}(0, 1) \\
q(\omega) &\sim \mathcal{N}(0, 1) \\
\end{align*}
$$

where $q(\omega)$ represents the prior probability of signal presence at $\omega$. 

90
An example of the MMSE estimates of the spectrum can be seen in Figure 6.14. In this example, the same noisy waveform is processed with the same MMSE enhancer but with $q(\omega)$ equal to 1.0 and 0.9. One can see that the overall spectral shape for both estimates are very similar. However, when $q(\omega)$ is lowered, the pitch harmonics are more clearly resolved.

![Gibbs Sampler Spectral Estimator, $P_v = 1.0$](image1)

![Gibbs Sampler Spectral Estimator, $P_v = 0.9$](image2)

**Figure 6.14:** Examples of Gibbs sampling spectral estimation methods on normal speech. The phrase spoken is “Shall we play a game?” under 0 dB SNR.

The other approach is to use a decision directed algorithm that is closer to the Ephraim-Malah method in Section 2.2. As in the standard method, an initial estimate of the a priori SNR is available, which can be improved by using the spectral estimates from the previous frame. This yields

$$
\hat{f}_k(t) = \alpha \frac{S(\omega_k; t - 1)^2}{P_e(\omega_k)} + (1 - \alpha) \frac{\delta^2(t)}{P_e(\omega_k)|A(\omega_k; t)|^2}.
$$

(6.54)

To properly resolve the pitch harmonics, it is necessary to consider speech presence again. For this type of estimator, the probability of speech is again reintroduced with

$$
p(\omega) = \frac{p(Y(\omega)|H_0)q(\omega)}{p(Y(\omega)|H_0)q(\omega) + p(Y(\omega)|H_1)(1 - q(\omega))},
$$

(6.55)

91
where $H_1$ refers to the hypothesis that speech is present in the spectral component, while $H_0$ refers to the absence of speech. It is useful to incorporate a presence tracking algorithm to find the prior probability of speech presence given by $q(\omega)$

$$ q(\omega; t) = \beta p(\omega; t - 1) + (1 - \beta) Q, $$

where $Q$ is a constant prior probability, which is set to 0.5 for further experiments. Finally, this is incorporated into the amplitude estimator by setting the speech amplitude to

$$ |\hat{S}(\omega)| = (G_{min})^{1-p(\omega)} |G_{LSA}|^{p(\omega)} |Y(\omega)|, $$

where $G_{LSA}$ is the original amplitude gain from the log-spectrum estimator, and $G_{min}$ is a constant that limits the attenuation. This value is set to 0.02 in the following experiments.

The results of this method can be seen in Figure 6.15. In the first two plots, one can compare the Ephraim-Malah method with the AR-based LSA estimator when signal presence is not accounted for. One can see that the proposed method has a very smooth spectrum and the pitch harmonics are well resolved. The noise is more attenuated in both estimators when the speech presence is accounted for in the lower two plots. However, the noise is further attenuated in the proposed method while keeping the overall spectral shape of the clean speech. If one compares this method with the spectrogram in Figure 6.14, one can see that the estimates are smoother from frame to frame.
Figure 6.15: Examples of spectral amplitude weighting enhancement methods on normal speech. The phrase spoken is “Shall we play a game?” under 0 dB SNR.
6.3 Comparison with Other methods

It is instructive to compare the proposed algorithms and their underlying assumptions with other model-based enhancement schemes. The HMM enhancement schemes of [25, 30] use a prior model outlined in Figure 6.16. Like the proposed method, the analysis blocks of the speech are assumed to be Gaussian autoregressive with parameters dependent on a Markov chain. However, in the HMM methods, the AR parameters are determined by a discrete distribution. In the proposed method, the state determines the current parameters in a dynamic linear model that generates the AR parameters. This results in different algorithms for parameter and signal estimation. This method also differs from [26], where a dynamic linear model is also used to enhance the speech. This method is a state dependent dynamic linear model of the speech waveform, not of the speech parameters. Nonstationary-state HMMs have also been proposed for speech enhancement algorithms [85]. In this model, the parameter trajectories are also derived from models that are dependent on a discrete Markov chain.

Figure 6.16: Prior model for AR-HMM enhancement schemes [25, 30].

6.4 Results

The test framework for noise removal is shown in Figure 6.17. In this system, the minimum statistics algorithm is used to estimate the noise floor. As a baseline system, the MELPs method is used to generate the parameters. The output is then modified by using different gain and LPC parameters from the proposed spectral estimators. This, in turn, is altered by a post-processor that reduces the effects of breath noise.
6.4.1 Optimizing Settings

All of the estimation schemes have different settings that need to be adjusted for best performance. In order to test a wider variety of configurations, it is necessary to look at objective measures of speech quality. For this purpose, the Itakura spectral distance described in Section 2.5.2 is used. To test performance at different SNRs, the DAM lists are corrupted with white noise using different noise levels using the ITU Recommendation P.56 [42].

For the EM algorithm, one of the key parameters is the noise model certainty parameter, $K$. In Table 6.1, the median and mean Itakura distances are listed for two noise estimators and two values of $K$. In addition, distances associated with using both the original MELP and MELPe methods are listed at the bottom. It is worth noting that the MELPe algorithm only slightly reduces the Itakura distance. In all cases, the EM algorithm actually increases the spectral distance to the clean waveform, which is expected due to the erratic nature of the ML estimator. Even when the noise PSD is known exactly, decreasing the value of $K$ to 32 improves performance. This decrease reduces the chance of creating false formants in spectral regions that are noise dominated.

One major issue with the Gibbs sampling algorithm is that it requires a large number of iterations to converge and get quality estimates. Since there are no analytical tools for choosing the number of iterations, $K$, this must be determined empirically. Again, white noise was synthetically added to the DAM lists with different SNRs. The MMSE-
Table 6.1: Effect of different settings for the EM algorithm on mean and median Itakura distances for white noise.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>20 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise known, $K = \infty$</td>
<td>1.840/1.5044</td>
<td>1.4229/1.0193</td>
<td>1.0302/0.6930</td>
<td>0.4125/0.0473</td>
</tr>
<tr>
<td>Noise known, $K = 32$</td>
<td>1.3990/1.1354</td>
<td>1.1172/0.8010</td>
<td>0.8035/0.5777</td>
<td>0.6145/0.0654</td>
</tr>
<tr>
<td>Min stat, $K = \infty$</td>
<td>1.0989/1.6880</td>
<td>1.5045/1.2104</td>
<td>1.3232/0.7799</td>
<td>0.6739/0.1495</td>
</tr>
<tr>
<td>Min stat, $K = 32$</td>
<td>1.4315/1.1819</td>
<td>1.1738/0.8668</td>
<td>0.9605/0.6511</td>
<td>0.7038/0.5522</td>
</tr>
<tr>
<td>Orig</td>
<td>1.1750/1.2926</td>
<td>0.9441/0.7657</td>
<td>0.7080/0.4778</td>
<td>0.3200/0.1169</td>
</tr>
<tr>
<td>MELPe</td>
<td>1.1308/0.9297</td>
<td>0.8864/0.6765</td>
<td>0.6914/0.4882</td>
<td>0.4694/0.3367</td>
</tr>
</tbody>
</table>

Table 6.2: Effect of number of iterations in the Gibbs sampler on mean and median Itakura distances for white noise.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Signal to Noise Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 dB</td>
</tr>
<tr>
<td>1000</td>
<td>1.7535/1.5439</td>
</tr>
<tr>
<td>2000</td>
<td>1.5731/1.3650</td>
</tr>
<tr>
<td>4000</td>
<td>1.0130/0.8524</td>
</tr>
<tr>
<td>6000</td>
<td>0.9421/0.7681</td>
</tr>
<tr>
<td>8000</td>
<td>0.9103/0.7572</td>
</tr>
<tr>
<td>Orig</td>
<td>1.1751/1.0263</td>
</tr>
<tr>
<td>MELPe</td>
<td>1.1308/0.9297</td>
</tr>
</tbody>
</table>

Algorithm was then applied to these waveforms with different numbers of iterations to create Table 6.2, where the mean and median Itakura distances of the outputs are listed. In all of these examples, the number of "burn-in" samples, $K_b$, is equal to 500. In general, the distortion drops quickly with the number of iterations until 4000 iterations, after which the benefit from additional samples is much smaller. Since the effect on the number of iterations required is nearly independent of the SNR, the number of iterations used for the Gibbs sampler is set to 6000 for the duration of this thesis.

Finally, the smoothing configurations must be determined. In order to do this, the smoothers were tested using the outputs from the ML and MMSE estimators. Again, the same four SNR levels were used for testing. For each noise condition, three different JMLS smoothers and the median filter were tested. For all configurations, the filter delay was varied from zero to two blocks. In addition, full and diagonal system matrices were tested for the JMLS smoothers. The median Itakura spectral distances for each configuration are listed in Tables 6.2 and 6.4. In these tables, the results without smoothing are listed under
Table 6.3: Median Itakura distance and the Gibbs sampler with 6000 iterations.

<table>
<thead>
<tr>
<th>SNR</th>
<th>Lag</th>
<th>Median</th>
<th>1 State</th>
<th>2 State</th>
<th>3 State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>diag</td>
<td>full</td>
<td>diag</td>
</tr>
<tr>
<td>0 dt</td>
<td>0</td>
<td>0.757</td>
<td>0.518</td>
<td>0.610</td>
<td>0.529</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.726</td>
<td>0.510</td>
<td>0.646</td>
<td>0.521</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.706</td>
<td>0.518</td>
<td>0.646</td>
<td>0.537</td>
</tr>
<tr>
<td>5 db</td>
<td>0</td>
<td>0.559</td>
<td>0.340</td>
<td>0.406</td>
<td>0.343</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.528</td>
<td>0.332</td>
<td>0.396</td>
<td>0.350</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.561</td>
<td>0.243</td>
<td>0.394</td>
<td>0.361</td>
</tr>
<tr>
<td>10 db</td>
<td>0</td>
<td>0.355</td>
<td>0.198</td>
<td>0.230</td>
<td>0.200</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.344</td>
<td>0.197</td>
<td>0.222</td>
<td>0.200</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.403</td>
<td>0.198</td>
<td>0.221</td>
<td>0.208</td>
</tr>
<tr>
<td>20 dt</td>
<td>0</td>
<td>0.048</td>
<td>0.040</td>
<td>0.047</td>
<td>0.041</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.130</td>
<td>0.039</td>
<td>0.045</td>
<td>0.041</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.210</td>
<td>0.039</td>
<td>0.045</td>
<td>0.048</td>
</tr>
</tbody>
</table>

The median filter with lag zero. In general, the median filters gave little or no improvement.

With the 3MLS smoothers, the best performance was usually accomplished by using the one-state diagonal model. By comparing scores with the unfiltered results, there is a gain of approximately 5 dB by using the smoother. However, additional gains from increasing the complexity were relatively small. In a few cases, adding a second state decreased the average distance, but it usually increased it. This is much worse for three states, where the distance actually increases. However, the two-state model still generates spectra with more realistic dynamics. For this reason, the two-state model with diagonal system matrices are used in further subjective listening experiments.

From the tests in this section, several conclusions can be drawn. First, with respect to the single block estimators:

- **MMSE-FD:E** and **ML-FD:E** are the most viable spectral estimators.
- The **MMSE estimator** outperforms the ML estimator by approximately 5 dB.
- The **MMSE estimator** produces closer spectral estimates without smoothing than the **MELPs algorithm**.
- The **MMSE algorithm** requires 6000 iterations per block to provide the optimal estimates.
Table 6.4: Median Itakura distance and the EM algorithm when the noise spectrum is known and $K = 32$

<table>
<thead>
<tr>
<th>SNR</th>
<th>Lag</th>
<th>1 State</th>
<th></th>
<th>2 State</th>
<th></th>
<th>3 State</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>median</td>
<td>diag.full</td>
<td></td>
<td>diag.full</td>
<td></td>
<td>diag.full</td>
</tr>
<tr>
<td>0 db</td>
<td>0</td>
<td>1.135</td>
<td>0.785</td>
<td>0.781</td>
<td>0.788</td>
<td>0.783</td>
<td>0.795</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.155</td>
<td>0.764</td>
<td>0.761</td>
<td>0.778</td>
<td>0.772</td>
<td>0.783</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.225</td>
<td>0.764</td>
<td>0.762</td>
<td>0.794</td>
<td>0.789</td>
<td>0.804</td>
</tr>
<tr>
<td>5 db</td>
<td>0</td>
<td>0.801</td>
<td>0.637</td>
<td>0.639</td>
<td>0.635</td>
<td>0.654</td>
<td>0.640</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.799</td>
<td>0.622</td>
<td>0.621</td>
<td>0.639</td>
<td>0.633</td>
<td>0.644</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.837</td>
<td>0.629</td>
<td>0.627</td>
<td>0.695</td>
<td>0.885</td>
<td>0.795</td>
</tr>
<tr>
<td>10 db</td>
<td>0</td>
<td>0.578</td>
<td>0.490</td>
<td>0.493</td>
<td>0.484</td>
<td>0.490</td>
<td>0.491</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.547</td>
<td>0.489</td>
<td>0.493</td>
<td>0.508</td>
<td>0.607</td>
<td>0.511</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.602</td>
<td>0.501</td>
<td>0.503</td>
<td>0.586</td>
<td>0.573</td>
<td>0.590</td>
</tr>
<tr>
<td>20 db</td>
<td>0</td>
<td>0.473</td>
<td>0.401</td>
<td>0.405</td>
<td>0.392</td>
<td>0.402</td>
<td>0.403</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.420</td>
<td>0.414</td>
<td>0.422</td>
<td>0.428</td>
<td>0.428</td>
<td>0.432</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.434</td>
<td>0.420</td>
<td>0.427</td>
<td>0.458</td>
<td>0.488</td>
<td>0.494</td>
</tr>
</tbody>
</table>

In addition, several conclusions can be made about the spectral smoothing algorithms:

- Spectral evolution can be modeled with different states to preserve rapidly spectral dynamics that occur during consonants.
- Using smoothers based on these states do not significantly change the average spectral distortion.
- Spectral smoothing improves performance of ML and MMSE algorithms by 5 dB.
- Better performance is generally achieved with diagonal models.
- Slight improvement is made by using a fixed lag filter with a single frame lag.

6.4.2 Example Spectrograms

To illustrate some of the differences between different enhancement methodologies, a series of spectrograms from these schemes are now presented. In all of the following experiments, the phrase shown in Figure 6.18 is corrupted with F-16 cockpit noise from the NOISEX database to create an overall waveform SNR of 0 dB.

The importance of spectral smoothing is shown in Figure 6.19. In the second plot, one can see the spectral estimates from (ML-FD:E,PV:N), while in the final plot the results of (ML-FD:E,PV:N,JML:S) are shown. The smoothing algorithm for the tests is a two-state
Figure 6.18: Clean LPC spectrum of whispered sample of male speaking "Shall we play a game?"

diagonal model of the LSF evolution. In each step, the spectral estimates are much closer to the clean waveform. In the process of smoothing estimates, the probabilities that the LSPs evolved from the high variance state are generated through time. When this value is close to one, the smoother assumes that the parameters are, in fact, rapidly changing, and does not smooth them as much. One can see that this method detects transitional periods with reasonable accuracy.

In Figure 6.20, three different estimators are compared prior to any smoothing. The three estimators are (MMSE-FD: E, PV:N), (ML-FD: E, PV:N), and (ML-COV: E, PV:N). In these, all three estimators improve the spectral estimates and the perceptual quality. The waveform-based maximum-likelihood based estimator (ML-FD: E) creates narrow bandwidth formants in the estimates that are perceptually annoying. As expected, this estimator also creates very erratic spectral estimates during periods of silence. The (MMSE-FD: E) algorithm produces spectral estimates that are more consistent between frames and in general are closer to the true spectrum. The (ML-COV: E) algorithm produces estimates that are far less accurate because less observed data is available.

After using the smoothing algorithm (MMSE: S), the spectrum estimates are all improved, as seen in Figure 6.21. In this plot, the (ML-FD: E) and (MMSE-FD: E) plots are replaced with smoothed spectral estimates. The (ML-COV: E) algorithm is not included since the results of this algorithm from the previous figure are not suitable for further study. Instead, these algorithms are compared with the results of the MELPe algorithm, which is shown...
in the lower plot. The biggest improvement occurs in the (ML-FD-E) estimator. This is because most of the incorrect narrowband formants have been removed. These spurious formants usually occur with LSF pairs with a high calculated variance. Knowledge of this high variance causes the spectral smoother to smooth these LSF's, which removes the offending formant. For the MMSE algorithm, there is an overall improvement in the consistency of the estimates from frame to frame. From these examples, one can see that there is promise in creating quality spectral estimates even from low intensity whispered speech.
Figure 6.19: Effect of smoothing on the MMSE-E spectrum estimator. Original waveform produced by male whispering “Shall we play a game?” with δ dB SNR.
Figure 6.20: Comparison of spectrum estimators. Original waveform produced by male whispering "Shall we play a game?" with 0 dB SNR.
Figure 6.21: Comparison of smoothed spectrum estimators. Original waveform produced by male whispering “Shall we play a game?” with 0 dB SNR.
6.4.3 Subjective Results

Although the objective results are useful for selecting parameters for the algorithms, it is necessary to perform actual listening tests. In the original listening test, diagnostic acceptability measure (DAM) tests and diagnostic rhyme tests (DRTs) were conducted on older versions of the proposed algorithms. In this section, these results will be given, as well as more recent degradation mean opinion scores (DMOSs) for waveforms processed by the optimized algorithms.

6.4.5.1 Original DAM and DRT Tests

The first listening tests were conducted on slightly older versions of the (ML:2, JMLS:3) and (MMSE:R, JMLS:3) enhancements schemes. These differ from the current method in several ways. First, the JMLS consisted of a two-state model with full, not diagonal, system matrices. The Gibbs sampler was also configured to use 200 iterations, instead of the 6000 used in the later experiments. Finally, the tests used a fixed filter for breath noise mitigation.

The DAM test results from these older experiments are included in Table 6.5. One can see that the ML estimation based methods performed similarly to the MELPc algorithm. However, the MMSE-based estimator performed slightly better across all experiments. The detailed DAM results in the street cafe environment for the MELPc, ML, and MMSE codecs are listed in Tables B.6, B.7, and B.8, respectively.

In detailed results for the ML estimator, the noise acceptance measure is improved at the expense of the signal acceptability when compared with MELPc. This was manifested in an improvement in all of the noise categories except for staticy and chirping (BS, BC), respectively. This improvement was made at the expense of making the signal more fluttered, interrupted, and thix (SF, SI, ST), respectively. These differences are less extreme in the MMSE results. The noise was mitigated better in the rushing, rumbling, and bubbling (BNM, BNL, BF) categories. The signal was characterized as having less flutter (SF) but having greater thinness (ST).

The intelligibility scores from these algorithms are included in Table 6.6. In this example,
the ML-based method produced results with much lower intelligibility, while the MMSE estimator produced similar, though slightly lower, results than the MELP results. The detailed results are in the appendix as Tables B.13, B.14, and B.15. The intelligibility of the EM output was lower across all of the distinctive features. This is most severe in the sibilation and voicing categories. In general, the listeners were very biased towards the presence of both of these features. A drop in both of these areas was seen in the MMSE algorithm, although the effect on the sibilation was less severe. Overall, the rest of the categories were similar to the MELP standard, with the exception of the nasality and compactness properties that saw a slight increase in performance.

6.4.3.2 Final Subjective Tests

In order to test the changes to the algorithms made since the previous tests, an additional listening test was conducted. This consisted of a DMOS test with the same procedure as in Section 5.6.2, but with noisy waveforms. For each test phrase, the MELP-coded clean whispers were presented as the reference waveform, while a processed MELP-coded waveform was used for the samples. Two different types of noise were used: the street cafe data described in Chapter 3 and electronically added car noise from the Aurora2 database.

The purpose of this is to compare the differences between enhancing speech corrupted by non-stationary cafe noise with speech added with more stationary automobile noise. For these environments, four estimators were tested. The first two are standard MELP and
Table 6.7: Noise Removal DIFOS test results.

<table>
<thead>
<tr>
<th>Debreath</th>
<th>Algorithm</th>
<th>Noise Condition</th>
<th>Mean</th>
<th>S.E.</th>
<th>Car Mean</th>
<th>S.E.</th>
<th>Street Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>MELP</td>
<td>Quiet</td>
<td>4.01</td>
<td>0.07</td>
<td>1.81</td>
<td>0.08</td>
<td>1.44</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>Gibbs</td>
<td></td>
<td>-</td>
<td>-</td>
<td>2.75</td>
<td>0.08</td>
<td>1.76</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>Gibbs+Smooth</td>
<td></td>
<td>-</td>
<td>-</td>
<td>3.13</td>
<td>0.08</td>
<td>1.83</td>
<td>0.08</td>
</tr>
<tr>
<td>Debreath</td>
<td>MELP</td>
<td>Quiet</td>
<td>3.98</td>
<td>0.08</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MELP</td>
<td></td>
<td>-</td>
<td>-</td>
<td>3.08</td>
<td>0.09</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Gibbs+Smooth</td>
<td></td>
<td>-</td>
<td>-</td>
<td>3.11</td>
<td>0.09</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

MELPe, which includes a noise preprocessing frontend. The second two are the Gibbs sampled MMSE estimates of the spectrum and the smoothed MMSE estimates. Finally, the breath removal algorithm was tested on a subset of these configurations. The results of these tests are included in Table 6.7. The configurations that were not tested are indicated with a dash.

In the table, one can see that for standard MELP, the two noise conditions create similar distortions. However, when enhancement is applied, these conditions behave very differently. In the street cafe environment, none of the methods provide more than 0.40 points of improvement over standard MELP, which is consistent with previous DAM testing. On the other hand, these schemes are very effective at improving the subjective quality in the car environment. The MELPe codec almost raises the quality by a whole point, while the Gibbs sampler is able to provide improvement of 0.38 points over the MELPe codec. Furthermore, by further smoothing of the spectrum, the subjective quality can be improved by another 0.14 points. The effect of the automatic debreath algorithm is a little surprising. Although the method reduces the overall quality of the speech processed by the Gibbs sampler, it improves the output of the MELPe codec.

Finally, in order to differentiate between similar configurations, a simple AB test was conducted on several pairs. The results of these tests are included in Table 6.8. This table contains the average number of people that preferred the second algorithm in the test, as well as the P-value for a two-tailed test of significance. As shown in the previous experiment, the average user preferred the Gibbs sampler spectral estimator to the MELPe codec in
<table>
<thead>
<tr>
<th>Test</th>
<th>Mean</th>
<th>P-val</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELP + Breath Remove vs. MELP (quiet)</td>
<td>0.40</td>
<td>0.04</td>
</tr>
<tr>
<td>Gibbs+Smooth+Breath Remove vs. Gibbs+Smooth (car noise)</td>
<td>0.45</td>
<td>0.32</td>
</tr>
<tr>
<td>MELPe+Breath Remove vs. MELPe (car noise)</td>
<td>0.37</td>
<td>4.46e-003</td>
</tr>
<tr>
<td>Gibbs+Smooth vs. MELPe - car noise</td>
<td>0.36</td>
<td>2.40e-003</td>
</tr>
<tr>
<td>Gibbs+Smooth vs. MELPe - street cafe</td>
<td>0.63</td>
<td>4.46e-003</td>
</tr>
</tbody>
</table>

The results also agree with the previous conclusion that the breath removal algorithm helped the MELPe algorithm in the car environment. The tests also show a slight preference towards the breath removal algorithms in the quiet environment. Finally, users chose the MELPe codec more often when street cafe noise was used.

These tests show that many of the proposed algorithms can improve the subjective quality of noisy speech. The breath removal algorithms have been shown to improve performance in some situations. For additive background noise, the results show that in highly non-stationary noise, there is little improvement that can be made using current techniques that will improve the subjective quality of coded whispered speech. On the other hand, in more stationary environments like automobile noise, the quality can be improved. In these cases, the methods proposed in this thesis yield better estimates of the speech spectrum.
CHAPTER 7

RECOGNITION

Recognition can mean several things in the context of speech. Usually, one is referring to automatic speech recognition; however, speaker recognition tasks are also common. In this chapter, three different recognition tasks are investigated. First, the effect of whispering on speech recognition performance is determined, as well as the viability of adapting recognition models to whispered speech. In the second section, the algorithms presented in this thesis are used to deal with noise robustness issues. This is an important recognition problem, with standard databases dedicated to its study. Finally, the problem of speaker recognition is addressed with an investigation of how algorithms developed in this thesis could be applied to this problem.

7.1 Speech Recognition - Recognizing Whispers

Automatic speech recognition (ASR) is an important application of speech processing. At the time of this thesis, the only ASR study of whispered speech was in Japanese [42]. In this study, the average cepstrum of different phonemes were compared, and whispered speech was tested with an ASR system. The average cepstral distance between normal and whispered speech was 4 dB for voiced phonemes and 2 dB for unvoiced phonemes. Using an ASR system trained on normal speech, the word accuracy drops from 82% to 18%. However, when the system was fully trained with whispered speech, the accuracy improved to 68%. The accuracy could also be increased to 44% by using 80 sentences of adaptation data.

This problem can be viewed as a specialized case of speaker adaptation for speech recognition. One commonly used method for this problem is maximum likelihood linear regression (MLLR) [56]. In this method, the HMM parameters are transformed to maximize the likelihood of some transcribed adaptation data. This method can be implemented by
using transcribed whispered speech. Another approach to speaker adaptation is vocal tract normalization (VTN). In this technique, one observes that linear frequency warping of the speech spectrum can compensate for the effect of physical size. This technique has been successfully combined with MLLR, producing better results than the individual techniques [80].

7.1.1 Baseline System

The speech recognition system used for the following experiments is a standard hidden Markov model based Viterbi recognizer. The system uses a two-state HMM to represent each monophone, while each state is represented by a 32-mixture Gaussian mixture model, with two separate gender-dependent models. These models were trained with 30 hours of HUB4 data using the Fast-Talk HMM training tool [13]. In order to accommodate the experiments, two model sets were trained: one for MFCCs and another for linear prediction cepstral coefficients (LPCC). The algorithm for generating the models, $\mathcal{M}_M$ and $\mathcal{M}_L$, is shown in Figure 7.1.

In addition to standard training, these models were also adapted to the six speakers in the whispered speech database. This was necessary since part of the experiments in this section includes adaptation to whispered speech data. If completely speaker independent models are used, it is impossible to determine if performance gains are made from adapting to whispering or from adapting to the speaker set. By first adapting to the speaker set, the improvements made by further adaptation can be attributed to matching whispering, not speaker, characteristics. This final training step was accomplished using the MLLR algorithm on the full voice DAM sentences.

7.1.2 Experimental Setup

The basic scheme for testing whispered ASR is shown in Figure 7.2. In these experiments, the whispered DAM lists are used for adaptation data, while the whispered DRT lists are used for test data. In this way, different waveform preprocessors can be used with different feature extraction schemes. In each configuration, both normal ASR and adapted ASR are used. In the adaptation scheme, the preprocessor and feature extractor for the adaptation
Figure 7.1: Block diagram of the baseline model training algorithm.

data always match the test data. The preprocessor either consists of the debeath algorithm or a simple pass through. The feature extraction portion consists of either MFCC- or LPCC-based features. For the MFCCs, both standard and frequency-warped feature vectors are used. For the LPCCs, the standard features are also used, along with features based on formant-shifted LPC representations. The full block diagram of these extraction algorithms is shown in Figure 7.3.

Figure 7.2: ASR test for whispered speech.

7.1.3 Frequency Warping

The frequency warping implementation is very similar to the vocal tract normalization method [55]. Using this method, the frequency axis is warped by altering the center frequencies of the filter banks. The center frequencies are altered as if they were formant frequencies altered using the function described in Section 5.2. In the top half of Figure 7.4,
the weights for the standard filter banks are plotted. The modified weights are shown underneath. To highlight the differences, the second, seventh, and fifteenth filter banks are emphasized.

7.1.4 Experiment Results

Before starting the recognition experiments, the effectiveness of the feature transformation was tested by comparing aligned feature vectors from the whispered and normal DAM lists. The whispered speech included both raw data and waveforms that had the breath noise automatically attenuated. The average Euclidean distance between the feature sets are listed in Table 7.1. Overall, the changes in distance were relatively small. For the MFCC vectors, the debreath algorithms did not reduce the distance, but the filter bank warping decreased the average distance. In the LPC vectors, the debreath also had a negligible average effect, except when the formant shifting algorithm was used. On the raw whispered speech, this method dramatically increased the average distance. However, the algorithm
only helped slightly on the de-breathed whispers.

Of course, the important results are the recognition accuracies. The first experiment was to run the DRT test using the automatic recognizer. The results of this test, which includes unadapted and adapted models, are in Table 7.2. In general, every attempt to improve the feature vectors reduced the performance. However, the adaptation was successful in improving the accuracy. The detailed results for these tests are in Appendix B in Tables B.16-B.21. In this experiment, the adaptation improved every area of the DRT test except for voicing, which is essentially left to chance.

The DRT only provides information about the initial consonants. In order to get a better representation of the recognition performance, general CVC recognition was conducted. In these experiments, all possible CVC and CV syllables were considered by the recognizer. In addition, several two-consonant endings were also considered. This was done to incorporate all of the endings that occur in the DRT lists. These endings include /t ʃ , tʃ l , dʃ l , n tʃ l , tʃ k , n k l , n dʃ l , and s tʃ l . The CVC results, which include both percent correct and accuracy, are given in Table 7.3. In this task, the feature modification produces small gains in the recognition accuracy. However, these changes are dominated by the improvements made by model adaptation. Again, the breach removal algorithm reduces the recognition
Table 7.1: Average weighted Euclidean distances between normal speech features and enhanced whispered features.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Enhance Method</th>
<th>Preprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>MFCC</td>
<td>None</td>
<td>Debreath</td>
</tr>
<tr>
<td></td>
<td>Warp</td>
<td>23.73</td>
</tr>
<tr>
<td></td>
<td>21.46</td>
<td>21.62</td>
</tr>
<tr>
<td></td>
<td>Warp</td>
<td>16.33</td>
</tr>
<tr>
<td></td>
<td>16.16</td>
<td>15.92</td>
</tr>
<tr>
<td></td>
<td>Shift</td>
<td>19.16</td>
</tr>
<tr>
<td></td>
<td>Shift+Tilt</td>
<td>16.21</td>
</tr>
</tbody>
</table>

Table 7.2: Results of the BRT test using ASR.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>No Debreath</th>
<th>Debreath</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Adapt</td>
<td>Adapt</td>
</tr>
<tr>
<td>MFCC</td>
<td>38.19</td>
<td>51.91</td>
</tr>
<tr>
<td>MFCC warp</td>
<td>38.02</td>
<td>48.44</td>
</tr>
<tr>
<td>LPCC</td>
<td>41.32</td>
<td>54.69</td>
</tr>
<tr>
<td>LPCC shift</td>
<td>38.57</td>
<td>51.91</td>
</tr>
<tr>
<td>LPCC shift+tilt</td>
<td>35.85</td>
<td>51.04</td>
</tr>
</tbody>
</table>

Full Voice MFCC: 69.36, Full Voice LPCC: 68.92

performance.

Since the acoustic differences that are compensated for are in the vowels, it is helpful to look at the vowel recognition accuracy in the /CVC/ experiments. In these methods, there is also improvement gained by using different feature extractors. The results of the experiments are in Table 7.4. The confusion matrix for normal speech with MFCC features are presented in Table 7.5. One can see from this table that there is some confusion between acoustically similar vowels.

The vowel error rate increases dramatically when whispered speech is used, as seen in

Table 7.3: Percent correct and accuracy of the recognizer using /CVC/ grammar.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>No Debreath</th>
<th>Debreath</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No Adapt</td>
<td>Adapt</td>
</tr>
<tr>
<td>MFCC</td>
<td>28.74/29.71</td>
<td>40.91/45.38</td>
</tr>
<tr>
<td>MFCC warp</td>
<td>33.19/24.46</td>
<td>50.97/48.71</td>
</tr>
<tr>
<td>LPCC</td>
<td>33.06/24.93</td>
<td>51.92/47.25</td>
</tr>
<tr>
<td>LPCC shift</td>
<td>35.32/24.20</td>
<td>51.87/46.07</td>
</tr>
<tr>
<td>LPCC shift+tilt</td>
<td>35.35/24.64</td>
<td>51.26/45.21</td>
</tr>
</tbody>
</table>

Full Voice MFCC: 67.71/52.70, Full Voice LPCC: 68.24/61.36

113
Table 7.4: Accuracy of vowel recognition using /CVC/ grammar.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>No Debreath No Adapt</th>
<th>Debreath No Adapt</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>23.78</td>
<td>19.90</td>
</tr>
<tr>
<td>MFCC warp</td>
<td>34.54</td>
<td>26.08</td>
</tr>
<tr>
<td>LPC</td>
<td>30.32</td>
<td>25.62</td>
</tr>
<tr>
<td>LPCC shift</td>
<td>35.82</td>
<td>32.90</td>
</tr>
<tr>
<td>LPCC shift-tilt</td>
<td>35.36</td>
<td>35.65</td>
</tr>
</tbody>
</table>

Full Voice: MFCC: 96.56, Full Voice LPCC: 66.92

Table 7.6. One of the main sources of error is a general bias towards /ɾ/. This is probably caused because /ɾ/ is used as the neutral vowel and has a large acoustic variation. This allows mismatched observations that do not fit any particular vowel well to be classified as /ɾ/. The other common error is for vowels to be confused with acoustic neighbors with a slightly higher first formant frequency. For example, the most common confusions are from /i/ to /e/, /e/ to /a/, /a/ to /e/, /æ/ to /e/, /a/ to /au/, and /a/ to /o/.

Many of the first formant errors are corrected by using features such as the warped MFCCs that account for the formant differences. These results, shown in Table 7.7, show that the recognition performance of /i/, /e/, /æ/, /o/, and /a/ are improved at the cost of /a/ and /æ/. The bulk of the performance improvement is gained by MLLR adaptation. Using this method on straight MFCC vectors, the confusions, which are in Table 7.8, to the neutral vowel, /i/, are nearly eliminated. However, many of the errors caused by the first formant still exist. By both warping the frequency axis and performing adaptation, the performance is once again improved as seen in Table 7.9.

Although no vowel intelligibility tests were made by human listeners on this data, the results can still be compared to previous whispered vowel recognition experiments. The two tests that have been conducted on human vowel recognition reported a drop from 82% to 65% [51] and from 92% to 82% [98] when the speakers whispered. The performance drop seen by the automatic speech recognizer from 66.9% to 57.6% is similar to the degradation seen for human listeners. These experiments confirm that adaptation and feature modification provide adequate compensation to the acoustic models for vowels.

114
Table 7.5: Vowel confusion matrix for normal speech with using MFCC features.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>o</th>
<th>e</th>
<th>i</th>
<th>u</th>
<th>% Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>259</td>
<td>0</td>
<td>2</td>
<td>99</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>o</td>
<td>0</td>
<td>279</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>179</td>
<td>0</td>
<td>7</td>
<td>151</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td>160</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>129</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>o</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.6: Vowel confusion matrix for whispered speech using MFCC features and no adaptation.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>o</th>
<th>e</th>
<th>i</th>
<th>u</th>
<th>% Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>71</td>
<td>29</td>
<td>36</td>
<td>8</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>o</td>
<td>5</td>
<td>56</td>
<td>62</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>e</td>
<td>73</td>
<td>10</td>
<td>32</td>
<td>17</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>i</td>
<td>2</td>
<td>50</td>
<td>68</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>u</td>
<td>1</td>
<td>30</td>
<td>39</td>
<td>4</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>o</td>
<td>15</td>
<td>3</td>
<td>27</td>
<td>21</td>
<td>43</td>
<td>2</td>
</tr>
<tr>
<td>u</td>
<td>7</td>
<td>3</td>
<td>40</td>
<td>4</td>
<td>17</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.7: Vowel confusion matrix for whispered speech using warped MFCC features and no adaptation.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>o</th>
<th>e</th>
<th>i</th>
<th>u</th>
<th>% Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>45</td>
<td>6</td>
<td>45</td>
<td>38</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>o</td>
<td>16</td>
<td>29</td>
<td>46</td>
<td>1</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>39</td>
<td>1</td>
<td>54</td>
<td>40</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>i</td>
<td>2</td>
<td>19</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>1</td>
<td>6</td>
<td>32</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td>0</td>
<td>2</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>o</td>
<td>0</td>
<td>1</td>
<td>27</td>
<td>22</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>u</td>
<td>2</td>
<td>1</td>
<td>23</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.8: Vowel confusion matrix for whispered speech using MFCC features and adaptation.

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>o</th>
<th>e</th>
<th>i</th>
<th>u</th>
<th>% Corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>152</td>
<td>0</td>
<td>4</td>
<td>20</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>o</td>
<td>1</td>
<td>143</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>e</td>
<td>102</td>
<td>0</td>
<td>9</td>
<td>33</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>i</td>
<td>121</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>o</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>u</td>
<td>1</td>
<td>9</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

115
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>o</th>
<th>e</th>
<th>i</th>
<th>u</th>
<th>% corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>112</td>
<td>0</td>
<td>7</td>
<td>48</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>o</td>
<td>0</td>
<td>118</td>
<td>1</td>
<td>0</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>e</td>
<td>61</td>
<td>0</td>
<td>5</td>
<td>75</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td>83</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>u</td>
<td>0</td>
<td>14</td>
<td>8</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>o</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>18</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
7.2 Speech Recognition - Noise Robustness

In this section, the noise removal algorithms in this thesis are applied to feature extraction for ASR. There are many standard databases in the field of noise-robust speech recognition. Among the most popular is the Aurora2 database [1], which consists of the TIDIGITS corpus corrupted with several types of electronically added noise under a variety of signal-to-noise ratios. In order to make comparisons with other methods, the algorithms are tested on this database, despite the fact that they were not whispered.

The use of the autoregressive (AR) or linear predictive coding model has been pervasive in speech processing areas such as speech coding and enhancement. Although these have been used as features in speech recognition, they were superseded by different cepstral representations such as mel-frequency cepstrum coefficients [27]. These were chosen not only because they increased clean performance, but because LPC-based features were found to be less robust to noise [45].

It might seem strange to focus on LPC, the very element that made the features less noise robust. However, recent gains in computational resources have made Monte Carlo methods useful for enhancing noise-corrupted speech [33]. In these methods, the speech is assumed to be autoregressive with slowly varying parameters. These algorithms operate in the time domain and their evaluation has focused on waveform enhancement for listening purposes.

There has been recent interest in finding direct statistical estimators of speech features given noise-corrupted speech [20]. In this section, an MMSE estimator of the feature vectors under the AR assumption is considered. These estimators are based on the single block estimator used in Section 7.2.2 to draw samples from the posterior distribution of the feature vectors as shown in Figure 7.5. The advantage of this method is that it automatically generates the variances, which can be applied directly to Uncertain Observation (UO) techniques [5, 19, 102].
Figure 7.5: Block diagram of the proposed feature extraction algorithm under the AR assumption.

7.2.1 Signal Model

Two models of the speech signal are considered: autoregressive (AR) and nonparametric (NP). In the AR model, the nth observed signal block, \( y_n[t] = s_n[t] + v_n[t], t = 1, \ldots, T \), is assumed to be the sum of an autoregressive speech signal, \( s_n[t] \), and a random noise signal, \( v_n[t] \), with power spectrum \( P_s(\omega) \). The speech signal is expressed by

\[
    s_n[t] = \sum_{k=1}^{P} a_n[k] s[t-k] + c_n[t], \quad c_n \sim N(0, \sigma^2 I). \tag{7.1}
\]

Under the NP model, the speech is assumed to be a Gaussian process with power spectrum \( P_s(\omega) \). The prior for this spectrum is given by an inverted gamma distribution

\[
P_s(\omega) \sim IG(\alpha, \beta). \tag{7.2}
\]

The actual quantity of interest is the MFCC feature vector, \( x_n \), which is a function of the speech, \( s_n \). Since direct estimation of AR parameters has been shown to produce erratic results [73], it is useful to leverage knowledge about the time evolution of these feature vectors. The feature vectors are modeled by a dynamic linear model,

\[
x_n = Ax_{n-1} + w_n + F, \quad w_n \sim N(0, Q), \tag{7.3}
\]

where the system matrices \( A \), \( F \), and \( Q \) are calculated using Aurora2 training data.
7.2.2 Feature Estimation

The goal of this section is to create an estimator of the true MFCC vector, $\mathbf{x}_n$, from the acoustic waveform, $y_n$. The MMSE estimates are $E(x_n|y_n, AR)$ and $E(x_n|y_n, NP)$. Because of the difficulties in finding these directly, Monte Carlo integration is performed on samples from the distributions.

One can generate spectral samples using the AR sampler is given by in Section 7.2.2. The NP sampler is given by

$$
P_{k}(\omega) | S_{k-1}(\omega), y_n(\omega) \sim \mathcal{N} \left( \frac{1}{2} \mathbf{s}^T \left( \mathbf{S}_{k-1} - \mathbf{x}_n \right) + \mathbf{y}_n(\omega), \frac{P_{k}(\omega)P_{k}(\omega)}{P_{k}(\omega) + \mathbf{y}_n(\omega) + \mathbf{y}_n(\omega)} \right),$$

(7.4)

$$
S_{k}(\omega) | P_{k}(\omega), y_n(\omega) \sim \mathcal{N} \left( \frac{P_{k}(\omega)P_{k}(\omega)}{P_{k}(\omega) + \mathbf{y}_n(\omega) + \mathbf{y}_n(\omega)} \right).$$

(7.5)

For each block, the samples create a sequence of $K$ samples from the posterior distributions. As in the noise removal algorithms, only samples $K_b$ through $K$ are considered. By using these samples, the conditional expectation of any function of the speech signal can be found by transforming the samples and averaging. Regardless of which sampler is used, the initial MFCC parameter estimates are

$$
k'_n = \frac{1}{K - K_b} \sum_{k = K_b+1}^{K} \mathcal{F} \left( S_k^{(n)} \right),$$

(7.6)

$$
S_n = \frac{1}{K - K_b} \sum_{k = K_b+1}^{K} \left( \mathcal{F} \left( S_k^{(n)} - \mathbf{x}_n \right) \right)',$n,$$

(7.7)

$$
|\mathcal{F}(S)|_k = \sum_{k = 0}^{K} d_{k,i} \log \left( \sum_{l = 0}^{L} w_{k,i} |S(\omega)| \right).$$

(7.8)

where $w_{k,i}$ are the filter bank weights and $d_{k,i}$ are the DCT coefficients for $\mathcal{F}$, the spectrum to MFCC transformation.

Finally, the underlying model in Equation 7.3 is combined with the observation equation $\mathbf{x}_n = \mathbf{x}_n + \mathbf{n}_n$, $\mathbf{n}_n \sim \mathcal{N}(0, \Sigma_n)$. It is straightforward to use this model with a fixed-lag Kalman smoother to get the final estimates, $\mathbf{x}_n$ and $\Sigma_n$.

In both of the methods, an estimate of the noise PSD is required. This information is collected for the experiments using one of two techniques: taking an ideal noise PSD by
windowing the nearest $L_1$ frames of the actual noise signal or by averaging over the initial $L_2$ signal frames before speech has begun.

7.2.3 Uncertain Observations

An interesting consequence of the speech enhancement scheme described above is that it allows for arbitrary PDF descriptions of the unknown clean feature to be generated with relative ease. As several techniques have recently been developed to use PDF descriptions of features for robust recognition instead of the standard points in the feature space [5, 6, 19, 102], the Monte Carlo feature enhancement methods are ideal for this decoding style.

In this section, the Monte Carlo feature enhancement algorithms in Section 7.2.2 are implemented as a front-end to the Uncertain Observation HMM decoding algorithms described in [6]. Instead of calculating the state $j$ output probability, $b_j(x)$, for a frame of speech by finding the probability of a single point, $x$, in space representing that speech frame, this decoding algorithm finds the probability of all possible observations weighted by their respective likelihoods. Thus, decoding is in general specified as

$$\Pr(y_n | q_o=j, W_n) = \int_{-\infty}^{\infty} f_x(\theta) b_j(\theta) d\theta,$$  \hspace{1cm} (7.9)

where $f_x(x)$ is some PDF describing $\Pr[x_n | y_n, W]$, the likelihood of unobserved clean speech feature vector, $x_n$, given the noisy observation $y_n$ and noise model $W_n$.

For the particular case of a $K$-mixture Gaussian speech model and a single Gaussian speech observation PDF with mean $\hat{s}_k$ and covariance $\Sigma_{jk}$, the decoding algorithm simplifies to

$$\sum_{k=1}^{K} c_{jk} N(\hat{s}_k, \mu_k, \Sigma_{jk} + \Sigma_n)$$  \hspace{1cm} (7.10)

It is worth noting that a more complex system can be obtained easily by extending to an $I$-mixture Gaussian describing the enhanced speech vector $x_n$:

$$\sum_{i=1}^{I} \sum_{k=1}^{K} c_{jk} N(\hat{s}_{ik}, \mu_k, \Sigma_{jk} + \Sigma_n)$$  \hspace{1cm} (7.12)

as the method of Section 7.2.2 readily extends to providing Gaussian mixture feature estimates.
7.2.4 Experimental Results

The algorithm parameters were all adjusted for the following experiments. The number of iterations $K$ and $K_0$ are set to 1000 and 200, respectively, for the NP algorithm, but are raised to 4000 and 500 for the AR algorithm. For the NP algorithm, $\alpha = 0.15$ and $\beta = 0$, while in the AR algorithm, $p = 10$. In both of the methods, an estimate of the noise PSD is required. This information was generated using one of two techniques: taking an ideal noise PSD by windowing the nearest $L_1$ frames of the actual noise signal, or by averaging over the initial $L_2$ signal frames before speech has begun.

7.2.4.1 Example Iteration

One can gain intuition about the algorithm by viewing results from a single block of noisy speech. In Figure 7.7, the results of running the Gibbs sampler under the AR condition on a vowel segment corrupted by car noise from Set A, noise condition 3 at -3 dB SNR, are displayed. In Figure 7.7a, the spectrum of the noisy and clean waveforms are plotted along with an estimate of the noise PSD. In the regions of the formant peaks, the signal and noise energy are comparable, while the region between 500 and 1500 Hz is completely dominated by noise.

In Figure 7.7b, the mean and variance estimates of the speech log spectrum are displayed. The distribution is represented by two gray bands that denote the regions within one and two standard deviations from the mean value. This effectively finds the shape of the spectral envelope without attempting to estimate the fine pitch structure. Of course, the goal is to create features for speech recognition, so the filter bank coefficients are included in Figure 7.7c, where each filter bank output is plotted at the location of its center frequency. Again, the estimated parameters are represented using the calculated means and variances. One can see that the shape of the filter bank outputs is well estimated by this method.

7.2.4.2 Aurora2 Recognition Experiments

The enhancement techniques of Section 7.2.2 were tested using the Aurora2 digit database [1]. This database consists of a series of connected digit strings that have been corrupted by
Table 7.10: Results of the AR and NP enhancement algorithms using an ideal noise estimator. The * denotes results prior to the final Kalman smoothing filter.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unenhanced</td>
<td>45.82</td>
<td>50.09</td>
<td>40.38</td>
<td>46.54</td>
<td></td>
</tr>
<tr>
<td>AR Sampler</td>
<td>77.40</td>
<td>79.88</td>
<td>74.97</td>
<td>77.90</td>
<td></td>
</tr>
<tr>
<td>AR Sampler*</td>
<td>75.69</td>
<td>78.51</td>
<td>71.46</td>
<td>76.09</td>
<td></td>
</tr>
<tr>
<td>NP Sampler</td>
<td>71.46</td>
<td>71.74</td>
<td>64.20</td>
<td>70.12</td>
<td></td>
</tr>
<tr>
<td>NP Sampler*</td>
<td>61.52</td>
<td>62.19</td>
<td>61.69</td>
<td>61.82</td>
<td></td>
</tr>
</tbody>
</table>

Electronically added background noise. Experiments were designed to compare the performance between the AR-constrained system and the nonparametric system and to show improved recognition performance when feature variance estimates are used in the Uncertain Observation algorithm. Models were trained using the Aurora2 clean-training scenario. For all experiments, recognition was performed using only the static MFCC coefficients and log energy, giving 13-dimensional feature vectors instead of the standard 39. This was because the increased feature variance from the enhancement algorithms caused the derivative features to hurt performance.

To get an upper bound on performance, experiments were first run using an ideal noise estimator. $\lambda$ smoothed spectral version of the actual noise signal was used as the noise estimate in the enhancement algorithm. Under these ideal conditions, the enhancement algorithm that assumes an underlying autoregressive model outperforms the unconstrained non-parametric enhancement algorithm. The performance of the AR and NP algorithms can be compared with the baseline Aurora2 recognition system from results in Table 7.10. Results are given for each algorithm both before and after the final Kalman smoothing filter.

As expected, the Set A and Set B results in Table 7.10 are similar, since the algorithm makes no use of the stereo training data, and hence, no assumption is made about the noise. Since neither method makes allowances for convolutional distortion as found in Set C, the performance is lower than that of Set A and B. By using the AR sampler, the overall performance is increased by 29.6%, which is much better than the 15.4% improvement gained from using the NP sampler. In both samplers, the Kalman smoother improved the
overall performance. For the AR sampler, the accuracy increased by 1.8%, while the NP sampler performance increased by 8.3%. The best overall configuration was the AR sampler with Kalman smoother, which yielded an overall accuracy of 77.9%.

In Figure 7.6, the results from the AR and NP algorithms are plotted across the range of Auroam2 SNRs. These results are averaged over Sets A and B only, as Set C is outside of the types of distortion for which this algorithm was designed. For the standard unenhanced MPCC features, the accuracy steadily decreases as the SNR is reduced. However, the performance of the AR and NP sampler only changes significantly when the SNR falls to 5 dB. For the higher SNRs, the accuracy differences between the two samplers are fairly small. For low SNR environments, the difference increases to 10%.

For a more realistic example, the noise PSD was estimated from nonspeech frames prior to the beginning of the utterance. The results for this method, shown in Table 7.11 for the AR Gibbs sampler, illustrate the necessity for an accurate noise PSD estimate. The performance for the AR sampler drops from 77.9% to 64.5%. Future work towards a practical implementation would require a more advanced noise estimation algorithm, where \( P_n(\omega) \) would be tracked in time and could be included in the sampler iterations.

The final set of experiments were performed again using the ideal noise PSD estimate and the AR Gibbs sampler, but this time, the second-order statistics of the clean feature

123
Table 7.11: AR enhancement results using nonspeech frames for noise PSD estimation.

<table>
<thead>
<tr>
<th>Enhancement method</th>
<th>Set A (%)</th>
<th>Set B (%)</th>
<th>Set C (%)</th>
<th>Overall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unenhanced</td>
<td>45.82</td>
<td>50.09</td>
<td>40.88</td>
<td>46.54</td>
</tr>
<tr>
<td>AR Sampler</td>
<td>66.53</td>
<td>65.96</td>
<td>59.22</td>
<td>64.48</td>
</tr>
</tbody>
</table>

estimates are used in the Uncertain Observation algorithm described in Section 7.2.3. The results, shown in Table 7.2.4.2, are averaged between Sets A and B only, and again use only static MFCC features. The results are compared with both baseline recognition using only the AR sampler means and the ideal case that the true feature variance is known (i.e., the exact value of \((y_n-x_n)^2\) is used as the diagonal of matrix \(\Sigma_n\)). By using this information, the accuracy is improved slightly by 1.5%. If the true variances are known, the performance can be raised by 5.1% to 87.4%.

Table 7.12: Comparison between standard HMM decoding, UO decoding using variances from the AR sampler, and UO decoding using ideal variances.

<table>
<thead>
<tr>
<th>Decoding Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Decoding</td>
<td>79.3</td>
</tr>
<tr>
<td>UO using AR sampler variances</td>
<td>80.8</td>
</tr>
<tr>
<td>UO using ideal variances</td>
<td>87.4</td>
</tr>
</tbody>
</table>
Figure 7.7: Example processing using the AR assumption on a block of noisy speech from Set A, noise condition 3, block SNR = -3 dB.
7.3 Speaker Recognition

A related problem to speech recognition is speaker recognition. Algorithms for this problem differ significantly by how the problem is posed. First, recognition systems are either text-dependent or text-independent. In addition, one is either interested in verifying a claimed identity, or automatically determining the identity from speech. A survey of methods used to solve these problems can be found in [11]. This research will compare whispered speech with normal speech for text-independent speaker identification. Gaussian mixture models (GMM) of speech have been shown to be an effective method for this problem [83]. In this work, GMMs and JMLRs trained using MFCCs are compared as models for speaker recognition.

7.3.1 Gaussian Mixture Model Recognition

To test these algorithms, the whispered and full-voice DAM lists were parameterized using 20-dimensional MFCC vectors sampled every 10 ms. Vectors that were classified as silence by their energy measure were excluded. Next, GMMs were created to describe each of the six speakers using varying quantities of training data and model parameters. Each GMM consisted of up to 32 Gaussian mixtures with diagonal covariance matrices. The training data consisted of either 20 seconds, 10 seconds, or 5 seconds of data that was separate from the phrase under test. In this method, the feature vectors are assumed to be independent with distribution given by

$$p_{k}(x_n) = \sum_{k=0}^{K-1} w_k \mathcal{N}(x_n; \mu_k, \Sigma_k), \quad (7.12)$$

where $K$ is the number of mixtures, $x_n$ is the observed MFCC vector at frame $n$, and $s$ is the speaker. The model parameters, which include $\mu_k$, $\Sigma_k$, and $w_k$, are trained using the EM algorithm. Each speaker model is initialized with a single mixture followed by progressive model splitting to get models with $K$ equal to 1, 2, 4, 8, 16, and 32. The speaker is chosen through a simple maximum likelihood test,

$$i = \arg \max \sum_{n=0}^{N-1} \log p_i(x_n), \quad (7.13)$$

where $N$ is the number of test frames.
In order to determine the effectiveness of this method one needs to test with different quantities of training and test data. In Figure 7.8, the training set for each speaker is set to 10 seconds while the query length is tested at 0.25 seconds, 0.5 seconds, 0.75 seconds, and 1.5 seconds on both normal and whispered speech. As expected, the performance improves as the query length increases, with the percent correct saturating to 100% for the longer queries. In addition, the performance tends to improve as the number of mixtures increases, with the exception of the jump from 16 to 32 where performance went down for whispered speech. In addition, the whispered speech performed consistently worse than the normal speech, with a more than 10% decrease for the short queries. This was expected, since humans were also found to have a more difficult time discriminating between whispering speakers [54].

Figure 7.8: Speaker recognition accuracy over six speakers using a GMM with 10 seconds of training data.
In Figure 7.9, the query length is fixed to 0.75 seconds, while the amount of training data is set to 5, 10, and 20 seconds. The relative decrease in performance is small from 20 to 15 seconds, but it is quite drastic from 10 to 5 seconds. Again, the whispered speech does not perform as well, and is also more sensitive to the amount of training data than the normal speech counterparts. In addition, the whispered speech recognizer reaches its peak performance at 8 mixtures when only 5 seconds of training data is available. From these experiments, one can conclude that speaker recognition is feasible for whispered speech, but there is a performance loss that is compounded by increased sensitivity to short queries and training sequences. For long queries and training sequences, the difference between normal and whispered speech is very small. However, if the training sequence and test queries are short the drop can be almost 15% as seen when the 10 seconds of training data are used to recognize 0.25 second test samples.

![Graph showing the effect of training data on recognition accuracy for normal and whispered speech.](image)

**Figure 7.9:** Speaker recognition accuracy over six speakers using a GMM with 0.75 second test queries.

128.
7.3.2 Jump Markov Linear System Recognition

In the enhancement section, jump Markov linear systems were trained and used to enhance speech. Here the JMLS's performance as a speaker recognizer is tested. To remain consistent with previous work, these will be trained using MFCC vectors. The recognition framework is the same as the GMM, but with a different model:

\[ x_{n+1} = A(r_{n+1}) x_n + B(r_{n+1}) w_{n+1} + F(r_{n+1}) \]
\[ p(w_n) = N(\mu_n, 0, I), \quad Pr(r_{n+1} = i | r_n = j) = p_{ji}. \]  

(7.14)  
(7.15)

There are a few details that arise from using a JMLS instead of a GMM. First, like HMMs, the likelihood of an observation coming from a speaker model can be calculated by using the forward-backward procedure [81]. In this framework, the continuous state variable, \( x_n \), must be directly observed, so this method is only directly applicable to clean speech. Another issue is the number of model parameters to use for recognition. Like the GMM, there are several model parameters that must be estimated from the training sequences. For the JMLS, the parameters include the state transition matrix, the continuous state transition matrices \( A(r) \), the state noise matrices \( B(r) \), and the state mean vectors \( F(r) \). To reduce the number of parameters involved, the \( A(r) \) and \( B(r) \) matrices can both be constrained to be diagonal. Training is initialized with a single JMLS, which is split using a similar procedure used for the GMM.

To test the feasibility of using the JMLS for speaker recognition, the JMLSs are trained for each speaker using 10 seconds of normal speech data. The results of this experiment are shown in Figure 7.10. With one exception, the accuracy reaches its maximum at 8 models. In general, the performance of this method is lower than the GMM. To compare the GMM with the JMLS in both whispered and normal speech, the accuracies of the best models generated with 10 seconds of training data are plotted together in Figure 7.11. In this environment, the GMM works very well at identifying speakers speaking normally, but goes down when they whisper. The JMLS does not perform nearly as well with the normal speech accuracy below the whispered speech levels for the GMM. The JMLS also does a very poor job of speaker recognition for whispered speech. This results in a performance...
Figure 7.10: Speaker recognition accuracy over six speakers using a JMIS with 10 seconds of training data.

Figure 7.11: Speaker recognition accuracy comparison between JMIS and GMM modeling. Each model is trained with 10 seconds of training data and the number of mixtures that optimizes performance.

loss in whispered speech that ranges from 20% for 0.25 second samples to 3% for 1.5 second test samples. Although the JMIS does a better job of modeling the feature vectors, it concentrates on the dynamics of the feature vectors, which is less important for speaker recognition.
CHAPTER 8

CONCLUSIONS AND FUTURE WORK

Whispered speech has been analyzed in the context of signal processing tasks. In general, many of the methods that are used for normal speech are also applicable to whispered speech with some performance loss. In coding environments, the effects of quantization are similar to those found in normal speech. However, standard noise removal methods are far less effective at improving noise robustness. However, by using methods that directly estimate the linear prediction coefficients for speech, this performance was found to improve when the background noise was nearly stationary. These noise removal methods were also found to be effective for noise robust feature extraction for ASR.

The differences between normal and whispered speech were explored in the context of creating phonated speech from whispers. This problem consisted of compensating for spectral differences between these two modes of speech and synthesizing the pitch and voicing parameters. Surprisingly, finding the pitch and voicing proved to be easier than modifying the spectrum. The subjective loss from these estimations was much lower than the effect of using the whispered spectrum. The methods for spectral modification were also applied to speech recognition where they were shown to improve performance when used in conjunction with adaptation schemes.

Jump Markov linear systems were also explored as a prior model for the parameters of speech. These models were found to be effective as models of the spectral parameters when used for reducing the variance of LPC estimators in noise. In addition, they made pitch estimation possible by creating models of how the pitch track evolves.

Although the algorithms presented in this thesis have focused on whispered speech, almost all of the spectral estimation schemes can be applied to general problems of speech in additive noise. In many cases, simple modifications have been proposed to make these algorithms better suited to normal phonated speech.
8.1 Contributions

Whispered Speech Analysis:

- Investigated the spectral differences between normal and whispered speech using both theoretical and empirical models [74].
- Developed a computationally efficient algorithm for modification of formant locations and bandwidths. This yielded several relationships between formant structures and the line spectrum [73].
- Used jump Markov linear systems to model pitch trajectories. Used these models to synthesize voicing and pitch information from whispered speech.
- Set up a test framework for combining parameters from whispered, normal, and synthetic speech. Used this framework to evaluate algorithms and determine importance of different aspects of speech.

Noise Removal Enhancement:

- Developed several novel algorithms for estimating spectral parameters from speech waveforms in noise. These include the derivation and comparison of maximum likelihood and minimum mean squared error estimates of LPC parameters.
- Created tools for using jump Markov linear systems for feature estimation. This includes development of algorithms for training parameters for jump Markov linear systems. Investigated the use of jump Markov linear systems for modeling the trajectories of the spectral parameters of speech.
- Investigated spectral estimation algorithms that combine smoothers based on jump Markov linear systems with the MMSE and ML spectral estimators [75].
- Created waveform enhancers that use spectral parameter estimates. This includes use of these parameters as prior information for spectral noise suppression algorithms and direct waveform enhancement using Monte-Carlo algorithms.

132
Speech and Speaker Recognition:

- Compared data-driven compensation methods, model-based adaptation, and signal enhancement schemes for improving automatic recognition of whispered speech.

- Combined the spectrum estimation algorithms with the Uncertain Observation algorithm to create new feature extraction algorithms for noise robust speech recognition.
  Performed tests on the Aurora 2 database [72].

- Compared performance of speaker recognition systems on normal and whispered speech.
  Also evaluated the JMLS as a method for identifying speakers.

8.2 Future Work

There are two main directions that one could extend this work in the future. The first is improvement in the area of whisper-to-speech conversion. All of the research performed in this area has used the MELP model to perform synthesis. Better speaker-dependent whisper to voice conversion might be performed using other methods. Also, spectrum alteration has proven to be the most difficult part of the estimation process.

The other area for future work is the extension of algorithms developed in this thesis. In the course of developing algorithms for processing whispered speech, several methods were created that could be used in general speech processing. While the use of the algorithms are not purely constrained to whispered speech, the possibilities for normal speech are numerous:

- The methods described for enhancing normal speech could be analyzed more rigorously. Evaluation of these algorithms has been constrained to whispered speech.

- Smoothing of recognition features was found to improve speech recognition. This could be a viable low complexity method to improve recognition performance.

- There is currently great interest in using alternate measurements such as general electromagnetic movement sensors (GEMS). The spectral modeling methods could be used to improve denoising from these devices.
# APPENDIX A

## IPA TABLE

Table A.1, adapted from Deller et al. [44], contains a list of the IPA symbols along with their corresponding ARPAAbet symbol and an example.

<table>
<thead>
<tr>
<th>IPA</th>
<th>ARPAAbet</th>
<th>Examples</th>
<th>IPA</th>
<th>ARPAAbet</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>–</td>
<td>German - Mann</td>
<td>t</td>
<td>t</td>
<td>tea</td>
</tr>
<tr>
<td>α</td>
<td>a</td>
<td>hood</td>
<td>d</td>
<td>d</td>
<td>deep</td>
</tr>
<tr>
<td>e</td>
<td>E</td>
<td>head</td>
<td>k</td>
<td>k</td>
<td>kick</td>
</tr>
<tr>
<td>e</td>
<td>e</td>
<td>hayed</td>
<td>g</td>
<td>g</td>
<td>go</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>heed</td>
<td>f</td>
<td>f</td>
<td>five</td>
</tr>
<tr>
<td>o</td>
<td>c</td>
<td>hawed</td>
<td>v</td>
<td>v</td>
<td>vice</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>hoed</td>
<td>θ</td>
<td>θ</td>
<td>thing</td>
</tr>
<tr>
<td>y</td>
<td>R</td>
<td>heard</td>
<td>θ</td>
<td>θ</td>
<td>then</td>
</tr>
<tr>
<td>u</td>
<td>U</td>
<td>hood</td>
<td>s</td>
<td>s</td>
<td>so</td>
</tr>
<tr>
<td>u</td>
<td>u</td>
<td>who'd</td>
<td>s</td>
<td>s</td>
<td>zebra</td>
</tr>
<tr>
<td>a</td>
<td>A</td>
<td>mud</td>
<td>s</td>
<td>Z</td>
<td>measure</td>
</tr>
<tr>
<td>æ</td>
<td>@</td>
<td>had</td>
<td>Z</td>
<td>m</td>
<td>mom</td>
</tr>
<tr>
<td>i</td>
<td>I</td>
<td>hid</td>
<td>n</td>
<td>n</td>
<td>noon</td>
</tr>
<tr>
<td>p</td>
<td>p</td>
<td>pea</td>
<td>l</td>
<td>l</td>
<td>love</td>
</tr>
<tr>
<td>w</td>
<td>w</td>
<td>want</td>
<td>r</td>
<td>r</td>
<td>race</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
<td>had</td>
<td>ə</td>
<td>ə</td>
<td>German - Glück</td>
</tr>
<tr>
<td>j</td>
<td>y</td>
<td>yard</td>
<td>x</td>
<td>x</td>
<td>German - Bach</td>
</tr>
<tr>
<td>ç</td>
<td>–</td>
<td>German - ich</td>
<td>Ø</td>
<td>–</td>
<td>German - schön</td>
</tr>
</tbody>
</table>

134
APPENDIX B

DAM AND DRT RESULTS

This appendix contains a series of results tables from the DAM and DRT tests conducted by ARCON corporation. In addition, the full details of the DRT tests conducted by the automatic speech recognizer are also included.

B.1 Human Listening

B.1.1 Diagnostic Acceptability Measure

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fretting</td>
<td>83.4</td>
<td>1.20</td>
<td>BNR</td>
<td>Hissing</td>
<td>90.8</td>
<td>0.61</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>84.2</td>
<td>1.18</td>
<td>BNM</td>
<td>Rushing</td>
<td>89.4</td>
<td>0.52</td>
</tr>
<tr>
<td>SD</td>
<td>Hash</td>
<td>85.9</td>
<td>2.46</td>
<td>BNL</td>
<td>Rumbling</td>
<td>89.6</td>
<td>1.00</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>90.5</td>
<td>2.94</td>
<td>BB</td>
<td>Buzzing</td>
<td>89.1</td>
<td>0.61</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>88.7</td>
<td>0.86</td>
<td>BF</td>
<td>Bubbling</td>
<td>88.1</td>
<td>0.35</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>81.1</td>
<td>3.44</td>
<td>BS</td>
<td>Staticy</td>
<td>87.7</td>
<td>0.53</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>84.6</td>
<td>1.40</td>
<td>BC</td>
<td>Chirping</td>
<td>88.9</td>
<td>0.22</td>
</tr>
<tr>
<td>SB</td>
<td>Dabble</td>
<td>77.5</td>
<td>2.37</td>
<td>IBA</td>
<td>Accept.</td>
<td>89.8</td>
<td>1.18</td>
</tr>
</tbody>
</table>

Table B.1: DAM scores - Full Voice, NULL, Quiet.

Composite Acceptability Estimate (CAE): Mean = 80.4, S.E. = 1.58
### Table B.2: DAM scores - Full Voice, MELP, Quiet

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>80.2</td>
<td>1.12</td>
<td>BNH</td>
<td>Hissing</td>
<td>92.2</td>
<td>0.55</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>83.3</td>
<td>1.82</td>
<td>BNM</td>
<td>Rubbling</td>
<td>99.7</td>
<td>0.50</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>75.0</td>
<td>2.10</td>
<td>BNL</td>
<td>Rubbling</td>
<td>88.7</td>
<td>0.81</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>80.5</td>
<td>1.04</td>
<td>BB</td>
<td>Buzzing</td>
<td>88.4</td>
<td>0.50</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>87.3</td>
<td>0.68</td>
<td>BF</td>
<td>Bubbling</td>
<td>88.1</td>
<td>0.47</td>
</tr>
<tr>
<td>SF</td>
<td>Nasal</td>
<td>67.7</td>
<td>1.83</td>
<td>BS</td>
<td>Staciy</td>
<td>87.5</td>
<td>3.64</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>77.7</td>
<td>1.38</td>
<td>BC</td>
<td>Chirping</td>
<td>88.0</td>
<td>0.39</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>67.4</td>
<td>1.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISA | Accept. | 61.6 | 2.37 | IBA    | Accept.   | 90.2 | 1.61 |

### Table B.3: DAM scores - Whispered, NULL, Quiet

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>92.3</td>
<td>0.61</td>
<td>BNH</td>
<td>Hissing</td>
<td>80.5</td>
<td>2.10</td>
</tr>
<tr>
<td>SF</td>
<td>Thin</td>
<td>74.3</td>
<td>2.17</td>
<td>BNM</td>
<td>Rubbling</td>
<td>86.6</td>
<td>0.70</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>72.4</td>
<td>2.54</td>
<td>BNL</td>
<td>Rubbling</td>
<td>89.1</td>
<td>0.62</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>83.2</td>
<td>2.23</td>
<td>BB</td>
<td>Buzzing</td>
<td>83.2</td>
<td>1.43</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>89.6</td>
<td>0.61</td>
<td>BF</td>
<td>Bubbling</td>
<td>88.0</td>
<td>0.39</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>80.2</td>
<td>2.73</td>
<td>BS</td>
<td>Staciy</td>
<td>84.7</td>
<td>0.82</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>77.6</td>
<td>3.18</td>
<td>BC</td>
<td>Chirping</td>
<td>88.6</td>
<td>0.42</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>74.7</td>
<td>2.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISA | Accept. | 61.7 | 2.70 | IBA    | Accept.   | 74.5 | 1.01 |

### Table B.4: DAM scores - Whispered, MELP, Quiet

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>89.5</td>
<td>0.68</td>
<td>BNH</td>
<td>Hissing</td>
<td>89.7</td>
<td>0.95</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>75.0</td>
<td>2.91</td>
<td>BNM</td>
<td>Rubbling</td>
<td>89.2</td>
<td>0.33</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>65.4</td>
<td>1.94</td>
<td>BNL</td>
<td>Rubbling</td>
<td>88.6</td>
<td>0.50</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>79.1</td>
<td>2.35</td>
<td>BB</td>
<td>Buzzing</td>
<td>88.1</td>
<td>0.42</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>88.2</td>
<td>0.46</td>
<td>BF</td>
<td>Bubbling</td>
<td>88.1</td>
<td>0.36</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>74.5</td>
<td>2.12</td>
<td>BS</td>
<td>Staciy</td>
<td>85.8</td>
<td>0.60</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>72.4</td>
<td>3.20</td>
<td>BC</td>
<td>Chirping</td>
<td>87.2</td>
<td>0.64</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>67.1</td>
<td>2.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ISA | Accept. | 51.8 | 1.50 | IBA    | Accept.   | 82.4 | 1.27 |

136
### Table B.5: DAM scores - Whispered, NULL, Street Cafe.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Signal Quality</th>
<th>Mean</th>
<th>S.E.</th>
<th>Background Quality</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>81.7</td>
<td>0.49</td>
<td></td>
<td>BNH</td>
<td>Hissing</td>
<td>86.6</td>
<td>2.12</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>63.9</td>
<td>2.06</td>
<td></td>
<td>BNM</td>
<td>Rushing</td>
<td>78.9</td>
<td>2.85</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>75.5</td>
<td>2.35</td>
<td></td>
<td>BNL</td>
<td>Rumbling</td>
<td>64.5</td>
<td>2.61</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>55.6</td>
<td>3.73</td>
<td></td>
<td>BB</td>
<td>Buzzing</td>
<td>80.9</td>
<td>2.10</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>89.6</td>
<td>0.55</td>
<td></td>
<td>BF</td>
<td>Babbling</td>
<td>81.3</td>
<td>2.17</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>81.0</td>
<td>2.25</td>
<td></td>
<td>BS</td>
<td>Staticy</td>
<td>85.1</td>
<td>1.15</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>71.5</td>
<td>4.37</td>
<td></td>
<td>BC</td>
<td>Chirping</td>
<td>83.1</td>
<td>1.45</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>75.8</td>
<td>1.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISA</td>
<td>Accept.</td>
<td>63.9</td>
<td>3.34</td>
<td></td>
<td>IBA</td>
<td>Accept.</td>
<td>58.5</td>
<td>1.67</td>
</tr>
</tbody>
</table>

### Table B.6: DAM scores - Whispered, MELP, Street Cafe.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Signal Quality</th>
<th>Mean</th>
<th>S.E.</th>
<th>Background Quality</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>74.6</td>
<td>1.24</td>
<td></td>
<td>BNH</td>
<td>Hissing</td>
<td>83.1</td>
<td>2.99</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>70.3</td>
<td>2.14</td>
<td></td>
<td>BNM</td>
<td>Rushing</td>
<td>82.2</td>
<td>1.62</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>63.2</td>
<td>2.14</td>
<td></td>
<td>BNL</td>
<td>Rumbling</td>
<td>79.4</td>
<td>2.46</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>76.0</td>
<td>2.52</td>
<td></td>
<td>BB</td>
<td>Buzzing</td>
<td>78.5</td>
<td>2.09</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>86.2</td>
<td>0.85</td>
<td></td>
<td>BF</td>
<td>Babbling</td>
<td>75.1</td>
<td>1.84</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>71.0</td>
<td>2.65</td>
<td></td>
<td>BS</td>
<td>Staticy</td>
<td>79.8</td>
<td>1.41</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>63.0</td>
<td>2.79</td>
<td></td>
<td>BC</td>
<td>Chirping</td>
<td>78.7</td>
<td>1.50</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>63.4</td>
<td>2.31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISA</td>
<td>Accept.</td>
<td>45.7</td>
<td>1.88</td>
<td></td>
<td>IBA</td>
<td>Accept.</td>
<td>54.9</td>
<td>1.54</td>
</tr>
</tbody>
</table>

### Table B.7: DAM scores - Whispered, EM, Street Cafe.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Signal Quality</th>
<th>Mean</th>
<th>S.E.</th>
<th>Background Quality</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>73.4</td>
<td>1.87</td>
<td></td>
<td>BNH</td>
<td>Hissing</td>
<td>85.1</td>
<td>1.63</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>72.3</td>
<td>2.27</td>
<td></td>
<td>BNM</td>
<td>Rushing</td>
<td>87.3</td>
<td>0.67</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>61.9</td>
<td>1.88</td>
<td></td>
<td>BNL</td>
<td>Rumbling</td>
<td>85.0</td>
<td>1.57</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>77.3</td>
<td>2.40</td>
<td></td>
<td>BB</td>
<td>Buzzing</td>
<td>83.4</td>
<td>1.56</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>84.6</td>
<td>1.02</td>
<td></td>
<td>BF</td>
<td>Babbling</td>
<td>82.7</td>
<td>1.09</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>73.3</td>
<td>2.30</td>
<td></td>
<td>BS</td>
<td>Staticy</td>
<td>77.2</td>
<td>1.77</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>54.2</td>
<td>5.22</td>
<td></td>
<td>BC</td>
<td>Chirping</td>
<td>77.5</td>
<td>1.30</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>63.0</td>
<td>2.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISA</td>
<td>Accept.</td>
<td>41.9</td>
<td>1.62</td>
<td></td>
<td>IBA</td>
<td>Accept.</td>
<td>59.6</td>
<td>1.05</td>
</tr>
</tbody>
</table>

137
### Table B.8: DAM scores - Whispered, Gibbs, Street Cafe.

**Composite Acceptability Estimate (CAE): Mean = 37.5, S.E. = 0.83**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
<th>Factor</th>
<th>Description</th>
<th>Mean</th>
<th>S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>Fluttering</td>
<td>79.1</td>
<td>0.93</td>
<td>ENH</td>
<td>Hissing</td>
<td>63.1</td>
<td>1.61</td>
</tr>
<tr>
<td>SH</td>
<td>Thin</td>
<td>72.0</td>
<td>2.27</td>
<td>BM</td>
<td>Rushing</td>
<td>85.0</td>
<td>0.86</td>
</tr>
<tr>
<td>SD</td>
<td>Harsh</td>
<td>62.5</td>
<td>1.99</td>
<td>BNL</td>
<td>Rumbling</td>
<td>82.5</td>
<td>2.11</td>
</tr>
<tr>
<td>SL</td>
<td>Muffled</td>
<td>77.9</td>
<td>2.60</td>
<td>BB</td>
<td>Buzzing</td>
<td>81.2</td>
<td>1.89</td>
</tr>
<tr>
<td>SI</td>
<td>Interrupted</td>
<td>86.4</td>
<td>0.86</td>
<td>BF</td>
<td>Bubbling</td>
<td>79.6</td>
<td>1.78</td>
</tr>
<tr>
<td>SN</td>
<td>Nasal</td>
<td>73.8</td>
<td>2.83</td>
<td>BS</td>
<td>Staticy</td>
<td>79.8</td>
<td>1.35</td>
</tr>
<tr>
<td>ST</td>
<td>Thin</td>
<td>58.8</td>
<td>3.49</td>
<td>BC</td>
<td>Chirping</td>
<td>77.2</td>
<td>1.94</td>
</tr>
<tr>
<td>SB</td>
<td>Babble</td>
<td>64.5</td>
<td>2.71</td>
<td>IBA</td>
<td>Accep.</td>
<td>57.8</td>
<td>0.93</td>
</tr>
</tbody>
</table>

ISA: Accept. | 44.9 | 1.89 | IBA: Accep. | 57.8 | 0.93 |
## B.1.2 Diagnostic Rhyme Test

Table B.9: DRT scores - Full Voice, Quiet, NULL codec.

<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td></td>
<td>Total</td>
<td>Mean</td>
<td>S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td>85.94</td>
<td>2.83</td>
<td></td>
<td>92.97</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-frictional</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td>75.00</td>
<td>6.68</td>
<td></td>
<td>87.50</td>
<td>3.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td>96.88</td>
<td>2.05</td>
<td></td>
<td>98.44</td>
<td>1.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td>53.75</td>
<td>4.09</td>
<td></td>
<td>69.88</td>
<td>2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td>99.22</td>
<td>0.78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>100.08</td>
<td>0.00</td>
<td></td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>96.88</td>
<td>1.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>96.88</td>
<td>2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td>81.25</td>
<td>4.09</td>
<td></td>
<td>89.84</td>
<td>1.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>62.50</td>
<td>8.18</td>
<td></td>
<td>79.69</td>
<td>3.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td>96.88</td>
<td>1.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>93.75</td>
<td>4.09</td>
<td></td>
<td>96.88</td>
<td>3.12</td>
<td></td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>95.18</td>
<td>0.73</td>
<td></td>
<td>95.23</td>
<td>1.16</td>
<td></td>
<td>95.70</td>
<td>0.41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present Mean</td>
<td>Present S.E.</td>
<td>Absent Mean</td>
<td>Absent S.E.</td>
<td>Total Mean</td>
<td>Total S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>96.88</td>
<td>2.05</td>
<td>96.88</td>
<td>3.12</td>
<td>96.88</td>
<td>1.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>96.88</td>
<td>3.12</td>
<td>93.75</td>
<td>6.25</td>
<td>95.31</td>
<td>3.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-frictional</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>100.00</td>
<td>0.00</td>
<td>90.62</td>
<td>2.05</td>
<td>95.31</td>
<td>1.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>100.00</td>
<td>0.00</td>
<td>81.25</td>
<td>4.00</td>
<td>90.62</td>
<td>2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td>85.94</td>
<td>4.98</td>
<td>71.88</td>
<td>2.05</td>
<td>78.91</td>
<td>2.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>78.12</td>
<td>7.38</td>
<td>50.00</td>
<td>6.68</td>
<td>64.06</td>
<td>4.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>93.75</td>
<td>4.09</td>
<td>93.75</td>
<td>4.09</td>
<td>93.75</td>
<td>3.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td>93.31</td>
<td>2.29</td>
<td>98.44</td>
<td>1.56</td>
<td>96.88</td>
<td>1.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>93.75</td>
<td>4.09</td>
<td>100.00</td>
<td>0.00</td>
<td>96.88</td>
<td>2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>90.88</td>
<td>3.12</td>
<td>96.88</td>
<td>3.12</td>
<td>96.88</td>
<td>2.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>84.38</td>
<td>6.58</td>
<td>75.00</td>
<td>4.72</td>
<td>79.69</td>
<td>1.96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>93.75</td>
<td>4.09</td>
<td>90.62</td>
<td>4.57</td>
<td>92.19</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>75.00</td>
<td>11.57</td>
<td>59.38</td>
<td>6.58</td>
<td>67.19</td>
<td>3.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>95.31</td>
<td>2.29</td>
<td>81.25</td>
<td>3.34</td>
<td>85.28</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>100.00</td>
<td>0.00</td>
<td>78.12</td>
<td>5.66</td>
<td>89.06</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>90.62</td>
<td>4.57</td>
<td>84.38</td>
<td>4.57</td>
<td>87.50</td>
<td>2.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>92.97</td>
<td>2.15</td>
<td>85.68</td>
<td>1.33</td>
<td>89.32</td>
<td>0.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td>Total Mean</td>
<td>S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>32.81</td>
<td>14.55</td>
<td>0.00</td>
<td>15.85</td>
<td>16.41</td>
<td>1.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>9.38</td>
<td>17.64</td>
<td>18.75</td>
<td>16.19</td>
<td>14.06</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-frictional</td>
<td>56.25</td>
<td>12.27</td>
<td>18.75</td>
<td>18.75</td>
<td>18.75</td>
<td>5.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>96.88</td>
<td>2.05</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustentation</td>
<td>87.50</td>
<td>3.34</td>
<td>98.44</td>
<td>1.56</td>
<td>92.97</td>
<td>1.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>84.38</td>
<td>6.58</td>
<td>96.88</td>
<td>3.12</td>
<td>90.62</td>
<td>3.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>90.02</td>
<td>4.57</td>
<td>100.00</td>
<td>0.00</td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td>96.88</td>
<td>2.05</td>
<td>100.00</td>
<td>0.00</td>
<td>95.44</td>
<td>1.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>87.50</td>
<td>2.36</td>
<td>89.06</td>
<td>2.83</td>
<td>88.28</td>
<td>2.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>96.88</td>
<td>3.12</td>
<td>100.00</td>
<td>0.00</td>
<td>98.44</td>
<td>1.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>78.12</td>
<td>3.12</td>
<td>78.12</td>
<td>5.66</td>
<td>78.12</td>
<td>3.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>98.44</td>
<td>1.56</td>
<td>93.75</td>
<td>2.36</td>
<td>96.09</td>
<td>1.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td>87.50</td>
<td>4.72</td>
<td>92.19</td>
<td>3.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>83.33</td>
<td>2.16</td>
<td>80.21</td>
<td>3.34</td>
<td>81.77</td>
<td>0.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present</td>
<td>Absent</td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>6.25</td>
<td>1.846</td>
<td>18.75</td>
<td>15.49</td>
<td>12.50</td>
<td>4.09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>-9.38</td>
<td>26.28</td>
<td>28.12</td>
<td>17.95</td>
<td>9.38</td>
<td>5.66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>89.06</td>
<td>4.38</td>
<td>93.19</td>
<td>3.29</td>
<td>90.62</td>
<td>2.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>81.25</td>
<td>7.83</td>
<td>90.62</td>
<td>4.57</td>
<td>85.94</td>
<td>4.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>96.88</td>
<td>3.12</td>
<td>93.75</td>
<td>4.09</td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td>67.19</td>
<td>5.76</td>
<td>89.06</td>
<td>3.69</td>
<td>78.12</td>
<td>2.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>71.88</td>
<td>7.38</td>
<td>84.38</td>
<td>8.10</td>
<td>78.12</td>
<td>5.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>62.50</td>
<td>9.45</td>
<td>93.75</td>
<td>4.09</td>
<td>78.12</td>
<td>5.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td>84.38</td>
<td>6.57</td>
<td>98.44</td>
<td>1.56</td>
<td>91.41</td>
<td>2.62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>84.38</td>
<td>6.58</td>
<td>100.00</td>
<td>0.00</td>
<td>92.19</td>
<td>3.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>84.38</td>
<td>6.58</td>
<td>96.88</td>
<td>3.12</td>
<td>90.62</td>
<td>3.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>78.12</td>
<td>3.12</td>
<td>60.94</td>
<td>4.98</td>
<td>69.53</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>71.88</td>
<td>5.66</td>
<td>96.88</td>
<td>3.12</td>
<td>84.38</td>
<td>3.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>84.38</td>
<td>6.58</td>
<td>25.00</td>
<td>10.55</td>
<td>54.69</td>
<td>6.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>100.00</td>
<td>0.00</td>
<td>89.06</td>
<td>4.38</td>
<td>94.53</td>
<td>2.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>100.00</td>
<td>0.00</td>
<td>87.50</td>
<td>4.72</td>
<td>93.75</td>
<td>2.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>100.00</td>
<td>0.00</td>
<td>90.62</td>
<td>4.57</td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>70.83</td>
<td>3.69</td>
<td>74.74</td>
<td>3.64</td>
<td>72.79</td>
<td>1.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present</td>
<td></td>
<td>Absent</td>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>56.25</td>
<td>12.94</td>
<td>12.50</td>
<td>12.05</td>
<td>34.38</td>
<td>5.41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>53.12</td>
<td>20.83</td>
<td>15.62</td>
<td>16.31</td>
<td>34.38</td>
<td>6.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-fictional</td>
<td>59.38</td>
<td>14.12</td>
<td>9.38</td>
<td>16.32</td>
<td>34.38</td>
<td>6.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>88.06</td>
<td>3.69</td>
<td>40.62</td>
<td>8.44</td>
<td>64.84</td>
<td>4.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>81.25</td>
<td>6.26</td>
<td>25.00</td>
<td>11.57</td>
<td>53.12</td>
<td>6.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>96.88</td>
<td>3.12</td>
<td>56.25</td>
<td>7.83</td>
<td>76.56</td>
<td>4.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td>73.44</td>
<td>4.38</td>
<td>59.38</td>
<td>8.76</td>
<td>66.41</td>
<td>3.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>65.62</td>
<td>8.10</td>
<td>50.00</td>
<td>10.56</td>
<td>57.81</td>
<td>4.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>81.25</td>
<td>4.09</td>
<td>68.75</td>
<td>10.30</td>
<td>75.00</td>
<td>5.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td>55.51</td>
<td>3.29</td>
<td>93.75</td>
<td>4.72</td>
<td>94.55</td>
<td>2.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>95.75</td>
<td>4.09</td>
<td>96.88</td>
<td>3.12</td>
<td>95.31</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td>90.62</td>
<td>6.50</td>
<td>93.75</td>
<td>3.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>81.25</td>
<td>4.09</td>
<td>70.31</td>
<td>2.29</td>
<td>75.78</td>
<td>2.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>90.62</td>
<td>4.57</td>
<td>46.88</td>
<td>7.38</td>
<td>68.75</td>
<td>4.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>71.88</td>
<td>5.66</td>
<td>93.75</td>
<td>4.09</td>
<td>82.81</td>
<td>2.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>53.75</td>
<td>2.36</td>
<td>59.06</td>
<td>4.98</td>
<td>51.41</td>
<td>2.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>87.50</td>
<td>4.75</td>
<td>100.00</td>
<td>0.00</td>
<td>93.75</td>
<td>2.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>100.00</td>
<td>0.00</td>
<td>78.12</td>
<td>9.95</td>
<td>89.06</td>
<td>4.98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>81.51</td>
<td>2.50</td>
<td>60.94</td>
<td>3.04</td>
<td>71.22</td>
<td>1.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present Mean</td>
<td>Present S.E.</td>
<td>Absent Mean</td>
<td>Absent S.E.</td>
<td>Total Mean</td>
<td>Total S.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>57.81</td>
<td>15.11</td>
<td>-14.06</td>
<td>13.25</td>
<td>21.98</td>
<td>4.26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frictional</td>
<td>62.50</td>
<td>14.17</td>
<td>-6.25</td>
<td>17.52</td>
<td>28.12</td>
<td>6.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-frictional</td>
<td>53.12</td>
<td>16.66</td>
<td>-21.88</td>
<td>11.99</td>
<td>15.62</td>
<td>5.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>57.81</td>
<td>10.00</td>
<td>48.44</td>
<td>6.85</td>
<td>53.12</td>
<td>3.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>53.12</td>
<td>11.99</td>
<td>31.25</td>
<td>10.30</td>
<td>42.19</td>
<td>5.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>62.50</td>
<td>9.45</td>
<td>65.62</td>
<td>6.58</td>
<td>64.06</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td>45.31</td>
<td>5.25</td>
<td>71.88</td>
<td>5.15</td>
<td>58.59</td>
<td>3.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>31.25</td>
<td>6.25</td>
<td>68.75</td>
<td>6.26</td>
<td>56.00</td>
<td>3.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>59.38</td>
<td>6.58</td>
<td>75.00</td>
<td>10.56</td>
<td>67.19</td>
<td>7.06</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilation</td>
<td>100.00</td>
<td>0.00</td>
<td>51.56</td>
<td>3.69</td>
<td>75.78</td>
<td>1.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>100.00</td>
<td>0.00</td>
<td>50.00</td>
<td>6.68</td>
<td>75.00</td>
<td>3.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>100.00</td>
<td>0.00</td>
<td>53.12</td>
<td>5.66</td>
<td>76.56</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>98.75</td>
<td>5.28</td>
<td>78.12</td>
<td>4.57</td>
<td>73.44</td>
<td>2.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>81.25</td>
<td>6.25</td>
<td>65.62</td>
<td>8.10</td>
<td>73.44</td>
<td>5.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>56.25</td>
<td>7.83</td>
<td>90.62</td>
<td>6.58</td>
<td>73.44</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>90.62</td>
<td>3.12</td>
<td>67.19</td>
<td>4.69</td>
<td>78.51</td>
<td>2.24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>93.75</td>
<td>4.09</td>
<td>78.12</td>
<td>5.66</td>
<td>85.94</td>
<td>2.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>87.50</td>
<td>6.68</td>
<td>56.25</td>
<td>9.15</td>
<td>71.88</td>
<td>5.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>70.05</td>
<td>3.54</td>
<td>60.52</td>
<td>2.57</td>
<td>60.29</td>
<td>1.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

144
<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present</th>
<th>Absent</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
</tr>
<tr>
<td>Voicing</td>
<td>60.94</td>
<td>11.91</td>
<td>-23.44</td>
</tr>
<tr>
<td>Frictional</td>
<td>46.88</td>
<td>22.36</td>
<td>-3.12</td>
</tr>
<tr>
<td>Non-frictional</td>
<td>75.00</td>
<td>6.68</td>
<td>-43.75</td>
</tr>
<tr>
<td>Nasality</td>
<td>82.61</td>
<td>5.76</td>
<td>53.12</td>
</tr>
<tr>
<td>Grave</td>
<td>87.50</td>
<td>6.68</td>
<td>56.25</td>
</tr>
<tr>
<td>Acute</td>
<td>78.12</td>
<td>7.38</td>
<td>50.00</td>
</tr>
<tr>
<td>Sustentation</td>
<td>71.88</td>
<td>7.75</td>
<td>60.94</td>
</tr>
<tr>
<td>Voiced</td>
<td>71.88</td>
<td>9.95</td>
<td>40.62</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>71.88</td>
<td>7.38</td>
<td>81.35</td>
</tr>
<tr>
<td>Similation</td>
<td>95.31</td>
<td>2.29</td>
<td>82.61</td>
</tr>
<tr>
<td>Voiced</td>
<td>93.75</td>
<td>4.09</td>
<td>87.50</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td>78.12</td>
</tr>
<tr>
<td>Graveness</td>
<td>76.56</td>
<td>3.69</td>
<td>71.88</td>
</tr>
<tr>
<td>Voiced</td>
<td>87.50</td>
<td>4.72</td>
<td>50.00</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>65.62</td>
<td>9.38</td>
<td>93.75</td>
</tr>
<tr>
<td>Compactness</td>
<td>92.19</td>
<td>4.09</td>
<td>95.51</td>
</tr>
<tr>
<td>Voiced</td>
<td>87.50</td>
<td>6.68</td>
<td>36.88</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>96.88</td>
<td>3.12</td>
<td>93.75</td>
</tr>
<tr>
<td>Totals</td>
<td>79.95</td>
<td>3.41</td>
<td>56.77</td>
</tr>
</tbody>
</table>
### B.2 Automatic Speech Recognition

#### B.2.1 Diagnostic Rhyme Test

<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present Mean</th>
<th>Present S.E.</th>
<th>Absent Mean</th>
<th>Absent S.E.</th>
<th>Total Mean</th>
<th>Total S.E.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voicing</td>
<td>66.67</td>
<td>5.38</td>
<td>58.33</td>
<td>5.86</td>
<td>62.50</td>
<td>3.98</td>
</tr>
<tr>
<td>Frictional</td>
<td>70.53</td>
<td>7.20</td>
<td>54.17</td>
<td>8.58</td>
<td>62.50</td>
<td>5.63</td>
</tr>
<tr>
<td>Non-frictional</td>
<td>62.50</td>
<td>7.97</td>
<td>62.50</td>
<td>7.97</td>
<td>62.50</td>
<td>5.63</td>
</tr>
<tr>
<td>Nasality</td>
<td>92.71</td>
<td>2.71</td>
<td>87.50</td>
<td>3.49</td>
<td>90.10</td>
<td>2.21</td>
</tr>
<tr>
<td>Grave</td>
<td>93.75</td>
<td>3.55</td>
<td>81.25</td>
<td>5.95</td>
<td>87.50</td>
<td>3.49</td>
</tr>
<tr>
<td>Acute</td>
<td>91.67</td>
<td>4.08</td>
<td>93.75</td>
<td>3.55</td>
<td>92.71</td>
<td>2.71</td>
</tr>
<tr>
<td>Sustention</td>
<td>42.71</td>
<td>6.53</td>
<td>69.79</td>
<td>5.17</td>
<td>56.25</td>
<td>4.22</td>
</tr>
<tr>
<td>Voiced</td>
<td>29.17</td>
<td>9.76</td>
<td>83.33</td>
<td>5.64</td>
<td>56.25</td>
<td>4.97</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>56.25</td>
<td>8.44</td>
<td>56.25</td>
<td>8.44</td>
<td>56.25</td>
<td>5.97</td>
</tr>
<tr>
<td>Sibilant</td>
<td>73.96</td>
<td>4.96</td>
<td>92.29</td>
<td>4.10</td>
<td>78.13</td>
<td>3.19</td>
</tr>
<tr>
<td>Voiced</td>
<td>81.25</td>
<td>5.05</td>
<td>85.42</td>
<td>5.31</td>
<td>83.33</td>
<td>3.99</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>66.67</td>
<td>7.61</td>
<td>79.17</td>
<td>6.24</td>
<td>72.92</td>
<td>4.94</td>
</tr>
<tr>
<td>Graveness</td>
<td>57.29</td>
<td>5.92</td>
<td>41.67</td>
<td>6.56</td>
<td>49.48</td>
<td>4.43</td>
</tr>
<tr>
<td>Voiced</td>
<td>56.25</td>
<td>8.44</td>
<td>41.67</td>
<td>9.28</td>
<td>48.96</td>
<td>6.29</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>58.33</td>
<td>8.29</td>
<td>41.67</td>
<td>9.28</td>
<td>50.00</td>
<td>6.25</td>
</tr>
<tr>
<td>Compactness</td>
<td>84.38</td>
<td>3.87</td>
<td>69.79</td>
<td>5.17</td>
<td>77.08</td>
<td>3.25</td>
</tr>
<tr>
<td>Voiced</td>
<td>87.50</td>
<td>4.94</td>
<td>70.83</td>
<td>7.20</td>
<td>79.17</td>
<td>4.41</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>81.25</td>
<td>5.95</td>
<td>68.75</td>
<td>7.41</td>
<td>75.00</td>
<td>4.77</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>69.62</strong></td>
<td><strong>2.12</strong></td>
<td><strong>68.23</strong></td>
<td><strong>2.15</strong></td>
<td><strong>68.92</strong></td>
<td><strong>1.51</strong></td>
</tr>
</tbody>
</table>

146
<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present</th>
<th>Absent</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
</tr>
<tr>
<td>Voicing</td>
<td>69.38</td>
<td>5.81</td>
<td>46.88</td>
</tr>
<tr>
<td>Fricational</td>
<td>68.75</td>
<td>7.41</td>
<td>47.92</td>
</tr>
<tr>
<td>Non-fricational</td>
<td>50.00</td>
<td>8.84</td>
<td>45.83</td>
</tr>
<tr>
<td>Nasality</td>
<td>87.50</td>
<td>3.49</td>
<td>92.71</td>
</tr>
<tr>
<td>Grave</td>
<td>89.58</td>
<td>4.54</td>
<td>89.58</td>
</tr>
<tr>
<td>Acute</td>
<td>85.42</td>
<td>5.31</td>
<td>90.83</td>
</tr>
<tr>
<td>Sustentation</td>
<td>30.21</td>
<td>6.88</td>
<td>79.17</td>
</tr>
<tr>
<td>Voiced</td>
<td>18.75</td>
<td>10.03</td>
<td>79.17</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>41.67</td>
<td>9.28</td>
<td>79.17</td>
</tr>
<tr>
<td>Sibilization</td>
<td>83.33</td>
<td>3.99</td>
<td>81.26</td>
</tr>
<tr>
<td>Voiced</td>
<td>96.83</td>
<td>2.92</td>
<td>77.08</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>70.83</td>
<td>7.20</td>
<td>85.42</td>
</tr>
<tr>
<td>Graveness</td>
<td>53.13</td>
<td>6.11</td>
<td>56.25</td>
</tr>
<tr>
<td>Voiced</td>
<td>54.17</td>
<td>8.58</td>
<td>54.17</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>52.08</td>
<td>8.71</td>
<td>58.33</td>
</tr>
<tr>
<td>Compactness</td>
<td>88.34</td>
<td>3.35</td>
<td>73.96</td>
</tr>
<tr>
<td>Voiced</td>
<td>93.75</td>
<td>3.55</td>
<td>68.75</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>83.33</td>
<td>5.64</td>
<td>79.17</td>
</tr>
<tr>
<td>Total</td>
<td>67.01</td>
<td>2.19</td>
<td>71.70</td>
</tr>
</tbody>
</table>
Table B.18: ASR DRT scores - Whispered, lPCC Features, No adaptation

<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present</th>
<th></th>
<th></th>
<th>Absent</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
</tr>
<tr>
<td>Voicing</td>
<td>2.08</td>
<td>10.20</td>
<td>-2.08</td>
<td>10.20</td>
<td>0.00</td>
<td>7.22</td>
</tr>
<tr>
<td>Nasality</td>
<td>43.75</td>
<td>9.18</td>
<td>31.25</td>
<td>9.76</td>
<td>37.50</td>
<td>6.69</td>
</tr>
<tr>
<td>Grave</td>
<td>62.50</td>
<td>11.27</td>
<td>41.67</td>
<td>13.12</td>
<td>52.08</td>
<td>8.71</td>
</tr>
<tr>
<td>Acute</td>
<td>25.00</td>
<td>13.98</td>
<td>20.83</td>
<td>14.12</td>
<td>22.92</td>
<td>9.93</td>
</tr>
<tr>
<td>Sustentio</td>
<td>22.92</td>
<td>9.93</td>
<td>30.58</td>
<td>9.37</td>
<td>51.25</td>
<td>6.86</td>
</tr>
<tr>
<td>Voiced</td>
<td>41.67</td>
<td>13.12</td>
<td>54.17</td>
<td>12.13</td>
<td>47.92</td>
<td>8.96</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>4.17</td>
<td>14.42</td>
<td>25.00</td>
<td>13.98</td>
<td>14.52</td>
<td>10.10</td>
</tr>
<tr>
<td>Sibilalion</td>
<td>60.62</td>
<td>4.13</td>
<td>79.17</td>
<td>6.24</td>
<td>69.79</td>
<td>5.37</td>
</tr>
<tr>
<td>Voiced</td>
<td>62.50</td>
<td>11.27</td>
<td>62.50</td>
<td>11.27</td>
<td>62.50</td>
<td>7.97</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>58.33</td>
<td>11.72</td>
<td>95.83</td>
<td>4.12</td>
<td>77.08</td>
<td>6.50</td>
</tr>
<tr>
<td>Graveness</td>
<td>52.08</td>
<td>8.71</td>
<td>41.67</td>
<td>9.28</td>
<td>46.88</td>
<td>6.37</td>
</tr>
<tr>
<td>Voiced</td>
<td>58.33</td>
<td>11.72</td>
<td>37.50</td>
<td>13.38</td>
<td>47.92</td>
<td>8.96</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>45.83</td>
<td>12.83</td>
<td>45.83</td>
<td>12.83</td>
<td>45.83</td>
<td>9.07</td>
</tr>
<tr>
<td>Compactness</td>
<td>70.83</td>
<td>7.29</td>
<td>54.17</td>
<td>8.58</td>
<td>62.50</td>
<td>5.63</td>
</tr>
<tr>
<td>Voiced</td>
<td>75.00</td>
<td>9.55</td>
<td>62.50</td>
<td>11.27</td>
<td>68.75</td>
<td>7.41</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>66.67</td>
<td>10.76</td>
<td>45.83</td>
<td>12.83</td>
<td>56.25</td>
<td>8.44</td>
</tr>
<tr>
<td>Total</td>
<td>42.01</td>
<td>3.78</td>
<td>40.63</td>
<td>3.81</td>
<td>41.32</td>
<td>2.68</td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present</td>
<td>Absent</td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voicing</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fricational</td>
<td>12.50</td>
<td>14.32</td>
<td>18.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-fricational</td>
<td>-12.50</td>
<td>14.32</td>
<td>-18.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nasality</td>
<td>35.42</td>
<td>35.62</td>
<td>35.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grave</td>
<td>54.17</td>
<td>58.33</td>
<td>56.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acute</td>
<td>16.67</td>
<td>12.56</td>
<td>14.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustention</td>
<td>22.92</td>
<td>52.08</td>
<td>37.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>37.50</td>
<td>54.17</td>
<td>45.83</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>8.33</td>
<td>50.00</td>
<td>29.17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sibilant</td>
<td>64.58</td>
<td>62.50</td>
<td>63.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>58.33</td>
<td>41.67</td>
<td>50.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>70.83</td>
<td>83.33</td>
<td>77.08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graveness</td>
<td>37.50</td>
<td>29.17</td>
<td>33.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>12.50</td>
<td>14.67</td>
<td>14.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>62.50</td>
<td>41.67</td>
<td>52.08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>72.92</td>
<td>45.83</td>
<td>59.38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voiced</td>
<td>75.00</td>
<td>41.67</td>
<td>58.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unvoiced</td>
<td>70.83</td>
<td>50.00</td>
<td>60.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>38.59</td>
<td>37.50</td>
<td>38.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main Attribute</td>
<td>Present Mean</td>
<td>Present S.E.</td>
<td>Absent Mean</td>
<td>Absent S.E.</td>
<td>Total Mean</td>
<td>Total S.E.</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Voicing</td>
<td>6.25</td>
<td>10.19</td>
<td>8.33</td>
<td>10.17</td>
<td>7.29</td>
<td>7.20</td>
</tr>
<tr>
<td>Frictional</td>
<td>33.33</td>
<td>13.61</td>
<td>33.33</td>
<td>13.61</td>
<td>33.33</td>
<td>9.62</td>
</tr>
<tr>
<td>Nasality</td>
<td>52.08</td>
<td>8.71</td>
<td>66.67</td>
<td>7.61</td>
<td>59.33</td>
<td>5.81</td>
</tr>
<tr>
<td>Grave</td>
<td>75.00</td>
<td>9.55</td>
<td>75.00</td>
<td>9.55</td>
<td>75.00</td>
<td>6.75</td>
</tr>
<tr>
<td>Acute</td>
<td>29.17</td>
<td>13.81</td>
<td>58.35</td>
<td>11.72</td>
<td>43.75</td>
<td>9.18</td>
</tr>
<tr>
<td>Sustentation</td>
<td>45.83</td>
<td>9.07</td>
<td>72.92</td>
<td>6.98</td>
<td>69.38</td>
<td>5.81</td>
</tr>
<tr>
<td>Voiced</td>
<td>54.17</td>
<td>12.13</td>
<td>70.50</td>
<td>9.55</td>
<td>64.58</td>
<td>7.79</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>37.50</td>
<td>13.38</td>
<td>70.83</td>
<td>10.19</td>
<td>54.17</td>
<td>8.58</td>
</tr>
<tr>
<td>Sibilant</td>
<td>68.75</td>
<td>7.41</td>
<td>70.83</td>
<td>7.29</td>
<td>60.79</td>
<td>5.17</td>
</tr>
<tr>
<td>Voiced</td>
<td>66.67</td>
<td>10.76</td>
<td>58.33</td>
<td>11.72</td>
<td>62.50</td>
<td>7.97</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>70.83</td>
<td>10.19</td>
<td>83.33</td>
<td>7.98</td>
<td>77.08</td>
<td>6.50</td>
</tr>
<tr>
<td>Graveness</td>
<td>56.25</td>
<td>8.44</td>
<td>60.42</td>
<td>8.13</td>
<td>58.33</td>
<td>5.86</td>
</tr>
<tr>
<td>Voiced</td>
<td>54.17</td>
<td>12.13</td>
<td>66.67</td>
<td>10.76</td>
<td>60.42</td>
<td>8.13</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>58.33</td>
<td>11.72</td>
<td>54.17</td>
<td>12.13</td>
<td>54.55</td>
<td>8.44</td>
</tr>
<tr>
<td>Compactness</td>
<td>83.33</td>
<td>5.64</td>
<td>64.58</td>
<td>7.79</td>
<td>73.96</td>
<td>4.86</td>
</tr>
<tr>
<td>Voiced</td>
<td>91.67</td>
<td>5.77</td>
<td>62.50</td>
<td>11.27</td>
<td>77.08</td>
<td>6.50</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>75.00</td>
<td>9.55</td>
<td>66.67</td>
<td>10.76</td>
<td>70.83</td>
<td>7.20</td>
</tr>
<tr>
<td>Total</td>
<td>52.08</td>
<td>3.56</td>
<td>57.29</td>
<td>3.42</td>
<td>54.69</td>
<td>2.47</td>
</tr>
</tbody>
</table>

150
<table>
<thead>
<tr>
<th>Main Attribute</th>
<th>Present</th>
<th></th>
<th>Absent</th>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valency</td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
<td>S.E.</td>
<td>Mean</td>
</tr>
<tr>
<td>Voicing</td>
<td>0.00</td>
<td>10.21</td>
<td>6.25</td>
<td>10.19</td>
<td>3.13</td>
</tr>
<tr>
<td>Nasality</td>
<td>64.58</td>
<td>7.79</td>
<td>64.58</td>
<td>7.79</td>
<td>64.58</td>
</tr>
<tr>
<td>Grave</td>
<td>79.17</td>
<td>8.82</td>
<td>79.17</td>
<td>8.82</td>
<td>79.17</td>
</tr>
<tr>
<td>Acute</td>
<td>50.00</td>
<td>12.50</td>
<td>50.00</td>
<td>12.50</td>
<td>50.00</td>
</tr>
<tr>
<td>Sustention</td>
<td>29.17</td>
<td>9.76</td>
<td>79.17</td>
<td>6.24</td>
<td>54.17</td>
</tr>
<tr>
<td>Voiced</td>
<td>50.00</td>
<td>12.50</td>
<td>91.67</td>
<td>5.77</td>
<td>70.83</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>8.33</td>
<td>14.38</td>
<td>66.67</td>
<td>10.76</td>
<td>37.50</td>
</tr>
<tr>
<td>Sibilation</td>
<td>72.92</td>
<td>6.98</td>
<td>62.50</td>
<td>7.97</td>
<td>67.71</td>
</tr>
<tr>
<td>Voiced</td>
<td>70.83</td>
<td>10.19</td>
<td>41.67</td>
<td>13.12</td>
<td>56.25</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>75.00</td>
<td>9.55</td>
<td>83.33</td>
<td>7.98</td>
<td>79.17</td>
</tr>
<tr>
<td>Graveness</td>
<td>54.17</td>
<td>8.58</td>
<td>35.42</td>
<td>9.54</td>
<td>44.79</td>
</tr>
<tr>
<td>Voiced</td>
<td>54.17</td>
<td>12.13</td>
<td>37.50</td>
<td>13.38</td>
<td>45.83</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>54.17</td>
<td>12.13</td>
<td>33.33</td>
<td>13.61</td>
<td>42.75</td>
</tr>
<tr>
<td>Compactness</td>
<td>81.25</td>
<td>5.99</td>
<td>72.92</td>
<td>6.98</td>
<td>77.08</td>
</tr>
<tr>
<td>Voiced</td>
<td>87.50</td>
<td>6.99</td>
<td>70.83</td>
<td>10.19</td>
<td>79.17</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>75.00</td>
<td>9.55</td>
<td>75.00</td>
<td>9.55</td>
<td>75.00</td>
</tr>
<tr>
<td>Total</td>
<td>50.35</td>
<td>3.60</td>
<td>53.47</td>
<td>3.52</td>
<td>51.91</td>
</tr>
</tbody>
</table>
REFERENCES


52


Robert W. Morris was born in Falls Church, Virginia on October 30, 1975. He graduated from the Thomas Jefferson High School for Science and Technology in 1994. He then attended the Virginia Polytechnic Institute and State University, where he received a Bachelor of Science degree in Computer Engineering in 1998. During this period, he worked as a co-op student at the Naval Research Laboratory. He continued to graduate school at the Georgia Institute of Technology, where he completed his Master of Science in 2000. During this period, he worked as a graduate research assistant, while also taking research internships at the Intel Corporation and Fast-Talk Communications.

His primary research has been on the study of whispered speech. This has led him to interests in statistical signal processing, speech coding, speech enhancement, and speech recognition.