University of California

Los Angeles

Frequency Warping by Linear Transformation, and Vocal Tract Inversion for Speaker Normalization in Automatic Speech Recognition

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Electrical Engineering

by

Sankaran Panchapagesan

2008

The dissertation of Sankaran Panchapagesan is approved.

Nhan Levan

Lieven Vandenberghe

Ying-Nian Wu

Abeer Alwan, Committee Chair

University of California, Los Angeles

2008

Table of Contents

1 Introduction **1**

1.1 Overview and Motivation 1

1.2 Automatic Speech Recognition using Hidden Markov Models . . . 3

1.3 The EM and Baum-Welch Algorithms 5

1.4 MFCC feature extraction 7

1.5 ML HMM Adaptation based on the EM algorithm 9

1.6 Frequency Warping for VTLN 11

1.7 Frequency Warping by Linear Transformation 13

1.8 Introduction to Vocal Tract Inversion and review of previous work 16

1.9 The Maeda Articulatory Model 21

1.10 Chain matrix computation of VT acoustic response 22

1.10.1 Chain Matrix for the Sondhi model of the vocal tract . . . 24

1.11 Outline of this dissertation 25

2 Frequency Warping as Linear Transformation of Standard MFCC 26

2.1 Brief Review and Motivation 26

2.2 Derivation of the Novel LT by Warping the Log Mel Filterbank Output 29

2.2.1 Linearity of the Cepstral Transformation 29

2.2.2 Computation of the Transform Matrix 31

2.2.3 Examples of Normalized Frequency Warping Functions . . 32
2.3 Adaptation with the LT and Estimation of the FW function . . . 33
2.3.1 Transformation of Features and HMM means 33
2.3.2 Combination with MLLR Bias and Variance Adaptation . 34
2.3.3 MLS Objective Criterion 35
2.3.4 The EM Auxiliary Function 36
2.3.5 Optimizing the EM auxiliary function 38
2.4 Comparison and relationships with previous transforms 40
2.4.1 McDonough's LT 40
2.4.2 Pitz et al.'s LT 42
2.4.3 Umesh et al.'s LT 42
2.4.4 Our LT 44
2.4.5 Claes et al. and Cui and Alwan's LTs for standard MFCCs 45
2.5 Summary 46

3 Experimental Results 48
3.1 Continuous Speech Recognition Experiments 48
3.2 Comparison with other LT approximations of VTLN for standard MFCCs 53
3.3 Speaker Adaptive Modeling Experiments 54
3.4 Speaker Adaptive Training Experiments 57
3.5 Unsupervised Adaptation 60
3.6 Summary 61

4 Vocal Tract Inversion by Cepstral Analysis-by-Synthesis using Chain Matrices 63

4.1 VT Inversion by Analysis-by-Synthesis 63

4.2 Choice of Acoustic Features 64

4.3 The Articulatory-to-Acoustic Mapping 65

4.3.1 Computation of cepstra 65

4.3.2 Liftering 68

4.3.3 Log Spectral Weighting 68

4.3.4 Mel Warping 69

4.3.5 The Cepstral Distance Measure 70

4.4 The Optimization Cost Function 70

4.5 Construction and efficient search of the Articulatory Codebook . . 72

4.5.1 Codebook Construction 72

4.5.2 Codebook Search 75

4.6 Convex optimization of the cost function 76

4.7 Chain matrix derivatives with respect to the area function 77

4.8 Results of VT Inversion Experiments 80

4.8.1 Codebook Search 81

4.8.2 Results of Optimization 84

4.9 Discussion 87

4.10 Summary 89

5 Summary and Future Work 91

5.1 Summary 91

5.2 Challenges and Outlook 94

A Calculations of Derivatives for Convex Optimization in Vocal Tract Inversion 97

A.1 Derivative of the Cost Function for VT Inversion 97

A.2 Derivatives of the transfer function with respect to the area function 98

References 101

List of Figures

1.1 Standard MFCC computation. 8

1.2 The shape of the Mel filter bank shown for the case when f_s is 8kHz and the number of filters is 15. 8

1.3 VT inversion using analysis-by-synthesis. 17

1.4 Maeda articulatory model [Mae90]: dependence of midsagittal VT outline on parameters (copied from [OL05] with author's permission). The parameters are: P1 - jaw (up/down), P2 - tongue body position (front/back), P3 - tongue body shape (arched/flat), P4 - tongue tip position (up/down), P5 - lip height (up/down), P6 - lip protrusion (front/back), and P7 - larynx height (up/down). 22

1.5 Vocal tract area function, for the neutral configuration (all zero parameters) of Maeda articulatory model 23

3.1 Discrete log filterbank outputs, unwarped (dotted line) and warped, with LT VTLN (thick solid line) and Regular VTLN (thin solid line). The speech frame is from the triphone 'S-AH+B' in the word 'sub', following phoneme transcriptions in the CMU Pronouncing Dictionary 51

3.2 Histograms of warping factors in Speaker Adaptive Modeling, with Regular and LT VTLN, for 72 adult speakers from the speaker independent training data in the RM1 database 56

4.1 Articulatory-to-acoustic mapping 65

4.2 Weighting function on log spectrum used in the cepstral distance measure 69

4.3 Results of codebook search for /au/ of JW46. (a) Unrealistic articulatory trajectory for a low value of c_{reg} in Equation 4.17 (b) More realistic articulatory trajectory obtained with a larger value for c_{reg}. The four measured XRMB tongue pellet positions are plotted using solid circles while the two shifted lip pellets are represented by empty circles. 82

4.4 Example of articulatory parameters before (dotted lines) and after (solid lines) optimization. 83

4.5 Speaker JW46, (a) (first row) /ai/ from 'side' (b) (second row) /oi/ from 'soyed' (c) (third row) /au/ from 'saud' - Measured XRMB tongue (solid circles) and shifted lip (empty circles) pellet positions plotted against inverted VT outlines (solid lines). Measured palate and pharyngeal outlines are plotted using dotted lines. 85

4.6 Speaker JW46, Natural (dotted lines) and computed (solid lines) log spectra (from truncated and liftered cepstra) for /au/. The frame indices are given to the left of the vertical axis. (see corresponding formants in Figure 4.7) 86

4.7 Speaker JW46, Natural (circles) and computed (lines) formants for /au/ (see corresponding log spectra in Figure 4.6) 87

4.8 Speaker JW11, (a) (first row) /ai/ from 'side' (b) (second row) /oi/ from 'soyed' (c) (third row) /au/ from 'saud' - Measured XRMB tongue (solid circles) and shifted lip (empty circles) pellet positions plotted against inverted VT outlines (solid lines). 88

List of Tables

3.1 Recognition Accuracy in VTLN Experiments using the RM1 database. FW parameters were estimated with the MLS criterion for both methods. Baseline Accuracy: 90.16 % 50

3.2 Recognition Accuracy in VTLN Experiments with Fixed Frame-State Alignment, using the RM1 database. Baseline Accuracy: 90.16 % 52

3.3 Recognition Accuracy in Global Speaker Adaptation Experiments with limited data on the RM1 database: LT Applied in the back-end and 3-block MLLR. Baseline Accuracy: 90.16 % 52

3.4 Comparison of different LT approximations for VTLN with MFCC features, on the RM1 database. FW parameters were estimated on 1 utterance with the MLS criterion for all methods. 53

3.5 Recognition Accuracy in SAM VTLN Experiments using the RM1 database. 10 iterations of warping factor estimation were performed for each VTLN method for the training speakers and testing was performed with the corresponding method. The baseline with SAM models was the same (86.82 %) for both Regular and LT VTLN. 57

3.6 Recognition Accuracy in Global (G-) CLTFW SAT Experiments with the PL FW using the RM1 database. 10 iterations of SAT warping factor estimation were performed for the training speakers. RT denotes the use of a regression tree to estimate transforms. * indicates insufficient data to estimate further transforms. . . . 58

3.7 Recognition Accuracy in Unsupervised VTLN and Adaptation Experiments on the RM1 database using models trained with LT Speaker Adaptive Modeling. Baseline Recognition Accuracy is 86.82 % 61

3.8 Recognition Accuracy in Experiments using the RM1 database. Summary of results with different FW methods. 62

Acknowledgments

I am deeply grateful to my advisor, Dr. Abeer Alwan for her guidance and support in all my years at UCLA. I would also like to express my gratitude to Professors Nhan Levan, Lieven Vandenberghe and Ying-Nian Wu for agreeing to serve on my Ph.D. committee and for their interest in my research.

This thesis would not have been possible without the love and support of my family - my father Sankaran and mother Muthulakshmi, my brothers Kartik and Aniruddhan and my sister-in-law Usha, my cousins and extended family - Paati, Athai, Kumar, Shyamala, Nandu, Prabha, Latha, Subra, Indu, Murthy, Ram and Anu. I also thank my guru Dr. K. R. Subramanyam for his musical teaching, and his advice, and his family for many Sunday meals. Thanks go to all my friends - Markus, Guru, Shyam, Anush, Cake, and many others, for being there for me over the years, and the labmates in the Speech Lab for creating a stimulating research environment and for their friendship.

Vita

1977	Born, Chennai, India.
1998	B. Tech. in Electrical Engineering Indian Institute of Technology (IIT) Madras, Chennai, India
7-12/2001	Research and development in echo cancellation algorithms Intel Inc., Irvine, California
2003	M.S. in Electrical Engineering University of California, Los Angeles, (UCLA)
1998–2006	Graduate Student Researcher, Teaching Assistant/Associate/Fellow Electrical Engineering Department, University of California, Los Angeles (UCLA)

Publications

S. Panchapagesan and A. Alwan,"Vocal Tract Inversion by Cepstral Analysis-by-Synthesis using Chain Matrices," accepted to *Interspeech 2008*.

S. Panchapagesan and A. Alwan,"Frequency Warping for VTLN and Speaker Adaptation by Linear Transformation of Standard MFCC," *Computer Speech and Language*, vol.23, pp.42-64, 2009. To appear.

S. Panchapagesan, "Frequency Warping by Linear Transformation of Standard MFCC", *Proceedings of Interspeech 2006, ICSLP*, pp. 397-400.

S. Panchapagesan and A. Alwan, "Multi-parameter Frequency warping for VTLN by gradient search," *Proceedings ICASSP 2006*, I-1181.

Abstract of the Dissertation

Frequency Warping by Linear Transformation, and Vocal Tract Inversion for Speaker Normalization in Automatic Speech Recognition

by

Sankaran Panchapagesan

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2008

Professor Abeer Alwan, Chair

Vocal Tract Length Normalization (VTLN) for standard filterbank-based Mel Frequency Cepstral Coefficient (MFCC) features is usually implemented by warping the center frequencies of the Mel filterbank, and the warping factor is estimated using the maximum likelihood score (MLS) criterion. A linear transform (LT) equivalent for frequency warping (FW) would enable more efficient MLS estimation. In this dissertation, we present a novel LT to perform FW for VTLN and model adaptation with standard MFCC features. Our formula for the transformation matrix is computationally simpler than previous LT approaches, with no required modification of the standard MFCC feature extraction scheme. In VTLN and Speaker Adaptive Modeling (SAM) experiments with the Resource Management (RM1) database, the performance of the new LT was comparable to that of regular VTLN by warping the Mel filterbank. This demonstrates that the approximations involved in the LT do not lead to any performance degradation. We also performed Speaker Adaptive Training (SAT) with feature space LT denoted CLTFW. Global CLTFW SAT gave results comparable to SAM and

VTLN. By estimating multiple CLTFW transforms using a regression tree, and including an additive bias, we obtained significantly improved results compared to VTLN, with increasing adaptation data.

In the second part of the dissertation, vocal tract (VT) inversion to recover the VT shape sequence from speech signals is performed for vowels by cepstral analysis-by-synthesis, using chain-matrix calculation of VT acoustics and the Maeda articulatory model. The derivative of the VT chain matrix with respect to the area function was calculated in a novel efficient manner, and used in the BFGS quasi-Newton method for optimizing a cost function that includes a distance measure between input and synthesized cepstral sequences, and regularization and continuity terms. Inversion is evaluated on data from the University of Wisconsin X-ray microbeam (XRMB) database, and good agreement was achieved between inverted midsagittal VT outlines and measured XRMB tongue and lip pellet positions, with smooth optimized articulatory trajectories, and an average relative error of less than 3% in the first three formants.

CHAPTER 1

Introduction

1.1 Overview and Motivation

The study of speech production and perception have resulted in many insights that have been useful in practical applications such as speech coding and speech recognition. Vocal Tract Length Normalization (VTLN), widely used to improve the accuracy of speech recognition systems, is one such technique motivated by knowledge of speech production [KAC95].

It is known that the acoustic resonances of the vocal tract (VT) are important for the perception of both vowel and consonant speech sounds [DCP06]. For vowel sounds, the vocal tract resonances (VTRs) usually correspond to peaks in the speech signal spectrum, and are called *formants*. It is also known that the resonances of an acoustic tube are approximately inversely proportional to its length. In VTLN, acoustic mismatch between speakers caused by variation in their vocal tract lengths is reduced by scaling or warping the frequency axis of the spectrum to better align the VTRs of different speakers for a given speech sound. This spectral frequency warping (FW) or its equivalent, for VTLN, is typically performed during the extraction of acoustic features from the signal to be used for speech recognition.

VTLN has proven to be effective in improving the performance of a speech recognition system even when only limited data are available to estimate the frequency warping parameter(s) for a particular test speaker. The estimation and implementation of spectral frequency warping for VTLN have therefore received some attention in recent years.

Since the VTRs, and the acoustic characteristics of the speech signal in general, depend greatly on the shape of the vocal tract, it is clear that knowledge of the vocal tract shape would also be very useful for all the applications mentioned above. Vocal tract inversion, the problem of recovering the sequence of vocal tract shapes that produced a given speech signal, has also been a topic of research for several decades. One approach to VT inversion has been analysis-by-synthesis, where the parameters of an articulatory synthesizer are adjusted to match acoustic features computed from the speech signal. This approach leads to a better understanding of speech production and of the limitations of current production models.

Variation between speakers, in the dynamics of the VT shape during the production of the same underlying speech sounds, leads to the wide variety of observed pronunciations, dialects and accents in a given language. If mappings can be found between VTR or VT shape patterns of speakers for a given speech sound, these would be very useful in making a speech recognition system robust to speaker variations.

The goal of this research is firstly to investigate linear transform (LT) equivalents for FW, that enable efficient estimation of VTLN FW parameters. LTs also allow the estimation of multiple parameter FWs that can warp different VTRs independently to adapt different subword models in the recognition system for improved performance [McD00].

Secondly, we investigate VT inversion for vowel sounds using analysis-by-synthesis, and develop efficient methods for achieving the inversion.

1.2 Automatic Speech Recognition using Hidden Markov Models

An automatic speech recognition (ASR) system is usually divided into two main components - the *front end* and the *back end*.

In the front end, the speech signal is processed to obtain a sequence of *features*. The back end contains models of speech that are used to find the sequence of words that best accounts for the features.

The features used for speech recognition are usually designed to be robust to variations in speaker and acoustic environment. Features designed using properties of human auditory perception have been successful in practice. Mel Frequency Cepstral Coefficients (MFCCs) [DM80] and Perceptual Linear Prediction (PLP) Cepstral Coefficients [Her90] are two such features that are commonly used in recognition systems. We use MFCC features; their computation is described in Section 1.4.

The statistical approach to ASR is usually formulated as finding the word sequence $\mathbf{W}_{recog}$ that has the maximum posterior probability given the observed feature sequence $\mathbf{X}$. This is usually reformulated using Bayes' rule as follows:

$$\begin{aligned} \mathbf{W}_{recog} &= \arg\max_{\mathbf{W}} P(\mathbf{W}|\mathbf{X}) && (1.1) \\ &= \arg\max_{\mathbf{W}} P(\mathbf{X}|\mathbf{W})P(\mathbf{W}) && (1.2) \end{aligned}$$

$P(\mathbf{X}|\mathbf{W})$ and $P(\mathbf{W})$ are called the acoustic and language models respectively. In this dissertation, we are concerned more with the acoustic models.

The acoustic models usually model different speech units, for example words in a limited vocabulary system, and subwords (monophones or context dependent phoneme units) in larger vocabulary recognition systems. Hidden Markov Models (HMMs) are the most popular choice for acoustic models, though Artificial Neural Networks are also used. In our work, we consider only HMM-based speech recognition systems.

A HMM consists of a set of *states*, together with a set of probabilities of transitions between states. Each state is also associated with a probability distribution for the output (or *emission*) of feature vectors from that state. There are sometimes assumed to be two non-emitting states, one of them the initial state and the other the final state. The reason for the terminology *hidden* in the name HMM, is that in practice, the state sequence is hidden or unknown and what is known is the observation sequence.

For a HMM, let the states be numbered $1 \leq i \leq N$, the transition probability from state i to state j be a_{ij}, the observation vectors (speech features) be $\mathbf{X} = \{\mathbf{x}_t,\ 1 \leq t \leq T\}$, and the output probability density of feature vector $\mathbf{x}$ from state i be $b_j(\mathbf{x})$.

The total likelihood of the observation sequence being produced by the model with parameters Λ is easily shown to be [RJ93]:

$$P(\mathbf{X}|\Lambda) = \sum_{\Theta} \prod_{t} a_{s_t,s_{t+1}} b_{s_t}(\mathbf{x}_t) \tag{1.3}$$

where the summation is over all possible state sequences $\Theta = \{s_1, s_2, \ldots, s_T\}$.

The output probability distribution is usually taken to be a Gaussian mixture distribution

$$b_j(\mathbf{x}) = \sum_{r=1}^{R} \frac{c_{jr}}{(2\pi)^{d/2}|\Sigma_{jr}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_{jr})^T \Sigma_{jr}^{-1}(\mathbf{x}-\mu_{jr})\right] \tag{1.4}$$

where R is the number of Gaussians in the mixture and $\sum_{r=1}^{R} c_{jr} = 1$. In practice, the covariance matrices are usually taken to be diagonal for computational efficiency during recognition.

The three main problems to be solved with HRMs for ASR are:

1. Efficient computation of the likelihood of a sequence of observations given a HMM $P(\mathbf{X}|\Lambda)$:

   This problem is solved using the *Forward-Backward* Algorithm [RJ93].

2. Efficient search for the most likely state sequence for a given HMM and observation sequence; i.e.,

   $$\arg\max_{\Theta} P(\mathbf{X}, \Theta|\Lambda) \tag{1.5}$$

   This problem is solved using the *Viterbi* algorithm and used during decoding (recognition) [RJ93].

3. Estimation of the parameters of the HMMs given speech training data:

   This is usually formulated as a Maximum Likelihood parameter estimation problem, and is solved using the Baum-Welch algorithm [Bau72], which is a special case of the Expectation-Maximization (EM) algorithm [DLR77]. Discriminative training criteria such as Maximum Mutual Information (MMI) and Minimum Classification Error (MCE), can give improvements in recognition accuracy for large vocabulary continuous speech recognition [RJ93].

1.3 The EM and Baum-Welch Algorithms

During HMM training, the problem is to estimate HMM parameters given a set of utterrances along with transcriptions. Therefore, the observation sequences

along with the identities of the model sequences producing them are given, while the state sequences of the HMMs are unknown.

The Expectation Maximization (EM) algorithm is an iterative algorithm to obtain increasing-likelihood estimates of model parameters from incomplete data ([DLR77]). Following [Bil97], let the distribution $p(\mathcal{X}, \mathcal{Y}|\Lambda)$ of data $(\mathcal{X}, \mathcal{Y})$ be known, but whose parameters Λ need to estimated given only $\mathcal{X}$. In the EM algorithm, given an initial estimate of the parameters $\Lambda^{(i-1)}$, we form the *auxiliary function*

$$\mathcal{F}(\Lambda, \Lambda^{(i-1)}) = E\left[\log p(\mathcal{X}, \mathcal{Y}|\Lambda)|\mathcal{X}, \Lambda^{(i-1)}\right] \tag{1.6}$$

A new estimate of the parameters is obtained as:

$$\Lambda^{(i)} = \arg\max_{\Lambda} \mathcal{F}(\Lambda, \Lambda^{(i-1)}) \tag{1.7}$$

It can be proved that the likelihood of the observed data is non-decreasing:

$$p(\mathcal{X}|\Lambda^{(i)}) \geq p(\mathcal{X}|\Lambda^{(i-1)}) \tag{1.8}$$

If the EM algorithm converges, then the limit is a local maximizer of the likelihood function.

Given an initial estimate of the parameters of an HMM and given data that was produced from the HMM, one can use the EM algorithm to derive a new estimate of the parameters that is guaranteed to increase the likelihood. For HMMs, the parameters are $\Lambda = \{\cup_g\{c_g, \mu_g, \Sigma_g\}, [a_{ij}]\}$, where g is a Gaussian mixture distribution in the HMM. The missing data is the state sequence Θ. The auxiliary function is therefore

$$\mathcal{F}(\Lambda, \Lambda^{(i-1)}) = \sum_{\Theta} P(\mathbf{X}, \Theta|\Lambda^{(i-1)}) \cdot \log P(\mathbf{X}, \Theta|\Lambda) \tag{1.9}$$

Maximizing this auxiliary function with respect the parameters results in the Baum-Welch equations.

Let $\gamma_{jm}(t)$ be the posterior probability of being in state j at time t and the output being produced by mixture r. $\gamma_{jr}(t)$ may be computed efficiently using the forward-backward algorithm. Then the new Baum-Welch estimates of the parameters are:

$$\hat{\mu}_{jr} = \frac{\sum_{t=1}^{T} \gamma_{jr}(t)\mathbf{x}_t}{\sum_{t=1}^{T} \gamma_{jr}(t)} \tag{1.10}$$

$$\hat{\Sigma}_{jr} = \frac{\sum_{t=1}^{T} \gamma_{jr}(t)(\mathbf{x}_t - \mu_{jr})(\mathbf{x}_t - \mu_{jr})^T}{\sum_{t=1}^{T} \gamma_{jr}(t)} \tag{1.11}$$

$$c_{jr} = \frac{\sum_{t=1}^{T} \gamma_{jr}(t)}{\sum_{t=1}^{T} \sum_{l=1}^{R} \gamma_{jl}(t)} \tag{1.12}$$

The re-estimation formulae for the transition probabilities a_{ij} may be found in [RJ93, YEK].

1.4 MFCC feature extraction

Mel Frequency Cepstral Coefficients (MFCCs) [DM80] are a very popular choice of features used for automatic speech recognition. Standard MFCCs are computed as shown in Figure 1.1, and the Mel filterbank is shown in Figure 1.2. The filters are assumed to be triangular and half overlapping, with center frequencies spaced equally apart on the Mel frequency scale. The Mel scale was derived from experiments on pitch perception (frequencies which are spaced equally apart according to pitch) and is calculated from the regular frequency scale using the formula [SVN37]:

$$\text{mel}(f) = 1127 \cdot \log\left(1 + \frac{f}{700}\right) \tag{1.13}$$

During MFCC feature extraction, the speech signal is pre-emphasized and divided into frames and each frame is first windowed using the Hamming window. The short-time power spectrum vector $\mathbf{S}$ is obtained from the squared magnitude of the FFT of the windowed frame.

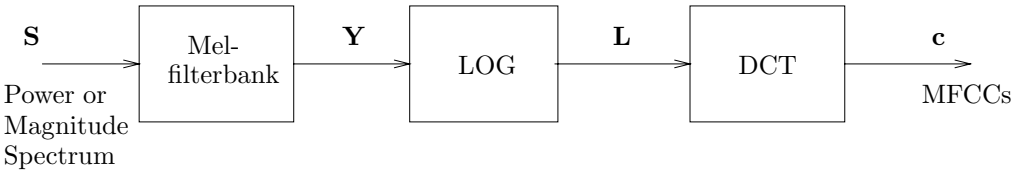

Figure 1.1: Standard MFCC computation.

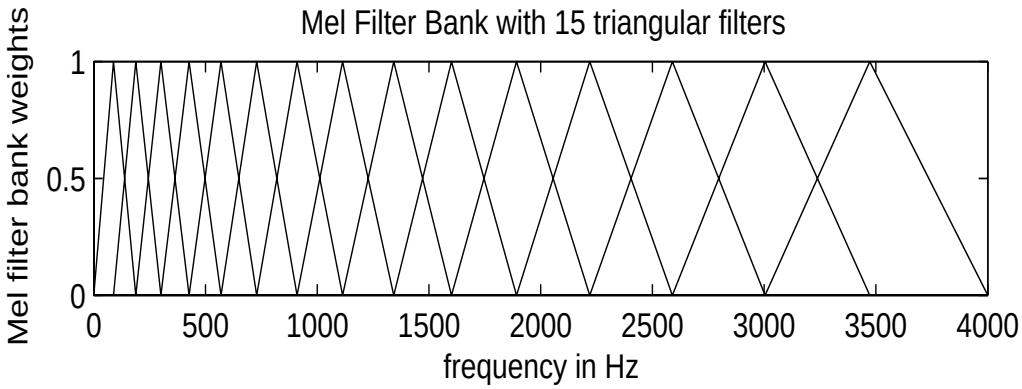

Figure 1.2: The shape of the Mel filter bank shown for the case when f_s is 8kHz and the number of filters is 15.

The log of the filterbank outputs is obtained as:

$$\mathbf{L} = \log(H \cdot \mathbf{S}) \tag{1.14}$$

where H is the Mel filterbank matrix. Here, we use the notation that the log of a vector is the log applied to each component.

The MFCCs are then given by

$$\begin{aligned} \mathbf{c} &= C \cdot \mathbf{L} && (1.15) \\ &= C \cdot \log(H \cdot \mathbf{S}) && (1.16) \end{aligned}$$

where C is a type-II DCT matrix.

We use a unitary type-II DCT matrix, for which we have $C^{-1} = C^T$, with

$$C = \left[\alpha_k \cos\left(\frac{\pi(2m-1)k)}{2M}\right)\right]_{\substack{0 \le k \le N-1 \\ 1 \le m \le M}} \tag{1.17}$$

where M is the number of filters in the filterbank, N is the number of cepstra used in the features, and

$$\alpha_k = \begin{cases} \sqrt{\frac{1}{M}}, & k = 0 \\ \sqrt{\frac{2}{M}}, & k = 1, 2, \ldots, N-1 \end{cases} \tag{1.18}$$

is a factor that ensures that the DCT is unitary. Similar expressions are valid for C and C^{-1} with a non-unitary type-II DCT matrix, but then $C^{-1} \neq C^T$ and two different sets of factors α_k and β_k would be required. Note that $N < M$ in practice. Typical values for a sampling rate of 8000Hz are $M = 26$ filters and $N = 13$.

The final feature vector $\mathbf{x}$ used for recognition, typically consists of the MFCCs and their first and second time derivatives, often called the *deltas* and *delta-deltas*:

$$\mathbf{x} = \begin{bmatrix} \mathbf{c} \\ \Delta\mathbf{c} \\ \Delta^2\mathbf{c} \end{bmatrix} \tag{1.19}$$

The delta cepstra are computed using the following formula [RJ93]:

$$\Delta\mathbf{c}_t = \frac{\sum_{k=1}^{K} k(\mathbf{c}_{t+k} - \mathbf{c}_{t-k})}{2\sum_{k=1}^{K} k^2} \tag{1.20}$$

This approximation of the time derivative is obtained by fitting a second order polynomial to a sequence of $2K + 1$ cepstral coefficients. $\Delta^2\mathbf{c}$ is similarly calculated from $\Delta\mathbf{c}$.

1.5 ML HMM Adaptation based on the EM algorithm

Speech recognition systems are usually trained on data from a large set of speakers so as to be robust to speaker variations, and also because of the practical

infeasibility of collecting large amounts of data from the potential user of the system. Therefore, much research is aimed at adapting a *speaker independent* speech recognition system to a given speaker using a limited amount of adaptation or enrollment data for that speaker.

In one formulation of stochastic matching of an ASR system to new test conditions, either the features are transformed (or *normalized*) to match the models, or the models are transformed (or *adapted*) to match the features, and the transformations may be estimated using a ML criterion and the EM algorithm [SL96]. Speaker normalization and adaptation are therefore commonly performed either by transforming features, or by transforming the means and variances of the Gaussian distributions in the HMMs.

Probably the most popular technique for speaker adaptation is Maximum Likelihood Linear Regression (MLLR), where the means and variances are transformed by ([LW95, Gal98]):

$$\hat{\mu} = A\mu + b \tag{1.21}$$

$$\hat{\Sigma} = H\Sigma H^T \tag{1.22}$$

If $H = A$, the transformation is said to be *constrained* MLLR or CMLLR, which is equivalent to feature transformation by A^{-1} [Gal98]. If H is independent of A, then the transformation is said to be *unconstrained* MLLR.

MLLR and CMLLR transforms are estimated by maximizing an EM auxiliary function. Calculating the derivatives of the auxiliary function with respect to A and H and setting them equal to zero results in a set of linear equations which can then be solved for A and H ([LW95], [Gal98]). We discuss the EM auxiliary function for CMLLR in Section 2.3.4.

The HMM distributions may be classified into a *regression tree* based on a distance measure, and individual MLLR transforms may be estimated for a node

depending on the amount of adaptation data available at the node. With limited amount of adaptation data, usually only a global transform is estimated. One may also choose to adapt only the means, and the variances may be adapted as more data becomes available. With limited adaptation data, the structure of the MLLR transformation matrix A in 1.21 may also be constrained to have a block diagonal, or n-diagonal form [YEK, CA07], as this results in fewer parameters to be estimated robustly.

1.6 Frequency Warping for VTLN

The motivating idea behind VTLN, and its implementation by spectral frequency warping was introduced in Section 1.1.

Briefly, if $X(f)$ is the speech signal spectrum, and if $w_\alpha(f)$ is the *inverse* frequency warping function to be applied, with parameter(s) α, then the warped spectrum is given by:

$$X_\alpha(f) = X(w_\alpha(f)) \tag{1.23}$$

A simple warping function is to scale the frequency axis uniformly: $w_\alpha(f) = \alpha f$, and the warped spectrum is:

$$X_\alpha(f) = X(\alpha f) \tag{1.24}$$

For MFCC features, FW for VTLN can be applied instead to the center frequencies of the filterbank [LR98], which is computationally more efficient since the warping only has to be performed once on the filterbank and not repeatedly for each frame of speech.

The parameters α controlling the FW are often estimated by optimizing a maximum likelihood (ML) criterion over the adaptation data. The ML criterion

could be the ASR likelihood score of the recognizer over the adaptation data [LR98, PMS01, PN03], the EM auxiliary function [DLR77, McD00, LNU06], or likelihoods of Gaussian mixture models (GMMs) trained specifically for FW parameter estimation [WMO96, LR98]. Another FW estimation method is by alignment of formants or formant-like spectral peaks between the test speaker and a reference speaker from the training set [GS97, CDB98, CA06].

The maximum likelihood score (MLS) criterion is commonly used for VTLN estimation [LR98, PMS01]. Here, the optimal FW parameter $\hat{p}$ is:

$$\hat{p} = \arg\max_{p} \left[\log P(\mathbf{X}^p, \Theta^p | W, \Lambda)\right] \tag{1.25}$$

where p is(are) the FW parameter(s), $\mathbf{X}^p = \{\mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_T^p\}$ is the normalized adaptation data, W is the word (or other unit) transcription, Λ are the corresponding HMMs, and Θ^p is the ML HMM state sequence with which $\mathbf{X}^p$ are aligned to Λ^p by the Viterbi algorithm during ASR decoding.

Maximizing the likelihood score is commonly performed by an exhaustive search over a grid of warping factors, when the FW is described by a single parameter that controls the scaling of the frequency axis [LR98].

Equation 1.25 is not strictly a ML criterion since the likelihood of the transformed feature vector is not normalized. The normalization factor for a given feature transformation would involve the determinant of the Jacobian matrix of the transformation. For VTLN by warping the center frequencies of the Mel filterbank for MFCCs, the transformation is not invertible, and the Jacobian matrix can not be computed.

Since the Viterbi re-alignment of utterances for each warping factor is computationally expensive, the MLS criterion is usually simplified by obtaining a frame-state alignment for the adaptation data once with unwarped features and

then maximizing the likelihood with a fixed alignment to estimate the warping parameters p [ZW97]. The simplified MLS objective function is:

$$\mathcal{F}(p) = \sum_{t=1}^{T} \log \left(\sum_{r=1}^{R} c_{tr} \mathcal{N}(\mathbf{x}_t^p; \mu_{tr}, \Sigma_{tr}) \right) \tag{1.26}$$

where $\sum_{r=1}^{R} c_{tr} = 1$ for the mixture Gaussian state output distribution at time t. A gradient search or quasi-Newton method may be used to optimize the simplified MLS objective function for multiple FW parameters [PA06].

1.7 Frequency Warping by Linear Transformation

Frequency warping of the spectrum has been shown to correspond to a linear transformation of cepstra [MBL98, PMS01]. This confers some advantages for speech recognition systems that use cepstral features.

- Firstly, one can apply the linear transform for a warping factor to previously computed unwarped features and not have to recompute features with different warp factors during VTLN estimation by MLS. This results in significant computational savings [UZN05], which would be important in embedded and distributed speech recognition (DSR) applications, where resources are limited. Given the recognition alignment of an utterance obtained with baseline models without VTLN, it can be shown by a rough calculation that parameter estimation for Regular VTLN is about 2.5 times as expensive as for LT VTLN, when the fixed alignment is used for VTLN estimation with the MLS criterion, with single Gaussian mixture HMMs and a grid search.

- The linear transform approach also has the advantage that one need not have access to any of the intermediate stages in the feature extraction dur-

ing VTLN estimation. This aspect would have definite advantages in DSR, where feature extraction is performed at the terminal and recognition is performed at the server. During VTLN estimation using a grid search over warping factors, since it would be impractical for the client to recompute and transmit features for each warping factor, warped features would have to be computed at the server. With a linear transform, only the cepstral transformation matrices for each warping factor need to be applied to unwarped features to choose the best warping factor, while with VTLN by spectral warping, the linear frequency spectrum needs to be reconstructed and the warped features recomputed for each warping factor.

- The linearity also enables one to take the expectation and thereby apply the linear transformation to the means of HMM distributions [CDB98, MB99]. Different transforms could then be estimated for different phonemes or classes of HMM distributions, unlike VTLN where the same global transformation is applied to all speech features. This can result in significantly improved recognition results [McD00, CA06, WCA07].

The equivalence of FW to linear transformation, though true also for cepstral features which are based on Perceptual Linear Prediction (PLP) or by Mel warping of the frequency axis [McD00, PN03], does not hold exactly for standard MFCC features computed using a filterbank and the DCT (Section 1.4). In fact, because of the non-invertible filterbank with non-uniform filter widths, even with the assumption of quefrency limitedness, the MFCC features after warping cannot even be expressed as a function (linear or non-linear) of the unwarped MFCC features. i.e., for a given warping of the linear frequency signal spectrum, there is not a single function (for all possible cepstra) that will give the warped cepstra from the unwarped cepstra. Hence, approximate linear transforms have

been developed for FW with MFCC features [CDB98, CA06, UZN05].

Claes et al. [CDB98] were the first to derive an approximate linear transform which was used to perform model adaptation with some success. Cui and Alwan [CA05, CA06] derived a simpler linear transform that is essentially an "index mapping" on the Mel filterbank outputs, i.e. one filterbank output is mapped to another. In fact, it may be shown to be mathematically a special case of Claes et al.'s transform (see Section 2.1) but was demonstrated to give better performance [CA05]. In both [CDB98] and [CA06], the FW was estimated by alignment of formants or formant-like peaks in the linear frequency domain.

[UZN05] showed that the formula for computing the linear transform for ordinary cepstra, derived in [PMS01], could be considerably simplified under the assumption of quefrency limitedness of the cepstra, when the log spectrum can be obtained from samples by sinc interpolation. They also developed non-standard filterbank based MFCC features, to which the linear transformation was extended. In their modified filterbank, the filter center frequencies were uniformly spaced in the linear frequency domain but filter bandwidths were uniform in the Mel domain. Their transformation formula (discussed further in Section 2.4) was, however, complicated by the use of two different DCT matrices, one for warping purposes and the other for computing the cepstra.

We proposed a novel LT to perform FW for VTLN and model adaptation with standard MFCC features in [Pan06, PA09]. The formula for our LT matrix is computationally simpler and unlike other previous linear transform approaches to VTLN with MFCC features, no modification of the standard MFCC feature extraction scheme is required. The mathematical derivation of our LT and interrelationships between different LTs for FW are presented in Chapter 2.

1.8 Introduction to Vocal Tract Inversion and review of previous work

Acoustic-to-articulatory inversion or vocal tract (VT) inversion is the problem of obtaining the vocal tract shapes that produced a given input speech signal. Potential benefits of successful VT inversion include the use of inverted articulatory parameters for efficient speech coding and improved speech recognition, computer-aided language learning using inverted VT outlines, and improved understanding of speech production, e.g. coarticulation.

Data-driven acoustic-to-articulatory inversion methods based on Artificial Neural Networks, Kalman filters, Mixture Density Networks and Hidden Markov Models have become popular in recent years [AR89, RGK93, PHT92, Dus00, Ric01, HH04]. These methods typically rely on simultaneously measured acoustic and articulatory data to train their respective models, which are then used to perform acoustic-to-articulatory mapping from acoustic data alone.

Here, we focus instead on analysis-by-synthesis methods where inversion is performed by adjusting the parameters of an articulatory synthesizer to match acoustic features computed from the input speech [ACM78, FIS80, SK86, SS94]. Such methods would lead to a better understanding of the speech process, and and help in improving current speech production models. An introduction and overview of several techniques may be may be found in [SS92, SS94, Rie97].

Figure 1.3 shows a block diagram of the different steps typically involved in VT inversion using analysis-by-synthesis.

The challenges faced in VT inversion by analysis-by-synthesis are:

(1) Complexity of speech production models

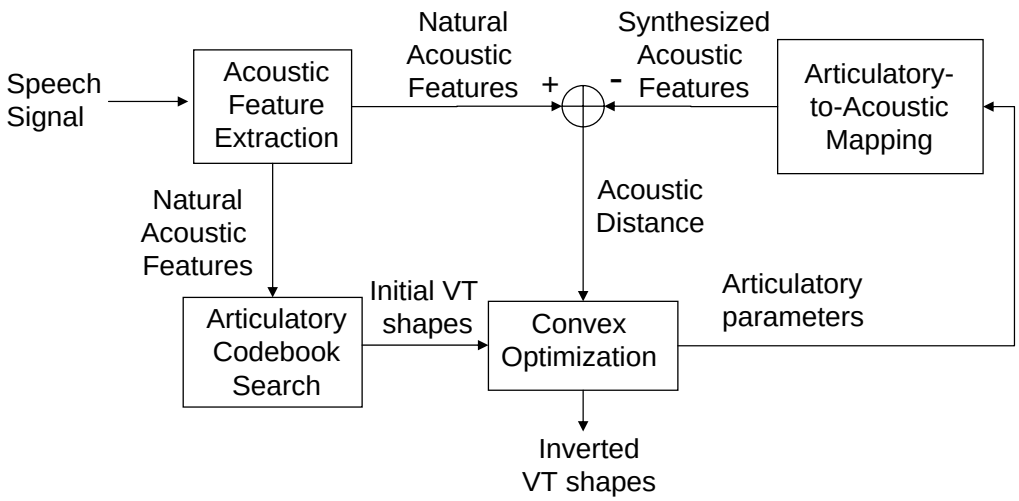

Figure 1.3: VT inversion using analysis-by-synthesis.

Since the articulatory-to-acoustic or *forward* mapping in the loop of Figure 1.3 is computationally expensive, efficient techniques need to be developed for optimizing articulatory parameters.

(2) Inherent non-uniqueness of the inverse mapping, and local optima of the cost function

It has been analytically shown that for a lossless acoustic tube, the same set of formant frequencies may be produced by an infinite number of different area functions [Sch67, Mer67]. For a vocal tract consisting of a series of uniform tubes modelled as lumped transmission lines, it has also been computationally verified that different area functions can produce identical first three formant frequencies and amplitudes with different formant bandwidths [ACM78]. Techniques found to be useful in resolving the non-uniqueness and local optima issues of the inverse problem are: use of articulatory models to constrain the vocal tract area function, inclusion of regularization and continuity terms in the cost function, and initialization using articulatory-acoustic linked codebooks [Rie97, SS92, SS94, ACM78,

Sor92, SK86].

(3) Incomplete knowledge about the shape and dynamics of the vocal tract for a given speaker, and

(4) Insufficient data to learn from or to evaluate the inversion results.

The main issues involved are therefore: choice of acoustic features, the articulatory-to-acoustic mapping used, the cost function to be optimized, construction and search of articulatory codebooks to initialize the optimization, the optimization techniques used, and evaluation of inversion results.

For vowels, the first three formant frequencies are important for the perception of vowel quality, and acoustic distance measures between natural and synthesized formants are often minimized [Sor92, OL05]. Cepstral distance measures are also useful and very flexible since the effects of peak emphasis, log spectral weighting and frequency warping can all be accounted for by simple linear weighting (liftering) and/or filtering of the cepstra [SK86, JRW87, SMP90]. Among a set of spectral distance measures, a cepstral distance was found, in [SK86], to give best performance when inverted articulatory parameters were used for vowel recognition. A liftered cepstral distance was also found to be most effective for searching articulatory codebooks (discussed below) [SMP90].

The first way to decrease non-uniqueness is to use articulatory models to constrain the area function to be similar to those from human talkers. The Mermelstein [Mer73] and Maeda [Mae90] articulatory models, describe the vocal tract midsagittal outline and area function using a relatively small number of parameters (10 for the Mermelstein model and 7 for the Maeda model) which control the shapes and positions of articulators such as the jaw, tongue, lips and larynx.

The non-uniqueness of the inverse solution can also be resolved by including regularization and continuity terms in the optimization cost function [Sor92, SK86, SS92, OL05]. The regularization term is designed to discourage vocal tract configurations farther from the mean or neutral position, and usually takes the form of the sum of squares of articulatory parameters minus their nominal values [Sor92, OL05]. The continuity term in the cost function can be the geometric distance from the articulatory parameters of the previous frame in the case of a frame-wise optimization [SS92], or sum of squares of the first time-derivatives of articulatory parameters over several frames, in the case of a global optimization over the speech segment [OL05]. The continuity terms are also useful in obtaining smoother trajectories for inverted articulatory parameters, which are desirable since human articulation is controlled by muscles of finite power and therefore human articulatory trajectories would necessarily be smooth.

An articulatory codebook is used to initialize the optimization, because of computationally intensive forward mapping, and to deal with the problem of local optima of the cost function [ACM78, SS92, SS94, Rie97]. The codebook consists of articulatory vectors and corresponding acoustic feature vectors computed using the forward mapping. The codebook is usually designed to cover both the articulatory and acoustic spaces well while having a low redundancy, and there is a tradeoff between codebook size and its resolutions in the articulatory and acoustic spaces. The issues involved in the design and search of codebooks are discussed in greater detail in [SS92, Rie97]. Codebooks specially constructed by dividing the articulatory parameter space into hypercube regions within which the articulatory-acoustic mapping is approximately linear, have also been used to obtain inverse solutions [OL05]. Since the cost function includes continuity terms, the codebook search involves dynamic programming [SS92, OL05].

Different techniques have been used for more refined optimization of the cost function after codebook initialization. These include direct search methods like the Hooke-Jeeves and coordinate descent methods which do not require the gradient of the cost function [FIS80, SS92, Sor92], gradient based methods [SK86], and iterative solutions of variational equations [OL05]. A finite difference approximation may be used for the gradient of the formants with respect to articulatory parameters [OL05] and gradients may be precomputed at each codevector in the case of the hypercube codebook where hypercube regions are identified around each code vector in which the articulatory-acoustic mapping is approximately linear [OL05]. Genetic algorithms can also be used to optimize the cost function without using a codebook for initialization [McG94].

If the goal of inversion is recovery of actual vocal tract shapes, then inverted vocal tract shapes would need to be compared against actual measured shapes. Inverted tongue outlines for static vowels and fricatives have been compared against x-ray microbeam measurements of gold pellets placed on the tongue [Sor92, ST96]. Examples of measured articulatory data along with simultaneously recorded acoustic data, that are publicly available, include the Edinburgh Multi-CHannel Articulatory (MOCHA) database [Moc] and the X-Ray Microbeam (XRMB) Speech Production Database from the University of Wisconsin, Madison [Wes94]. The MOCHA database includes data from Electromagnetic Articulography (EMA), where positions of coils placed at different points on the jaw, tongue and lips are measured during speech production. In both the XRMB and MOCHA databases, no information is available on the vocal tract in the pharyngeal region since all XRMB pellets or EMA coils were placed either in the oral cavity or on the face. However, except for the larynx, some information is available on the positions of all the other important articulators (jaw, tongue body and tip, lips). A reasonable geometric error measure for inversion

can therefore be obtained by comparing inverted VT outlines against measured positions of tongue and lip XRMB pellets. The available geometric information may also give clues as to the weights or constraints that need to be placed on the displacements of different articulators in order to more accurately recover the VT shape for a particular speaker and speech sound.

1.9 The Maeda Articulatory Model

The Maeda articulatory model was derived from a factor analysis of around 1000 frames of cineradiographic and labiofilm images of the vocal tracts of two speakers uttering ten French sentences [Mae90]. In this model, the exterior midsagittal VT outline consisting of the hard and soft palates, velum and rear pharyngeal walls is fixed for a speaker. The interior VT outline is controlled by seven parameters: jaw position, tongue body shape and position, tongue tip position, lip height and width, and larynx height as shown in Figure 1.4. The VT outlines are described using a system of semi-polar grid lines, and the offsets of the interior VT outline along the grid lines are obtained as a linear combination of basis offset vectors obtained from the factor analysis mentioned above.

Midsagittal widths $d(x)$ along the length of the tract x, are converted to areas using the heuristic formula [Mae90, HS64]:

$$A(x) = \alpha(x) d(x)^{\beta(x)} \tag{1.27}$$

where $\alpha(x)$ and $\beta(x)$ are *ad hoc* coefficients that vary along the tract. Using the semi-polar grid, the area function is obtained as a sequence of varying areas and lengths of 29 uniform tubes. The lengths of the tube sections in the area function are the distances between the midpoints of consecutive midsagittal grid line segments between the exterior and interior VT outlines. Figure 1.5 shows the

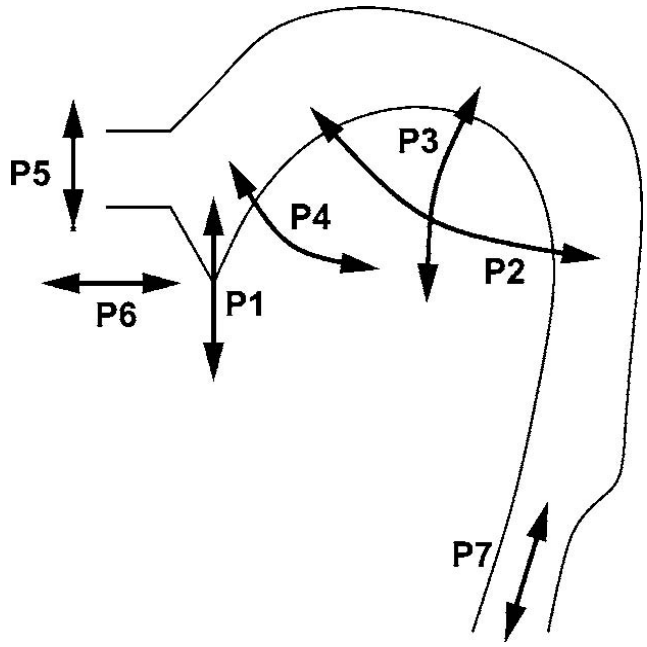

Figure 1.4: Maeda articulatory model [Mae90]: dependence of midsagittal VT outline on parameters (copied from [OL05] with author's permission). The parameters are: P1 - jaw (up/down), P2 - tongue body position (front/back), P3 - tongue body shape (arched/flat), P4 - tongue tip position (up/down), P5 - lip height (up/down), P6 - lip protrusion (front/back), and P7 - larynx height (up/down).

area function corresponding to the neutral configuration (all zero parameters) of the Maeda model.

1.10 Chain matrix computation of VT acoustic response

The chain matrix method is one of the preferred approaches for computing the acoustic response of the vocal tract given its area function [SS87, SS92]. Here, the pressure, P, and volume velocity, U, at the input and output of an acoustic

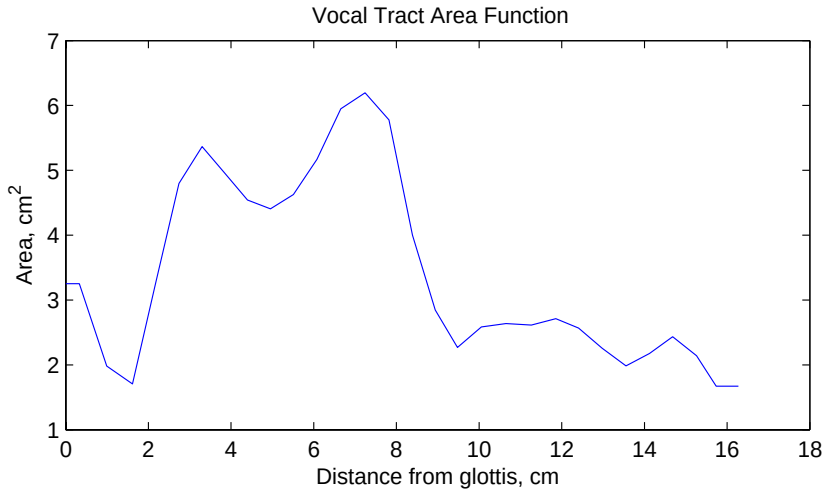

Figure 1.5: Vocal tract area function, for the neutral configuration (all zero parameters) of Maeda articulatory model

tube, for a linear wave, are related in the frequency domain by:

$$\begin{pmatrix} P_{out} \\ U_{out} \end{pmatrix} = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{pmatrix} \begin{pmatrix} P_{in} \\ U_{in} \end{pmatrix} \tag{1.28}$$

where the subscripts *in* and *out* denote the input and the output of the tube respectively. $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are referred to as the chain or transmission parameters of the tube, and the matrix formed is called the chain matrix (CM).

The computational convenience of using the chain matrix to characterize a tube arises from the property that the overall chain matrix of a concatenation of two tubes is just the product of the individual chain matrices. If the vocal tract for a vowel sound is approximated as the concatenation of N uniform tubes starting at the glottis and ending at the lips, and if K_n is the chain matrix of the nth tube, the overall chain-matrix of the vocal tract would then be:

$$K = K_N \cdot K_{N-1} \cdot \cdots \cdot K_1 \tag{1.29}$$

The transfer function of the vocal tract may then be shown to be:

$$H(s) = \frac{U_L(s)}{U_G(s)} = \frac{1}{(\mathcal{A} - \mathcal{C}Z_L)} \tag{1.30}$$

where U_G and U_L are the volume velocities at the glottis and lips, respectively, $\mathcal{A}$ and $\mathcal{C}$ are the elements of the chain-matrix of the overall vocal tract, and Z_L is the radiation impedance at the lips. Z_L is often approximated by that of a pulsating sphere with a radius equal to that of the mouth opening [Fla72]:

$$Z_L = \frac{\rho\omega^2}{2\pi c} + j\frac{8\rho\omega}{3\pi^2 r} \tag{1.31}$$

where $\omega = 2\pi f$ and r is the radius of the lip opening.

1.10.1 Chain Matrix for the Sondhi model of the vocal tract

In our work, we follow [SS94, SS87, SS92] in using the Sondhi model [Son74] for wave propagation in a vocal tract, where frequency dependent losses due to air viscosity, heat conduction and yielding tract walls are taken into account. For this model the chain matrix parameters of a uniform lossy cylindrical tube of area A (not to be confused with the chain-matrix parameter $\mathcal{A}$) and length L at a given angular frequency ω are given by ([SS87]):

$$\mathcal{A} = \cosh(\sigma L/c) \tag{1.32}$$

$$\mathcal{B} = -\frac{\rho c}{A}\,\gamma\,\sinh(\sigma L/c) \tag{1.33}$$

$$\mathcal{C} = -\frac{A}{\rho c}\,\frac{\sinh(\sigma L/c)}{\gamma} \tag{1.34}$$

$$\mathcal{D} = \cosh(\sigma L/c) \tag{1.35}$$

where ρ is the density of air, and c is the speed of sound in air. Details on the values of the different parameters and the formulae for calculating γ and σ are given in [SS87]. The important thing to be noted is that γ and σ are only functions of ω and do not depend on the area or the length of the tube.

The chain matrix and the transfer function are typically computed for a set of equally spaced frequencies, and these may be used to compute further quantities of interest like the all-pole LPC spectral envelope and formant frequencies. The values of formants computed using the chain matrix method with area functions obtained by magnetic resonance imaging (MRI), have been verified to lie within 5-10% of their actual values obtained by analyzing recorded signals from the same speaker for static speech sounds [STH96].

1.11 Outline of this dissertation

The rest of the dissertation is organized as follows.

Chapter 2 presents our novel linear transform (LT) equivalent of frequency warping (FW) for the standard Mel frequency cepstral coefficient (MFCCs) features. The matrix for the new LT is derived and compared with previous approaches. The estimation of FWs using MLS and EM auxiliary functions as objective criteria is considered and formulae for convex optimization of the EM auxiliary function for multiple FW parameters are derived.

Chapter 3 discusses results of speech recognition experiments using the LT presented in Chapter 2 for VTLN and adaptation.

In Chapter 4 we describe our approach for vocal tract inversion by cepstral analysis-by-synthesis using chain matrices. Methods of optimization of the cost function for inversion are discussed along with a novel efficient calculation of derivatives of the VT chain matrix with respect to its area function.

Finally, Chapter 5 presents a summary of the dissertation and future research directions.

CHAPTER 2

Frequency Warping as Linear Transformation of Standard MFCC

In this chapter, the mathematical derivation of our novel Linear Transform (LT) for frequency warping (FW) with MFCC features is presented, and the formula for computing the LT for any given FW function and parameter is expressed in a simple and compact form. Our LT is compared with other LTs earlier proposed for FW with cepstral features and their inter-relationships are clarified. We also discuss FW parameter estimation using MLS and EM auxiliary function as objective criteria, and optimization of the EM auxiliary function.

2.1 Brief Review and Motivation

In Section 1.7 we discussed Claes et al.'s [CDB98] and Cui and Alwan's [CA06] derivation of approximate LTs for FW with MFCCs, using approximations for the Mel filterbank matrix and its inverse. The two approximate transforms are briefly derived to motivate the development of our LT.

As discussed in Section 1.6, VTLN for standard MFCC features is commonly

implemented by warping the center frequencies of the filterbank [LR98]. For this discussion, we consider direct frequency warping of the spectrum, $\mathbf{S}$ (Section 1.4). Since the filterbank H and DCT C result in significant reduction of dimensionality and are non-invertible, $\mathbf{S}$ can only be approximately recovered from the Mel cepstrum $\mathbf{c}$:

$$\mathbf{S} \approx H^{-1} \cdot \exp(C^{-1}\mathbf{c}) \tag{2.1}$$

where H^{-1} and C^{-1} are approximate inverses of H and C respectively. A (partial) IDCT matrix is a natural choice for C^{-1}, while different choices have been proposed for H^{-1} by [CDB98] and [CA06], as discussed below.

Between the two approximate inverse operations, the application of C^{-1} is less severe since it only corresponds to a smoothing of the log filterbank output by cosine interpolation. Since the spectrum is already smoothed and warped by the Mel filterbank operation, the cepstral truncation and application of C^{-1} would result in the recovery of a reasonable Mel-warped log spectrum which can be used for further VTLN warping. The FFT spectrum recovered using an approximate filterbank inverse H^{-1}, however, would probably only be a gross approximation of the original FFT spectrum since there is large dimensionality reduction due to application of H (256×26 in our case). However, the use of a particular choice of H^{-1} to perform VTLN warping can be empirically justified by the improvement in recognition results.

By applying a warping W to the approximate linear spectrum $\mathbf{S}$ from Equation 2.1 and recomputing Mel cepstra, a non-linear FW transform for MFCCs may be derived as in [CDB98]:

$$\hat{\mathbf{c}} = C \cdot \log\{H \cdot W \cdot H^{-1} \cdot \exp(C^{-1}\mathbf{c})\} \tag{2.2}$$

Claes et al. [CDB98] also showed that for small frequency scaling factors, the non-linear cepstral transformation of Equation 2.2 may be approximately

linearized to:

$$\hat{\mathbf{c}} \approx (C\bar{B}C^{-1}) \cdot \mathbf{c} + C\mathbf{d} \tag{2.3}$$

where $\bar{B}$ is the matrix obtained from $B = H \cdot W \cdot H^{-1}$ by normalizing each row of B so that the sum of the elements in each row is 1: $\bar{B}(i,j) = B(i,j)/\sum_j B(i,j)$, and $\mathbf{d}(i) = \log \sum_j B(i,j)$. For the choice of H^{-1}, [CDB98] used a special matrix M that satisfied $HM = I$, and which was found to give better results than just using the pseudo-inverse of H.

Cui and Alwan [CA06] obtained a transform that has a simpler form than that in Equation 2.3, and was shown to give even better results, by approximating H, W and H^{-1} in Equation 2.2 by carefully chosen index mapping (IM) matrices, which are matrices in which each row contains only one nonzero element which is 1. Then, $B = H \cdot W \cdot H^{-1}$ is also an IM matrix, and the exponential and the logarithm in Equation 2.2 cancel each other out [CA06]. The cepstral transformation then becomes linear:

$$\hat{\mathbf{c}} = (CHWH^{-1}C^{-1}) \cdot \mathbf{c} \tag{2.4}$$

In fact, when B is an IM matrix, $\bar{B} = B$ and $\mathbf{d} = 0$ in Equation 2.3, and Equation 2.3 also reduces to Equation 2.4. Cui and Alwan's linear transform is therefore mathematically a special case of Claes et al's transform.

We can rewrite Equation 2.4 as

$$\hat{\mathbf{c}} = C \cdot \hat{\mathbf{L}} \tag{2.5}$$

where

$$\hat{\mathbf{L}} = HWH^{-1} \cdot \mathbf{L} = B \cdot \mathbf{L} \tag{2.6}$$

with

$$\mathbf{L} \approx C^{-1}\mathbf{c} \tag{2.7}$$

Considered from the point of view of the log Mel filterbank output $\mathbf{L}$, since B is an IM matrix, we can see from Equation 2.6 that Cui and Alwan's transform therefore amounts to an index mapping.

In [CA06], the warping W was estimated by alignment of formant-like peaks in the linear frequency spectrum $\mathbf{S}$, and the cepstral linear transform was demonstrated to give excellent results when used for model adaptation. This raises the possibility of obtaining the same success by estimating and applying warping directly on the log Mel spectrum $\mathbf{L}$ without reconstructing the linear frequency spectrum $\mathbf{S}$ using an approximate inverse of the filterbank. This simplifies the warping transform, and also has other advantages over peak alignment as discussed in Section 2.4.5.

We will next discuss how to implement and estimate continuous warping on $\mathbf{L}$, the log Mel filterbank output, and show that it naturally results in a linear transformation on the MFCCs.

2.2 Derivation of the Novel LT by Warping the Log Mel Filterbank Output

2.2.1 Linearity of the Cepstral Transformation

Equation 2.7 describes how the smoothed log filterbank output may be approximately recovered from the truncated cepstra using the IDCT. For a unitary DCT matrix as in Equation 1.17, $C^{-1} = C^T$, and Equation 2.7 therefore becomes $\mathbf{L} = C^{-1}\mathbf{c} = C^T\mathbf{c}$ (the approximation being understood implicitly). This may be written in expanded form as

$$\mathbf{L}(m) = \sum_{k=0}^{N-1} \mathbf{c}(k)\alpha_k \cos\left(\frac{\pi(2m-1)k}{2M}\right),\ m = 1, 2, \ldots, M \tag{2.8}$$

where $\mathbf{c}(k), k = 0, 1, \ldots, N-1$, are the MFCCs.

Using the idea of cosine interpolation one can consider the IDCT approximation of Equation 2.8 to describe a continuous log Mel spectrum $L(u)$, where u is a continuous (scaled) Mel frequency variable:

$$L(u) = \sum_{k=0}^{N-1} \mathbf{c}(k)\alpha_k \cos\left(\frac{\pi(2u-1)k}{2M}\right) \tag{2.9}$$

with

$$\mathbf{L}(m) = L(u)|_{u=m}, m = 1, 2, \ldots, M \tag{2.10}$$

We can now apply continuous warping to u. Let us take the *inverse* of the warping function to be applied, to be $\psi(u)$. The warped continuous log Mel spectrum is then:

$$\hat{L}(u) = L(\psi(u)) \tag{2.11}$$

The warped *discrete* log filterbank output is obtained by sampling $\hat{L}(u)$:

$$\begin{aligned}
\hat{\mathbf{L}}(m) &= \hat{L}(u)|_{u=m},\ m = 1, 2, \ldots, M && (2.12)\\
&= L(\psi(u))|_{u=m},\ m = 1, 2, \ldots, M && (2.13)\\
&= \sum_{k=0}^{N-1} \mathbf{c}(k)\alpha_k \cos\left(\frac{\pi(2\psi(m)-1)k}{2M}\right),\ m = 1, 2, \ldots, M && (2.14)
\end{aligned}$$

by Equations 2.11 and 2.9.

Therefore, in vector form,

$$\hat{\mathbf{L}} = \tilde{C} \cdot \mathbf{c} \tag{2.15}$$

where $\tilde{C}$ is the *warped IDCT matrix*:

$$\tilde{C} = \left[\alpha_k \cos\left(\frac{\pi(2\psi(m)-1)k}{2M}\right)\right]_{\substack{1 \le m \le M \\ 0 \le k \le N-1}} \tag{2.16}$$

The transformed MFCCs are given by

$$\begin{aligned}
\hat{\mathbf{c}} &= C\ \hat{\mathbf{L}} = (C\tilde{C})\ \mathbf{c}\\
&= T\ \mathbf{c} && (2.17)
\end{aligned}$$

Hence, the MFCCs corresponding to the warped log Mel spectrum are naturally obtained by a linear transformation of the original MFCCs, and the transformation matrix is given by

$$T = C\tilde{C} \tag{2.18}$$

where $\tilde{C}$ is the warped IDCT matrix given in Equation 2.16.

2.2.2 Computation of the Transform Matrix

In the above derivation, one needs to specify the warping $\psi(u)$ before the transform matrix can be computed from Equations 1.17, 2.16 and 2.18. The first detail is the range of values that u can take. $L(u)$ as described in Equation 2.9 above is periodic with a period of $2M$, and is symmetric about the points $u = \frac{1}{2}$ and $u = M + \frac{1}{2}$. Therefore, the range of u to be be warped is $\frac{1}{2} \leq u \leq M + \frac{1}{2}$.

Frequency warping functions on u may be obtained using a normalized frequency variable λ with $0 \leq \lambda \leq 1$. We can pass from the continuous Mel domain u to the normalized frequency domain λ, and vice versa, by the affine transformations:

$$u \;\rightarrow\; \lambda = \frac{u - 1/2}{M}, \quad \frac{1}{2} \leq u \leq M + \frac{1}{2} \tag{2.19}$$

$$\lambda \;\rightarrow\; u = \frac{1}{2} + \lambda M, \quad 0 \leq \lambda \leq 1 \tag{2.20}$$

Let $\theta_p(\lambda)$ be a normalized FW function controlled by parameter(s) p (see Equations 2.26, 2.27 and 2.28 for examples). The only practical constraint required for $\theta_p(\lambda)$ to be usable is that $0 \leq \theta_p(\lambda) \leq 1$ for $0 \leq \lambda \leq 1$. Then we can obtain a warping $\psi(u) = \psi_p(u)$ on u, using

$$\psi_p(u) = \frac{1}{2} + M \cdot \theta_p\left(\frac{u - 1/2}{M}\right) \tag{2.21}$$

Note that if $\lambda = 0$ and $\lambda = 1$ are fixed points of $\theta_p(\lambda)$ (i.e. $\theta_p(0) = 0$ and

$\theta_p(1) = 1$), then $u = \frac{1}{2}$ and $u = M + \frac{1}{2}$ are fixed points of $\psi_p(u)$.

By Equation 2.21,

$$\frac{2\psi_p(u) - 1}{2M} = \theta_p\left(\frac{2u-1}{2M}\right) \tag{2.22}$$

and the warped IDCT matrix of Equation 2.16 can be rewritten as:

$$\tilde{C}_p = \left[\alpha_k \cos\left(\pi k\ \theta_p\left(\frac{2m-1}{2M}\right)\right)\right]_{\substack{1\le m\le M \\ 0\le k\le N-1}} \tag{2.23}$$

Comparing Equations 2.17 and 2.18 with Equation 2.4, we see that the warping of the log Mel spectrum has been embedded into the IDCT matrix. In fact, if we let $\lambda_m = \frac{2m-1}{2M}$ for $1 \le m \le M$, then Equations 1.17 and 2.23 may be rewritten as:

$$C^T = \left[\ \alpha_k \cos\left(\pi k\ \lambda_m\right)\ \right]_{\substack{1\le m\le M \\ 0\le k\le N-1}} \tag{2.24}$$

$$\tilde{C}_p = \left[\ \alpha_k \cos\left(\pi k\ \theta_p\left(\lambda_m\right)\right)\ \right]_{\substack{1\le m\le M \\ 0\le k\le N-1}} \tag{2.25}$$

This last equation shows clearly the simplest way of computing the warped IDCT matrix for a given normalized warping function $\theta_p(\lambda)$ and warping parameter p. We next look at some examples for $\theta_p(\lambda)$.

2.2.3 Examples of Normalized Frequency Warping Functions

1. *Piecewise Linear:* These are the type of FW functions that are commonly used in VTLN [WMO96, PMS01].

$$\theta_p(\lambda) = \begin{cases} p\lambda, & 0 \le \lambda \le \lambda_0 \\ p\lambda_0 + \left(\frac{1-p\lambda_0}{1-\lambda_0}\right)(\lambda - \lambda_0), & \lambda_0 < \lambda \le 1 \end{cases} \tag{2.26}$$

   where λ_0 is a fixed reference frequency, around 0.7 in our experiments.

2. *Linear:* This FW can be used for adaptation from adult models to children's models, where the original models have more spectral information than necessary for children's speech [CA06, Pan06].

   For $p \leq 1$,

$$\theta_p(\lambda) = p\lambda, \ 0 \leq \lambda \leq 1 \tag{2.27}$$

3. *Sine-Log Allpass Transforms (SLAPT):* SLAPT frequency warping functions introduced in [McD00], are capable of approximating any 1-1 arbitrary frequency warping function, and are therefore suitable for multi-class adaptation or the adaptation of individual distributions. The K-parameter SLAPT, denoted SLAPT-K, is given by:

$$\theta_p(\lambda) \ = \ \lambda + \sum_{k=1}^{K} p_k \sin(\pi k \lambda) \tag{2.28}$$

2.3 Adaptation with the LT and Estimation of the FW function

2.3.1 Transformation of Features and HMM means

The final feature vector $\mathbf{x}$ consists of the MFCCs and their first and second time derivatives as discussed in Section 1.4. The transform on the time derivatives of the cepstral features will also be linear [CDB98, MB99, CA06]:

$$\widehat{\Delta \mathbf{c}} \ = \ T_p \ \Delta \mathbf{c} \tag{2.29}$$

$$\widehat{\Delta^2 \mathbf{c}} \ = \ T_p \ \Delta^2 \mathbf{c} \tag{2.30}$$

Therefore, the feature vector $\mathbf{x} = \begin{bmatrix} \mathbf{c} \\ \Delta\mathbf{c} \\ \Delta^2\mathbf{c} \end{bmatrix}$ may be transformed as:

$$\mathbf{x}^p = A_p\ \mathbf{x}, \quad \text{where} \quad A_p = \begin{bmatrix} T_p & 0 & 0 \\ 0 & T_p & 0 \\ 0 & 0 & T_p \end{bmatrix} \tag{2.31}$$

where the transformed feature vector $\mathbf{x}^p$ is now a function of the FW parameters, p. Taking the expectation, the mean μ of a given HMM distribution may be transformed as [CDB98, MB99, CA06]:

$$\hat{\mu} = A_p\ \mu \tag{2.32}$$

2.3.2 Combination with MLLR Bias and Variance Adaptation

After estimating the LT (see Section 2.3 below), a bias vector b and an *unconstrained* variance transform matrix H may be estimated according to Maximum Likelihood Linear Regression (MLLR, see Section 1.5) [LW95, Gal96]. The adapted mean and covariance matrix $\{\hat{\mu}, \hat{\Sigma}\}$ of a Gaussian distribution $\{\mu, \Sigma\}$ are given by:

$$\hat{\mu} = A_p\ \mu + b \tag{2.33}$$

$$\hat{\Sigma} = B^T H B \tag{2.34}$$

where $\Sigma = CC^T$ and $B = C^{-1}$. This form of covariance transformation is equivalent to the one presented in Section 1.5.

The MLLR formulae for estimating the bias and variance transforms are [Gal96, McD00, CA06]:

$$b = \left(\sum_g \sum_u \sum_t \gamma_{gut} \Sigma_g^{-1} \right)^{-1} \left(\sum_g \sum_u \sum_t \gamma_{gut} \Sigma_g^{-1} (\mathbf{x}_{ut} - A_p \mu_g) \right) \tag{2.35}$$

$$H = \frac{\sum_g C_g^T \left[\sum_u \sum_t \gamma_{gut}(\mathbf{x}_{ut} - \mu_g)(\mathbf{x}_{ut} - \mu_g)^T\right] C_g}{\sum_g \sum_u \sum_t \gamma_{gut}} \tag{2.36}$$

In the above equations, g is summed over the Gaussian distributions that are being transformed together, u is summed over the set of adaptation utterances and t is the time index over a given adaptation utterance u. γ_{gut} is the posterior probability that a speech frame $\mathbf{x}_{ut}$ was produced by Gaussian g, for the given transcription of the adaptation data. In the case of diagonal covariance matrices, the off-diagonal elements of H from Equation 2.36 above are simply ignored and zeroed out.

2.3.3 MLS Objective Criterion

For a feature space transform, the maximum likelihood score (MLS, see Equation 1.25 of Section 1.6) criterion to estimate the optimal FW parameters $\hat{p}$ is [LR98, PMS01]:

$$\hat{p} = \arg\max_p \left[\log P(\mathbf{X}^p, \Theta^p | W, \Lambda) + T \log |A_p|\right] \tag{2.37}$$

where p is(are) the FW parameter(s), $\mathbf{x}^p = A_p\mathbf{x}$ is a normalized feature vector, $|A_p|$ is the determinant of A_p, $\mathbf{X}^p = \{\mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_T^p\}$ is the normalized adaptation data, W is the word (or other unit) transcription, Λ are the corresponding HMMs, and Θ^p is the ML HMM state sequence with which $\mathbf{X}^p$ are aligned to Λ^p by the Viterbi algorithm during ASR decoding.

The determinant term in Equation 2.37 is required to properly normalize the likelihood when the feature space is transformed. For regular VTLN by Mel bin center frequency warping [LR98], the objective function only includes the first term in Equation 2.37 since the second term is not defined. In our experiments with the Linear Transformation, the determinant term was found to be important during training with Speaker Adaptive Modeling (SAM, see Section 3.3), but was

not used in testing, since slightly better results were obtained without it.

The simplified MLS criterion (see Equation 1.26) becomes:

$$\mathcal{F}(p) = \sum_{t=1}^{T} \log \left(\sum_{r=1}^{R} c_{tr} \mathcal{N}(\mathbf{x}_t^p; \mu_{tr}, \Sigma_{tr}) \right) + T \log |A_p| \tag{2.38}$$

where $\sum_{r=1}^{R} c_{tr} = 1$ for the mixture Gaussian state output distribution at time t.

The MLS criterion can also to be used to estimate LT FW to transform the means of the HMMs in the back end as in Equation 2.32:

$$\hat{p} = \arg\max_{p} \left[\log P(\mathbf{X}, \Theta^p | W, \Lambda^p) \right] \tag{2.39}$$

where the variables are as explained above for Equation 2.37 except that here it is not the adaptation data but the HMMs Λ that are modified to Λ^p for FW parameters p.

2.3.4 The EM Auxiliary Function

The FW parameters can also be estimated by maximizing the EM auxiliary function over the adaptation data [McD00, LNU06]. This objective function is identical to the one used for MLLR and CMLLR (constrained MLLR, [Gal98]), except the linear transformation to be estimated is constrained by the FW parametrization. Speaker Adaptive Training (SAT) also uses iterative maximization of the EM auxiliary function to alternately estimate FW parameters and HMM parameters [AMS96].

Here we consider only estimation of a feature transform, which we denote CLTFW similar to CMLLR. The basic auxiliary function to be *minimized* may be expressed as:

$$\mathcal{F}(p) = \frac{1}{2} \sum_{g} \sum_{t} \gamma_g(t) \left[(A_p \mathbf{x}_t - \mu_g)^T \Sigma_g^{-1} (A_p \mathbf{x}_t - \mu_g) - \log(|A_p|^2) \right] \tag{2.40}$$

where g varies over the set of Gaussian distributions for which the transform is to be estimated, t is time or frame index of the adaptation data, and $\gamma_g(t)$ is the posterior probability that feature frame $\mathbf{x}_t$ was generated by Gaussian g for the given transcription of the adaptation utterances.

For diagonal covariance models, this can be simplified to:

$$\mathcal{F}(p) = \frac{1}{2}\sum_{i=1}^{d}\left[a_i G^{(i)} a_i^T - 2a_i k^{(i)T}\right] - \beta\log(|A_p|) \tag{2.41}$$

where d is the feature vector size, a_i is the ith row of A_p, and

$$G^{(i)} = \sum_g \frac{1}{\sigma_{gi}^2}\sum_t \gamma_g(t)\mathbf{x}_t\mathbf{x}_t^T \tag{2.42}$$

$$k^{(i)} = \sum_g \frac{\mu_{gi}^2}{\sigma_{gi}^2}\sum_t \gamma_g(t)\mathbf{x}_t^T \tag{2.43}$$

$$\beta = \sum_g\sum_t \gamma_g(t) \tag{2.44}$$

The computations involved in this approach are mostly during the accumulation of the statistics (i.e. computing $G^{(j)}$ and $k^{(j)}$). Once the statistics have been accumulated, the computational cost of optimizing the objective function is significantly smaller since it is twice differentiable and typically convex, and a few iterations of Newton's method are found to be sufficient to optimize it for a reasonably small number of FW parameters (10 or so). Different CLTFW transforms can also be estimated for different classes of distributions similar to CMLLR, without much increase in computations, since it is seen from Equation 2.41 that the accumulator values for a set of Gaussians is the sum over the individual Gaussians. The accumulator method of optimizing the EM auxiliary function for CLTFW may be extended in a very natural manner for the estimation of an aditive bias on top of the CLTFW transform.

Loof et al. [LNU06] also discuss briefly how this accumulator based approach may be extended to the case with a global feature space LDA/HLDA transform.

The approach can also be extended to the multi-class semi-tied covariance (STC, [Gal99]) case, as long as all the Gaussians considered for CLTFW estimation share the same STC transformation.

2.3.5 Optimizing the EM auxiliary function

For the estimation of multiple FW parameters like with the SLAPT FW using the EM auxiliary function, it is efficient to use a convex optimization method. Newton's method can be used since the auxiliary function is twice differentiable [McD00]. We consider the diagonal covariance case and derive the formulae for calculating the first derivative of the objective function as follows.

Differentiating $\mathcal{F}(p)$ in Equation 2.41 with respect to p, we have:

$$\frac{\partial \mathcal{F}(p)}{\partial p_k} = \sum_{i,j=1}^{d} \frac{\partial \mathcal{F}(p)}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial p_k} \tag{2.45}$$

If we let

$$\mathcal{F}(p) = \mathcal{F}_1(p) - \beta \log(|A_p|) \tag{2.46}$$

where

$$\mathcal{F}_1(p) = \frac{1}{2} \sum_{i=1}^{d} \left[a_i G^{(i)} a_i^T - 2 a_i k^{(i)T} \right] \tag{2.47}$$

then

$$\frac{\partial \mathcal{F}(p)}{\partial A} = \frac{\partial \mathcal{F}_1(p)}{\partial A} - \beta \frac{\partial \log(|A_p|)}{\partial A} \tag{2.48}$$

where for a function f, $\frac{\partial f}{\partial A}$ denotes the matrix of partial derivatives $\frac{\partial f}{\partial a_{ij}}$. It can be shown (for example, Section 5.1, [McD00]), that

$$\frac{\partial \log(|A|)}{\partial A} = (A^{-1})^T \tag{2.49}$$

We have:

$$\frac{\partial \mathcal{F}_1(p)}{\partial a_i} = a_i G^{(i)} - k^{(i)} \tag{2.50}$$

where $\frac{\partial \mathcal{F}_1(p)}{\partial a_i}$ is the vector of partial derivatives $\frac{\partial \mathcal{F}_1(p)}{\partial a_{ij}}$. Therefore $\frac{\partial \mathcal{F}(p)}{\partial A}$, can be computed from Equations 2.49, 2.48 and 2.50. We also need $\frac{\partial A_p}{\partial p_k}$ to compute $\frac{\partial \mathcal{F}(p)}{\partial p_k}$ from Equation 2.45. We have:

$$\frac{\partial A_p}{\partial p_k} = \begin{bmatrix} \frac{\partial T_p}{\partial p_k} & 0 & 0 \\ 0 & \frac{\partial T_p}{\partial p_k} & 0 \\ 0 & 0 & \frac{\partial T_p}{\partial p_k} \end{bmatrix} \tag{2.51}$$

By Equation 2.18,

$$\frac{\partial T_p}{\partial p_k} = C \cdot \frac{\partial \tilde{C}_p}{\partial p_k} \tag{2.52}$$

To compute $\frac{\partial \tilde{C}_p}{\partial p_k}$, recall Equation 2.25 by which we have:

$$\tilde{C}_p(i,j) = \alpha_j \cdot \cos\left[\pi j \ \theta_p(\lambda_i)\right] \tag{2.53}$$

for $1 \leq i \leq M,\ 0 \leq j \leq N-1$, and where $\lambda_i = \frac{2i-1}{2M}$. Then,

$$\frac{\partial \tilde{C}_p(i,j)}{\partial p_k} = -\alpha_j \cdot \pi \cdot j \cdot \sin\left[\pi \theta_p(\lambda_i) j\right] \cdot \frac{\partial \theta_p(\lambda_i)}{\partial p_k} \tag{2.54}$$

For the frequency warping functions used (Equations 2.26 to 2.28), the derivative with respect to the parameter is easily computed. For example, for the piecewise-linear warping (Equation 2.26), we have:

$$\frac{\partial \theta_p(\lambda)}{\partial p} = \begin{cases} \lambda, & 0 \leq \lambda \leq \lambda_0 \\ \lambda_0 \cdot \frac{1-\lambda}{1-\lambda_0}, & \lambda_0 < \lambda \leq 1 \end{cases} \tag{2.55}$$

The gradient of the objective function in Eq. 2.38 with respect to the FW parameters p, $\nabla_p \mathcal{F}(p)$, can therefore be calculated using Equations 2.45 to 2.55.

Formulae for the Hessian matrix of second derivatives of the objective function with respect to FW parameters were also derived, and used in Newton's method for optimizing $\mathcal{F}(p)$.

2.4 Comparison and relationships with previous transforms

As discussed in Section 1.8, several cepstral linear transforms have earlier been derived in the literature as equivalents of frequency warping for use in speaker normalization and adaptation. Some of them were derived for plain or PLP cepstra [MBL98, PMS01] and extended to non-standard MFCC features [PN03, UZN05]. Although our LT was derived for standard MFCCs by warping the log filterbank output, motivated by the work of [CA06], it is closely related to the earlier transforms for cepstral features.

In fact, we have verified that for the SLAPT-1 warping function, the different cepstral LTs (McDonough's [MBL98], Umesh et al.'s [UZN05] and ours) are numerically identical except in the first row, up to numerical accuracy in Matlab. Since this is not readily apparent from their mathematical formulations, we now wish to clarify the relationships between these different cepstral linear transforms for frequency warping. We first briefly describe the assumptions and formulae involved in the calculation of the LTs of McDonough, Pitz et al. and Umesh et. al., and then compare them with our LT.

2.4.1 McDonough's LT

McDonough derived his LT using the strict definition of cepstra as Laurent series coefficients of the log spectrum (see, for example, [MBL98]). With this definition, the LT can be computed for analytic transformations that preserve the unit circle in the complex plane, such as the rational and sine-log all-pass transforms (RAPT and SLAPT). If $Q(z)$ is the warping transformation, then the transformation

matrix is given by:

$$a_{nm} = \begin{cases} 1 & \text{for } n = 0, m = 0 \\ 2q^{(m)}[0], & \text{for } n = 0, m > 0 \\ 0, & \text{for } n > 0, m = 0 \\ q^{(m)}[n] + q^{(m)}[-n], & \text{for } n > 0, m > 0 \end{cases} \tag{2.56}$$

where $q^{(m)}[n]$ are obtained from $q[n]$ using $q^{(m)}[n] = q^{(m-1)}[n] * q[n], m \geq 1$, with $q^{(0)}[n] = \delta[n]$, the unit sample sequence. This matrix differs in the first row from the one given in [MBL98], since that was for the causal minimum-phase cepstra ($x[n]$ in McDonough et al., 1998), while this is for the plain real cepstra ($c[n]$ in McDonough et al., 1998).

Since we will later compare the computations involved in our LT with that of McDonough's, we now briefly list the steps involved in calculating McDonough's LT. For the K-parameter SLAPT FW,

$$Q(z) = zG(z) = z \exp F(z) \tag{2.57}$$

where

$$F(z) = \left(\frac{\pi}{2}\right) \sum_{k=1}^{K} \alpha_k (z^k - z^{-k}) \tag{2.58}$$

If $f^{(m)}[n]$ are defined using $f[n]$, similar to $q^{(m)}[n]$ using $q[n]$ above, then

$$g[n] = \sum_{m=0}^{\infty} \frac{1}{m!} f^{(m)}[n] \tag{2.59}$$

and

$$q[n] = g[n-1], n = 0, \pm 1, \pm 2, \ldots \tag{2.60}$$

The transformation matrix can then be calculated as shown in Equation 2.56. The matrix is, in theory, doubly-infinite-dimensional.

2.4.2 Pitz et al.'s LT

Pitz et al. [PMS01] used the definition of cepstra as inverse discrete-time Fourier transform (IDTFT) coefficients of the log power spectrum to derive their cepstral LT. The transformation matrix was shown to be:

$$a_{nm} = \frac{2}{\pi} \int_0^{\pi} \cos(\omega n) \cos(\phi(\omega) m) d\omega \tag{2.61}$$

where $\phi(\omega)$ is a warping function on ω.

By comparing their derivation with that of McDonough's, it becomes clear that the derivations are equivalent except that in [PMS01], all the complex integrals have been performed on the unit circle, and the assumption is made that the original unwarped cepstra are quefrency limited. For APT FW functions, Pitz et al.'s LT would therefore be identical to McDonough's LT. Note that this is theoretically true even though it may not be possible to evaluate the above integral anlytically for the APT FW function. It has been numerically verified as discussed below. Interestingly, this has not been noted in the literature.

With Pitz et al.'s treatment of cepstra as the IDTFT of the log spectrum, non-analytic FW functions like the popular piecewise-linear (PL) FW can also be used, while such functions cannot be used with McDonough's LT since they would not result in valid cepstra according to his stricter definition of cepstra as Laurent series coefficients of a function analytic in an annular region that includes the unit circle.

2.4.3 Umesh et al.'s LT

The integral involved in the computation of Pitz et al.'s LT (Equation 2.61) can be analytically evaluated only for some simple cases such as the linear and PL FWs. [UZN05] showed that a discrete approximation of the integral would

become exact under the assumption of quefrency limitedness of cepstra. In this case, we can show that the LT matrix is given by

$$A = C_1 \tilde{C}_{1p} \tag{2.62}$$

where C_1 is a type-I DCT matrix, $\tilde{C}_{1p}$ is a type-I warped IDCT matrix, and p are FW parameters. Note that this specific expression was formulated by us and is equivalent to the one given in [UZN05] where IDFT and warped DFT matrices have been used. From our formulation it is seen more clearly by comparing Equations 2.61 and 2.62 that Umesh et al.'s matrix is a discrete version of Pitz et al.'s.

Umesh et al.'s approach is still only an approximation since it involves the assumptions of quefrency limitedness of both the unwarped and warped cepstra. This assumption cannot be valid since it can be seen from McDonough's and Pitz et al.'s derivation, that even if the original cepstra were quefrency limited, the transformed cepstra would not necessarily be. However, it is a very good approximation, and we have verified that for the SLAPT-1 FW function, Umesh et al's matrix (Equation 2.62) is numerically identical to that of McDonough's (Equation 2.56) up to numerical accuracy in Matlab. This has also not been noted earlier in the literature.

Umesh et al. (2005) applied their LT derived for FW with plain cepstra, to a non-standard MFCC feature extraction scheme with a modified filterbank whose filters were uniformly spaced in the linear frequency domain, but of uniform bandwidth in the Mel domain. Their formulae for computing Mel and VTLN warped cepstral coefficients were complicated by the use of two different DCT matrices C_1 and C_2. We can show that their warping transformation matrix for MFCCs is:

$$T = C_2 C_1 \tilde{C}_{1p} C_2^{-1} \tag{2.63}$$

where C_2 is a type-II DCT matrix.

2.4.4 Our LT

We have expressed the equation for our LT in Equation 2.18. To be clearer, we may write it as:

$$T = C_2 \tilde{C}_{2p} \tag{2.64}$$

where C_2 is a type-II DCT, $\tilde{C}_{2p}$ is a type-II warped IDCT matrix, and p are FW parameters. We have given compact formulae for calculating C_2 and $\tilde{C}_{2p}$ in Equations 2.24 and 2.25.

We now see that there is a close relationship between our LT and McDonough-Umesh's LT for plain cepstra. In fact, though different types of DCT matrices have been used in our LT and Umesh's LT, because of the combination of DCT and warped IDCT matrices in both, the final transform matrices are identical in all rows except the first. This, however is only numerically true for values of M (the number of filters) that are not small. In our experiments, we used a value of $M = 26$ for computing our LT and $M = 256$ for computing Umesh et al.'s LT.

It therefore follows from the previous discussion of Umesh et al.'s transform, that for the SLAPT-1 FW, except for the first row, our LT is also an approximation of McDonough's LT. Note that the version of McDonough's LT for minimum-phase cepstra is different from both Umesh's LT and our LT in the first row.

Our approach has two advantages over McDonough's and Umesh et al.'s:

- Our LT (and Umesh et al.'s LT) can be calculated using compact closed form expressions for any FW function as in Equations 2.18, 2.24 and 2.25, unlike McDonough's original LT which is more complicated to calculate since it requires approximate summation of an infinite series and several

iterations of discrete sequence convolution as in Equations 2.56 to 2.60. If the computation of derivatives during optimization of the objective function is also considered (as in Section 2.3.5), the closed-form formulae would be even more convenient.

- By using a warped type-II IDCT, we have applied our LT directly to standard MFCC features, without modifying the feature extraction like [UZN05] have done. Comparing our linear transform in Equation 2.64 with that of Umesh et al. in Equation 2.63, it is clear that our linear transform matrix for MFCCs is mathematically simpler and easier to calculate.

2.4.5 Claes et al. and Cui and Alwan's LTs for standard MFCCs

Claes et al. [CDB98] and Cui and Alwan [CA06] derived transforms for standard MFCCs which were discussed in some detail in Section 2.1. As shown there, Cui and Alwan's transform is a special case of Claes et al.'s transform, but is mathematically simpler. It was also found to give better performance for connected digit recognition of children's speech using the TIDIGITS database. In Section 2.1, we motivated our proposal to perform continuous warping of the log filterbank output based on the success of the transform in [CA06] which was basically a discrete mapping on the log filterbank outputs. In [CA06], the FW was estimated in the linear frequency domain by alignment of formant like peaks, hence the name Peak Alignment (PA) for their method.

Estimatation of FW parameters directly using the MLS or other objective criterion would eliminate the need for access to the intermediate linear frequency spectrum during feature extraction, and the estimation can be performed entirely using just the previously extracted unwarped features. This would be an advantage in DSR as mentioned in Section 1.7. In Section 3.2, we show that when

the MLS criterion is used to estimate the FW parameter, our LT gives better performance than the LTs of Claes et al. and Cui and Alwan.

Computationally, FW estimation based on formant-like peak alignment can be more efficient than MLS estimation, depending on how the peaks are estimated. The most expensive part of using the MLS criterion to estimate a speaker specific warp factor, is the Viterbi forced alignment of frames and HMM states for the adaptation data, which may be performed for each warp factor, or once with unwarped features in the simplified criterion. Forced alignment with a known transcription of the adaptation data can be performed much faster than ASR decoding. Since forced alignment is already part of the ASR decoder algorithms, MLS is simpler to implement, which may be useful in some applications. In [CA06], the EM algorithm is used to fit Gaussian mixtures to the linear frequency DFT spectrum and the formant-like peaks are estimated from these Gaussians for each frame of voiced speech. There, it is necessary to detect voicing and to specify the number of peaks used to fit the spectrum, which may depend on the age and gender of the test speaker and also the bandwidth of the speech signal used in the recognizer. With the MLS criterion, these considerations are not necessary and the FW estimation is automatic and robust for any test speaker.

2.5 Summary

In this chapter, we introduced a novel LT for FW with MFCCs. The main idea was to directly warp the continuous log filterbank output obtained by cosine interpolation with the IDCT. This approach can be viewed as using the idea of spectral interpolation of [UZN05], to perform a continuous warping of the log filterbank outputs instead of the discrete mapping in [CA06]. However, a single warped IDCT matrix was used to perform both the interpolation and

warping, thus resulting in a simpler mathematical formula for computing the transform compared to [UZN05]. No modification of the standard MFCC feature extraction scheme is required unlike some previous approaches [PN03, UZN05]. Also, the warping in the IDCT matrix is parametrized and the parameter can be estimated directly by optimizing an objective criterion, without using the intermediate linear frequency spectrum as in the Peak Alignment method of [CA06]. This would be advantageous in distributed speech recognition, where intermediate variables in the feature extraction have to be reconstructed at the server. We also discussed estimation of FW parameters for VTLN and speaker adaptation using the MLS criterion and EM auxiliary function. Formulae were also derived for calculation of derivatives of the EM auxiliary function for the estimation of several FW parameters. We also showed that different LTs earlier proposed for FW with cepstral features are all closely related, and these LTs were found to be identical in all rows except the first, for the all-pass transform warping functions. In fact, the earlier proposed LTs can be more easily computed using the closed form expressions that were given in this chapter.

CHAPTER 3

Experimental Results

In this chapter, we present the results of recognition experiments with our LT for FW with MFCCs developed in Chapter 2. We validate the LT by testing it on a continuous speech recognition task and comparing the performance with that of regular VTLN by warping the filterbank center frequencies (hereafter referred to as Regular VTLN). The main advantages of using the LT over Regular VTLN, as discussed in Section 1.7, are computational savings and flexibility of implementation. The spectral information available during LT parameter estimation consists only of the smoothed log Mel spectrum that can be computed from the truncated unwarped cepstra, and the corresponding HMM means. More spectral information is available to Regular VTLN since it can use the linear frequency spectrum of each speech analysis frame. In the results below, one of our main aims is to show that VTLN and adaptation using the LT, while being computationally superior and working with less available information, can give recognition performance comparable to that of Regular VTLN.

3.1 Continuous Speech Recognition Experiments

We also performed experiments on continuous speech recognition using the 1000 word vocabulary DARPA Naval Resource Management (RM1) database [PFB88].

The speech data was downsampled to 8000 Hz in our experiments and context dependent triphone (a phoneme with specified preceeding and following phonemic contexts) models were trained on speech from 72 adult speakers in the speaker independent training set. All triphone HMMs contained 3 emitting states and 6 Gaussian mixtures per state. The Mel filterbank contained 26 filters, and the feature vectors consisted of the first 13 MFCCs with the corresponding first and second derivatives. Cepstral Mean Subtraction (CMS) was also performed on each utterance.

Recognition experiments were performed on 50 test utterances from each of 10 speakers from the speaker dependent test data in the database. The baseline recognition accuracy was 90.16 %.

VTLN and back-end adaptation were tested with varying amounts of adaptation data to validate the effectiveness of the new linear transform in improving accuracy in continuous speech recognition. Experiments were performed with 1, 5 and 10 adaptation utterances from each test speaker. For adaptation with a single utterance, the 10 utterances marked for rapid adaptation in the RM1 database were used. For more than one adaptation utterance, ten different combinations of utterances were randomly selected for each speaker and results were obtained for each combination of adaptation utterances using each of the adaptation techniques. The results were then averaged over the adaptation combinations and the speakers. The pool of adaptation utterances was separate from the set of test utterances for each speaker.

Table 3.1 shows the results of VTLN experiments comparing LT VTLN with Regular VTLN. A speaker-specific warp factor for the piecewise-linear (PL) FW was estimated from the adaptation data for each test speaker, using a grid search to optimize the MLS criterion of Section 2.3. The warping factor step size in the

Algorithm	No. of adaptation utterances		
	1	5	10
LT VTLN	91.46	91.59	91.54
Regular VTLN	91.42	91.60	91.66

Table 3.1: Recognition Accuracy in VTLN Experiments using the RM1 database. FW parameters were estimated with the MLS criterion for both methods. Baseline Accuracy: 90.16 %

grid was 0.01. It was again observed that slightly better results were obtained without the Jacobian Normalization term in the MLS criterion during the estimation of the parameter for LT VTLN and these are the results shown. With LT VTLN, the PL FW gave slightly better results than the linear and the SLAPT-1 FWs.

The performance of LT VTLN is seen to be comparable to that of Regular VTLN.

In Figure 3.1 sample discrete log filterbank outputs, before and after warping with LT and Regular VTLN are shown. The speech frame is from the triphone 'S-AH+B' in the word 'sub'. The features of the utterance were normalized with the corresponding estimated PL FW parameter for each VTLN method from the particular utterance. The warped log filterbank outputs of the two VTLN methods are seen to be very similar, which explains the very similar performance seen in Table 3.1. This seems to imply that most of the spectral information required for VTLN is already contained in the unwarped truncated cepstra, which is why LT VTLN may be as successful as Regular VTLN.

We then performed VTLN estimation with the simplified MLS objective func-

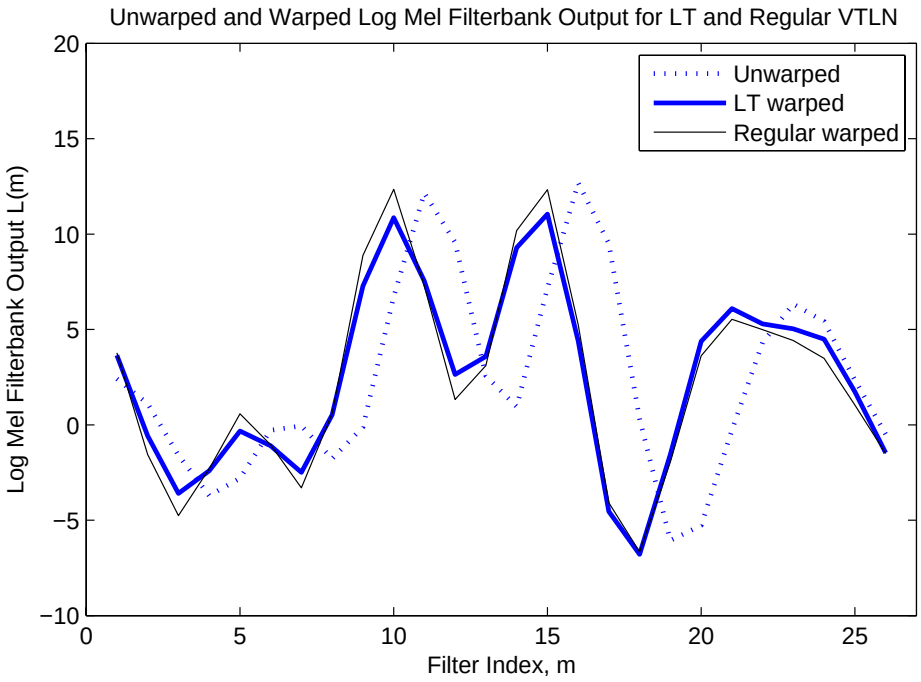

Figure 3.1: Discrete log filterbank outputs, unwarped (dotted line) and warped, with LT VTLN (thick solid line) and Regular VTLN (thin solid line). The speech frame is from the triphone 'S-AH+B' in the word 'sub', following phoneme transcriptions in the CMU Pronouncing Dictionary

tion as in Equation 2.38 of Section 2.3.3, with fixed frame-state alignment obtained with unwarped features. Again, the PL FW with a grid step size of 0.01 was used. The results are shown in Table 3.2. As can be seen, both Regular and LT VTLN have comparable results, with the results for both being slightly worse with the simplified objective function, as compared to the results in Table 3.1.

Table 3.3 shows the results of global speaker adaptation experiments on the RM1 database. The LT was used to adapt HMM Means as in Equation 2.32, and is combined with MLLR mean bias and unconstrained variance transforms as described in Section 2.3.2. The results of standard MLLR with a 3-block

Algorithm	No. of adaptation utterances		
	1	5	10
LT VTLN	91.33	91.33	91.33
Regular VTLN	91.29	91.28	91.34

Table 3.2: Recognition Accuracy in VTLN Experiments with Fixed Frame-State Alignment, using the RM1 database. Baseline Accuracy: 90.16 %

Algorithm	No. of adaptation utterances		
	1	5	10
Back End LT FW + MLLR bias & var	91.58	91.74	91.76
MLLR	84.89	92.38	92.43

Table 3.3: Recognition Accuracy in Global Speaker Adaptation Experiments with limited data on the RM1 database: LT Applied in the back-end and 3-block MLLR. Baseline Accuracy: 90.16 %

mean transformation matrix and unconstrained variance transformation are also shown for comparison. Comparing Tables 3.3 and 3.1 we see that back end HMM mean adaptation with the LT combined with unconstrained MLLR bias and variance adaptation, gives results comparable to VTLN in the front end. The results confirm earlier observed trends [CA06, McD00] that FW based methods are definitely superior to MLLR for very limited adaptation data (1 utterance), where MLLR actually gives worse performance than the baseline. With increased adaptation data, MLLR gives better performance.

3.2 Comparison with other LT approximations of VTLN for standard MFCCs

As discussed in Section 2.1, Claes et al. (1998) and Cui and Alwan (2005, 2006) have earlier proposed linear transforms for approximating VTLN with standard MFCC features. In Table 3.4 we show results comparing our LT with those of Cui and Alwan's Peak Alignment (PA) LT, and Claes et al's LT. The recognition results shown are on the RM database with VTLN estimated on 1 utterance, since it is desirable in practice to estimate the VTLN parameter with limited data. The MLS criterion was used to estimate the PL FW parameter for all methods. The results of Regular VTLN are also shown.

Algorithm	**Recognition Accuracy**, %
Baseline	90.16
Regular VTLN	91.42
Our LT VTLN	91.46
PA LT	90.82
Claes et al.'s LT	90.79

Table 3.4: Comparison of different LT approximations for VTLN with MFCC features, on the RM1 database. FW parameters were estimated on 1 utterance with the MLS criterion for all methods.

It is seen that our LT performs as well as Regular VTLN, while the PA LT and Claes et al.'s LT do not perform as well, when the FW parameter is estimated using the MLS criterion with 1 utterance. The statistical significance levels of our LT compared to PA LT and Claes et al.'s LT, computed using the matched-pairs

test [GC89], were 0.023 and 0.26 respectively. This shows that the improvements obtained with our LT compared to PA LT were statistically significant while the improvements compared to Claes et al.'s LT were not statistically very significant. The latter could be due to the amount of data used for recognition and needs to be investigated using a larger number of test utterances. By the comparison with PA LT, we may conclude that the parametrization of the transform is important since it determines the behavior of the objective function and performance of the VTLN parameter estimated using the criterion.

As we have discussed in Section 2.4, our LT is numerically almost identical to McDonough's and Umesh et al.'s LTs, except in the first row. Therefore, the performance of these LTs was very similar to that of our LT.

3.3 Speaker Adaptive Modeling Experiments

It is well known that the effectiveness of VTLN is greatly improved when it is performed also during training [McD00, WNK02]. In this way, the trained models capture more of the phonetic variability and less of the inter-speaker variability in the training data. Speaker Adaptive Modeling (abbreviated as SAM here, [WNK02]) and Speaker-Adapted Training (SAT, [AMS96, McD00]) are two techniques for incorporating VTLN during the training process.

We first performed VTLN during training along the SAM framework. The main feature of this technique is that the optimal warping factor for each training speaker is selected iteratively using single Gaussian mixture HMMs and the MLS criterion. Initial models are trained without any warping, and then at each iteration the optimal warping factor for each speaker in the training set is obtained by MLS over the training data from that speaker, and models are retrained with

the new warping factors. The use of single Gaussians mixtures during the iterative warp factor estimation is important because that gives the best results. After a certain number of iterations or when the warping factors converge, the final models are trained with the best warping factor for each speaker, and with the desired number of Gaussians per mixture.

Ten iterations were performed during SAM VTLN parameter estimation with the PL FW for both Regular and LT VTLN. One important observation was that when the Jacobian Normalization (JN, see Section 2.3.3) term was not included in the MLS objective function, the performance of the LT was very poor, even worse than without any SAM. This was investigated and it was found that the warping factor did not converge during the iterations, and the mean warping factor (which should presumably be close to 1, the initial value corresponding to no warping) continuously decreased to around 0.93 in ten iterations without the JN term. After including the JN term in the warping parameter estimation, the training speakers' warping factors were observed to converge, and the mean value at the end of ten iterations was around 0.99. However, during testing, it was again observed that slightly better results were obtained without the JN term in the MLS estimation and these are the results that are shown.

The histograms of estimated warping factors of the 72 training speakers for both Regular VTLN and LT VTLN with the PL FW are shown in Figure 3.2. For each VTLN method, ten bins over the corresponding ranges of warping factor were used for calculating the histogram, but both histograms are plotted over the same range of warping factors, from 0.85 to 1.25, for comparison. It is observed that the range of the warping factors for LT VTLN is significantly smaller than that of Regular VTLN, probably due to the fact that warping in LT VTLN is being performed on an already Mel warped log spectrum.

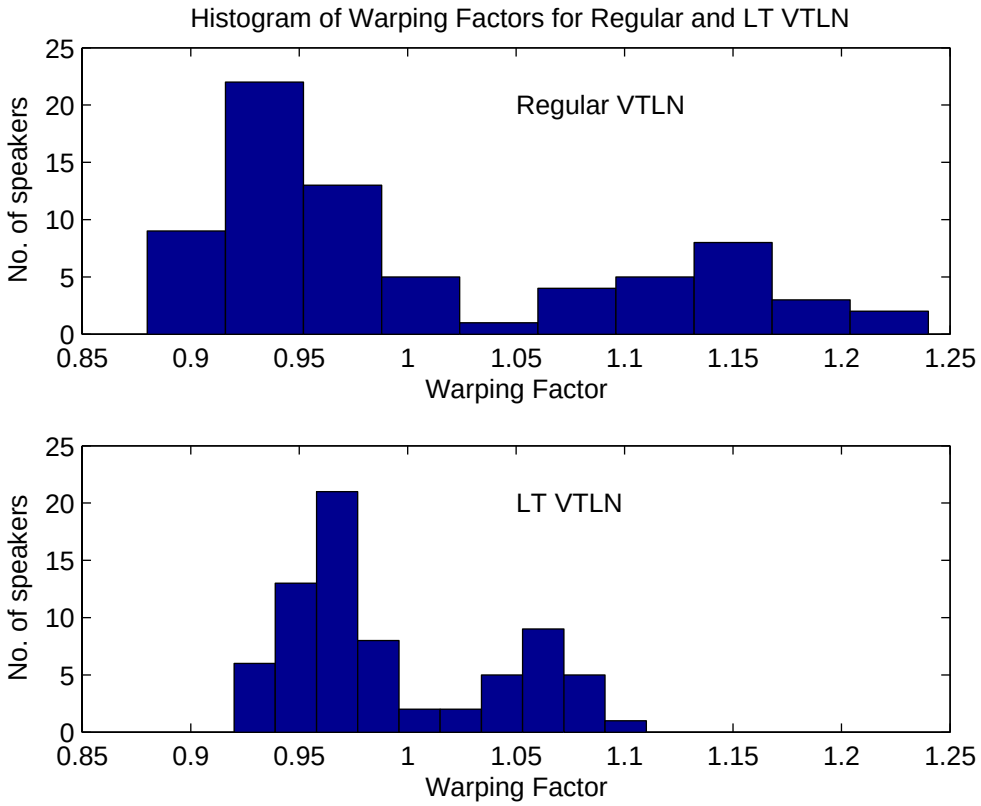

Figure 3.2: Histograms of warping factors in Speaker Adaptive Modeling, with Regular and LT VTLN, for 72 adult speakers from the speaker independent training data in the RM1 database

The results of SAM VTLN experiments are shown in Table 3.5.

We first notice that when SAM is performed, the baseline accuracy is lower than without SAM, but once the test speaker is normalized, the accuracy is significantly better than without SAM.

The performances of the two VTLN methods are comparable when normalization is performed also during training. The important results here are those for adaptation with 1 utterance, since MLLR would be preferred when more utterances of adaptation data are available. Here, the difference in accuracies is small, around 0.17% absolute. However, better results were obtained with back end

Algorithm	No. of adaptation utterances		
	0	1	5
Regular VTLN	86.82	92.81	93.07
LT VTLN	86.82	92.64	92.79
Back End LT FW + MLLR Bias & Var.	86.82	92.87	93.31

Table 3.5: Recognition Accuracy in SAM VTLN Experiments using the RM1 database. 10 iterations of warping factor estimation were performed for each VTLN method for the training speakers and testing was performed with the corresponding method. The baseline with SAM models was the same (86.82 %) for both Regular and LT VTLN.

LT FW combined with MLLR bias and variance adaptation, tested on models trained with LT VTLN, which are also shown in Table 3.5.

Therefore, in all cases, results comparable to Regular VTLN can be obtained with the LT, by applying it in the back end instead of the front end.

We have also verified that with a global Semi-Tied Covariance (STC) matrix included, the performance of LT VTLN SAM models tested with LT VTLN is still comparable to that of Regular VTLN SAM models tested with Regular VTLN.

3.4 Speaker Adaptive Training Experiments

We also implemented SAT with feature space LT which we denoted CLTFW similar to CMLLR (constrained MLLR which is equivalent to feature space MLLR) and tested it on the RM1 database. CLTFW parameters were estimated by

Train/Test Conditions	No. of adaptation utterances		
	1	5	10
G-CLTFW PL SAT / G-CLTFW PL	92.82	92.91	92.94
G-CLTFW PL SAT / RT CLTFW (SLAPT-5)	92.82*	93.03	93.31
G-CLTFW PL SAT / RT CLTFW (SLAPT-5) + Bias	92.82*	93.30	94.07

Table 3.6: Recognition Accuracy in Global (G-) CLTFW SAT Experiments with the PL FW using the RM1 database. 10 iterations of SAT warping factor estimation were performed for the training speakers. RT denotes the use of a regression tree to estimate transforms. * indicates insufficient data to estimate further transforms.

optimizing the EM auxiliary function as discussed in Sections 2.3.4 and 2.3.5. SAT uses the iterative maximization of the EM auxiliary function to jointly estimate speaker transforms and HMM parameters. Ten iterations of SAT were performed with global LT and the PL FW on single mixture HMMs and the final single-mixture SAT speaker transforms were used to retrain 6-mixture HMMs using the baseline models and single-pass retraining. Multiple iterations of model re-estimation were then performed keeping the transforms fixed.

We tested the CLTFW SAT models with CLTFW adaptation with 1, 5 and 10 utterances, and the recognition results are shown in Table 3.6.

It is seen that when the Global (G-) CLTFW SAT models were tested with G-CLTFW, the performance was comparable to that obtained with VTLN SAM

(refer Table 3.5), and the performance saturates for larger number of utterances. However, improved results for more adaptation data were obtained when multiple parameter SLAPT-5 CLTFW was estimated for multiple classes using a regression tree (RT). A frame count threshold of 400 for estimating a transform at a regression node was found to be effective. During estimation, 5 iterations of CLTFW parameter estimation were performed on a single utterance to first estimate a global PL CLTFW transform (similar to VTLN estimation), and this global transform was used to obtain alignments for two iterations of multi-class RT SLAPT-5 CLTFW estimation. It is seen that the performance of RT CLTFW improves with more data. An additive bias was included in the transform, and the performance improved significantly. The statistical significance levels of RT SLAPT-5 CLTFW with additive bias compared to global PL CLTFW, computed using the matched-pairs test [GC89], were 0.035 and 0.005 with 5 and 10 adaptation utterances, respectively. This shows that the improvements obtained with RT SLAPT-5 CLTFW with bias compared to global PL CLTFW are statistically significant.

Therefore, multiple parameter SLAPT-5 CLTFW-Bias transforms estimated using the EM auxiliary function and a regression tree, can give significantly better performance than global VTLN, and improving performance with increasing data.

Since Regular VTLN is not a non-invertible operation on standard MFCCs, the Jacobian determinant term required in the EM auxiliary function for SAT cannot be computed (McDonough, 2000; Sankar and Lee, 1996). Also, even if the Jacobian determinant term were neglected, the accumulator based approach (Gales, 1998) for efficient optimization of the EM auxiliary function with CLTFW cannot be used with Regular VTLN. For multiple class adaptation to be performed with Regular VTLN, features would have to be recomputed with

different warping factors for different distributions. As we have shown, recomputation of features is expensive and this is not practical.

Experiments with multi-class CLTFW SAT and comparisons and combination with HMM mean adaptation (MLLR for example) and LDA/STC would be the topic of future work.

3.5 Unsupervised Adaptation

We have so far given the results of supervised adaptation experiments, where the transcription of the adaptation data is known. Frequency warping methods are known to be effective in adaptation in an unsupervised mode as well [McD00, CA06]. This was confirmed for VTLN and back end model adaptation using our LT, for the case of the speaker adaptive models trained as discussed in the previous section. The results are shown in Table 3.7. In these experiments, an initial recognition pass was first performed over the adaptation data, and the resulting transcriptions were then used to estimate the FW parameter using the MLS criterion and the MLLR mean bias and variance transforms.

Comparing Tables 3.5 and 3.7, it is seen that the results of unsupervised LT VTLN are not much different from those of supervised LT VTLN. In fact, the warping factors estimated with supervised and unsupervised adaptation were only slightly different. This is probably because of our already high baseline recognition accuracy where the transcription produced by the initial recognition pass is close to the actual transcription. With a worse baseline, one may have to use confidence measures calculated from the likelihoods obtained with the initial recognition pass, to select a subset of the adaptation data for warping factor estimation. However, since the VTLN parameter estimated with very little data

Algorithm	No. of adaptation utterances	
	1	5
LT VTLN	92.63	92.86
Back End LT FW + MLLR Bias & Var.	92.75	93.16

Table 3.7: Recognition Accuracy in Unsupervised VTLN and Adaptation Experiments on the RM1 database using models trained with LT Speaker Adaptive Modeling. Baseline Recognition Accuracy is 86.82 %

also performs well the LT would be very effective in unsupervised adaptation.

3.6 Summary

In this chapter, we presented the results of recognition experiments with our LT for FW with MFCCs developed in Chapter 2. We validated the LT with continuous speech recognition experiments using the DARPA Resource Management (RM1) database, and the results are summarized in Table 3.8. These included experiments with front end VTLN and back end adaptation of HMM means, as well as speaker adaptive modeling (SAM) and training (SAT) using the LT [WNK02, AMS96]. We showed that in all cases, LT VTLN can give results comparable to those of Regular VTLN. This shows that the LT, while being only an approximation, and computationally more efficient, does not lead to performance degradation. The results with SAM and SAT using a global transform were comparable. We also showed that results significantly better than with global VTLN can be obtained for increasing amounts of adaptation data by estimating multiple parameter SLAPT-5 FW transforms using a regression tree. Finally, we showed

Algorithm	No. of adaptation utterances		
	0	1	5
LT VTLN	90.16	91.46	91.59
Regular VTLN	90.16	91.42	91.60
Regular VTLN SAM	86.82	92.81	93.07
LT VTLN SAM	86.82	92.64	92.79
LT VTLN SAM/ Back End LT FW + Bias & Var.	86.82	92.87	93.31
G-CLTFW PL SAT / G-CLTFW PL	86.82	92.82	92.91
G-CLTFW PL SAT / RT CLTFW (SLAPT-5)	86.82	92.82	92.91
G-CLTFW PL SAT / RT CLTFW (SLAPT-5) + Bias	86.82	92.82	93.30

Table 3.8: Recognition Accuracy in Experiments using the RM1 database. Summary of results with different FW methods.

that the LT can perform almost just as well when FW parameters for VTLN and adaptation are estimated in an unsupervised mode.

CHAPTER 4

Vocal Tract Inversion by Cepstral Analysis-by-Synthesis using Chain Matrices

4.1 VT Inversion by Analysis-by-Synthesis

We introduced VT inversion using analysis-by-synthesis in Section 1.8 (see Figure 1.3) and described the different challenges and issues involved in achieving inversion. In this chapter, we discuss the details of our inversion method for vowel sounds, specifically the choice of acoustic features, the articulatory-to-acoustic mapping, the cost function to be optimized, construction and search of articulatory codebooks to initialize the optimization, and convex optimization of the cost function using an efficient computation of the derivative of the articulatory-to-acoustic mapping by chain matrices. Finally, we present some results of inversion of diphthong vowels from the University of Wisconsin X-ray Microbeam database.

4.2 Choice of Acoustic Features

As discussed in the introduction to this dissertation, Section 1.1, the vocal tract resonances (VTRs) or formants have a close relationship with the vocal tract shape, and are important for the perception of vowel quality. The first three formants are therefore often used as acoustic features for inversion of vowels [Sor92, OL05]. However, VTR estimation can be difficult for high-pitched talkers, consonants, and semi-vowels.

As described in Section 1.10, during articulatory synthesis, the acoustic quantity that is calculated first is the VT transfer function. The calculation of formants from the VT transfer function would involve either locating maxima of the transfer function using an optimization method (such as Newton's method), or by finding the roots of an all-pole model fitted to samples of the transfer function at a set of uniformly spaced frequencies. It would therefore be computationally simpler to match the computed VT transfer function with natural speech signal spectra, than matching computed and natural formants. Matching spectra would also effectively result in matching the formant spectral peaks, and explicit formant estimation is not necessary.

However, it is difficult to directly compare computed spectral magnitude values with estimated natural values. The natural spectrum first needs to be smoothed, the voice source spectral tilt needs to be removed, and sensitivity to formant bandwidths needs to be decreased due to inaccuracies in the speech production model. Mel frequency warping is also used to account for the fact that perturbations of the logarithm of the area function more linearly affect the logarithms of the formant frequencies (as a first order approximation) [Sch67]. These operations are all performed more conveniently in the cepstral domain [SMP90, SK86].

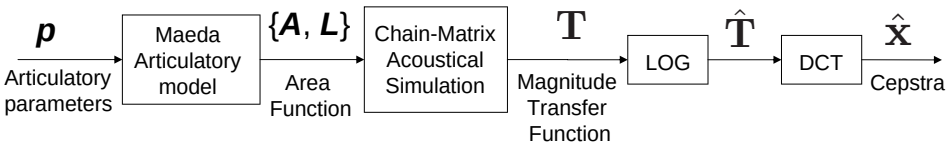

Figure 4.1: Articulatory-to-acoustic mapping

We performed VT inversion by matching the truncated cepstrum, which is equivalent to matching the smoothed log spectral envelope. The first 20 cepstral coefficients were used, excluding the zeroth cepstral coefficient as it is only related to signal energy. De-emphasis of spectral tilt and formant bandwidths, log spectral weighting, and Mel frequency warping can all be captured in a linear weighting matrix on cepstra, as will be discussed below in Section 4.3 [SMP90, SK86].

We used matching of natural LPC cepstra and synthesized DFT cepstra to perform VT inversion for vowels. Since the shape of the computed transfer function for vowels is very well fitted by an LP all-pole model, the synthesized LPC cepstrum is very well approximated by the DFT cepstrum. The difference between log spectra described by liftered LPC and DFT cepstra was verified to be negligible for $f_s = 8000Hz$ and a DFT size as low as 64.

We next discuss the articulatory-to-acoustic mapping, and the acoustic distance measure between natural and computed cepstra.

4.3 The Articulatory-to-Acoustic Mapping

4.3.1 Computation of cepstra

Figure 4.1 shows the block diagram of the articulatory-to-acoustic mapping used in our work. The seven-parameter Maeda articulatory model and the chain matrix method for computing VT acoustics using the Sondhi VT model were discussed

in Sections 1.9 and 1.10 respectively. Recall that the Maeda model computes the VT area function for a given configuration of articulatory parameters as a sequence of uniform tubes of varying areas and lengths:

$$\mathbf{A} = [A_1\ A_2\ \ldots\ A_N] \tag{4.1}$$

$$\mathbf{L} = [L_1\ L_2\ \ldots\ L_N] \tag{4.2}$$

The chain matrix method is used to compute the VT transfer function ($H(s)$, Equation 1.30) from the area function $\{\mathbf{A}, \mathbf{L}\}$. Considering only the imaginary axis, $s = j(2\pi f)$, the magnitude of the VT transfer function is:

$$T(f) = |H(f)| = \frac{1}{|\mathcal{A} - \mathcal{C}Z_L|} \tag{4.3}$$

where $\mathcal{A}$ and $\mathcal{C}$ are the elements of the overall chain-matrix of the vocal tract, and Z_L is the radiation impedance at the lips.

First, $T(f)$ is computed at frequencies:

$$f_i = i \cdot \frac{F_{max}}{N_f},\ 0 \leq i \leq N_f \tag{4.4}$$

where F_{max} is a maximum frequency and (N_f+1) is the number of frequency samples. For comparison with natural acoustic features, $F_{max} = f_s/2$, where f_s is the sampling frequency of the speech signal.

Let

$$\mathbf{T} = [T(f_0)\ T(f_1)\ \ldots\ T(f_{N_f})]^T \tag{4.5}$$

and

$$\hat{\mathbf{T}} = [\hat{T}(f_0)\ \hat{T}(f_1)\ \ldots\ \hat{T}(f_{N_f})]^T \tag{4.6}$$

where

$$\hat{T}(f) = \log(T(f)) \tag{4.7}$$

i.e.,

$$\hat{\mathbf{T}} = \log(\mathbf{T}) \tag{4.8}$$

with elementwise logarithm of the vector $\mathbf{T}$.

The DFT cepstrum $\hat{\mathbf{x}}$ of the computed VT magnitude transfer function is obtained as the truncated IDFT of the vector:

$$[\hat{T}(f_0)\ \hat{T}(f_1)\ \ldots\ \hat{T}(f_{N_f-1})\ \hat{T}(f_{N_f})\ \hat{T}(f_{N_f-1})\ \ldots\ \hat{T}(f_1)]^T$$

Note that it has even symmetry, and the first $(N_f + 1)$ elements comprise $\hat{\mathbf{T}}$. Therefore the IDFT can be expressed as a DCT of $\hat{\mathbf{T}}$. The DFT cepstrum $\hat{\mathbf{x}}$ is therefore given by:

$$\hat{\mathbf{x}} = C \cdot \hat{\mathbf{T}} \tag{4.9}$$

where C (not to be confused with $\mathcal{C}$, the chain matrix parameter) is a DCT matrix that may be easily be shown to be:

$$C(k,n) = \begin{cases} \frac{1}{2N_f}, & 1 \le k \le M, n = 0 \\ \frac{1}{N_f}\cos\left(\frac{\pi n k}{N_f}\right), & 1 \le k \le M, 1 \le n \le N_f - 1 \\ \frac{(-1)^k}{2N_f}, & 1 \le k \le M, n = N_f \end{cases} \tag{4.10}$$

and $M = 20$ is the number of cepstral coefficients used.

Calculating the cepstrum of the computed vocal tract transfer function in this way allows us to analytically compute its derivative with respect to the area function and the articulatory parameters, as will be discussed in Section 4.6.

The formants of the VT can be computed from the roots of the LP polynomial fitted to the $T(f_i)$.

The most computationally intensive step in Figure 4.1 is the calculation of the VT chain matrix using Equations 1.29 to 1.35 since there may be up to $N = 30$

sections in the area function, and $T(f)$ may be desired at $N_f = 30$ or more frequency points depending on the sampling rate and frequency resolution.

4.3.2 Liftering

We used the raised sine lifter introduced in [JRW87] to decrease the spectral tilt resulting from the voice source, and to emphasize the formant peaks [SA97]. This lifter has earlier been found to give good performance when used for articulatory codebook search [SMP90]. The coefficients of the lifter are given by:

$$w_k = 1 + 0.5M \sin(k\pi/M) \tag{4.11}$$

Liftering of the cepstra may be represented as multiplication by $W_{lifter} = \text{diag}(\vec{w})$ where $\vec{w}$ is the lifter vector.

4.3.3 Log Spectral Weighting

Log spectral weighting is also performed to de-emphasize spectral values below 150 Hz and above 3500Hz, which are not reliably measured for the sampling rate of 8000Hz used. The weighting as a function of frequency is shown in Figure 4.2.

To apply the log spectral weighting, the approximate inverse of the partial DCT matrix in 4.10, taken to be the corresponding partial IDCT matrix, is used to obtain the log spectrum. The IDCT matrix is easily shown to be:

$$C^{-1}(n,k) = \begin{cases} 1, & n = 0, 1 \le k \le M \\ 2\cos\left(\frac{\pi nk}{N_f}\right), & 1 \le n \le N_f - 1, 1 \le k \le M \end{cases} \tag{4.12}$$

If $r(f)$ is the log spectral weighting function, $r_n = r(f_n)$, and $R = \text{diag}(\vec{r})$ is the diagonal matrix of weights, then the weighted log spectrum is obtained from the liftered cepstral vector using the matrix $R \cdot C^{-1}$.

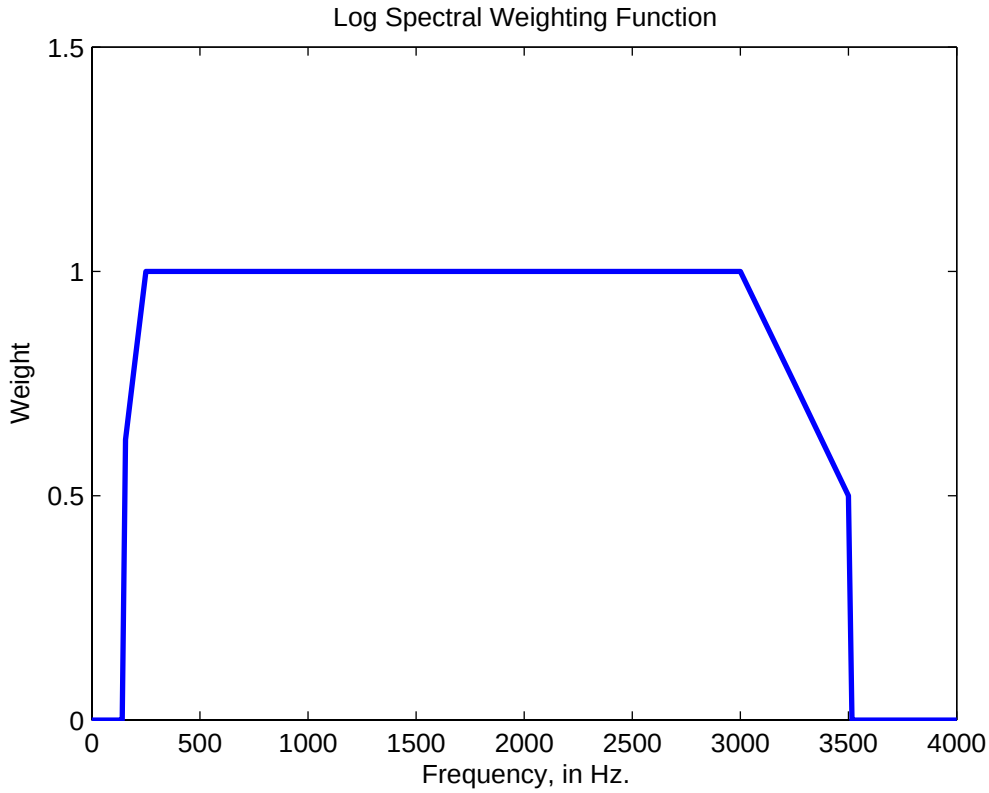

Figure 4.2: Weighting function on log spectrum used in the cepstral distance measure

4.3.4 Mel Warping

We also perform Mel frequency warping of the log spectrum to account for the fact that log area function perturbations linearly affect log formant frequencies [Sch67]. As in our LT developed in Chapter 2, the Mel warping is incorporated into the IDCT matrix.

$$C_{mel}^{-1}(n,k) = \begin{cases} 1, & n = 0, 1 \leq k \leq M \\ 2\cos\left(k\pi\phi_{imel}\left(\frac{n}{N_f}\right)\right), & 1 \leq n \leq N_f - 1, 1 \leq k \leq M \end{cases} \tag{4.13}$$

where $\phi_{imel}(\cdot)$ is a normalized inverse Mel warping function.

4.3.5 The Cepstral Distance Measure

The distance measure between input and computed cepstra is calculated as:

$$d(\hat{\mathbf{x}}, \hat{\mathbf{x}}_{in}) = (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{in})^T W_{cep} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{in}) \tag{4.14}$$

where W_{cep} the cepstral weighting matrix, incorporates the operations of liftering, log-spectral weighting and Mel warping in that order, and is obtained as:

$$W_{cep} = D^T \cdot D \tag{4.15}$$

where

$$D = C \cdot C_{mel}^{-1} \cdot C \cdot R \cdot C^{-1} \cdot W_{lifter} \tag{4.16}$$

To preserve as much information as possible, the intermediate operations were performed with full versions of the DCT and IDCT matrices of Equations 4.10, 4.12 and 4.13.

It should be noted that W_{cep}, though complicated, needs to be computed only once.

4.4 The Optimization Cost Function

As discussed in Section 1.8, the objective function to be minimized (E) is the sum of acoustic (E_{acou}), regularization (E_{reg}) and geometric continuity (E_{geo}) terms [Sor92, SS94, OL05]:

$$E = E_{acou} + c_{reg} E_{reg} + c_{geo} E_{geo} \tag{4.17}$$

where c_{reg} and c_{geo} are weights. We use:

$$E_{acou} = \sum_{t=1}^{T} \left([\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{in}(t)]^T W_{cep} [\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{in}(t)] \right)^{\gamma} \tag{4.18}$$

$$E_{reg} = \sum_{t=1}^{T} [\mathbf{p}(t) - \mathbf{p}_0(t)]^T W_{par} [\mathbf{p}(t) - \mathbf{p}_0(t)] \tag{4.19}$$

$$E_{geo} = \sum_{t=1}^{T-1} ||\mathbf{p}(t+1) - \mathbf{p}(t)||^{2\eta} \tag{4.20}$$

where $\{\mathbf{p}(t),\ 1 \leq t \leq T\}$ is the articulatory vector sequence being optimized, $\{\hat{\mathbf{x}}_{in}(t),\ 1 \leq t \leq T\}$ and $\{\hat{\mathbf{x}}(t),\ 1 \leq t \leq T\}$ are the target and synthesized cepstral sequences, W_{cep} is the cepstral weighting matrix, γ and η are exponents, $\{\mathbf{p}_0(t),\ 1 \leq t \leq T\}$ is a "regularizing" sequence, and W_{par} is an articulatory parameter weighting matrix. In the literature, $W_{par} = I$, $\mathbf{p}_0(t) = 0$, and $\gamma = \eta = 1$ [SK86, Sor92, OL05].

The values of c_{reg}, c_{geo}, γ, $\mathbf{p}_0(t)$, W_{par} and η may be chosen to better achieve the competing goals of acoustic match, realistic VT shapes and smooth articulatory trajectories. Increasing values of γ and η result in lower maximum values accross frames, of the acoustic and geometric distances respectively. We used $\gamma = 3$ and $\eta = 2$. E_{reg} helps in eliminating unrealistic VT shapes during the codebook search by discouraging VT configurations farther from the mean position (nominally $\mathbf{p}_0(t) = 0$ for the Maeda model), which are less likely to occur. In the subsequent optimization we use $\mathbf{p}_0(t) = \mathbf{p}_{init}(t)$, the initial sequence obtained from the codebook search, since we are more interested in improving the acoustic match close to the initial sequence. W_{par} and $\mathbf{p}_0(t)$ may be used to place constraints on the articulatory parameters, either based on phonetic considerations or in an *ad hoc* manner to improve results for a specific speaker. The choices for the different parameters are discussed in Section 4.8.

4.5 Construction and efficient search of the Articulatory Codebook

As discussed in Section 1.8, a codebook is needed in VT inversion by analysis-by-synthesis because of the problems of non-uniqueness of the inverse mapping, the computation-intensive nature of the articulatory-to-acoustic or forward mapping, and local optima in the optimization.

An articulatory codebook $\mathcal{C}$ consists of linked pairs $(\mathbf{x}_m, \mathbf{p}_m)$ of articulatory vectors and corresponding calculated/synthesized acoustic vectors [ACM78, SS92, SS94, Rie97]. i.e., $\mathbf{x}_m = f(\mathbf{p}_m)$, where $f(\cdot)$ is the articulatory-to-acoustic mapping. For a given target natural acoustic vector sequence $\mathbf{x}_{in}(t), t = 1, 2, \ldots, T$ computed from the speech signal, the codebook is searched to retrieve a sequence of articulatory vectors $\mathbf{p}(t), t = 1, 2, \ldots, T$, that corresponds to an acoustic vector sequence $\mathbf{x}(t), t = 1, 2, \ldots, T$ close to the target sequence. The codebook articulatory vector sequence may itself be taken as the inverted sequence, or used to initialize a convex optimization method.

4.5.1 Codebook Construction

The main considerations in the design of an articulatory codebook would be to cover the articulatory and acoustic spaces with desired resolutions, while maintaining a low redundancy. It is not difficult to see that the size of the codebook would grow exponentially with the resolution of the codebook; i.e. the higher the desired resolution in a given space (acoustic or articulatory), the larger, exponentially, the number of samples needed to achieve that resolution and cover the range in that space [Rie97]. Ideally during the search, given an input acoustic vector $\mathbf{x}$ which was produced by articulatory vector $\mathbf{p}$, we want to recover code-

book vectors $(\mathbf{x}_n, \mathbf{p}_n)$ such that $||\mathbf{x} - \mathbf{x}_n|| \leq d_x$ and $||\mathbf{p} - \mathbf{p}_n|| \leq d_p$, where d_x and d_p are maximum distances (resolutions) in acoustic and articulatory spaces, respectively.

Constructing a codebook that is ideal according to the above consideration would require a large amount of computation involving the systematic sampling of articulatory space, first to achieve the desired articulatory resolution and then to explore regions of articulatory space where a small change in articulatory parameter leads to larger changes in acoustics. The hypercube method of codebook construction is a refinement of this idea, where the articulatory parameter space is devided into hypercube regions within which the articulatory-acoustic mapping is approximately linear [OL05].

We followed the method of codebook construction using log formant bins described in [SMP90]. As stated above, we use cepstra and not formants as acoustic features to access the codebook. However, formants can still be calculated for any articulatory configuration and used to construct and organize the codebook since they are important for characterizing VT acoustics and for the perception of vowel quality.

First, cubical bins are formed in log formant space, with width corresponding to a desired relative error in each formant. Starting with a large number of random samples in articulatory space with valid VT area functions, the corresponding first three formants are computed, to get training pairs $(\mathbf{p}_i, \mathbf{F}_i)$. The training pairs are sequentially considered and a training pair $(\mathbf{p}_i, \mathbf{F}_i)$ is added to the formant bin containing $\mathbf{F}_i$ unless that bin already contains a pair $\{\mathbf{p}_j, \mathbf{F}_j\}$ with $||\mathbf{p}_i - \mathbf{p}_j|| \leq d_p$. This avoids redundancy within a bin, although there is some redundancy between adjacent bins. Although this redundancy may be avoided by more careful consideration, it may be useful in efficient search as explained

below.

We first obtained 2×10^6 random pairs of articulatory and acoustic vectors, with the constraints that each articulatory configuration must have a minimum area along the VT greater than 0.05 cm^2, and have total VT length between 14 cm and 19 cm (the VT length for the nominal configuration of the Maeda model is around 16.3 cm). These area and length limits are wide for vowels, which usually have minimum areas greater than 0.15 cm^2, and areas smaller than 0.1 cm^2 typically result in frication [Rie97]. We investigate pruning of the codebook to improve results in Section 4.8.

Using the 2×10^6 training pairs, with a log formant bin width corresponding to 20% relative error, and an ∞-norm of 1 in articulatory space, the codebook size was around 82,000 vectors. Cepstra are computed for the codebook articulatory configurations as in Section 4.3.

The exhaustive coverage of articulatory and acoustic spaces by systematic or random sampling has serious drawbacks. It results in large codebook sizes, and includes many unrealistic articulatory configurations in the codebook which may hinder the retrieval of realistic articulatory trajectories for an input acoustic vector sequence. One reason is that this method does not take into account information about correlations between articulatory parameters, as actually observed in human speech. While the Maeda model imposes a degree of realistic constraints on VT shapes, combinations of extreme values of Maeda model parameters often result in unrealistic or unlikely configurations, which could be eliminated with more information about human VT geometry during speech.

This is one of the challenges faced in VT inversion that was discussed in Section 1.8, that insufficient information is available about VT geometry during speech production. It is not clear, for example, as to how X-ray microbeam

measurements of gold pellets placed on the tongue, lips, teeth, etc. as in the XRMB database [Wes94] could be used to infer information about the possibility or likelihood of different articulatory parameter combinations.

4.5.2 Codebook Search

The bin structure of the codebook in the formant domain can also be exploited for efficient search since it is equivalent to a tree organization in acoustic space. The cepstral centroids of the bins are used to first identify the bin containing an input cepstral vector $\mathbf{x}_{in}(t)$, and the search at time t then continues only in it and neighbouring bins. Since the cepstral centroids were observed to retain formant peak information clearly, and there is some redundancy in articulatory space between adjacent bins in the codebook, further refinement of cepstral clusters was not considered necessary, and search results were satisfactory.

For dynamic speech segments, since the cost function includes the geometric distance, the search for the optimal codevector sequence involves dynamic programming (DP) [SS94]. For the DP search, we used two kinds of pruning. At each time t, from the identified bins for $\mathbf{x}_{in}(t)$, only the best n_1 codevectors according to $E_{acou} + E_{reg}$ were considered for the DP iteration, and after the iteration, only n_2 codevectors were retained for the next iteration. Good search results were obtained even with $n_1 = 200$ and $n_2 = 20$, for a fraction of the original search time. The DP search may be further improved by using distance beams to prune paths instead of n_1-best and n_2-best sorting.

The values of c_{reg}, c_{geo}, $\mathbf{p}_0(t)$, and W_{par} in the cost function need to carefully chosen, sometimes in an ad hoc manner in order to achieve a balance between the simultaneous goals of acoustic match (E_{acou}), realistic inverted VT shapes (E_{reg}) and smooth articulatory trajectories (E_{geo}), and to improve results for a specific

speaker. In particular, a minimum value of c_{reg} was found to be necessary to obtain realistic trajectories for vowels with lip rounding, as seen in Section 4.8.

4.6 Convex optimization of the cost function

After obtaining initial VT shapes using the codebook, further optimization is needed to obtain both a better acoustic match with the input speech, and smoother articulatory trajectories, because of the trade-off between the acoustic and articulatory resolutions of the codebook and the size of the codebook.

We developed an efficient way of calculating the derivative of the CM of the VT with respect to the area function, since the computation of the VT CM is the most expensive step in synthesis as noted at the end of Section 4.3. This was then used in the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method to optimize the cost function of Equation 4.17 [Goc05]. The BFGS method has better (superlinear) asymptotic convergence than some other methods used in the past for optimization of area functions. The direct search methods of [SS94, Sor92] and the iteration in the variational approach of [OL05] which appears to be a type of fixed point method, have linear convergence.

The BFGS method requires $\frac{\partial E}{\partial \mathbf{p}}$, the gradient of the cost function with respect to articulatory parameters (time dependence is ignored for the sake of clarity). $\frac{\partial E_{reg}}{\partial \mathbf{p}}$ and $\frac{\partial E_{geo}}{\partial \mathbf{p}}$ can easily be calculated from Equations 4.19 and 4.20, as shown in Equations A.2 and A.4 of Appendix A. The functional dependencies in computing E_{acou} are (see Figure 4.1):

$$\mathbf{p} \rightarrow \{\mathbf{A}, \mathbf{L}\} \rightarrow \mathbf{T} \rightarrow \hat{\mathbf{T}} \rightarrow \hat{\mathbf{x}} \rightarrow E_{acou} \tag{4.21}$$

$\frac{\partial E_{acou}}{\partial \mathbf{p}}$ can be computed by applying the chain rule.

$$\frac{\partial E_{acou}}{\partial \mathbf{p}} = \frac{\partial E_{acou}}{\partial \hat{\mathbf{x}}} \cdot \frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{T}}} \cdot \frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{T}} \cdot \left(\frac{\partial \mathbf{T}}{\partial \mathbf{A}} \cdot \frac{\partial \mathbf{A}}{\partial \mathbf{p}} + \frac{\partial \mathbf{T}}{\partial \mathbf{L}} \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{p}} \right) \tag{4.22}$$

where the notation $\frac{\partial \mathbf{x}}{\partial \mathbf{y}}$ is used to denote the matrix of partial derivatives $\left[\frac{\partial \mathbf{x}(i)}{\partial \mathbf{y}(j)}\right]$ when $\mathbf{x}$ and $\mathbf{y}$ are both vectors.

$\frac{\partial E_{acou}}{\partial \hat{\mathbf{x}}}$, $\frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{T}}}$ and $\frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{T}}$ are relatively straightforward to calculate from Equations 4.18, 4.9 and 4.8 respectively. The details may be found in Appendix A.

$\frac{\partial \mathbf{A}}{\partial \mathbf{p}}$ and $\frac{\partial \mathbf{L}}{\partial \mathbf{p}}$ can be calculated from the equations of the Maeda articulatory model, which as discussed in Section 1.9 involve the calculation of the midsagittal interior VT outline as a linear combination of basis outline vectors and $\mathbf{p}$, and then calculation of the area function using Equation 1.27.

We focus on the step $\{\mathbf{A}, \mathbf{L}\} \rightarrow \mathbf{T}$, i.e., the chain matrix calculation of the VT transfer function, which is the most computationally intensive step.

4.7 Chain matrix derivatives with respect to the area function

By Equation 4.3, $\mathbf{T}$ depends on the CM parameters $\mathcal{A}$ and $\mathcal{C}$ of the VT and the radiation impedance Z_L. Therefore, to compute $\frac{\partial \mathbf{T}}{\partial \mathbf{A}}$ and $\frac{\partial \mathbf{T}}{\partial \mathbf{L}}$, we need to compute the derivatives of $\mathcal{A}$ and $\mathcal{C}$, which are given by Equations 1.29 to 1.35, with respect to $\{\mathbf{A}, \mathbf{L}\}$. Note that $\mathcal{A}$ and $\mathcal{C}$ are elements of the matrix K in Equation 1.29. The details of the calculation of $\frac{\partial \mathbf{T}}{\partial \mathbf{A}}$ and $\frac{\partial \mathbf{T}}{\partial \mathbf{L}}$ from $\frac{\partial K}{\partial \mathbf{A}}$ and $\frac{\partial K}{\partial \mathbf{L}}$ are given in Section A.2 of Appendix A.

We first calculate $\frac{\partial K}{\partial A_n}$. Observe from Equations 1.32 to 1.35, that the CM of each section depends only on its own area and length, and not on those of other

sections. This simplifies the derivative calculation from Equation 1.29:

$$\frac{\partial K}{\partial A_n} = [K_N \cdots K_{n+1}] \cdot \frac{\partial K_n}{\partial A_n} \cdot [K_{n-1} \cdots K_1] \tag{4.23}$$

If we define:

$$P_n = K_{n-1} K_{n-2} \cdots K_1, \quad 2 \leq n \leq N \tag{4.24}$$

$$Q_n = K_N K_{N-1} \cdots K_{n+1}, \quad 1 \leq n \leq N-1 \tag{4.25}$$

and let:

$$P_1 = Q_N = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.26}$$

then:

$$\frac{\partial K(\mathbf{A}, \mathbf{L})}{\partial A_n} = Q_n \cdot \frac{\partial K_n}{\partial A_n} \cdot P_n, \quad 1 \leq n \leq N \tag{4.27}$$

From Equations 1.32 to 1.35, we can show:

$$\frac{\partial \mathcal{A}_n}{\partial A_n} = 0 \tag{4.28}$$

$$\frac{\partial \mathcal{B}_n}{\partial A_n} = \frac{\rho c}{A_n^2} \gamma \sinh(\sigma L_n / c) = -\frac{1}{A_n} \cdot \mathcal{B}_n \tag{4.29}$$

$$\frac{\partial \mathcal{C}_n}{\partial A_n} = -\frac{1}{\rho c} \frac{\sinh(\sigma L_n / c)}{\gamma} = \frac{1}{A_n} \cdot \mathcal{C}_n \tag{4.30}$$

$$\frac{\partial \mathcal{D}_n}{\partial A_n} = 0 \tag{4.31}$$

Therefore, $\frac{\partial K_n}{\partial A_n}$ is very easily obtained from the elements of K_n.

The partial derivatives with respect to the lengths of the area function can also similarly be calculated from

$$\frac{\partial K(\mathbf{A}, \mathbf{L})}{\partial L_n} = Q_n \cdot \frac{\partial K_n}{\partial L_n} \cdot P_n, \quad 1 \leq n \leq N \tag{4.32}$$

From Equations 1.32 to 1.35:

$$\frac{\partial \mathcal{A}_n}{\partial L_n} = \frac{\sigma}{c} \cdot \sinh(\sigma L_n / c) \tag{4.33}$$

$$\frac{\partial \mathcal{B}_n}{\partial L_n} = -\frac{\rho}{A_n} \cdot (\gamma\sigma) \cdot \cosh(\sigma L_n / c) \tag{4.34}$$

$$\frac{\partial \mathcal{C}_n}{\partial L_n} = -\frac{A_n}{\rho c^2} \cdot \frac{\sigma}{\gamma} \cdot \cosh(\sigma L_n / c) \tag{4.35}$$

$$\frac{\partial \mathcal{D}_n}{\partial L_n} = \frac{\sigma}{c} \cdot \sinh(\sigma L_n / c) \tag{4.36}$$

Many of the quantities involved in the calculation of $\frac{\partial K_n}{\partial L_n}$ are also already available from the calculation of K_n.

Note that the calculation of K involves, already, the calculation of either $\{P_n,\ 2 \leq n \leq N\}$ or $\{Q_n,\ 1 \leq n \leq N-1\}$, which, performed in a recursive manner, require $(N-2)$ chain matrix multiplications (CMMs) each. If we assume that $\{P_n\}$ has already been obtained during the calculation of K, then $\{Q_n\}$ requires another $(N-2)$ CMMs to compute. Computing $\frac{\partial K}{\partial \mathbf{A}}$ using Equation 4.27 requires another $2(N-1)$ CMMs (since $P_1 = Q_N = I$). Similarly for $\frac{\partial K}{\partial \mathbf{L}}$ from Equation 4.32. In total, we need approximately $5N$ additional CMMs to compute both $\frac{\partial K}{\partial \mathbf{A}}$ and $\frac{\partial K}{\partial \mathbf{L}}$, which is around five times the number of CMMs required for the computation of K (which requires $(N-1)$ CMMs).

Computational efficiency:

A careful count of the real multiplications involved shows that computation of $\frac{\partial E_{acou}}{\partial \mathbf{p}}$ using the above analytical calculation of $\frac{\partial \mathbf{T}}{\partial \mathbf{A}}$ and $\frac{\partial \mathbf{T}}{\partial \mathbf{L}}$ is around 2.4 times as efficient as a finite-difference approximation for the 7 parameter Maeda model, even assuming that the finite difference derivatives are computed using the efficient forms of Equations 4.27 and 4.32. If the analytical calculation was used in conjunction with an articulatory model with more parameters (like the Mermelstein model [Mer73]), the advantage in efficiency would be higher.

4.8 Results of VT Inversion Experiments

The inversion method was evaluated on diphthongs from the University of Wisconsin X-ray microbeam (XRMB) speech production database [Wes94], which was briefly described in Section 1.8. As explained there, in the XRMB database, articulatory data are available in the form of x-ray microbeam measurements of gold pellets placed on the tongue, teeth/jaw, and lips, along with simultaneously recorded acoustic data, for several speakers uttering a series of tasks. We evaluate the inversion geometrically by comparing inverted VT outlines against measured positions of tongue and lip XRMB pellets.

Results were obtained for two speakers, one female and one male, from the XRMB database. For the female speaker ("JW46") the VT external outline (palate and rear pharyngeal wall) was similar in scale and shape to that of the Maeda articulatory model, and the model was used without any adaptation. For the male speaker ("JW11"), limited adaptation of the Maeda model was performed by overall scaling of the VT, and modifying the palate and pharyngeal wall outlines according to the measured outlines provided in the database. A speaker-specific codebook was therefore also constructed. Detailed speaker adaptation of the Maeda model would probably also involve separate scaling factors for the oral and pharyngeal regions of the VT, and modifying the coefficients used to convert mid-sagittal widths to cross-sectional areas ($\alpha(x)$ and $\beta(x)$ in Equation 1.27) and the basis vectors used to compute the tongue outline from the jaw and tongue parameters [ML97, Mae90]. This would be a topic of future work.

We evaluated the inversion on three diphthongs: /ai/, /au/ and /oi/, taken from the middle of words of the form /sVd/ where V is the vowel, as contained in utterrance task #13 of the XRMB database. We downsampled speech signals to

8kHz, and computed 20 linear prediction cepstral coefficients (LPCCs) from 20ms frames with an LPC order of 10, for inversion. Frames were centered around times at which XRMB pellet positions were measured, with a frame rate of around 146Hz. A lower frame rate would suffice and will be explored in the future. Natural formants were also manually extracted from the LPC analysis of the speech signals, and used for acoustic evaluation of inversion results. For matching the natural LPCCs, synthetic DFT cepstra were computed with a DFT size of 64, i.e., using the transfer function computed at 33 frequency points between 0 and 4kHz (included) as in Section 4.3.

4.8.1 Codebook Search

The goals of VT inversion are to obtain a good match between input and synthetic acoustic features (i.e., low E_{acou}), realistic inverted VT shape sequences (related to E_{reg}) and smooth articulatory trajectories (low E_{geo}). The values of c_{reg}, c_{geo}, $\mathbf{p}_0(t)$, and W_{par} in the cost function are carefully chosen, as discussed in Section 4.4, to achieve a balance between these three simultaneous goals.

The codebook search needs to return an initial articulatory sequence that is realistic, close to the expected optimal sequence, and serves as a good starting point for the subsequent optimization stage. Therefore, the codebook search needs to resolve the non-uniqueness issue of the inverse mapping to a large extent, and reject unlikely or unrealistic articulatory trajectories.

The geometric continuity term in the cost function, E_{geo} alone was found to be insufficient to resolve the non-uniqueness of the acoustic-to-articulatory mapping for /au/ and /oi/, where more than one articulatory trajectory was observed to correspond to the same trajectories for the first three formants. For low values of c_{reg}, the codebook search often selected unrealistic VT configurations with

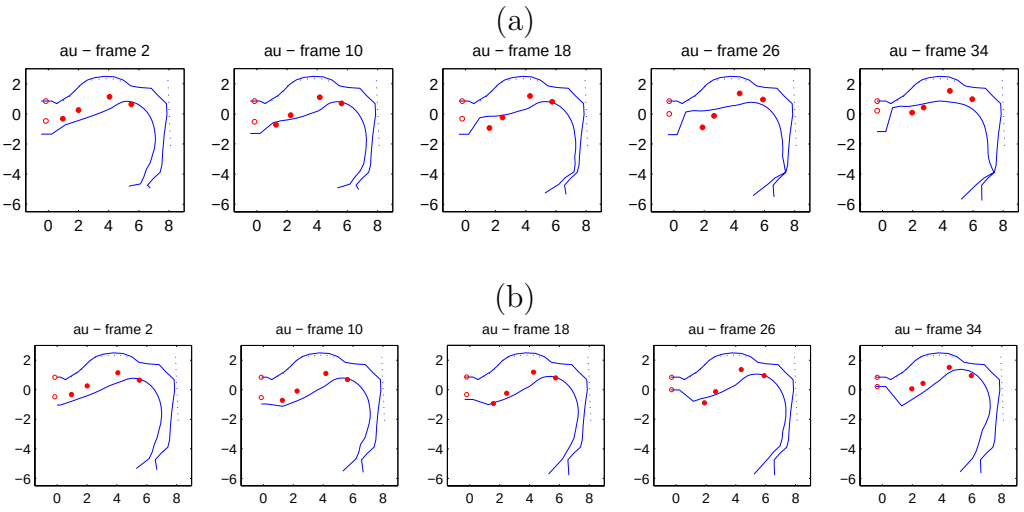

Figure 4.3: Results of codebook search for /au/ of JW46. (a) Unrealistic articulatory trajectory for a low value of c_{reg} in Equation 4.17 (b) More realistic articulatory trajectory obtained with a larger value for c_{reg}. The four measured XRMB tongue pellet positions are plotted using solid circles while the two shifted lip pellets are represented by empty circles.

an elongated larynx and wide open mouth where lip-rounding was expected, as shown in Figure 4.3 (a). To obtain more realistic VT configurations with lip rounding that are closer to the measured pellet positions as in Figure 4.3 (b), a minimum value of c_{reg} was found to be necessary, along with greater penalty (between 5 and 10) on the larynx height parameter relative to other parameters in W_{par}.

For the plots comparing inverted Maeda model VT outlines with measured XRMB pellet positions, the model outlines were shifted so that the model and measured palate ends behind the teeth are aligned. Lip pellets were shifted vertically by the approximate height between them during a token of /m/ for the speaker, and horizontally averaged and shifted by an ad hoc speaker-specific distance.

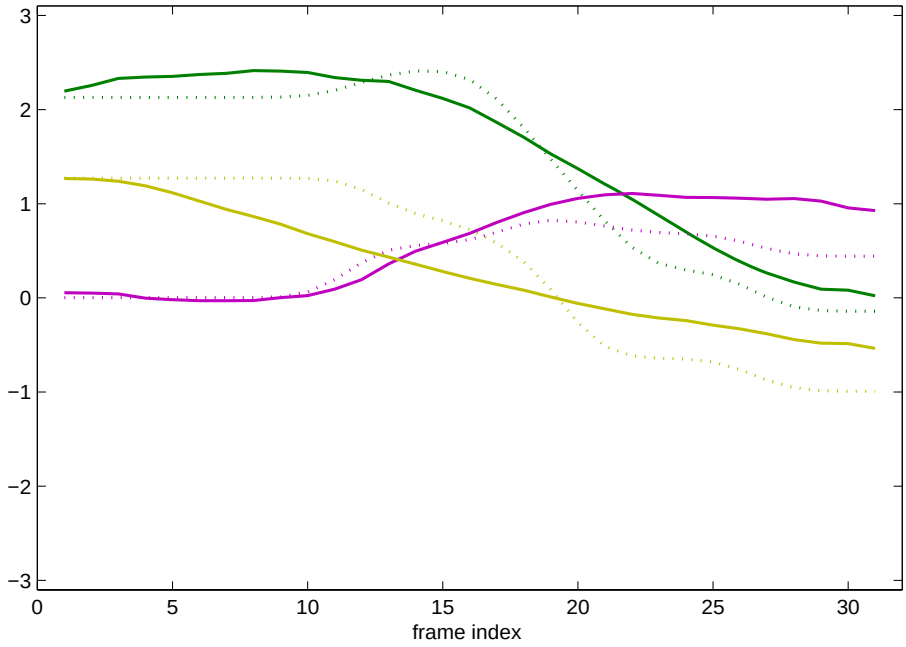

Figure 4.4: Example of articulatory parameters before (dotted lines) and after (solid lines) optimization.

Adding a bias to the larynx height regularization parameter to shorten the larynx was also found to be necessary for JW11 and useful for JW46. Increasing the penalty on the jaw parameter (to 3) also improved results for JW11. As explained earlier, such ad hoc measures are needed because of insufficient information about the vocal tract geometry and of vocal tract dynamics. For example, the bias to the larynx height parameter to shorten it is needed probably because the larynx for the speaker is shorter than that in the model. Since no information about the laryngeal region is available in the XRMB database, the total length of the combined pharyngeal and laryngeal regions could not be adapted separately from the oral region as suggested in [ML97, OL05].

4.8.2 Results of Optimization

The strategy for improving acoustic match and smoothening articulatory trajectories after codebook search was as follows. The parameter trajectories obtained from codebook lookup are first smoothed using a short hamming window as this helps to reduce E_{geo} without affecting E_{acou} much. In the BFGS optimization, a smaller value of c_{reg} and a larger value of c_{geo} were used compared to their corresponding values in codebook search, and the articulatory parameter sequence $\mathbf{p}_{init}(t)$ after codebook search and smoothing was used for regularization as mentioned in Section 4.4. The BFGS iterations were stopped when the decrease in acoustic cost fell below a threshold (1%). The total inversion time for a speaker, with 3 diphthongs of around 30 frames each, was around 45-55 seconds in Matlab running on an AMD Athlon 4.2GHz processor, with each codebook search taking around 1 second for $n_1 = 500$ and $n_2 = 50$. After optimization, inverted articulatory trajectories varied smoothly as would be expected for a human talker, and the average relative errors in the first three formants for the three diphthongs were around 3% and 2% for JW46 and JW11 respectively.

Figure 4.4 shows an example of articulatory parameters before (dotted lines) and after (solid lines) optimization. It can be seen that the parameters vary more smoothly after optimization.

Measured XRMB gold pellet positions are plotted against the VT outlines obtained from inversion and shown for five evenly spaced frames each from /ai/, /oi/ and /au/ of JW46 in Figure 4.5. The match between inverted VT outlines and measured pellet positions is observed to be very good for /oi/, reasonably good for /au/, and okay for /ai/ of JW46.

Sample plots comparing natural and computed log spectra and formants are shown in Figures 4.6 and 4.7 respectively. It is seen that cepstral matching

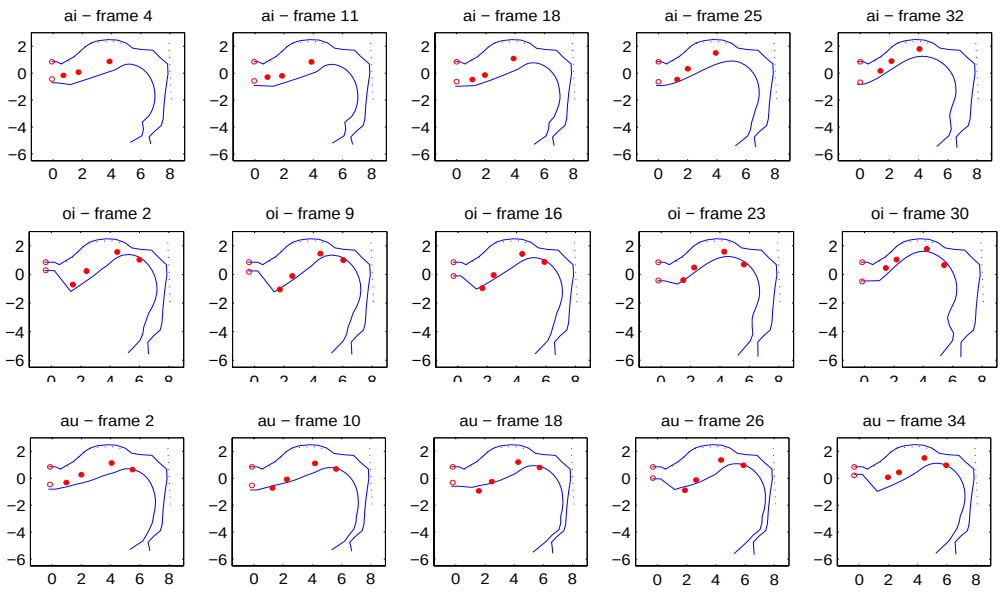

Figure 4.5: Speaker JW46, (a) (first row) /ai/ from 'side' (b) (second row) /oi/ from 'soyed' (c) (third row) /au/ from 'saud' - Measured XRMB tongue (solid circles) and shifted lip (empty circles) pellet positions plotted against inverted VT outlines (solid lines). Measured palate and pharyngeal outlines are plotted using dotted lines.

effectively results in formant matching.

For speaker JW11, measured XRMB gold pellet positions are plotted against the VT outlines obtained from inversion and shown for five evenly spaced frames each from /ai/, /oi/ and /au/ in Figure 4.8.

For JW11, although the acoustic match was excellent (around 2% average relative errors in the first three formants) and the inverted VT outline approximately followed the curve of tongue pellet positions, the pellets were found to lie slightly away from the inverted outline for all three diphthongs. This is probably due to lack of appropriate constraints in the regularization and inadequate adaptation of the Maeda model to the speaker, and needs to be investigated.

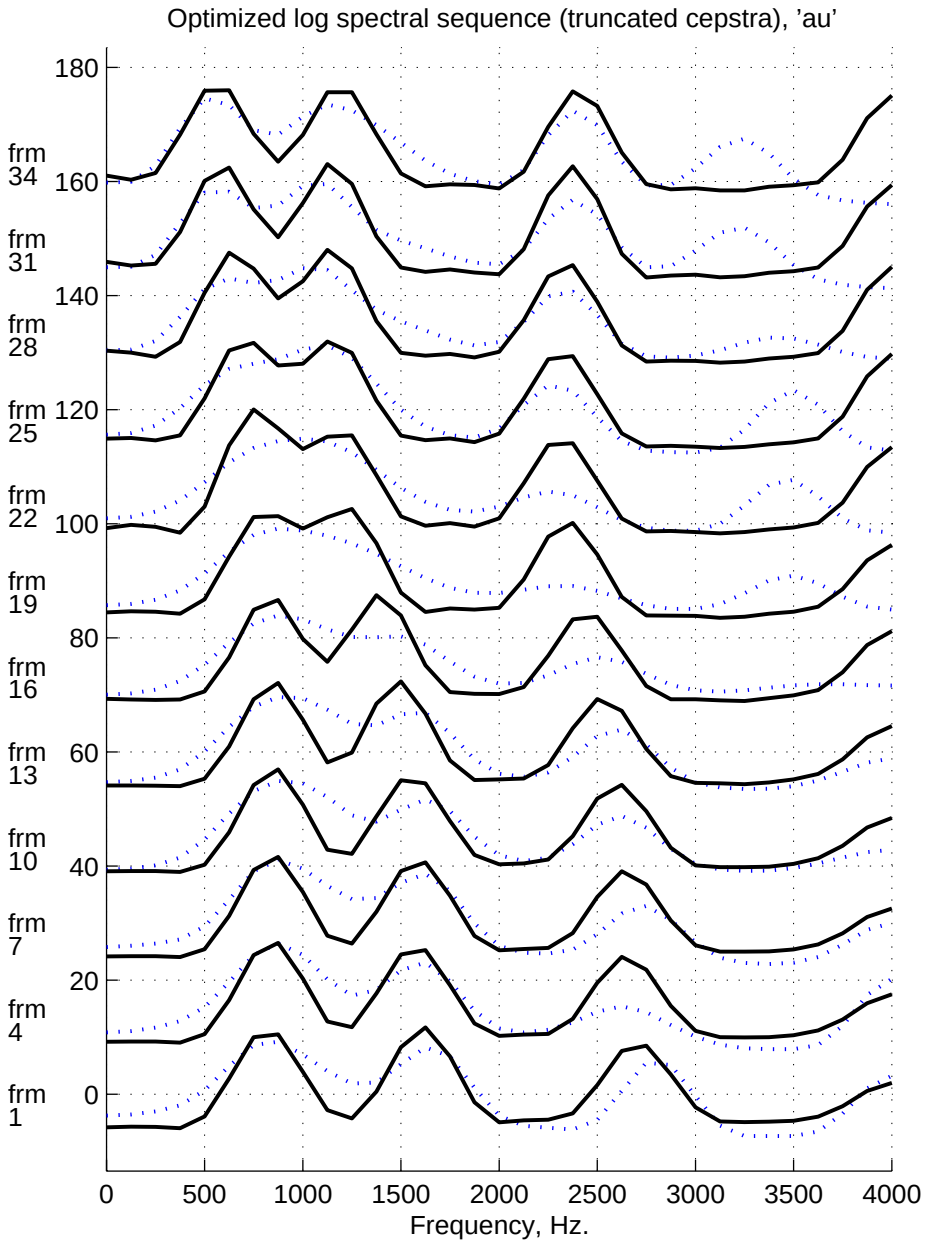

Figure 4.6: Speaker JW46, Natural (dotted lines) and computed (solid lines) log spectra (from truncated and liftered cepstra) for /au/. The frame indices are given to the left of the vertical axis. (see corresponding formants in Figure 4.7)

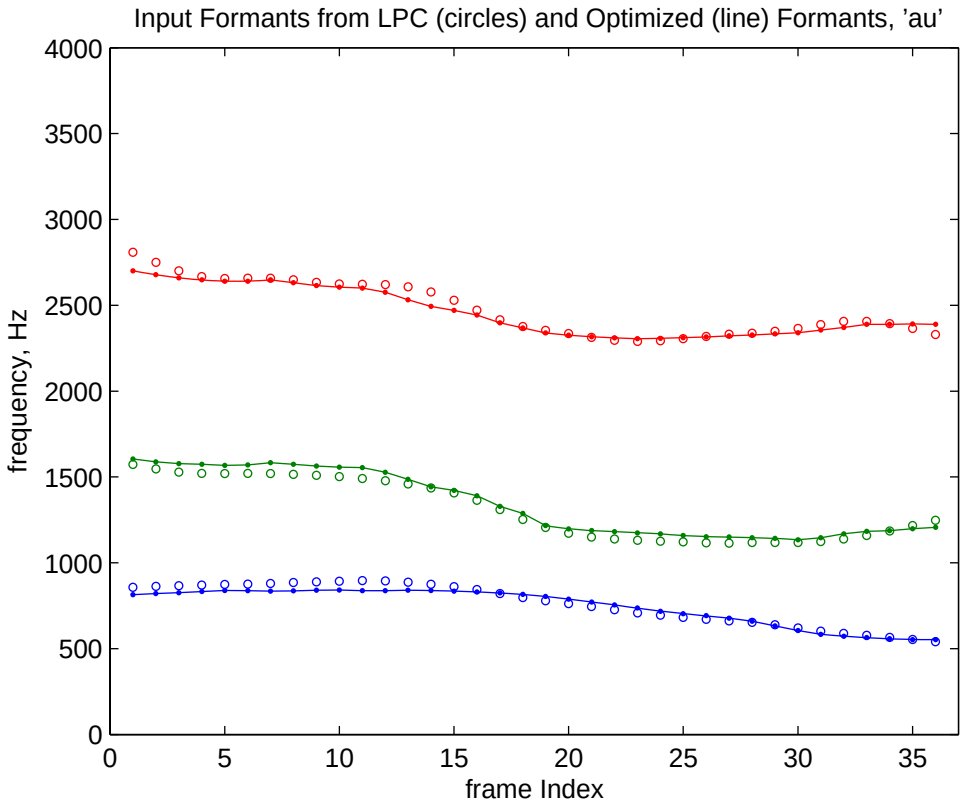

Figure 4.7: Speaker JW46, Natural (circles) and computed (lines) formants for /au/ (see corresponding log spectra in Figure 4.6)

4.9 Discussion

In Chapter 1, we had listed the main challenges faced in VT inversion to be (1) complexity of speech production models, (2) inherent non-uniqueness of the inverse mapping, and local optima of the cost function, (3) incomplete knowledge about the shape and dynamics of the vocal tract for a given speaker, and (4) insufficient data to learn from or to evaluate the inversion results.

We can now assess the effect of each of these issues on the successes and failures of our inversion method. It is clear that all four factors remain big challenges in VT inversion.

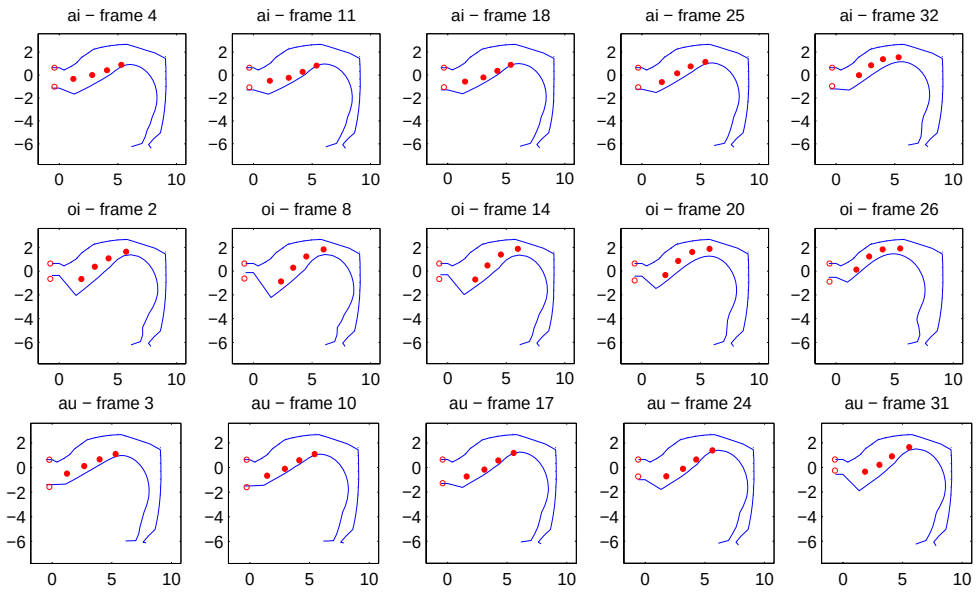

Figure 4.8: Speaker JW11, (a) (first row) /ai/ from 'side' (b) (second row) /oi/ from 'soyed' (c) (third row) /au/ from 'saud' - Measured XRMB tongue (solid circles) and shifted lip (empty circles) pellet positions plotted against inverted VT outlines (solid lines).

We developed efficient optimization techniques to deal with the complexity of the articulatory-to-acoustic mapping to some extent.

Much improvement could be made with the codebook used to initialize the optimization. First, some knowledge of correlations between articulatory parameters would enable us to prune out unrealistic VT configurations from the codebook such as those encountered in the inversion of /au/ and /oi/. A mapping from XRMB pellet positions to Maeda articulatory parameters would be very useful in this regard, so that better articulatory constraints could be easier learned. With such a mapping, inverted articulatory parameter trajectories could also be compared with actual ones.

We have also not used any a priori model of articulatory dynamics, and used

only the constraints provided by the articulatory model and the regularization and continuity terms in the cost function. The inversion could be improved by using a model of articulatory dynamics such as the task dynamic model from gestural phonology, where the fundamental units of speech production are modeled as gestures, which are the coordinated action of articulators [SM89].

Results can also be improved with more information about the VT geometry for the given speaker, mainly the entire exterior VT outline consisting of the hard and soft palates and rear pharyngeal wall extending down to the laryngeal region. The XRMB database does not include information on the soft palate (velum) and on the laryngeal region, which are limiting factors in our experiments since the length of the pharyngeal region could not be adapted.

The coefficients $\alpha(x)$ and $\beta(x)$ used to convert midsagittal widths to cross-sectional areas in the Maeda model would also vary from speaker to speaker, and need to be adapted for improved results. The parameters used in calculating the chain matrix of a tube section may also be adapted.

The optimization approach we have developed in this chapter has the advantage that it can be extended without much difficulty to optimize all these different parameters.

4.10 Summary

In this chapter, we performed VT inversion for vowels by cepstral analysis-by-synthesis using chain matrices and the Maeda articulatory model. We first discussed the computational advantage of optimizing cepstra rather than formants, and then gave the details of the articulatory-to-acoustic mapping to compute DFT cepstra from Maeda model parameters. The equation for the optimization

cost function consisting of acoustic, regularization and articulatory continuity terms was given, and the form of the cost function was carefully chosen to obtain a balance between acoustic match, realistic VT shapes, and smooth articulatory trajectories. We described the construction of the articulatory codebook organized using cubical bins in log formant space, and efficient pruning for dynamic programming search of the codebook for initializing the optimization. We developed a novel efficient calculation of the derivative of the VT chain matrix with respect to the area function which was used to compute the gradient of the cost function. The BFGS quasi-Newton method was used to optimize the cost function given its gradient. The inversion method was evaluated on diphthongs of two speakers from the X-ray microbeam database and limited adaptation of the Maeda model was performed for one speaker. A minimum weight on the regularization term, and related constraints on the articulatory parameters were found to be necessary to obtain realistic VT shapes for /au/ and /oi/. Good geometric match was achieved between inverted midsagittal VT outlines and measured XRMB tongue and lip pellet positions, with smooth optimized articulatory trajectories and an average relative error of less than 3% between the first three synthesized and natural formants.

CHAPTER 5

Summary and Future Work

5.1 Summary

In this dissertation, we present a novel linear transform (LT) equivalent for frequency warping (FW) with standard Mel frequency cepstral coefficient (MFCC) features in speech recognition, and also develop efficient techniques for vocal tract inversion of vowel sounds by cepstral analysis-by-synthesis using chain matrices.

Chapter 1 presents motivations for our investigations, explains the fundamentals of statistical speech recognition and the computation of standard filterbank based MFCC features, and introduces frequency warping (FW) for vocal tract length normalization (VTLN). The important advantages of using a LT for FW are: VTLN estimation by optimizing the Maximum Likelihood Score (MLS) criterion is performed computationally more efficiently with a LT; the transform can also be estimated and applied in the back end to HMM means; and one need not have access to or reconstruct the intermediate linear frequency spectrum in order to apply the FW, which would be useful in distributed speech recognition (DSR). Chapter 1 also introduces vocal tract inversion using analysis-by-synthesis, and the challenges and issues involved. The Maeda articulatory model and chain matrix calculation of the VT transfer function are discussed.

Chapter 2 presents our novel LT for FW with standard MFCC features. The main idea is to directly warp the smoothed log Mel spectrum obtained by cosine interpolation of the log Mel filterbank output with the IDCT. This results in a linear transformation in the Mel cepstral domain. The warping was parametrized and incorporated into a warped type-II IDCT matrix, which can be easily calculated using a compact formula. Estimation of FW parameters for VTLN and speaker adaptation using the MLS criterion and EM auxiliary function were discussed, and formulae for calculating the gradient of the EM auxiliary function with respect to the warping parameters were derived. Our LT for MFCCs was also shown to be closely related to earlier proposed plain cepstral LTs of McDonough, Pitz et al. and Umesh et al. [McD00, PN03, UZN05]. In fact, these LTs for FW are all found to be numerically almost identical for the sine-log all-pass transform (SLAPT) warping functions, which had not been observed earlier in the literature. Our LT matrix formula is, however, computationally simpler and unlike some other previous linear transform approaches to VTLN with MFCC features, no modification of the standard MFCC feature extraction scheme is required.

Chapter 3 presents results of speech recognition experiments using our LT for VTLN and speaker adaptation. We validated our LT on continuous speech recognition with the Resource Management (RM1) database. In VTLN and VTLN Speaker Adaptive Modeling (SAM, see Section 3.3) experiments with the RM1 database, the performance of the new LT VTLN was comparable to that of Regular VTLN. For the LT, the inclusion of the Jacobian normalization term in the MLS criterion was found to be quite important for convergence of the FW parameters during training using SAM. During testing, however, better results were obtained without the Jacobian determinant term in the MLS criterion. Our LT was also found to perform better than the earlier proposed transform of Cui and Alwan [CA06] for approximate VTLN with MFCC features, when the MLS

criterion was used to estimate the FW parameter. This would be an advantage in DSR where only recognition features are available and the linear frequency spectrum needs to be reconstructed in order to locate formant-like peaks for FW estimation as in [CDB98, CA06].

LT adaptation of HMM means combined with MLLR (Maximum Likelihood Linear Regression) mean bias and variance adaptation typically gave results that were comparable to the front end VTLN methods. The FW based methods were found to be significantly better than MLLR for limited adaptation data. We also performed Speaker Adaptive Training (SAT) with feature space LT denoted CLTFW. Global CLTFW SAT models with the piecewise-linear (PL) FW, tested with global PL CLTFW gave results comparable to SAM and VTLN, and the performance saturates with increasing adaptation data. By estimating multiple parameter SLAPT-5 (5-parameter sine-log all-pass transform FW) CLTFW transforms using a regression tree, and including an additive bias, we obtained significantly better performance than global VTLN, and improving results with increasing adaptation data. Warping factors estimated in an unsupervised mode were almost identical with those from supervised estimation, and therefore the performance of unsupervised VTLN and model adaptation with the LT were almost as good as with supervised VTLN and adaptation.

In Chapter 4 we describe our approach for vocal tract (VT) inversion by cepstral analysis-by-synthesis using chain matrices and the Maeda articulatory model. The different issues addressed include the choice of acoustic features, the articulatory-to-acoustic mapping, the optimization cost function, construction and search of articulatory codebooks, and optimization of the cost function. The computation of DFT cepstra and the incorporation of liftering, log spectral weighting and Mel warping into a linear matrix on cepstra were discussed. The

forms of the acoustic, regularization and articulatory continuity terms, and the various parameters in the cost function are carefully chosen to obtain a balance between acoustic match, realistic VT shapes, and smooth articulatory trajectories. The construction of the formant bin codebook, and codebook search using dynamic programming were described. We developed a novel efficient calculation of the derivative of the VT chain matrix with respect to the area function which was used to compute the gradient of the cost function. The BFGS quasi-Newton method was used to optimize the cost function given its gradient. Results of inversion on diphthongs of two speakers from the X-ray microbeam (XRMB) database were presented, and issues involved in adaptation of the Maeda model to a specific speaker were addressed. A minimum weight on the regularization term, was found to be necessary to obtain realistic VT shapes with lip rounding for /au/ and /oi/. Some ad hoc constraints had to be placed using the regularizing parameter sequence and the parameter weighting matrix to improve results. Good geometric match was achieved between inverted midsagittal VT outlines and measured XRMB tongue and lip pellet positions, for /oi/ and /au/ of female speaker JW46 and /oi/ of male speaker JW11. Further improvement is needed for the other cases. The optimized articulatory trajectories varied smoothly and the average relative error between the first three synthesized and natural formants after optimization were around 3% for JW46 and 2% for JW11.

5.2 Challenges and Outlook

Our experimental results with LT VTLN are only comparable in performance to regular VTLN on the RM1 database. Since our aim was to obtain a linear transform equivalent for VTLN with standard MFCC features, it is important to demonstrate that the involved approximations do not lead to performance

degradation. It is probably also not to be expected that an approximation would perform better than the original method. By estimating multiple transforms using the EM auxiliary function and a regression tree, we have also shown that it is possible to obtain results better than global VTLN. It would be the topic of future work to compare and/or combine multi-class CLTFW with MLLR adaptation.

Though the computations required for VTLN implementation may be small compared to the overall effort for training and testing, the computational advantage of LT VTLN over regular VTLN discussed in Section 1.6 becomes significant when the VTLN parameter has to be estimated in real time. For example, in DSR, the computational savings during FW parameter estimation, the ability to estimate and implement VTLN directly on the features without having access to the feature extraction modules and the flexibility of application (front-end or back-end) would be a significant advantage of LT over regular VTLN. We believe that the proposed linear transform would prove very useful in practice in embedded and distributed speech recognition applications, where resources are limited.

Future work would also be aimed at extending the inversion method to other speech sounds such as nasals and fricatives. In combination with a phonological model of articulatory dynamics such as the task dynamic model [SM89], inversion could be performed in an improved manner for a given entire speech signal. As discussed in Chapter 4, a method of mapping XRMB pellet positions to Maeda model parameters would be very useful in learning or pruning the codebook used in inversion, and in more detailed adaptation of the articulatory model to different speakers. The results of Chapter 4 also suggest that cepstral analysis-by-synthesis could be used to estimate VT resonances (VTRs) for vowels and other speech sounds.

Finally, as noted in the introduction to this dissertation, if mappings can be found between VTR or VT shape patterns of different speakers for a given speech sound, these could be used to make a speech recognition system more robust to speaker variations.

APPENDIX A

Calculations of Derivatives for Convex Optimization in Vocal Tract Inversion

In this appendix, we fill in some of the details of the calculation of the derivative of the optimization cost function in VT inversion, referred to in Chapter 4.

A.1 Derivative of the Cost Function for VT Inversion

From Equation 4.17, we have:

$$\frac{\partial E}{\partial \mathbf{p}(t)} = \frac{\partial E_{acou}}{\partial \mathbf{p}(t)} + c_{reg} \cdot \frac{\partial E_{reg}}{\partial \mathbf{p}(t)} + c_{geo} \cdot \frac{\partial E_{geo}}{\partial \mathbf{p}(t)} \tag{A.1}$$

From Equation 4.19,

$$\frac{\partial E_{reg}}{\partial \mathbf{p}(t)} = 2W_{par}[\mathbf{p}(t) - \mathbf{p}_0(t)] \tag{A.2}$$

Writing Equation 4.20 as:

$$E_{geo} = \sum_{t=1}^{T-1} ||\Delta \mathbf{p}(t)||^{2\eta} \tag{A.3}$$

where $\Delta\mathbf{p}(t) = \mathbf{p}(t+1) - \mathbf{p}(t)$, it is easy to see that

$$\frac{\partial E_{geo}}{\partial \mathbf{p}(t)} = \begin{cases} (-2\eta||\Delta\mathbf{p}(t)||^{2(\eta-1)}) \cdot \Delta\mathbf{p}(t) & t = 1 \\ [(2\eta||\Delta\mathbf{p}(t-1)||^{2(\eta-1)}) \cdot \Delta\mathbf{p}(t-1) & \\ \quad - (2\eta||\Delta\mathbf{p}(t)||^{2(\eta-1)}) \cdot \Delta\mathbf{p}(t)] & 2 \leq t \leq T-1 \\ (2\eta||\Delta\mathbf{p}(t-1)||^{2(\eta-1)}) \cdot \Delta\mathbf{p}(t-1) & t = T \end{cases} \tag{A.4}$$

In the computation of $\dfrac{\partial E_{acou}}{\partial \mathbf{p}}$ using the chain rule shown in Equation 4.22, we have:

$$\frac{\partial E_{acou}}{\partial \hat{\mathbf{x}}(t)} = \gamma(E_{acou}(t))^{\gamma-1} 2W_{cep}[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{in}(t)] \tag{A.5}$$

where $E_{acou}(t) = \left([\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{in}(t)]^T W_{cep}[\hat{\mathbf{x}}(t) - \hat{\mathbf{x}}_{in}(t)]\right)^{\gamma}$

From Equation 4.9, we have

$$\frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{T}}} = C \tag{A.6}$$

and from Equation 4.8

$$\frac{\partial \hat{\mathbf{T}}}{\partial \mathbf{T}} = \text{diag}\left(\frac{1}{T(f_0)}, \frac{1}{T(f_1)}, \ \ldots, \frac{1}{T(f_{N_f})}\right) \tag{A.7}$$

A.2 Derivatives of the transfer function with respect to the area function

In this section, we give some of the details in the calculation of $\dfrac{\partial \mathbf{T}}{\partial \mathbf{A}}$ and $\dfrac{\partial \mathbf{T}}{\partial \mathbf{L}}$ for the chain-matrix approach, which were referred to in Section 4.7.

Assume, initially, that $\mathbf{L}$ is fixed and does not vary. Then, the transfer function depends only on $\mathbf{A}$. If the transfer function is $H = H(f; \mathbf{A})$, the magnitude is

$$T(f; \mathbf{A}) = |H(f; \mathbf{A})| = \sqrt{H_R^2 + H_I^2}$$

where $H_R = \mathrm{Re}[H(f)]$ and $H_I = \mathrm{Im}[H(f)]$. Then,

$$\frac{\partial T}{\partial A_n} = \frac{1}{T}\left[H_R \frac{\partial H_R}{\partial A_n} + H_I \frac{\partial H_I}{\partial A_n}\right] \tag{A.8}$$

Since

$$\frac{\partial H_R}{\partial A_n} = \mathrm{Re}\frac{\partial H}{\partial A_n} \tag{A.9}$$

$$\frac{\partial H_I}{\partial A_n} = \mathrm{Im}\frac{\partial H}{\partial A_n} \tag{A.10}$$

we need to calculate $\dfrac{\partial H}{\partial A_n}$.

By Equation 1.30,

$$\frac{\partial H}{\partial A_n} = \frac{-1}{(\mathcal{A} - \mathcal{C}Z_L)^2} \cdot \left[\frac{\partial \mathcal{A}}{\partial A_n} - Z_L \frac{\partial \mathcal{C}}{\partial A_n} - \mathcal{C}\frac{\partial Z_L}{\partial A_n}\right] \tag{A.11}$$

$$= -H^2 \cdot \left[\frac{\partial \mathcal{A}}{\partial A_n} - Z_L \frac{\partial \mathcal{C}}{\partial A_n} - \mathcal{C}\frac{\partial Z_L}{\partial A_n}\right] \tag{A.12}$$

The calculation of $\dfrac{\partial \mathcal{A}}{\partial A_n}$ and $\dfrac{\partial \mathcal{C}}{\partial A_n}$ was shown in Section 4.7.

Z_L, the radiation impedance at the lips was given in Equation 1.31:

$$Z_L = \frac{\rho\omega^2}{2\pi c} + j\frac{8\rho\omega}{3\pi^2 r}$$

where $\omega = 2\pi f$ and r is the radius of the lip opening. Using $r = \sqrt{\dfrac{A_N}{\pi}}$ where A_N is the area of the lip opening, we have:

$$Z_L = \frac{\rho\omega^2}{2\pi c} + j\frac{8\rho\omega}{3\pi^{3/2}} A_N^{-1/2} \tag{A.13}$$

Therefore,

$$\frac{\partial Z_L}{\partial A_N} = j\frac{8\rho\omega}{3\pi^{3/2}} \cdot \left(-\frac{1}{2} A_N^{-3/2}\right) \tag{A.14}$$

$$= -j\frac{4\rho\omega}{3\pi^3 r^3} \tag{A.15}$$

again using $A_N = \pi r^2$.

Since Z_L depends only on A_N, we also have:

$$\frac{\partial Z_L}{\partial A_n} = 0,\ 1 \leq n \leq N-1 \tag{A.16}$$

References

[ACM78] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey. "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique." *JASA*, **63**(5):1535–1555, 1978.

[AMS96] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. "A Compact Model for Speaker-Adaptive Training." In *Proc. ICSLP '96*, volume 2, pp. 1137–1140, Philadelphia, PA, 1996.

[AR89] B. S. Atal and O. Rioul. "Neural networks for estimating articulatory positions from speech." *J. Acoust. Soc. Am. Suppl.1*, **86**:S67, 1989.

[Bau72] L. E. Baum. "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes." *Inequalities*, **3**:1–8, 1972.

[Bil97] J. A. Bilmes. "A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.", 1997.

[CA05] X. Cui and A. Alwan. "MLLR-Like Speaker Adaptation Based on Linearization of VTLN with MFCC features." In *Proc. Interspeech*, pp. 273–276, 2005.

[CA06] X. Cui and A. Alwan. "Adaptation of Children's Speech with Limited Data Based on Formant-like Peak Alignment." *Computer Speech and Language*, **20**(4):400–419, October 2006.

[CA07] X. Cui and A. Alwan. "Robust speaker adaptation by weighter averaging based on the Minimum Description Length criterion." *IEEE Transactions on Audio, Speech and Language Processing*, **15**(2):652–660, February 2007.

[CDB98] T. Claes, I. Dologlou, L. ten Bosch, and D. Van Compernolle. "A novel feature transformation for vocal tract length normalization in

automatic speech recognition." *IEEE Transactions on Speech and Audio Processing*, **6**(6):549–557, November 1998.

[DCP06] Li Deng, Xiaodong Cui, Robert Pruvenok, Jonathan Huang, Safiyy Momen, Yanyi Chen, and Abeer Alwan. "A Database of Vocal Tract Resonance Trajectories for Reasearch in Speech Processing." In *Proceedings of IEEE ICASSP*, volume I, p. 369, 2006.

[DLR77] A. Dempster, N. Laird, and D. Rubin. "Maximum Likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society B*, **39**(1):1–38, 1977.

[DM80] S. B. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." *IEEE Transactions on Acoustics, Speech and Signal Processing*, **28**:357–366, Aug 1980.

[Dus00] S. Dusan. *Statistical estimation of articulatory trajectories from the speech signal using dynamic and phonological constraints.* Ph.d. thesis, University of Waterloo, 2000.

[FIS80] J. L. Flanagan, K. Ishizaka, and K. L. Shipley. "Signal models for low bit-rate coding of speech." *J. Acoust. Soc. Am.*, **68**:780–791, September 1980.

[Fla72] J. Flanagan. *Analysis, synthesis, and perception of speech.* Springer-Verlag, Berlin, 2nd edition, 1972.

[Gal96] M. J. F. Gales. "Mean and variance adaptation within the MLLR framework." *Computer Speech and Language*, **10**:249–264, 1996.

[Gal98] M. J. F. Gales. "Maximum likelihood linear transformations for HMM-based speech recognition." *Computer Speech and Language*, **12**(2):75–98, Apr 1998.

[Gal99] M. J. F. Gales. "Semi-tied covariance matrices for hidden Markov models." *IEEE Transactions Speech and Audio Processing*, **7**:272–281, 1999.

[GC89] L. Gillick and S. Cox. "Some statistical issues in the comparison of speech recognition algorithms." In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, p. 532535, 1989.

[Goc05] M. S. Gockenbach. "Online lectures on Numerical Optimization." Department of Mathematical Sciences, Michigan Technological University, Spring 2005. http://www.math.mtu.edu/ msgocken/ma5630spring2005/lectures.html.

[GS97] E. B. Gouvea and R. M. Stern. "Speaker normalization through formant-based warping of the frequency scale." In *Proc. Eurospeech*, volume 3, pp. 1139–1142, 1997.

[Her90] H. Hermansky. "Perceptual Linear Prediction (PLP) Analysis of Speech." *Journal of the Acoustical Society of America*, **87**(4):1738–1752, 1990.

[HH04] S. Hiroya and M. Honda. "Estimation of articulatory movements from speech acoustics using an HMM-based speech production model." *Speech and Audio Processing, IEEE Transactions on*, **12**(2):175–185, 2004.

[HS64] J. M. Heinz and K. N. Stevens. "On the derivation of area functions and acoustic spectra from cineradiographic films of speech." *The Journal of the Acoustical Society of America*, **36**:1037, 1964.

[JRW87] B.H. Juang, L. Rabiner, and J. Wilpon. "On the use of bandpass liftering in speech recognition." *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, **35**(7):947–954, 1987.

[KAC95] T. Kamm, G. Andreou, and J. Cohen. "Vocal tract normalization in speech recognition: Compensating for systematic speaker variability." In *Proceedings of the 15th Annual Speech Research Symposium, Johns Hopkins University, Baltimore, MD*, p. 161167, 1995.

[LNU06] J. Loof, H. Ney, and S. Umesh. "VTLN Warping Factor Estimation Using Accumulation of Sufficient Statistics." In *Proc. ICASSP*, volume 1, pp. 1–4, 2006.

[LR98] L. Lee and R. C. Rose. "A frequency warping approach to speaker normalization." *IEEE Trans. Speech and Audio Processing*, **6**(1):49–60, 1998.

[LW95] C. J. Leggetter and P.C. Woodland. "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models." *Computer Speech and Language*, **9**:171–185, 1995.

[Mae90] S. Maeda. "Compensatory articulation during speech: evidence from the analysis and synthesis of vocal tract shapes using an articulatory model." In W. J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modeling*, pp. 131–149. Kluwer, 1990.

[MB99] J. McDonough and W. Byrne. "Speaker adaptation with all-pass transforms." In *Proc. ICASSP*, volume 2, pp. 757–760, 1999.

[MBL98] J. McDonough, W. Byrne, and X. Luo. "Speaker normalization with all-pass transforms." In *Proc. ICSLP*, volume 6, pp. 2307–2310, 1998.

[McD00] J. W. McDonough. *Speaker compensation with all-pass transforms.* Ph.d. dissertation, Johns Hopkins University, Baltimore, Maryland, 2000.

[McG94] R. S. McGowan. "Recovering articulatory movement from formant frequency trajectories using task dynamics and a genetic algorithm: Preliminary model tests." *Speech Communication*, **14**:19–48, 1994.

[Mer67] P. Mermelstein. "Determination of vocal tract shapes from measured formant frequencies." *J. Acoust. Soc. Am.*, **41**(5):1283–1294, 1967.

[Mer73] P. Mermelstein. "Articulatory model for the study of speech production." *J. Acoust. Soc. Am.*, **53**(4):1070–1082, 1973.

[ML97] B. Mathieu and Y. Laprie. "Adaptation of Maeda's model for acoustic to articulatory inversion." In *Proceedings of Eurospeech*, pp. 2015–2018, 1997.

[Moc] http://www.cstr.ed.ac.uk/research/projects/artic/mocha.html.

[OL05] S. Ouni and Y. Laprie. "Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion." *J. Acoust. Soc. Am.*, **118**(1):444–460, July 2005.

[PA06] S. Panchapagesan and A. Alwan. "Multi-parameter Frequency warping for VTLN by gradient search." In *ICASSP*, volume I, p. 1181, 2006.

[PA09] S. Panchapagesan and A. Alwan. "Frequency Warping for VTLN and Speaker Adaptation by Linear Transformation of Standard MFCC." *Computer Speech and Language*, **23**:42–64, 2009. (to appear).

[Pan06] S. Panchapagesan. "Frequency Warping by Linear Transformation of Standard MFCC." In *Proceedings of Interspeech*, pp. 397–400, 2006.

[PFB88] P. Price, W. M. Fisher, J. Bernstein, and D.S. Pallett. "The DARPA 1000-word resource management database for continuous speech recognition." In *Proceedings of ICASSP*, pp. 651–654, April 1988.

[PHT92] G. Papcun, J. Hochberg, T. Thomas, F. Laroche, J. Zacks, and S. Levy. "Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data." *J. Acoust. Soc Am*, **92**(2, Pt.1):688–700, 1992.

[PMS01] M. Pitz, S. Molau, R. Schlueter, and H. Ney. "Vocal Tract normalization equals linear transformation in cepstral space." In *Eurospeech*, pp. 721–724, 2001.

[PN03] M. Pitz and H. Ney. "Vocal Tract normalization as linear transformation of MFCC." In *Proc. Eurospeech*, pp. 1445–1448, 2003.

[RGK93] M. G. Rahim, C. C. Goodyear, W. B. Kleijn, J. Schroeter, and M. M. Sondhi. "On the use of neural networks in articulatory speech synthesis." *J. Acoust. Soc. Am.*, **93**(2):1101–1121, February 1993.

[Ric01] K. Richmond. *Estimating Articulatory Parameters from the Acoustic Speech Signal*. Ph.d. thesis, U. Edinburgh, 2001.

[Rie97] E. L. Riegelsberger. *The Acoustic-to-Articulatory Mapping of Voiced and Fricated Speech*. Ph. d. dissertation, The Ohio State University, 1997.

[RJ93] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood, New Jersey, 1993.

[SA97] B. Strope and A. Alwan. "A model of dynamic auditory perception and its application to robust word recognition." *IEEE Transactions on Speech and Audio Processing*, **5**(5):451–464, 1997.

[Sch67] M. R. Schroeder. "Determination of the geometry of the human vocal tract by acoustic measurements." *J. Acoust. Soc. Am.*, **41**(4, Part 2):1002–1010, 1967.

[SK86] K. Shirai and T. Kobayashi. "Estimating articulatory motion from speech wave." *Speech Communication*, **5**:159–170, 1986.

[SL96] A. Sankar and C. Lee. "A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition.", 1996.

[SM89] E. L. Saltzman and K. G. Munhall. "A dynamical approach to gestural patterning in speech production." *Ecological Psychology*, **1**:333–382, 1989.

[SMP90] J. Schroeter, P. Meyer, and S. Parthasarathy. "Evaluation of improved articulatory codebooks and codebook access distance measures." In *Proc. IEEE ICASSP*, pp. 393–396, 1990.

[Son74] M. M. Sondhi. "Model for wave propagation in a lossy vocal tract." *J. Acoust. Soc. Am.*, **55**(5), May 1974.

[Sor92] V. N. Sorokin. "Determination of vocal tract shape for vowels." *Speech Communication*, **11**:71–85, 1992.

[SS87] M. M. Sondhi and J. Schroeter. "A Hybrid Time-Frequency Domain Articulatory Speech Synthesizer." *IEEE Trans. ASSP*, **35**(7):955–967, July 1987.

[SS92] J. Schroeter and M. M. Sondhi. "Speech coding based on physiological models of speech production." In S. Furui and M. M. Sondhi, editors, *Advances in Speech Signal Processing*, pp. 231–267. Marcel Dekker, New York, 1992.

[SS94] J. Schroeter and M. M. Sondhi. "Techniques for estimating vocal tract shapes from the speech signal." *IEEE Trans. SAP*, **2**(1):133–150, Jan 1994.

[ST96] V. N. Sorokin and A. V. Trushkin. "Articulatory-to-acoustic mapping for inverse problem." *Speech Communication*, **19**:105–118, 1996.

[STH96] B.H. Story, I.R. Titze, and E.A. Hoffman. "Vocal tract area functions from magnetic resonance imaging." *The Journal of the Acoustical Society of America*, **100**:537–554, 1996.

[SVN37] S. S. Stevens, J. Volkman, and E. Newman. "A scale for the measurement of the psychological magnitude of pitch." *Journal of the Acoustical Society of America*, **8**(3):185–190, 1937.

[UZN05] S. Umesh, A. Zolnay, and H. Ney. "Implementing frequency-warping and VTLN through linear transformation of conventional MFCC." In *Prof. INTERSPEECH*, pp. 269–272, 2005.

[WCA07] S. Wang, X. Cui, and A. Alwan. "Speaker adaptation with limited data using regression tree based spectral peak alignment." *IEEE Transactions on Speech, Audio and Language Processing*, **15**(8):2454–2464, 2007.

[Wes94] J. R. Westbury. *X-ray Microbeam Speech Production Database User's Handbook, Version 1.0*, June 1994. http://www.medsch.wisc.edu/ milenkvc/pdf/ubdbman.pdf.

[WMO96] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin. "Speaker normalization on conversational telephone speech." In *Proc ICASSP*, pp. 339–341, 1996.

[WNK02] L. Welling, H. Ney, and S. Kanthak. "Speaker Adaptive Modeling by Vocal Tract Normalization." *IEEE Trans. Speech and Audio Processing*, **10**(6):415–426, 2002.

[YEK] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book version 3.2.*

[ZW97] P. Zhan and A. Waibel. "Vocal tract length normalization for large vocabulary continuous speech recognition." Technical report, Carnegie Mellon University, May 1997. CMU-CS-97-148.