A BANDPASS TRANSFORM FOR SPEAKER NORMALIZATION

by

Pierre L. Dognin

B.S. in E.E., École Supérieure de Chimie, Physique et Électronique de Lyon, 1997

M.S. in E.E., University of Pittsburgh, 1999

Submitted to the Graduate Faculty of
the School of Engineering in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2003

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This dissertation was presented

by

Pierre L. Dognin

It was defended on

July 23, 2003

and approved by

Amro A. El-Jaroudi, Associate Professor, Electrical Engineering

J. Robert Boston, Associate Chairman, Electrical Engineering

Luis F. Chaparro, Associate Professor, Electrical Engineering

Ching-Chung Li, Associate Professor, Electrical Engineering

Mihai Anitescu, Assistant Professor, Mathematics

Dissertation Advisor: Amro A. El-Jaroudi, Associate Professor,
Electrical Engineering

A BANDPASS TRANSFORM FOR SPEAKER NORMALIZATION

Pierre L. Dognin, Ph. D.

University of Pittsburgh, 2003

One of the major challenges for Automatic Speech Recognition is to handle speech variability. Inter-speaker variability is partly due to differences in speakers' anatomy and especially in their Vocal Tract geometry. Dissimilarities in Vocal Tract Length (VTL) are a known source of speech variation. Vocal Tract Length Normalization is a popular Speaker Normalization technique that can be implemented as a transformation of a spectrum frequency axis. We introduce in this document a new spectral transformation for Speaker Normalization. We use the Bilinear Transformation to introduce a new frequency warping resulting from a mapping of a prototype Band-Pass (BP) filter into a general BP filter. This new transformation called the Bandpass Transformation (BPT) offers two degrees of freedom enabling complex warpings of the frequency axis that are different from previous works with the Bilinear Transform. We then define a procedure to use BPT for Speaker Normalization based on the Nelder-Mead algorithm for the estimation of the BPT parameters. We present a detailed study of the performance of our new approach on two test sets with gender dependent and independent systems. Our results demonstrate clear improvements compared to standard methods used in VTL Normalization. A score compensation procedure is presented and results in further improvements of our results by refining our BPT parameter estimation.

Keywords: Automatic Speech Recognition, Analytical Function, Bilinear Transformation, Feature Transformation, Frequency Warping, Front End Processing, Model Adaptation, Nelder-Mead Optimization, Non-Linear Transformation, Speaker Normalization, Speech Processing, Vocal Tract Length Normalization.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to express sincere thanks to my advisor, Dr. Amro A. El-Jaroudi, for his guidance during the preparation of this dissertation. Thanks also goes to Drs. J. Robert Boston, Luis F. Chaparro, Ching-Chung Li and Mihai Anitescu for taking their time to serve on my Ph.D. thesis committee and for all their suggestions.

I would also like to thank the Speech and Language Processing Department at BBN Technologies for providing funding to support the research that is presented in this dissertation. Thanks goes to all the members of the LVCSR group and the EARS project at BBN Technologies. I would like to thank John Makhoul, Herb Gish and Owen Kimball for welcoming me into the LVCSR group as a Visiting Researcher starting May 2000. I would like to thank all the members of the LVCSR group and of the more recent EARS project. Among them, past and present members, I would like to thank Jayadev Billa, Bill Belfield, Thomas Colthurst, Rukmini Iyer, Jeff Ma, Carl Quillen, Jim Van Sciver and Dongxin Xu. From the "Rough 'n Ready" project, I would like to thank Francis Kubala, Amit Srivastava, Daben Liu, Daniel Kiecza and Mohamed Noamany. I would like to thank all the students working on their research at BBN. Thanks also goes to Josh Bers and Marie-Hélène Talon. One person at BBN had a significant impact on the thoughts behind this dissertation work. Spyros Matsoukas took upon himself to help me in my research on top of his own work at BBN. I would like to particularly thank him for his guidance, mentorship and friendship over the past three years.

Special thanks goes to Keith Davidson, J. Angela Beauford and the many other friends I made during my years at the University of Pittsburgh. Thanks to Laurent Arnaud, Niels Jacobsen and his family, Vincent and Karine Ruet , Sharmila Manglani, Guillaume Fréchette and family, Stéphane Peysson and all my friends scattered all over the world whose support

have been valuable over the past years. I would like to thank my family: Jeanne, Danielle, Paul, Karine, Philippe, Jean-Claude, Christine, Céline, Émilie and the two latest additions Oscar and Carla for their support and their understanding over the past 7 years I have been living in the USA. Thanks goes also to Michèle and Paul Bacot.

Finally, these acknowledgments would not be complete without the person who changed my life for the better over the past two years. I would like to express my gratitude to Joanna and her unconditional support.

1.0 INTRODUCTION

In the past decade *Automatic Speech Recognition* (ASR) has offered a new basis for human-machine interaction that was once described only in science fiction literature. In his book *"2001: A Space Odyssey"*, Arthur C. Clarke offers the description of a future where machines and humans talk to each other, a technical achievement that is now commonly accepted in most futuristic visions of what is to become of our world[1]. Even if we are far from this engineering accomplishment, the latest improvements in ASR have made possible new applications where information is gathered directly from its speech form. For instance, this dissertation could have been dictated to a personal computer using one of the software now readily available to the public. On a different scale, applications such as real-time close captioning for TV programs and the creation of information databases gathered from radio and TV broadcast have become available to the corporate world. These accomplishments signal that the transcription task of speech information has reached an adequate level of accuracy, enabling further work on the content of the collected data.

1.1 SPEECH VARIABILITY

Among all the difficulties that ASR has encountered, a major challenge has been to handle the multitude of ways people speak. The diversity of the causes of speech variation is quite impressive, for instance it is easily noticeable that people do not talk to close friends the same way they talk to coworkers. This variability of speech can become a significant source of performance degradation for an ASR system. Consequently many researchers have taken

[1]The Author's view was indeed quite optimistic in term of speech recognition advances. The book was written in 1968 and aimed at describing a not-so-distant future: the year 2001, now our not-so-distant past.

interest in the task of making speech recognition more robust to speech variability. Speech variability is generated by diverse factors whose nature can be purely physical, cultural or sociological. Therefore variations in speech not only appear for a single speaker but also within a group of speakers. These two types of variability are usually referred to as *Intra-Speaker* and *Inter-Speaker Variability*.

Intra-Speaker Variability stems from the natural randomness in the pronunciation of *phonemes*, the smallest constituents of speech. A person will rarely produce the same phoneme in an identical manner twice. Furthermore, speech is constituted of a series of phonemes, each one of them being pronounced differently depending on the neighboring phonemes. This is known as the *coarticulation effect*. Further, the intra-speaker variability combines all variations of speech, including the effects of mood, stress or even health.

Inter-Speaker Variability accounts for the fact that speech is different among speakers. For instance two individuals will not produce the same speech even if they are asked to say the same sentences. Differences in size, age, gender, speaking rate and accentuation are sources of variations on the phoneme level between individuals. Regional and local accents are also sources of variability among the same language. Residents of New Jersey and Georgia share the same language and the same set of phonemes (53 for American-English) but their speech will be quite different. Finally, the social context in which speech is produced creates more sources of speech variations. In his work on politics, Aristotle argued that humans are by nature "social animals" and that any understanding of human behavior must include social considerations. It is then not surprising that some of the intra-speaker differences come inherently from our social structure. A formal conversation between an employee and his/her boss and a relaxed chat between friends offer examples of the differences in "quality" between formal and informal speech; this creates distinction which an ubiquitous ASR system needs to handle[2].

[2] For phone conversation, the level of formality between two speakers will directly condition the amount of cross-talk, variations of intonations that will be observed.

1.2 MODELS AND FEATURES FOR SPEECH RECOGNITION

ASR is often interpreted as an extensive pattern recognition task that tries to recognize sequences of words from speech waveforms. To accomplish this task, two fundamental and separate steps are required. First, *Acoustic Models* need to be created from transcribed speech. Statistical properties of observed speech will be modeled in this step, commonly referred to as *Acoustic Training*, or simply *Training*. The second step is called *Recognition* and its goal is to find the most likely sequence of words that were uttered given the observed speech and the previously trained Acoustic Models. The Acoustic Models are chosen to be *Hidden Markov Models* (HMMs) because they offer a powerful tool for building parameterized models for speech recognition[1].

In theory, ASR could work directly on speech signals. However, the large variability of speech makes it preferable to work on features extracted from speech waveforms. The difficulty remains in finding the best features, those which contain the most information on the produced phonemes as well as those that are sufficiently uncorrelated and robust to amplitude scaling and time shift, etc. *Cepstral features* based on homomorphic filtering of speech have been the most successful at representing speech in ASR.

HMMs and cepstral features are at the heart of current ASR systems. A Hidden Markov Model is an extension of the Markov Chains used to model data with minimum memory. A Markov chain is a finite state process with transition probabilities from one state to another. The probability of being in state s_t at time t depends only on the previous state s_{t-1} at time $t-1$. Each one of the states is associated with an observation. Unlike Markov Chain Models, HMMs allow the observation for each state to follow a *probability density function* or pdf. This pdf associated with the observation of a state s is often chosen to be modeled as a combination of Gaussians known as *Gaussian Mixture Model* (GMM). Therefore, the observation probability of an observation vector $\boldsymbol{x}_t$ for a state s at time t for a GMM of K Gaussians is given by

$$P^s(\boldsymbol{x}_t; \boldsymbol{\Gamma}^s) = \sum_{k=0}^{K-1} w_k^s \mathcal{N}^s(\boldsymbol{x}_t; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s) \tag{1.1}$$

where $\mathcal{N}^s(\boldsymbol{x}; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$ is the k^{th} multivariate Gaussian distribution consisting of mean vector

$\boldsymbol{\mu}_k^s$ and covariance matrix $\boldsymbol{\Sigma}_k^s$. w_k^s is a normalized weight associated to the k^{th} Gaussian for state s and $\boldsymbol{\Gamma}^s$ is the set of all the Model Parameters (mean vectors $\boldsymbol{\mu}_k^s$ and covariance matrices $\boldsymbol{\Sigma}_k^s$) for the GMM at state s. The observation probability for each Gaussian for the state s is defined by

$$\mathcal{N}^s(\boldsymbol{x}; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s) = \frac{1}{(2\pi)^{\frac{n}{2}} \, |\boldsymbol{\Sigma}_k^s|^{\frac{1}{2}}} \quad \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k^s)^T \boldsymbol{\Sigma}_k^{s\,-1} (\boldsymbol{x}-\boldsymbol{\mu}_k^s)}. \tag{1.2}$$

For each utterance to be transcribed, we have an observed acoustic information sequence $\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ and a sequence of possible words $\boldsymbol{W} = \{\boldsymbol{w}_0, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$. Recognition is trying to find the most likely word sequence $\hat{\boldsymbol{W}}$ that satisfies

$$P(\hat{\boldsymbol{W}}|\boldsymbol{X}) = \max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{X}) \tag{1.3}$$

In simpler terms, given an acoustic observation sequence, what is the word sequence that has the *Maximum Likelihood* (ML) to have been uttered? The Bayes' rule can let us express the term $P(\boldsymbol{W}|\boldsymbol{X})$ as

$$P(\boldsymbol{W}|\boldsymbol{X}) = \frac{P(\boldsymbol{W})P(\boldsymbol{X}|\boldsymbol{W})}{P(\boldsymbol{X})}. \tag{1.4}$$

One can notice that the term $P(\boldsymbol{X})$ is independent of $\boldsymbol{W}$. Therefore, it is not relevant to the Recognition task, and so Equation (1.3) can then be rewritten as

$$P(\hat{\boldsymbol{W}}|\boldsymbol{X}) = \max_{\boldsymbol{W}} P(\boldsymbol{W})P(\boldsymbol{X}|\boldsymbol{W}). \tag{1.5}$$

In Equation (1.5), the *prior* $P(\boldsymbol{W})$ assigns probabilities to a word sequence $\boldsymbol{W}$ which clearly demonstrates the need for a *Language Model*[2]. $P(\boldsymbol{X}|\boldsymbol{W})$ comes from an *Acoustic Model* that evaluates the probability of the acoustic information $\boldsymbol{X}$ given a particular word sequence $\boldsymbol{W}$.

1.3 SPEAKER NORMALIZATION

When ASR systems need to handle speech variability, two different approaches are commonly used: model the variability or eliminate it. Modeling variability means incorporating the different forms of speech into the model. The Acoustic Models' parameters are changed to better fit the statistics of observed speech. Eliminating speech variability is based on transforming the speech features to compensate for differences, making them closer to Acoustic Models.

In order to reduce speech variability, we need to recognize which variability accounts for the most of performance drop in ASR. Among all the variability sources that have been stated earlier, the inter-speaker variability has a more significant impact on ASR than the intra-speaker one. A *Speaker Dependent* (SD) system always performs better than its *Speaker Independent* (SI) counterpart. The two techniques mainly used to handle inter-speaker variability are *Model Adaptation* (also referred to as *Speaker Adaptation* or SA) which works on the model parameters, and *Speaker Normalization* (SN), which will transform the speech features. Both techniques are part of a broader task known as *Speaker Compensation*. Speaker Adaptation works on the model parameters level. Models parameters (mean vectors and covariance matrices for HMMs) are modified in a constrained or unconstrained way[3] in order to make the models better fit the observed features. Applications of this technique includes adaptation of models to new unseen speech. Speaker Normalization works on the feature level. Before recognition, SN is used to transform speech features, adjusting them to fit better to Acoustic Models, and eliminating some inter-speaker differences in the process. Since the features are more "discriminative", recognition is expected to be more accurate. Our work involves taking this SN approach and applying it to Speaker Compensation.

The inter-speaker differences in speech are partly due to differences in speakers anatomy especially in the *Vocal Tract* (VT) geometry. More precisely, one parameter of its geometry known as the *Vocal Tract Length* (VTL) creates variations in the resonance frequencies of identical phonemes. To understand this phenomenon, researchers often chose to model the Vocal Tract with a lossless tube. An acoustic excitation at one end of the tube will reveal the resonance frequencies of this acoustic system at the other end of the tube. Spectral analysis

will clearly reveal the resonance frequencies' location on the frequency axis. The location of these resonance frequencies depends directly on the length of the tube. Variations of the tube's length will shift the resonance frequencies along the frequency axis. For instance, the longer the tube gets, the more resonance frequencies are shifted to the lower end of the spectrum. Decrease the length and the frequencies will be shifted to "higher" frequencies. This simple phenomenon is well known in music as it is often utilized to tune wind instruments. For example, clarinets are finely tuned by changing the length of their acoustic tube. It is accomplished by pulling the components of the clarinet closer or further together[3].

This phenomenon is also observable among speakers. Differences in VTL can partially explain why some people have deeper voices than others. In a chorus, male singers are more likely to be tenors while female singers will be more numerous in the alto section. Knowledge of the gender is important here as adult male speakers have longer VTL (16.9 cm in average) than adult female speakers (14.1 cm)[4]. The differences in VTL create a clear division between male and female speakers[4]. Trying to compensate for this difference is know as *Vocal Tract Length Normalization* or VTLN for short.

1.4 VOCAL TRACT LENGTH NORMALIZATION

Vocal Tract Length Normalization (VTLN) is a popular speaker normalization technique among ASR systems[5, 6]. The effect of VTL variations on the Short-Time Fourier Transform of speech waveform can be interpreted as a distortion or *warping* of the spectrum frequency axis. It seems therefore natural to imagine a compensation method based entirely on an "opposite" frequency transformation. VTLN aims to compensate for the variability in formants location due to VTL differences by working directly on the Fourier representation of speech. VTLN is often implemented as a direct transformation of the Fourier Spectrum of speech and is referred to as *Frequency Warping* or often *Spectral Warping*. This family of transformation focuses on warping the frequency axis of the Fourier spectrum in order

[3]For clarinets, it is performed between the part below the mouthpiece and the rest of the body of the instrument. By changing the distance between the mouthpiece and the body of the instrument, one can perform a really precise tuning necessary to assure that the clarinet is in tune with an accompanying piano.

[4]This difference between male and female speakers is often utilized in gender detection applications.

to compensate for the natural warping that occurs from variations in VTL across speakers. Such transformation is speaker dependent (each speaker has different VT properties) and is often chosen to be parametric with only a few parameters. A set of parameters needs to be estimated for each speaker so that the spectral characteristics of his/her transformed speech is closer to the ones of a "canonical" speaker. The goals of a VTLN procedure is to estimate the set of parameters and transform the speech of each speaker accordingly. VTLN can be implemented by working directly on the spectrum of speech or by transforming the cepstral features in such a way that some frequency warping of the speech spectrum occurs.

A simple linear warping of the frequency axis could be used in VTLN if our vocal tract was a lossless tube of varying length. Unfortunately, the physical shape of our VT is a lot more complex and formants in speech are not linearly warped across voiced phonemes when the VTL varies. To address this fact, researchers started to look into nonlinear parametric spectrum transformations for VTLN. One example of these nonlinear transformations is the one proposed by Eide[7] . This transformation was defined to accommodate most of the voiced phonemes. It is controlled by only one parameter called *warping factor* whose choice is based on formant estimation. By changing the warping factor, the transformation modifies the formants' location.

The frequency warping properties of the *Bilinear Transformation* (BLT) did not go unnoticed and BLT has focused some research effort to its application to VTLN as well. BLT was the transformation that Acero considered in his thesis[8] specifically to eliminate inter-speaker variations. In this case, the transformation is also controlled by only one parameter that need to be estimated for each speaker, the estimation procedure being based on trying to minimize a Vector Quantization (VQ) distortion measure. John McDonough has used the *All-Pass Transform* or APT in his work[9, 10, 11] in order to transform cepstral sequences[12] to perform VTLN.

1.5 BILINEAR TRANSFORMATION

The Bilinear Transformation has blossomed in the signal processing field due to its use in analog-to-digital filter design, digital filter transformation and transformation of discrete time

series[13]. In the area of Speaker Normalization, the main interest in BLT is its ability to offer an efficient technique to perform frequency warping. Indeed, BLT offers a mean to transform a discrete-time sequence into another sequence. The transformed sequence has the same Fourier Transform as the original sequence but with a warped frequency axis. If the sequence to transform is chosen to be the cepstral speech features, then VTLN can be performed using BLT[8, 10]. The Bilinear Transform offers an elegant mathematical framework to Speaker Normalization and is the keystone of the work presented in this document.

1.6 THE BYBLOS SYSTEM

The experimental environment for our work was provided by the BYBLOS system. BYBLOS is a state of the art Large Vocabulary Conversational Speech Recognition (LVCSR) system developed by BBN Technologies (BBN), a world leading company in Speech Recognition technology.

1.7 DISSERTATION ORGANIZATION

Chapter 2 presents in detail the *Front End* of the BYBLOS system and the different signal processing steps required to obtain cepstral speech features. This chapter covers extensively the framework in which VTLN is performed in BYBLOS . Chapter 3 introduces the new frequency warping we propose to use for Speaker Normalization and more precisely for VTLN. This chapter describes the different steps that lead to the definition of our frequency warping based on complex analysis theory and digital design. This frequency warping is called the *Bandpass Transform* or BPT. A complete system description is found in Chapter 4. First, a performance measure needed to compare our experimental results is defined. Second, the BYBLOS system is presented with a focus on the motivations for the choice of the Training and Decoding sets. Finally, the settings used in the extraction of speech features, training of the acoustic and language models are discussed. In Chapter 5, the parameter estimation procedure used to find the BPT parameters is described in details. The definition of an

objective function is followed by the presentation of the Nelder-Mead algorithm that is used to maximize this objective function. A compensation score method is then introduced. This novel compensation method relies on approximating the transformation that the BPT performs on the speech features by a linear transformation. From the expression of this linear transformation, a compensation factor is found that can be used to make our feature transformation similar to a constrained adaptation of the Acoustic Models. All the steps of the score compensation method are described and finally an overview of the resulting Speaker Normalization procedure is presented. Experimental results are the main focus of Chapter 6. First, improvements from the BPT are discussed, then followed by results from the use of our score compensation method. Finally, Chapter 7 summarizes the contributions of this work and presents possible future research directions.

2.0 ANALYSIS OF SPEECH

2.1 MODELS AND FEATURES

One early problem that ASR had to face was to chose the "right" speech features and the "right" acoustic models to work with. Acoustics models were chosen to be *Hidden Markov Models* (HMMs) as they are assumed to offer a good fit to the statistical nature of speech. One important assumption on the speech feature HMMs is the fact that they are uncorrelated. The acoustics models are often assumed to have diagonal covariance matrices. There was a need to have speech features that are by nature uncorrelated or more precisely decorrelated enough so that the assumption of diagonal covariance matrices can still hold.

2.2 THE SOURCE-FILTER MODEL FOR SPEECH

A common approach to speech production model is to consider that speech is the result of the *excitation* of an acoustic system composed of the VT. At one end of the system, the glottis creates a sequence of quasi-periodic pulses of airflow that offers a wide-band excitation of the VT system. Speech is the result of this acoustic excitation that goes through the time-varying linear filter constituted by the VT. This model is known as the *Source-Filter model*[14] and is a simple but reasonable interpretation of speech production particularly for *voiced phonemes* like vowels. As a consequence of this model, any speech signal can be written as a function of continuous time like

$$s(t) = e(t) * v(t) \quad \forall\, t \in \mathbb{R} \tag{2.1}$$

where $e(t)$ is the excitation and $v(t)$ a time-varying filter whose properties are determined entirely by the VT shape changes during speech production. The resulting speech signal $s(t)$ in Equation (2.1) is the only one we have access to. ASR works on discrete-time version of $s(t)$ and Equation (2.1) can be rewritten as

$$s(n) = e(n) * v(n) \quad \forall\, n \in \mathbb{N}. \tag{2.2}$$

Both Equation (2.1) and Equation (2.2) are extreme simplifications of the speech production system but they serve the purpose to show that excitation $e(n)$ and $v(n)$ can be considered as independent from each other. $v(n)$ depends directly on the VT shape and contains important information on which phoneme is being uttered. ASR systems try to define a set of features that represents the best $v(n)$ and its variations in time. ASR systems work directly on those speech features in order to find a phoneme sequence from speech. The information contained in $e(n)$ is linked to the "tone" of the produced speech. If the period of $e(n)$ is decreased, the pitch of speech is increased and the produced speech will sound "higher" in frequency. However for a same pitch, small changes in $v(n)$ can make the phoneme /i/ turn into /a/ or /o/ quickly by simply shaping the spectrum of the produced sounds. The main frequency for $e(n)$ is referred as F_0. The VT is characterized by the resonances of its frequency response called *formants*. They follow the notation F_n for $n = 1, 2, ..., \infty$ among which F_1, F_2, F_3, F_4 are the ones of higher amplitudes. The VT shapes the frequency spectrum of the excitation imposing its own formants locations on the spectrum of $s(n)$.

During speech production, the VT shape does not change rapidly. $v(n)$ is therefore a "slow" time-varying filter. It is important to know the frequency composition, or spectrum, of a speech signal as it is changing in time. A time-dependent frequency analysis is therefore a good approach to get features for speech. One of them, the Short-Time Fourier Analysis is of great use in ASR.

2.3 SHORT-TIME FOURIER ANALYSIS

2.3.1 Fourier Analysis

ASR systems use *Short-Time Fourier Analysis*[15] to analyze speech periodically in time. It is motivated by the need to study the local frequency properties of speech at every short interval of time $t, t+1, \ldots, t+N$. In order to do so, it is required to emphasize the signal at time t and suppress the rest of the signal. This is achieved by windowing the speech signal before performing a *Fourier Analysis* on the windowed signal. The Fourier Analysis is performed using the well-known *Fourier Transform* (FT), denoted by $\mathcal{F}$ in this document. The Fourier Transform of a continuous time signal $x(t)$ defines a relationship between the time domain signal and its representation in the frequency domain:

$$x(t) \overset{\mathcal{F}}{\rightleftharpoons} X(\mathrm{e}^{\jmath\omega}) \quad t \in \mathbb{R},\, \omega \in \mathbb{R} \tag{2.3}$$

where ω is an angular frequency[1] (in rad.s^{-1}). $X(\mathrm{e}^{\jmath\omega})$ is the representation of $x(t)$ in the Fourier domain. The Fourier Transform is given by

$$X(\mathrm{e}^{\jmath\omega}) = \mathcal{F}\left(x(t)\right) = \int_{t=-\infty}^{\infty} x(t)\, \mathrm{e}^{\jmath\omega t} \mathrm{d}t. \tag{2.4}$$

The relation in Equation (2.3) implies that there exists an inverse transform to go from the frequency domain to the time domain. This *Inverse Fourier Transform* (IFT), denoted by $\mathcal{F}^{-1}$, is defined by

$$x(t) = \mathcal{F}^{-1}\left(X(\mathrm{e}^{\jmath\omega})\right) = \frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} X(\mathrm{e}^{\jmath\omega})\, \mathrm{e}^{\jmath\omega t} \mathrm{d}\omega \tag{2.5}$$

Since ASR is truly working with discrete-time signals, it is more accurate to consider the *Discrete-Time Fourier Transform* (DTFT) defined by

$$x(n) \overset{\mathcal{F}}{\rightleftharpoons} X(\mathrm{e}^{\jmath\omega}) \qquad n \in \mathbb{N}, w \in \mathbb{R} \tag{2.6}$$

where the DTFT of $x(n)$ is

[1] This document will use in the same manner the angular frequency as well as the oscillation frequency notation f defined by $\omega \equiv 2\pi f$ and measured in s^{-1} or Hz.

$$X(\mathrm{e}^{\jmath\omega}) = \sum_{n=-\infty}^{\infty} x(n)\,\mathrm{e}^{-\jmath\omega n} \tag{2.7}$$

and the inverse DTFT of $X(\mathrm{e}^{\jmath\omega})$ is

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\mathrm{e}^{\jmath\omega})\,\mathrm{e}^{\jmath\omega n}\mathrm{d}\omega. \tag{2.8}$$

Because speech signals are limited in duration, it is more precisely the *Discrete Fourier Transform* (DFT) that is utilized in ASR systems. The analysis equation for the DFT is

$$\mathring{X}(k) = \sum_{n=0}^{N-1} \mathring{x}(n)\,\mathrm{e}^{-\jmath\frac{2\pi}{N}nk} \quad n\in\mathbb{N}, k\in\mathbb{N}. \tag{2.9}$$

Recovering the time-sequence $\mathring{x}(n)$ from $\mathring{X}(k)$ is done through the synthesis equation

$$\mathring{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \mathring{X}(k)\,\mathrm{e}^{\jmath\frac{2\pi}{N}nk} \quad n\in\mathbb{N}, k\in\mathbb{N} \tag{2.10}$$

where $\mathring{x}(n)$ is a periodic sequence with period N. $\mathring{x}(n)=x(n+rN)$ for any integer value r. The periodicity of $\mathring{x}(n)$ comes as a direct effect of the discretization of its Fourier domain representation $\mathring{X}(k)$. For all k, $\mathring{X}(k)$ is a complex number that can be written as a function of its modulus and argument:

$$\mathring{X}(k) = \left|\mathring{X}(k)\right| \mathrm{e}^{\jmath\angle\mathring{X}(k)} \quad \forall\, k \tag{2.11}$$

In ASR, it is generally assumed that the information coming from the phase $\angle\mathring{X}(k)$ can be discarded[2] and ASR systems work on the *Power Spectrum* rather than the complex (Fourier) spectrum. The power spectrum provides information about the energy distribution of a signal in the frequency domain. It is defined as $\left|\mathring{X}(k)\right|^2$ which can be easily computed from the real and imaginary part of $\mathring{X}(k)$:

$$\left|\mathring{X}(k)\right|^2 = \Re\left\{\mathring{X}(k)\right\}^2 + \Im\left\{\mathring{X}(k)\right\}^2 \quad \forall k\in[0,1,\ldots,N-1] \tag{2.12}$$

Analyzing a speech signal and computing its windowed power spectrum at equally spaced times t is known in Time-Frequency Analysis as the *spectrogram*[15] of a signal. To get the spectrogram of a signal, the signal is first decomposed into frames, each of them windowed before computing its power spectrum.

[2]Speech features allowing phase information tend to be dependent on time shift, which is generally not desired. Hence the assumption that the phase can be discarded.

2.3.2 Frame Blocking

In order to implement a Short-Time Fourier Analysis, the speech signal $s(n)$ from Equation (2.2) is first blocked into *frames* of M samples before computing the DFT for each frame. Adjacent frames are lagged by Q samples. In ASR, it is chosen to have $Q < M$ so that adjacent frames overlap to ensure the capture of VT characteristics changes with great detail in the time domain. A sequence of P frames $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{P-1}\}$ is extracted from the speech signal $s(n)$ such that each vector $\boldsymbol{x}_p$ in the frame sequence is defined by

$$\boldsymbol{x}_p = \left\{ \begin{array}{c} x_p(0) \\ x_p(1) \\ \vdots \\ x_p(M-1) \end{array} \right\} \tag{2.13}$$

where $x_p(m)$ is linked to $s(n)$ by

$$x_p(m) = s(Qp+m) \quad \text{where} \quad \left\{ \begin{array}{l} m = 0, 1, \ldots, M-1, \\ p = 0, 1, \ldots, P-1, \\ Q < M. \end{array} \right. \tag{2.14}$$

The notation $x_p(m)$ will refer to the time-sequence representation of a frame whereas $\boldsymbol{x}_p$ will be used for its vector representation. Both representations are equivalent and linked by Equation (2.13). They describe the same set of samples from our original speech signal $s(n)$. Each speech frame $\boldsymbol{x}_p$ must go through preprocessing steps before any windowing can take place.

2.3.3 Waveform Mean Removal

Before $x_p(m)$ is windowed, it must go through a *Mean Removal* step. Mean Removal has for goal to rid $x_p(m)$ of its constant component (or DC component) for each frame p. This mean component brings no relevant information for the Fourier Analysis. For each frame p, the mean μ_p defined by

$$\mu_p = \frac{1}{M}\sum_{m=0}^{M-1} x_p(m) \tag{2.15}$$

is removed to the frame points. The sequence $x_p(m)$ is transformed such that

$$\underline{x}_p(m) = x_p(m) - \mu_p \quad \forall m \in [0, 1, \ldots, M-1] \tag{2.16}$$

or in vector notation

$$\begin{aligned} \underline{\boldsymbol{x}}_p &= \boldsymbol{x}_p - \boldsymbol{\mu}_p && (2.17)\\ &= \boldsymbol{x}_p - \frac{1}{M}\left(\boldsymbol{J}\boldsymbol{x}_p\right) \\ &= \boldsymbol{I}\boldsymbol{x}_p - \frac{1}{M}\boldsymbol{J}\boldsymbol{x}_p \\ &= \left(\boldsymbol{I} - \frac{1}{M}\boldsymbol{J}\right)\boldsymbol{x}_p && (2.18)\\ &= \boldsymbol{R}\boldsymbol{x}_p && (2.19) \end{aligned}$$

where $\boldsymbol{\mu}_p$ is a $(M\times 1)$ vector with all its components equal to the scalar μ_p and $\boldsymbol{J}$ is a $(M\times M)$ matrix whose components are equal to one. $\boldsymbol{R}$ is the matrix what once multiplied to a vector gives the zero mean version of this vector. Once we have this zero-mean frame, a window is applied to it[3].

2.3.4 Windowing

The zero-mean signal $\underline{x}_p(m)$ is multiplied to a window $w(m)$ that is often chosen to be either a Hamming or a Blackman window. The Byblos Front End uses a Hamming window defined by the following equation for $w(m)$:

$$w(m) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi m}{M-1}\right), & 0 \le m \le M-1, \\ 0, & \text{otherwise}, \end{cases} \tag{2.20}$$

[3]It can be noticed that once a zero mean signal is windowed, it is not necessarily zero mean. Another Mean Removal step should be performed if we want to guarantee to have a zero mean windowed signal.

where M is the length of our window equal to the length of our frame. The window length for ASR is typically 25 ms and the interval between frames is 10 ms. Since we are working with telephone quality speech signal, we know that the sampling frequency F_s is set to $F_s = 8000\,\text{Hz}$. $s(n)$ is then considered as narrow-band since only the frequencies in the range $[0\,\text{Hz}, 4000\,\text{Hz}]$ follow the Shannon Theorem. For wide-band, $F_s = 16000\,\text{Hz}$ and a range of $[0\,\text{Hz}, 8000\,\text{Hz}]$ is achieved. A 25 ms window has the number of samples of $M = 25\,\text{ms} \times 8000\,\text{Hz} = 200$. For each frame p, the zero-mean signal $\underline{x}_p(m)$ is windowed using $w(m)$ such that

$$\underline{x}_p^w(m) = \underline{x}_p(m) \times w(m) \quad \text{for } m = 0, 1, \ldots, M-1, \tag{2.21}$$

which becomes in vector notation:

$$\underline{\boldsymbol{x}}_p^w = \boldsymbol{D}_w \times \underline{\boldsymbol{x}}_p \tag{2.22}$$

where $\boldsymbol{D}_w$ is the following $(M \times M)$ diagonal matrix:

$$\boldsymbol{D}_w = \text{diag}(\boldsymbol{w}) = \begin{bmatrix} w(0) & 0 & \cdots & 0 \\ 0 & w(1) & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & w(M-1) \end{bmatrix}. \tag{2.23}$$

In order to simplify our notations, we will refer to our zero-mean, windowed and block framed signal $\underline{x}_p^w(m)$ by using $x_p(m)$ from now on. Similarly, $\boldsymbol{x}_p$ will be used in place of $\underline{\boldsymbol{x}}_p^w$.

2.3.5 Fourier Spectrum

For each one of the frames, the DFT of the zero-mean windowed signal $x_p(m)$ is computed:

$$\mathring{X}_p(k) = \text{DFT}\left(\mathring{x}_p(m)\right), \quad \forall\, p \tag{2.24}$$

where $\mathring{x}_p(m)$ is a periodic version of $x_p(m)$ due to the discretization of its frequency representation. Equation (2.24) can be rewritten in vector notation using the DFT matrix $\boldsymbol{F}$ defined by

$$\boldsymbol{F} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{\jmath\frac{2\pi}{N}} & e^{\jmath\frac{4\pi}{N}} & e^{\jmath\frac{6\pi}{N}} & \dots & e^{\jmath\frac{2\pi}{N}(N-1)} \\ 1 & e^{\jmath\frac{4\pi}{N}} & e^{\jmath\frac{8\pi}{N}} & e^{\jmath\frac{12\pi}{N}} & \dots & e^{\jmath\frac{2\pi}{N}(N-1)\times 2} \\ 1 & e^{\jmath\frac{6\pi}{N}} & e^{\jmath\frac{12\pi}{N}} & e^{\jmath\frac{18\pi}{N}} & \dots & e^{\jmath\frac{2\pi}{N}(N-1)\times 3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\jmath\frac{2\pi}{N}(N-1)} & e^{\jmath\frac{2\pi}{N}2(N-1)} & e^{\jmath\frac{2\pi}{N}3(N-1)} & \dots & e^{\jmath\frac{2\pi}{N}(N-1)(N-1)} \end{bmatrix} \tag{2.25}$$

The $(M \times M)$ matrix $\boldsymbol{F}$ multiplied to any vector will provide its DFT. It is clear that there is a symmetry between the variables k and n resulting in $\boldsymbol{F}^T = \boldsymbol{F}$. The inverse matrix $\boldsymbol{F}^{-1}$ defined such that $\boldsymbol{F}^{-1} \times \boldsymbol{F} = \boldsymbol{I}$ is

$$\boldsymbol{F}^{-1} = \frac{1}{N}\overline{\boldsymbol{F}} \tag{2.26}$$

where $\overline{\boldsymbol{F}}$ is the complex conjugate of $\boldsymbol{F}$. Equation (2.24) using the DFT matrix $\boldsymbol{F}$ can be rewritten in vector notation as

$$\tilde{\boldsymbol{x}}_p = \boldsymbol{F}\boldsymbol{x}_p$$

where $\tilde{\boldsymbol{x}}_p$ is the resulting DFT spectrum vector. All the different steps that have discussed up to now involve simple linear algebra expression that can be streamlined to obtain our final DFT spectrum:

$$\begin{aligned} \tilde{\boldsymbol{x}}_p &= \boldsymbol{F}\boldsymbol{D}_w\underline{\boldsymbol{x}}_p \\ &= \boldsymbol{F}\boldsymbol{D}_w(\boldsymbol{x}_p - \boldsymbol{\mu}_p) \\ &= \boldsymbol{F}\boldsymbol{D}_w\boldsymbol{R}\boldsymbol{x}_p \\ &= \boldsymbol{F}\boldsymbol{D}_w\left(\boldsymbol{I} - \frac{\boldsymbol{J}}{M}\right)\boldsymbol{x}_p \end{aligned} \tag{2.27}$$

2.3.6 Power Spectrum

From the DFT spectrum $\mathring{X}_p$, the computation of the power spectrum is straightforward. The power spectrum is a function of discrete frequency k defined by

$$P_{X_p}(k) = \left|\mathring{X}_p(k)\right|^2 \quad \forall\, k \tag{2.28}$$

The power spectrum computation can be expressed in vector notation as

$$\boldsymbol{p}_{\tilde{\boldsymbol{x}}} = \mathrm{diag}(\tilde{\boldsymbol{x}})\,\overline{\tilde{\boldsymbol{x}}} \tag{2.29}$$

where $\boldsymbol{x}$ corresponds to $\boldsymbol{x}_p$ to simplify notations. If we use the complex conjugate transpose defined by $\boldsymbol{A}^* = \overline{\boldsymbol{A}}^T$ for a complex matrix A, we can compute $\boldsymbol{p}_{\tilde{\boldsymbol{x}}}$ with

$$\boldsymbol{p}_{\tilde{\boldsymbol{x}}} = \mathrm{diag}(\boldsymbol{x}\boldsymbol{x}^*) \tag{2.30}$$

where $\boldsymbol{p}_{\tilde{\boldsymbol{x}}}$ is a vector composed by the diagonal elements of $\boldsymbol{x}\boldsymbol{x}^*$.

The power spectrum $P_{X_p}(k)$ provides information about the energy distribution in frequencies for each frame p. However, P_{X_p} is still the power spectrum of $s(n)$, the convolution of the excitation $e(n)$ and $v(n)$, windowed with $w(n)$. For ASR, the information of $e(n)$ is less important than the information carried by $v(n)$. The need to separate $e(n)$ and $v(n)$ to obtain an approximation of $v(n)$ lead ASR to utilize *Cepstral Processing* to obtain a sequence of features for speech signals. However, prior to any cepstral processing, the power spectra need to be modified to compensate for VTL differences and to mimic the human perception of sounds.

2.3.6.1 LPC Spectrum

Instead of creating our cepstral features from a power Fourier spectrum, it is often chosen to work on the LPC spectrum of the speech signal. *Linear Predictive Coding* (LPC) was defined by Makhoul[16]. LPC enables the creation of a smooth power spectrum for any time-domain signal based on a very simple assumption. The definition of the LPC spectrum is derived directly from the assumption that the value of a signal at time t can be expressed as a linear transformation of a finite number of the previous samples that occurred before t. The consequence of this assumption is to define a power spectrum that corresponds to the envelope of the Fourier Spectrum of the original time-domain signal. This power spectrum, often called LPC spectrum, has several poles that control its "smoothness". This spectral smoothing property is of great use for ASR. For ASR, it is the difference of formants' location between voiced phonemes that allows their recognition. Since formants can be interpreted as poles in the Fourier Spectrum of voiced speech, the use of LPC spectrum is relevant to ASR.

2.4 SPECTRUM TRANSFORMATIONS

2.4.1 Band-Limiting

For an ASR system working on phone-quality speech, the frequency bandwidth for the DFT spectrum is directly limited by the sampling frequency $F_s = 8000\,\text{Hz}$. The available range of frequency is $[0\,\text{Hz}, 4000\,\text{Hz}]$ from Shannon's Theorem. Unfortunately, the phone channel as well as other undesired noises such as the $60/50\,\text{Hz}$ noise coming from power lines decrease the effective bandwidth of usable frequencies. A band-limiting of the spectrum is required to avoid those corrupted frequencies. The usable frequency range for our ASR system is chosen to be

$$R = [125\,\text{Hz}, 3750\,\text{Hz}] \tag{2.31}$$

in order to eliminate most of the noises and still keep enough of the spectrum of the speech signal. All the frequencies outside this range are discarded. From a vector notation point of view, this can be interpreted as a dimension reduction of the power spectrum vector.

2.4.2 Frequency Warping for VTLN

In order to compensate for spectrum variations due to VTL differences among speakers, a frequency transformation is performed for each frame. This transformation of the frequency axis of the spectrum is speaker dependent and is referred to as Frequency Warping. The goal of this transformation is basically to modify the spectrum by transforming the frequency axis. The result is a warped spectrum with a changed energy distribution. A frequency axis transformation can be defined by the equation

$$\omega = g(\psi) \tag{2.32}$$

that gives the relation between the new frequency ω and the original frequency ψ. If we consider the unwarped spectrum $X(\psi)$, the transformation into the warped spectrum $Y_g(\omega)$ is given by

$$\begin{aligned} Y_g(\omega) &= X\big(g(\psi)\big) \\ &= X(\omega) \end{aligned} \tag{2.33}$$

where $Y_g(\omega)$ is the warped version of $X(\psi)$. Similarly, Equation (2.33) holds for the power spectra $P_{Y_g}(\psi)$ and $P_X(\psi)$:

$$P_{Y_g}(\omega) = P_X(\omega) = P_X\big(g(\psi)\big) \tag{2.34}$$

It is important to notice that the transformation is performed on ψ only and not on the values of $X(\psi)$. There is *a priori* no restriction on the type of frequency transformation we can perform. Those functions can be divided into two main categories: linear and non-linear transforms.

2.4.2.1 Linear Transforms A simple linear frequency warping can theoretically compensate for VTL variations in the case of the lossless tube model of the VT. Let g_{LT} be the following frequency warping:

$$\begin{aligned} \omega &= g_{\text{LT}}(\psi) \\ &= \beta_s \times \psi \quad \text{for} \quad \beta_s > 0 \end{aligned} \tag{2.35}$$

where $\beta_s \in \mathbb{R}$ is a speaker dependent stretching factor. The original power spectrum $P_X(\psi)$ is warped to become $P_{Y_{\text{LT}}}(\omega)$ such as

$$P_{Y_{\text{LT}}}(\omega) = P_X(\beta\psi). \tag{2.36}$$

No warping is performed for $\beta_s = 1$. For $\beta_s < 1$, the power spectrum $P_{Y_{\text{LT}}}(\omega)$ results in an compression of $P_X(\psi)$ whereas $\beta_s > 1$ results in an expansion of $P_X(\psi)$. Linear warping is often utilized because of its implementation simplicity. It can be generalized by a piecewise-linear transform where several scalars are defined for different part of the spectrum.

2.4.2.2 Non-Linear Transforms BYBLOS utilizes a non-linear frequency warping function [7] defined in angular frequency by

$$\omega = k_s^{\left(\frac{3\psi}{16{,}000\pi}\right)} \times \psi \tag{2.37}$$

or

$$f_{\text{w}} = k_s^{\left(\frac{3f}{8{,}000}\right)} \times f \tag{2.38}$$

where f is the original frequency axis and f_{w} the warped (hence the subscript w) frequency axis. k_s is referred to as a "warp factor" or "warp" which is speaker dependent. A warp factor is assigned to each speaker, offering a specific spectrum transformation to each speaker. In Figure (1) we can see some of the frequency warping accomplished by Equation (2.38) for several values of the warp factor k_s.

2.4.3 The Mel-Scale Warping

The human perception of sounds does not follow a linear scale[17, 18, 4, 19, 20] as the field of psychoacoustics has demonstrated. Each frequency component of a sound is perceived in its subjective frequency different from its measurable value. This subjectivity is linked to the human ear processing of sounds and is all the more apparent in speech perception. It is believed that perceptually motivated transformations of the spectrum of a speech signal will increase its discriminatory power in a ASR sense, leading to better recognition results. Several mappings have been defined between a measured frequency and its subjective counterpart. One of them widely known as the *Mel-Scale* provides a subjective pitch measured in "mel" for each tone measured in Hz. It was defined originally as a function trying to accommodate experimental data points. The mel-warping function that provides the subjective frequency f_{mel} (in mel) for the frequency f in Hz is widely used in ASR and has for equation

$$f_{\text{mel}}(f) = \frac{1000}{\log(2)} \log\left(1 + \frac{f}{1000}\right) \tag{2.39}$$

that can be generalized into

$$f_{\text{mel}}(f, \alpha) = \frac{1000}{\log(\alpha)} \log\left(1 + (\alpha - 1)\frac{f}{1000}\right) \quad \forall\, \alpha > 1. \tag{2.40}$$

α is a parameter that controls the shape of the function. Increasing the value of α will compress more the high frequencies of the spectrum. The mel-warping function has one fixed-point for $f = 1000\,\text{Hz}$, then $f_{mel}(1000, \alpha) = 1000\,\text{mel}$ for any values of α. The frequencies below 1000 Hz are expanded while the rest of the spectrum is compressed. When $\alpha \to \infty$, it can be noticed that $f_{mel}(f, \alpha) \to 1000$, the mapping becomes a constant (horizontal line).

The Byblos Front End of our ASR system utilizes Equation (2.40) but some other ASR systems like HTK (HMM Tool Kit) from Cambridge University and Sphinx from Carnegie Mellon University system employ the following Mel-scale:

$$\begin{aligned} f_{\text{mel}}(f) &= 2595 \log_{10}\left(1 + \frac{f}{700}\right) \\ &= \frac{2595}{\log(10)} \log\left(1 + \frac{f}{700}\right) \\ &= 1127 \log\left(1 + \frac{f}{700}\right) \end{aligned} \tag{2.41}$$

which corresponds to Equation (2.40) for $\alpha = \left(\frac{1000}{700} + 1\right)$ or approximately $\alpha = 2.429$. Figure (2) presents the different Mel-Scale warping transformations that can be achieved for several values of α.

2.4.4 Implementation Challenges

The Band-Limiting, VTLN Frequency Warping and Mel-Scale Warping are all transformations of the frequency axis of our Short-Time spectrum (being LPC, Fourier or Power Spectrum). These transformations can be merged into a single transformation being the composition of these three transformations. Let $\mathcal{W}$ be the final frequency transformation. Each frequency f will be transformed into its warped version f_{w} such that

$$f_{\text{w}} = \mathcal{W}(f). \tag{2.42}$$

The implementation issue comes from the fact that we do not have access to a continuous frequency axis but to a discrete one. The previous equation becomes

$$k_{\text{w}} = \mathcal{W}(k). \tag{2.43}$$

where k is the discrete frequency in the spectrum from the Short-Time Fourier Analysis. It is possible that in our new spectrum, a frequency k_{w} originally comes from a frequency k that is in-between two known frequencies. Therefore the value of the original spectrum at this value of k is unknown. In order to remedy to this problem, the new warped spectrum is obtained via interpolation of the original spectrum. In the case of our Front End, a simple first order interpolation is considered sufficient to compute the warped spectrum.

2.5 CEPSTRAL PROCESSING

If we recall the Equation (2.2) on page 11, an ASR system can have access only to $s(n)$, convolution of the excitation signal $e(n)$ and the vocal tract contribution $v(n)$, but not to $v(n)$ alone. An *Homomorphic Deconvolution* approach[21] offers a solution to recover $v(n)$ from $s(n)$. It is accomplished by means of an homomorphic transformation $\mathcal{H}$ that converts a convolution into a sum. If we define $\hat{s}(n)$ as the "$\mathcal{H}$-Transform" of $s(n)$, $\hat{s}(n)$ can be expressed as

$$s(n) = e(n) * h(n) \overset{\mathcal{H}}{\rightleftharpoons} \hat{s}(n) = \hat{e}(n) + \hat{h}(n). \tag{2.44}$$

$\hat{s}(n)$ is a new discrete-time sequence where filter and source are not convoluted anymore. This property of $\mathcal{H}$ takes all its significance in the Fourier Domain. If we compute the DTFT spectrum $S(\mathrm{e}^{\jmath\omega})$ of $s(n)$, we obtain

$$s(n) = e(n) * h(n) \overset{\mathcal{F}}{\rightleftharpoons} S(\mathrm{e}^{\jmath\omega}) = E(\mathrm{e}^{\jmath\omega}) \times H(\mathrm{e}^{\jmath\omega})$$

and we compare the result to the DTFT spectrum of the transformed sequence $\hat{s}(n)$,

$$\hat{s}(n) = \hat{e}(n) + \hat{h}(n) \overset{\mathcal{F}}{\rightleftharpoons} \hat{S}(\mathrm{e}^{\jmath\omega}) = \hat{E}(\mathrm{e}^{\jmath\omega}) + \hat{H}(\mathrm{e}^{\jmath\omega})$$

If $\hat{E}(\mathrm{e}^{\jmath\omega})$ and $\hat{H}(\mathrm{e}^{\jmath\omega})$ are spectrally well separated, a simple filtering could eliminate in $\hat{s}(n)$ any contribution from $\hat{e}(n)$ and therefore $e(n)$. Most ASR systems chose to use cepstral processing to perform this Homomorphic Deconvolution. The *complex cepstrum* of a time sequence $s(n)$ is the time sequence $c_{\mathrm{x}}(n)$ defined by

$$c_{\mathrm{x}}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log\left(S(\mathrm{e}^{\jmath\omega})\right) \mathrm{e}^{\jmath\omega n} \mathrm{d}\omega \tag{2.45}$$

where $\log(x)$ is the natural logarithm of x and $S(\mathrm{e}^{\jmath\omega})$ is the DTFT of $s(n)$. We mentioned earlier that the phase information is often regarded as not important for speech recognition. This lead to utilize in ASR the *"real"* complex cepstrum (or real cepstrum) instead. The real cepstrum is defined by

$$c(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log\left|S(\mathrm{e}^{\jmath\omega})\right| \mathrm{e}^{\jmath\omega n} \mathrm{d}\omega \tag{2.46}$$

where the phase information of $S(e^{\jmath\omega})$ has clearly been discarded. For discretized frequencies from the DFT, the equation becomes

$$\mathring{c}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log\left|\mathring{S}(k)\right| \, e^{\jmath\frac{2\pi}{N}nk}. \tag{2.47}$$

where $\mathring{c}(n)$ is a periodic version of $c(n)$. In ASR it is chosen to work on the power spectrum instead on the magnitude spectrum. The magnitude spectrum $|S(e^{\jmath\omega})|$ is replaced by $|S(e^{\jmath\omega})|^2$ in Equation (2.46). The new real cepstrum sequence $c_{\text{pow}}(n)$ that we obtain is in fact just a scaled version of the one in Equation (2.46):

$$\begin{aligned} c_{\text{pow}}(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|S(e^{\jmath\omega})|^2 \, e^{\jmath\omega n} d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} 2\log|S(e^{\jmath\omega})| \, e^{\jmath\omega n} d\omega \\ &= 2c(n) \end{aligned} \tag{2.48}$$

Using the power spectrum creates only a scaling of the cepstral sequence which is not of much importance as the features are normalized later on. Another implementation modification of Equation (2.47) comes from the fact that the log function is not defined for log(0). To avoid this problem, the implementation of Equation (2.46) in our ASR system is making use of the DFT and becomes in fact

$$\mathring{c}(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log\left(\left|\mathring{S}(k)\right|^2 + 1\right) \, e^{\jmath\frac{2\pi}{N}nk}. \tag{2.49}$$

For each frame p, only the N_c first cepstral coefficients from $\mathring{c}(n)$ are considered to be used with our ASR system. It is often chosen to take $N_c = 14$ to get rid of the excitation contribution. The coefficients $c(1), \ldots, c(14)$ will be part of the speech feature vector for each frame p. We discard $c(0)$ as it is sensitive to scaling.

2.6 SPEECH FEATURE VECTOR

The cepstral coefficients defined in Equation (2.49) are the speech features ASR works on to build acoustic models and perform speech recognition. Previously, we mentioned that each frame will bring a set of 14 cepstral coefficients features $(c(1), \ldots, c(14))$ called the "base features". The total number of features for our feature vector for a frame p is increased by incorporating the energy E_s of the time signal $s(n)$. For the frame p, the "base" feature vector $\boldsymbol{v}_{b_p}$ becomes

$$\boldsymbol{v}_{b_p} = \begin{Bmatrix} c(1) \\ \vdots \\ c(14) \\ E_s \end{Bmatrix} \tag{2.50}$$

The final speech feature vector is composed of the $\boldsymbol{v}_p$ and of its first and second derivative. The complete speech feature vector is a 45-dimensional vector

$$\boldsymbol{v}_p = \begin{Bmatrix} v_{b_p} \\ \Delta v_{b_p} \\ \Delta^2 v_{b_p} \end{Bmatrix} \tag{2.51}$$

where Δ is a discrete derivative operator, Δ^2 being the second order discrete derivative operator.

2.7 FEATURE NORMALIZATION

The cepstral features are normalized over the length of a conversation through two commonly used techniques: *Cepstral Mean Subtraction* (CMS) and *Variance Normalization*. CMS is used to eliminate the channel influence on the spectrum of the speech sequence. Variance normalization removes any scaling issues of the features by imposing unit variance on the cepstral features.

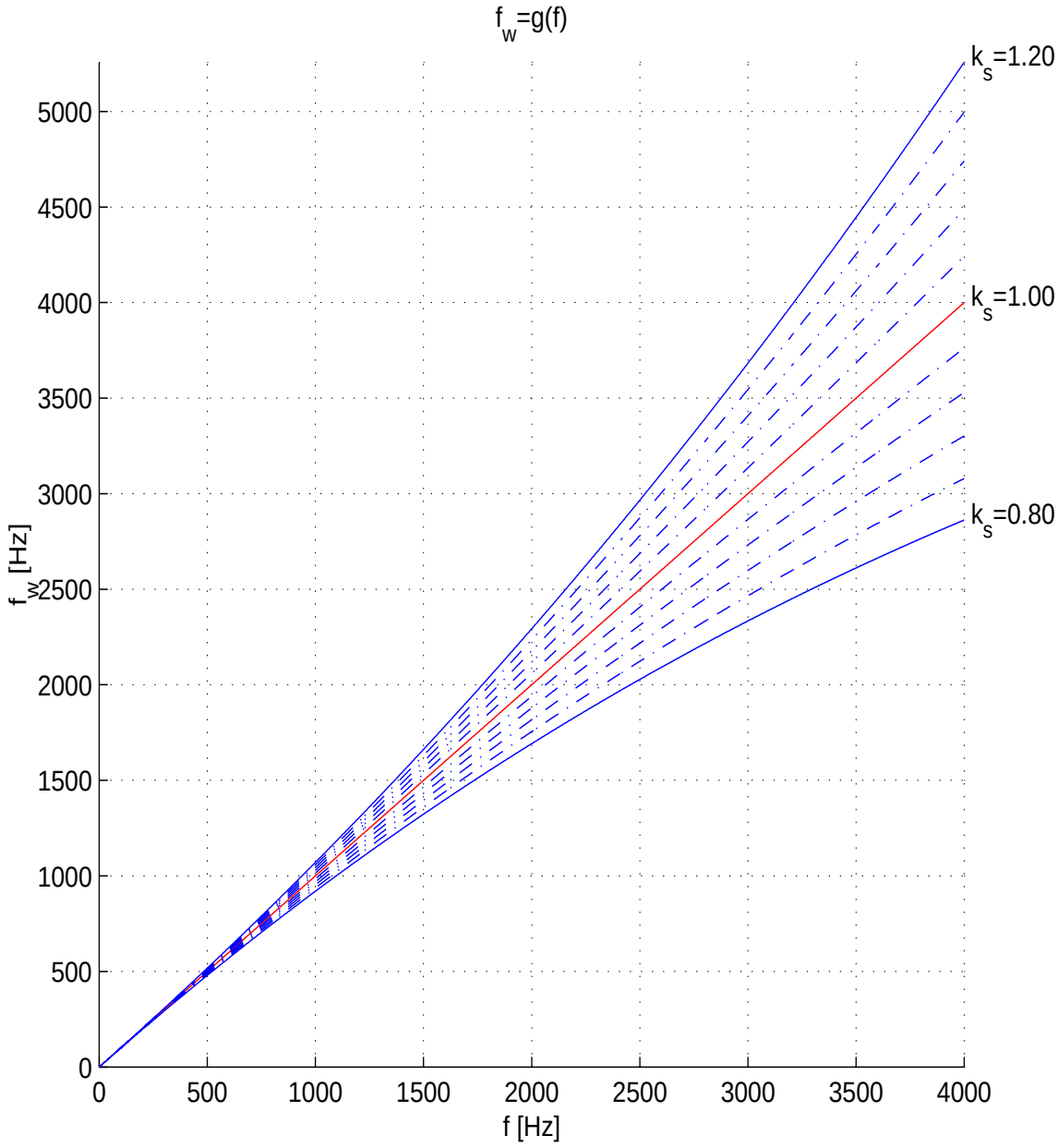

Figure 1: Eide's non-linear frequency warping for VTLN for $k_s = 0.80$ to $k_s = 1.20$ with a step $\Delta k_s = 0.04$.

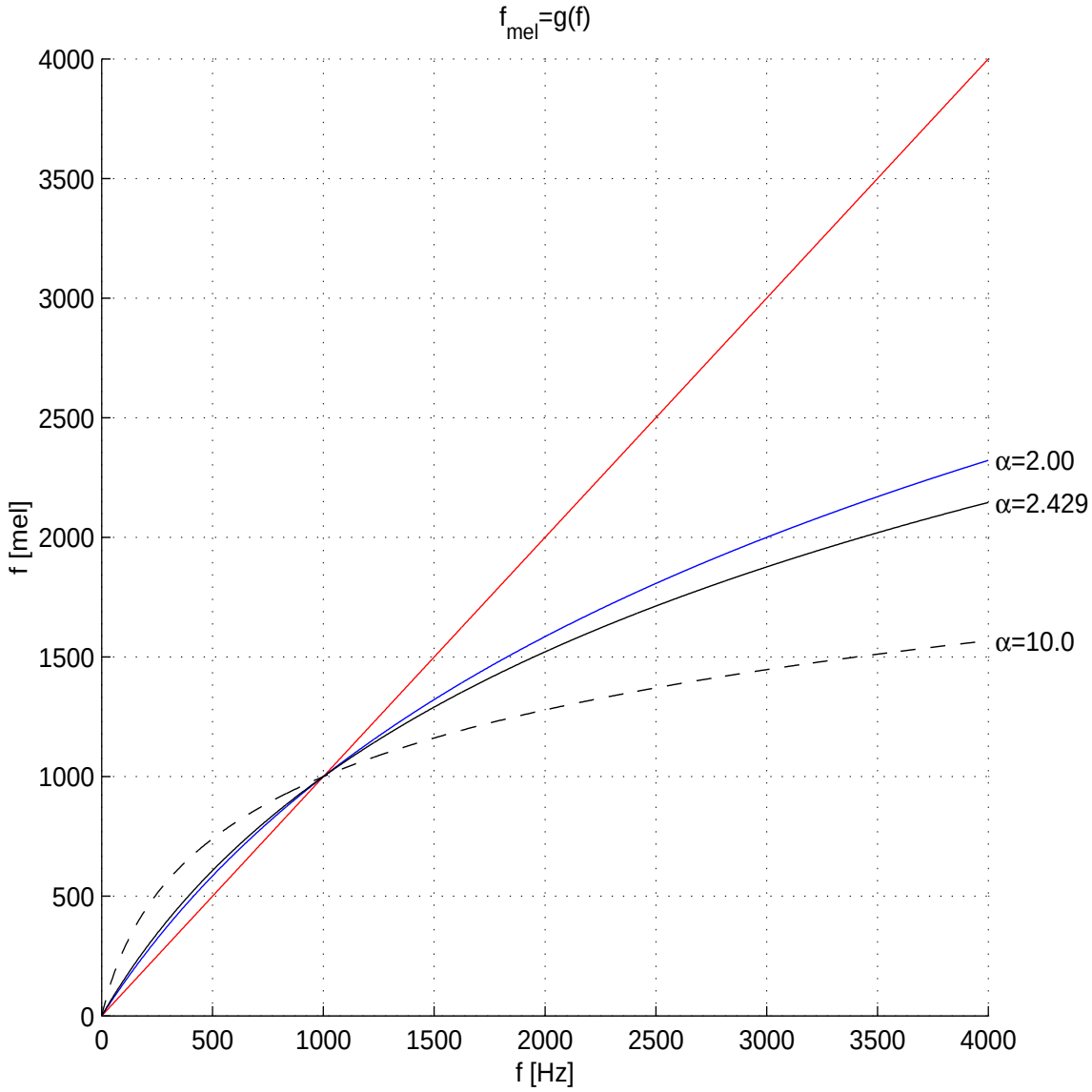

Figure 2: Mel-Scale frequency warping for $\alpha = 2.0$, $\alpha = (\frac{1000}{700} + 1)$ and $\alpha = 10$.

3.0 FREQUENCY WARPING FOR SPEAKER NORMALIZATION

All VTLN techniques based on frequency warping need to assess two important points: First, among all possible frequency transformations find a class of transformations believed to better compensate for VTL variations. Second, define an approach to estimate the SD parameters for the chosen transformation. This chapter will focus entirely on the first point: the choice of a frequency transformation.

The frequency warping performed by the Bilinear Transformation (BLT) is a good candidate for VTLN[8, 10]. In earlier work with BLT[8], the frequency transformation is controlled by only one parameter and is equivalent to converting a simple prototype LP digital filter into a desired LP filter. More recently, a generalization of this approach[10] was offered allowing the use of an *All-Pass Transformation* or APT that allows more complex frequency warpings. In regards to filter design, this approach is still a transformation of a LP prototype filter into some "other" filter. APT is controlled by few parameters leading to frequency transformation of complex shapes. However, the previous BLT and APT approach focuses mainly on utilizing BLT and APT to transform directly the cepstral feature sequences. Our work proposes to expand this approach on three points:

1. Our frequency transformation should not be limited to the transformation of a Low-Pass (LP) prototype to some "other" filter such as LP, Band-Pass (BP) or filter or more complicated transfer function. We propose instead to use the class of BLT that converts a prototype BP of our choice to a desired BP Filter. This allows to have a complex mapping controlled by two parameters allowing two degrees of freedom. This frequency transformation is a novel transformation for VTLN.

2. The BP to BP frequency mapping we propose maps a frequency band to another. This

approach seems a lot more appealing for Speaker Normalization than mapping only a frequency point to another, as seen for previous work with BLT. This property of defining a band of frequencies to be mapped instead of a single frequency point confers a lot more finesse to the spectral transformation that is performed. This enables to move selectively frequency band that are of interest for ASR, especially the frequency bands where formants are present.

3. The new form of BLT we define will not be used to transform the cepstral sequence but to transform the Short-Time Fourier spectrum of the speech signal prior to the computation of cepstral coefficients. One limitation of transforming the cepstral sequence is that the number of cepstral coefficient used to compose the base speech features is generally limited to $N_c = 14$ as mentioned in Section 2.6.

BLT is referred to in Complex Analysis as the *Möbius Transformation*[1] is the center of our work on defining a novel frequency transformation to perform VTLN.

3.1 THE MÖBIUS TRANSFORMATION

The Möbius Transformation is a mapping in the complex plane $\mathbb{C}$ of the complex variable $z = re^{j\omega}$ to the complex variable $w = \rho e^{j\psi}$ defined by

$$z \mapsto w = M(z) = \frac{az + b}{cz + d} \tag{3.1}$$

where the constants $a,b,c,d \in \mathbb{C}$. The variable z maps to w through the mapping[2] $M(z)$. If $(ad-bc) = 0$, the mapping becomes singular and all points z are sent to the same image point $\left(\frac{a}{c}\right)$ which is of no interest to us. Therefore, we will consider only the mappings for which $(ad-bc) \neq 0$ as they are non-singular. The main properties of this class of transformation can be summarized in three points:

[1] In this document, we refer to the "Möbius Transformation" whenever we are clearly in the context of Complex Analysis theory and gradually switch to use "Bilinear Transformation" when discussing it in a purely signal processing context.

[2] Throughout this document, the variable $z = re^{j\omega}$ will always refer to the variable mapping to the variable $w = \rho e^{j\psi}$. We will try to avoid any confusion to the reader between the argument of z (which is ω) and the complex variable w.

- Möbius Transformations map circles to circles
- Möbius Transformations are conformal (they preserve angles)
- Möbius Transformations preserve symmetry

Moreover, a Möbius Transformation is one-to-one and conformal. A more important point to us is that there exists a unique Möbius Transformation sending any 3 points to any 3 points in the complex plane $\mathbb{C}$[22].

3.1.1 Automorphism of the Unit Disc

We are most interested in the specific class of Möbius Transformations that performs an *automorphism* of the unit disc. An automorphism of a region R in the complex plane $\mathbb{C}$ is a one-to-one and conformal mapping of R to itself. If R is a disc, there exists a Möbius Transformation $M(z)$ that can perform this mapping of R to itself. Since $M(z)$ is one-to-one and conformal, it is by definition an automorphism. If we chose R to be the unit disc then $M(z)$ is an automorphism of the unit disc. From Complex Analysis[22], a general form for this automorphism is given by

$$z \mapsto w = M_a^{\phi}(z) = \mathrm{e}^{\jmath\phi}\frac{z-a}{a^* z - 1} \tag{3.2}$$

where a is a complex constant inside the unit disc ($|a|<1$) and a^* is its complex conjugate (inside the unit disc as well). The term $\mathrm{e}^{\jmath\phi}$ can be interpreted as a rotation of angle ϕ. This equation has clearly a zero at $z=a$ and a pole at $z=\frac{1}{a^*}$. Interestingly, if the rotation angle ϕ is set to zero then $M_a^0(z)$ imposes that $a\mapsto 0$ and $0\mapsto a$. This is the only automorphism of the unit disc with the property of swapping 0 and a[22].

3.1.2 Mapping of the Unit Circle

The automorphism in Equation (3.2) has the property of mapping the unit circle to itself. This mapping is of great interest to us because the Fourier Transform of a discrete sequence corresponds to the Z-Transform of this sequence evaluated on the unit circle. The mapping of the unit circle to itself is therefore performing a frequency axis distortion of the Fourier

spectrum of a discrete sequence. The mapping of the complex variable $z = re^{\jmath\omega} = e^{\jmath\omega}$ lying on the unit circle can be derived from Equation (3.2) as

$$\begin{aligned} z \mapsto w = M_a^{\phi}(z) &= e^{\jmath\phi}\frac{z-a}{a^*z-1} \\ &= e^{\jmath\phi}\frac{e^{\jmath\omega}-a}{a^*e^{\jmath\omega}-1} \\ &= -\left(\frac{e^{\jmath\phi}}{e^{\jmath\omega}}\right)\left(\frac{e^{\jmath\omega}-a}{e^{-\jmath\omega}-a^*}\right) \end{aligned}$$

where the terms $e^{\jmath\phi}$ and $e^{\jmath\omega}$ of the left ratio have unity magnitudes. The numerator and denominator of the right ratio are complex conjugates of each other implying that this ratio has also unity magnitude. Therefore, the unit circle maps onto itself independently of the value chosen for ϕ and a since $\left|M_a^{\phi}(z)\right| = 1$; as a consequence, $w = \rho e^{\jmath\psi} = e^{\jmath\psi}$.

In order to use the mapping from Equation (3.2) as a frequency warping for VTLN, we need to impose some constrains on the parameters ϕ and a. For instance, it would be preferable that the point $z = 1$ (for $\omega = 0$) maps onto itself. This will preserve the DC frequency of the original spectrum in the new warped spectrum as it will map to itself. As a consequence, we need to have $M_a^{\phi}(1) = 1$ which is only possible for $\phi = \pi + 2\pi k$ (with $k \in \mathbb{N}$) and $a = a^*$, this latter implying that a must be real. The same constrains appear in order to ensure that $M_a^{\phi}(-1) = -1$. If $z = -1$ (for $\omega = \pi$) is mapped to itself, the highest frequency in our original spectrum is preserved in the new warped spectrum. Under these additional constrains, Equation (3.2) becomes

$$\begin{aligned} z \mapsto w = M_a^{\pi}(z) &= e^{\jmath\pi}\frac{z-a}{az-1} \\ &= -\frac{z-a}{az-1} \\ &= \frac{z-a}{1-az} \end{aligned} \tag{3.3}$$

where $a \in \mathbb{R}$ and $|a| < 1$. This mapping of the unit circle preserving the low frequency and high frequency of our Fourier Spectrum is defined by

$$z \overset{\mathcal{M}_a}{\longmapsto} w$$

where $\mathcal{M}_a$ is the our notation for the Möbius Transformation with real parameter a and $\phi{=}\pi$. The absence of the value for ϕ in the $\mathcal{M}_a$ notation will *always* imply that $\phi{=}\pi$. From Equation (3.3) we have the relation between w and z such that

$$w = \frac{z-a}{1-az}. \tag{3.4}$$

We can derive from Equation (3.4) the following relationship:

$$\begin{aligned} \frac{1}{w} &= \frac{1-az}{z-a} \\ &= \left(\frac{z^{-1}}{z^{-1}}\right)\frac{1-az}{z-a} \\ \Leftrightarrow w^{-1} &= \frac{z^{-1}-a}{1-az^{-1}} \\ &= \mathcal{M}_a(z^{-1}). \end{aligned} \tag{3.5}$$

An interesting point is the symmetry of the function $w^{-1} = \mathcal{M}_a(z^{-1})$ and its inverse $(\mathcal{M}_a)^{-1}$:

$$\begin{aligned} w^{-1} &= \mathcal{M}_a(z^{-1}) \\ &= \frac{z^{-1}-a}{1-az^{-1}} \\ \Leftrightarrow z^{-1} &= \frac{w^{-1}+a}{1+aw^{-1}} \\ &= \mathcal{M}_{-a}(w^{-1}). \end{aligned}$$

It is clear that the relation between the Möbius Transformation $\mathcal{M}_a$ and its inverse is

$$(\mathcal{M}_a)^{-1} = \mathcal{M}_{-a}. \tag{3.6}$$

Equation (3.5) is the Bilinear Transformation of z into w that transforms a prototype LP filter into another LP filter[23]. This class of Möbius automorphisms is known in Electrical Engineering as the "Bilinear Transformation" or "Bilinear Transform".

3.2 THE BILINEAR TRANSFORM

The frequency warping properties of the Bilinear Transform or BLT have been useful in digital filter design[21]. The frequency warping is the consequence of a mapping in the complex plane of the unit circle to itself. Such mapping is defined by

$$z^{-1} \mapsto w^{-1} = G(z^{-1}) \tag{3.7}$$

where the mapping sends $z^{-1} = \mathrm{e}^{-\jmath\omega}$ to $w^{-1} = \mathrm{e}^{-\jmath\psi}$. The mapping $G(z^{-1})$ is constrained in its form. In order to ensure causality and stability, $G(z^{-1})$ needs to be a rational fraction of z^{-1}[21]. The unit circle should map to itself and the region inside the unit circle should map to itself. All those constrains have been shown to be achieved with the following general form of mapping[23]:

$$z^{-1} \mapsto w^{-1} = G(z^{-1}) = \pm \prod_{k=1}^{N} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}}. \tag{3.8}$$

where $\alpha_k \in \mathbb{R}$ and $|\alpha_k| < 1$ From Equation (3.8), different mapping can be defined offering different warping properties. If we choose to have $N = 1$ and impose a positive sign on the fraction, we obtain the same form as our Möbius automorphism in Equation (3.5). In this form, $G(z^{-1})$ is a fraction of polynomial of z^{-1} of first order. We will refer to it as a "First Order Bilinear Transform" or 1st-BLT.

3.2.1 The First Order Bilinear Transform

A First Order Bilinear Transformation is a mapping[3] in the complex plane $\mathbb{C}$ that transforms a prototype LP filter into a desired LP filter. Let $w^{-1} = \mathrm{e}^{-\psi}$ be the complex variable on the unit circle associated with the prototype LP filter and $z^{-1} = \mathrm{e}^{-\jmath\omega}$ the variable associated to the desired filter. In order to transform the prototype filter into the desired filter, we need to replace the variable w^{-1} by $G(z^{-1})$ in the Z-Transform representation of the prototype filter. This mapping of z^{-1} to w^{-1} is the same as in Equation (3.5):

$$z^{-1} \mapsto w^{-1} = G(z^{-1}) = \frac{z^{-1} - a}{1 - az^{-1}} \quad a \in \mathbb{R},\ |a| < 1 \tag{3.9}$$

[3]From now on, we will call "mapping" an automorphism of the Unit Disc to itself that maps the unit circle to itself as well.

3.2.1.1 Frequency Warping A direct consequence of the mapping $G(z^{-1})$ takes shape into a warping occurring on the frequency axis of the Fourier spectrum. It is straightforward to derive this frequency warping from the mapping in Equation (3.9) by replacing w^{-1} and z^{-1} with their polar expressions:

$$\begin{aligned} \mathrm{e}^{-\psi} &= \frac{\mathrm{e}^{-\jmath\omega} - a}{1 - a\mathrm{e}^{-\jmath\omega}} \\ &= \mathrm{e}^{-\jmath\omega}\frac{1 - a\mathrm{e}^{\jmath\omega}}{1 - a\mathrm{e}^{-\jmath\omega}} \\ &= \mathrm{e}^{-\jmath\omega}\frac{1 - a\mathrm{e}^{\jmath\omega}}{(1 - a\mathrm{e}^{\jmath\omega})^*}. \end{aligned} \tag{3.10}$$

We are interested in the relation between the arguments of the Right Hand Side (RHS) and the Left Hand Side (LHS) of this equation. This relation is the analytical form of the frequency warping that occurs during the mapping. In the RHS of this equation, we have the ratio of complex numbers that are clearly the complex conjugates of each other. The argument of such a ratio is double the argument of the numerator[4]. The relationship between the arguments in Equation (3.10) in the light of this result becomes

$$\begin{aligned} \arg(\mathrm{e}^{-\psi}) &= \arg(\mathrm{e}^{-\jmath\omega}) + 2\arg(1 - a\mathrm{e}^{\jmath\omega}) \\ \Leftrightarrow -\psi &= -\omega + 2\arg(1 - a\mathrm{e}^{\jmath\omega}) \\ \Leftrightarrow \psi &= \omega - 2\arg(1 - a\cos(\omega) - \jmath a\sin(\omega)) \\ \Leftrightarrow \psi &= \omega - 2\arctan\left(\frac{-a\sin\omega}{1 - a\cos\omega}\right) \\ \Leftrightarrow \psi &= \omega - \rho(\omega, a) = g^{-1}(\omega, a) \end{aligned} \tag{3.11}$$

where $a \in \mathbb{R}$ and $|a| < 1$. This equation gives the "old" frequency ψ for all "new" frequency ω given the term $\rho(\omega, a)$ that can be interpreted as a "correction term". It seems that it would be more interesting for us to find the function $\omega = g(\psi, a)$ rather than its inverse form[5]. Finding the function $g(\psi, a)$ has a straightforward solution if we remember the symmetry property given in Equation (3.6) on page 33. The inverse mapping of w^{-1} to z^{-1} is found

[4] Let z be a complex number such that $z = r\mathrm{e}^{\jmath\omega}$, the ratio $\frac{z}{z^*} = \mathrm{e}^{\jmath 2\omega} = \mathrm{e}^{\jmath 2\arg z}$. Therefore $\arg\left(\frac{z}{z^*}\right) = 2\arg(z)$.

[5] It is in fact not an important issue. Our implementation of VTLN can accommodate having either $\psi = g^{-1}(\omega)$ or $\omega = g(\psi)$ or both.

by replacing a by $-a$ and w^{-1} by z^{-1} in Equation (3.9). In a similar manner we can find the frequency warping by performing the same changes in Equation (3.11) so it becomes

$$\omega = \psi - 2\arctan\left(\frac{a\sin\psi}{1+a\cos\psi}\right) \tag{3.12}$$

This relation between ω and the old frequency ψ is in fact a simpler rewriting of the more commonly known equation[21]:

$$\omega = \arctan\left(\frac{(1-a^2)\sin\psi}{2a+(1+a^2)\cos\psi}\right) \tag{3.13}$$

The one parameter a in both Equation (3.12) and Equation (3.13) controls completely the frequency warping characteristics as you can see in Figure (3).

3.2.1.2 Parameter Study Since the values of the constant a condition entirely the mapping, a can be chosen so that the cutoff frequency ψ_c of the prototype LP filter matches the cutoff frequency ω_c of the desired LP filter. It has been previously derived[23] that the function $a = f(\psi_c, \omega_c)$ is given by

$$a = \frac{\sin\left(\frac{\psi_c - \omega_c}{2}\right)}{\sin\left(\frac{\psi_c + \omega_c}{2}\right)} \tag{3.14}$$

where a is clearly a real constant. Figure (3) presents the warping obtained for different values of a. If $\psi_c = \omega_c$, $a = 0$ and no frequency warping is performed regardless of the values of ψ_c.

3.2.2 The Second Order Bilinear Transform

The new approach we propose in this work is based on a mapping $G(z^{-1})$ that offers a more complex shape for its resulting frequency warping than the one from a 1st-BLT. Instead of using a mapping that transforms a LP prototype filter into a desired LP filter, we became interested on working on a transformation of a BP prototype filter into a desired BP filter. In order to present this BP to BP mapping, we need first to understand the mechanisms behind the more common mapping of a LP filter into a BP filter. Then, the extension to the BP to BP transformation will be straightforward. In order to convert a LP filter into a BP filter, the mapping $G(z^{-1})$ does not need to be restricted to first order polynomials of z^{-1} for its numerator and denominator. Second order polynomials are sufficient and offer what we call the "2nd order BLT" or "2nd BLT".

The 2nd BLT converts a LP filter with cutoff frequency θ_p into a BP filter with lower cutoff frequency ω_{p1} and higher cutoff frequency ω_{p2}. The resulting mapping is given by

$$\begin{aligned} w^{-1} &= G(z^{-1}) \\ \Leftrightarrow w^{-1} &= -\frac{z^{-2} - \dfrac{2\alpha k}{k+1} z^{-1} + \dfrac{k-1}{k+1}}{\dfrac{k-1}{k+1} z^{-2} - \dfrac{2\alpha k}{k+1} z^{-1} + 1} \end{aligned} \tag{3.15}$$

where the parameters α and k are defined by

$$\alpha = \frac{\cos\left(\dfrac{\omega_{p2} + \omega_{p1}}{2}\right)}{\cos\left(\dfrac{\omega_{p2} - \omega_{p1}}{2}\right)} \tag{3.16}$$

and

$$\begin{aligned} k &= \frac{\tan\left(\dfrac{\theta_p}{2}\right)}{\tan\left(\dfrac{\omega_{p2} - \omega_{p1}}{2}\right)} \\ &= \cot\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right) \tan\left(\frac{\theta_p}{2}\right) \end{aligned} \tag{3.17}$$

These are well-known results whose derivations were first formulated by Constantinides[23].

3.2.2.1 Parameters Study If we define $\mu = \frac{\omega_{p2}+\omega_{p1}}{2}$ and $\delta = \frac{\omega_{p2}-\omega_{p1}}{2}$, μ is the mean of the two cutoff frequencies while δ is half the distance between them (half the bandwidth of the filter). Equation (3.16) and Equation (3.17) can be rewritten into

$$\alpha = \frac{\cos(\mu)}{\cos(\delta)} \tag{3.18}$$

and

$$k = \frac{\tan(\frac{\theta_p}{2})}{\tan(\delta)} \tag{3.19}$$

From Equation (3.18) it becomes clear that $\alpha = 0$ as soon as $\mu = \frac{\pi}{2}$. In other words, if the cutoff frequencies are symmetrically located on the right and left of $\frac{\pi}{2}$, α will always be zero. The term $\cos(\mu)$ is a bounded number directly linked to the position of the mean of the cutoffs frequencies compared to the value $\frac{\pi}{2}$. We know that $|\alpha| < 1$ and $|\gamma| < 1$. Since $\gamma = \frac{k-1}{k+1}$, it imposes that $k > 0$.

3.2.2.2 Equation Analysis At first look, the Equation (3.15) does not give an intuitive sense of what are the different steps necessary to transform a LP filter into a BP filter. This equation can be rewritten in a simpler form with use of two new parameters γ_1 and γ_2:

$$w^{-1} = -\frac{z^{-2} + \gamma_1 z^{-1} + \gamma_2}{\gamma_2 z^{-2} + \gamma_1 z^{-1} + 1} \tag{3.20}$$

where

$$\gamma_2 = \frac{k-1}{k+1} = \gamma \tag{3.21}$$

and

$$\begin{aligned} \gamma_1 &= -\frac{2\alpha k}{k+1} \\ &= -\alpha\left(\frac{k+1+k-1}{k+1}\right) \\ &= -\alpha\left(1+\frac{k-1}{k+1}\right) \\ \Leftrightarrow \gamma_1 &= -\alpha(1+\gamma_2) \\ &= -\alpha(1+\gamma) \end{aligned} \tag{3.22}$$

(3.23)

By using the newly defined parameter γ instead of γ_2 and replacing γ_1 by $-\alpha(1+\gamma)$, we obtain the following form for Equation (3.20):

$$w^{-1} = -\frac{z^{-2} - \alpha(1+\gamma)z^{-1} + \gamma}{\gamma z^{-2} - \alpha(1+\gamma)z^{-1} + 1} \tag{3.24}$$

or

$$w^{-1} = -\frac{z^{-2} - \alpha(1+\gamma)z^{-1} + \gamma}{1 - \alpha(1+\gamma)z^{-1} + \gamma z^{-2}}. \tag{3.25}$$

If we expand this last equation, we obtain the following form[23]:

$$w^{-1} = -\left\{ \frac{z^{-1}\left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right) + \gamma}{1 + \gamma z^{-1}\left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right)} \right\} \tag{3.26}$$

that offers an intuitive understanding of the different steps necessary to transform a LP filter into a BP filter. If we define $H_1(z^{-1})$ and $H_2(z^{-1})$ such that

$$H_1(z^{-1}) = z^{-1}\left(\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\right) \tag{3.27}$$

$$= z^{-1}\mathcal{M}_\alpha(z^{-1}) \tag{3.28}$$

where $\alpha \in \mathbb{R}$ and $|\alpha| < 1$ and

$$H_2(z^{-1}) = -\left(\frac{z^{-1} + \gamma}{1 + \gamma z^{-1}}\right) \tag{3.29}$$

$$= -\mathcal{M}_{-\gamma}(z^{-1}) \tag{3.30}$$

where $\gamma \in \mathbb{R}$ and $|\gamma| < 1$. It is clearly noticeable that Equation (3.26) is in fact

$$\begin{aligned} w^{-1} &= G(z^{-1}) \\ &= H_2(H_1(z^{-1})) \\ &= H_2 \circ H_1(z^{-1}) \\ &= (-\mathcal{M}_{-\gamma}) \circ (z^{-1}\mathcal{M}_\alpha)(z^{-1}) \end{aligned} \tag{3.31}$$

where $G(z^{-1})$ is the composition of a negative Möbius Transformation with parameter $-\gamma$ and the product of a Möbius Transformation with parameter α and z^{-1}. These two steps are necessary to transform a LP filter into a BP filter.

3.2.2.3 Frequency Warping The frequency warping that results of the transformation in Equation (3.24) is derived from

$$\begin{aligned} w^{-1} &= -\frac{z^{-2}+\gamma_1 z^{-1}+\gamma_2}{1+\gamma_1 z^{-1}+\gamma_2 z^{-2}} \\ &= -z^{-2}\frac{1+\gamma_1 z+\gamma_2 z^2}{1+\gamma_1 z^{-1}+\gamma_2 z^{-2}} \\ &= -z^{-2}\frac{1+\gamma_1 z+\gamma_2 z^2}{(1+\gamma_1 z+\gamma_2 z^2)^*} \end{aligned}$$

We are interested in the relation between the arguments of the RHS and the LHS of the equation. Here also we use the fact that for a complex number z, $\arg\left(\frac{z}{z^*}\right) = 2\arg(z)$ to obtain

$$\begin{aligned} -\psi &= -\pi - 2\omega + 2\arg(1+\gamma_1 e^{\jmath\omega}+\gamma_2 e^{\jmath\omega^2}) \\ \Leftrightarrow \psi &= \pi + 2\omega - 2\arctan\left(\frac{\gamma_1\sin\omega+\gamma_2\sin 2\omega}{1+\gamma_1\cos\omega+\gamma_2\cos 2\omega}\right) \end{aligned}$$

If we replace γ_1 and γ_2 by their equation respective, we have

$$\psi = \pi + 2\omega - 2\arctan\left(\frac{-\alpha(1+\gamma)\sin\omega+\gamma\sin 2\omega}{1-\alpha(1+\gamma)\cos\omega+\gamma\cos 2\omega}\right). \tag{3.32}$$

3.2.2.4 A Special Case of Transformation For the special case of $k=1$ ($\gamma=0$), $H_2(z^{-1})$ reduces to a simple rotation of angle $\frac{\pi}{2}$:

$$H_2(z^{-1})\big|_{\gamma=0} = -z^{-1} \tag{3.33}$$

For $\alpha=0$, $H_1(z^{-1})$ reduces to

$$H_1(z^{-1})\big|_{\alpha=0} = z^{-2} \tag{3.34}$$

and their composition $H_2 \circ H_1(z^{-1})$ becomes

$$\begin{aligned} H_2 \circ H_1(z^{-1})\big|_{\alpha,\gamma=0} &= -z^{-2} \\ \Leftrightarrow w^{-1} &= -z^{-2} \end{aligned} \tag{3.35}$$

This transformation is a special case of Equation (3.15) on page 37. It is the simplest expression you can get for $G(z^{-1}) = H_2 \circ H_1(z^{-1})$. Our next step is to recover the characteristics of the LP filter and BP filter involved in Equation (3.35).

Since the value of α and k (and therefore γ) define the prototype LP filter and the desired BP filter we can recover expressions of their characteristics, meaning their cutoff frequencies. For α to be zero, we saw earlier that μ needs to be $\frac{\pi}{2}$. The BP filter has w_{p1} and w_{p2} symmetric to $\frac{\pi}{2}$. Since $k=1$, Equation (3.19) becomes

$$\begin{aligned} k = \frac{\tan\left(\frac{\theta_p}{2}\right)}{\tan(\delta)} &= 1 \\ \Leftrightarrow \frac{\theta_p}{2} &= \delta \\ &= \frac{w_{p2} - w_{p1}}{2} \\ \Leftrightarrow w_{p2} &= \theta_p + w_{p1} \end{aligned} \tag{3.36}$$

One can notice that normally when we have $\tan(y)=\tan(x)$, the relation between y and x is $y = x + r\pi, r \in \mathbb{N}$. In our case, for whichever w_{p1} and w_{p2} and as long as $w_{p1} < w_{p2}$, we have $0<\delta<\frac{\pi}{2}$. Furthermore, $0<\theta_p<\pi$ which means that $0<\frac{\theta}{2}<\frac{\pi}{2}$. Then the relation $\frac{\theta_p}{2}=\delta$ without the modulo π. Since $\mu=\frac{\pi}{2}$, we can write that

$$\begin{aligned} \mu = \frac{w_{p2} + w_{p1}}{2} &= \frac{\pi}{2} \\ \Leftrightarrow w_{p2} &= \pi - w_{p1} \end{aligned} \tag{3.37}$$

From Equation (3.36) and Equation (3.37) we obtain the following set of equations:

$$\begin{cases} w_{p2} = \theta_p + w_{p1} \\ w_{p2} = \pi - w_{p1} \end{cases} \tag{3.38}$$

which can be rewritten into

$$\begin{cases} w_{p2} = \frac{\pi}{2} + \frac{\theta_p}{2} \\ w_{p1} = \frac{\pi}{2} - \frac{\theta_p}{2} \end{cases} \tag{3.39}$$

This set of equations is the direct consequence of $\alpha = 0$ and $k = 1$. It is interesting to see that the mapping is the same for various values of θ_p as long as the equations to find w_{p1} and w_{p2} are respected. The choice of θ_p conditions entirely the values of the cut-off frequencies. Among all the possible values for θ_p, we chose the value for which the prototype LP filter has a cutoff frequency of $\theta_p=\frac{\pi}{2}$ which imposes from Equation (3.39) that $w_{p1}=\frac{\pi}{4}$

and $w_{p2} = \frac{3\pi}{4}$ Therefore, the mapping defined in Equation (3.35) transforms a LP filter with cutoff frequency $\theta_p = \frac{\pi}{2}$ to a $\left[\frac{\pi}{4}, \frac{3\pi}{4}\right]$ BP filter which corresponds to a $[1000\,\text{Hz}, 3000\,\text{Hz}]$ BP filter for a sampling frequency $F_s = 8000\,\text{Hz}$.

3.3 A NOVEL APPROACH TO VTLN

3.3.1 The Bandpass Transform

When BLT is used to perform frequency warping, it is often chosen to use an equation that modifies a prototype LP filter. Instead of a prototype LP filter, our approach utilizes a prototype BP filter as starting point. In order to define our new mapping, we need to recall some of our previous results.

In order to transform a LP prototype filter of variable Z^{-1} with cutoff frequency $\theta_p = \frac{\pi}{2}$ to a BP filter of variable w^{-1} with cutoff frequencies w_1, w_2 respectively equal to $\frac{\pi}{4}$ and $\frac{3\pi}{4}$, we need to replace every Z^{-1} with the mapping

$$Z^{-1} = -w^{-2}$$

similar to the one in Equation (3.35). This mapping will provide our prototype BP filter.

If we want to transform the prototype LP to a general BP of variable z^{-1} with no assumptions on the cutoff frequencies, we use the mapping

$$Z^{-1} = -\frac{z^{-2} - \alpha(1+\gamma)z^{-1} + \gamma}{1 - \alpha(1+\gamma)z^{-1} + \gamma z^{-2}} \tag{3.40}$$

similar to Equation (3.25). Finally, the transformation of the prototype BP into the general BP filter is achieved by

$$-w^{-2} = -\frac{z^{-2} - \alpha(1+\gamma)z^{-1} + \gamma}{1 - \alpha(1+\gamma)z^{-1} + \gamma z^{-2}} \tag{3.41}$$

$$\Leftrightarrow w^{-2} = \frac{z^{-2} - \alpha(1+\gamma)z^{-1} + \gamma}{1 - \alpha(1+\gamma)z^{-1} + \gamma z^{-2}} \tag{3.42}$$

where $w^{-1} = \mathrm{e}^{-\jmath\psi}$ is the old variable replaced by $z^{-1} = \mathrm{e}^{-\jmath\omega}$, the new variable. This new transformation that is defined in Equation (3.42) is the *Bandpass Transform* (BPT).

3.3.2 Frequency Warping

The frequency warping that comes from this mapping can be directly derived from Equation (3.42) in the same manner as previously seen in Section 3.2.2.3 on page 40. We obtain the following results

$$\begin{aligned} \mathrm{e}^{-\jmath 2\psi} &= \frac{\mathrm{e}^{-\jmath 2\omega} - \alpha(1+\gamma)\mathrm{e}^{-\jmath\omega} + \gamma}{1 - \alpha(1+\gamma)\mathrm{e}^{-\jmath\omega} + \gamma\mathrm{e}^{-\jmath 2\omega}} \\ &= \mathrm{e}^{-\jmath 2\omega}\frac{1 - \alpha(1+\gamma)\mathrm{e}^{\jmath\omega} + \gamma\mathrm{e}^{\jmath 2\omega}}{1 - \alpha(1+\gamma)\mathrm{e}^{-\jmath\omega} + \gamma\mathrm{e}^{-\jmath 2\omega}} \\ &= \mathrm{e}^{-\jmath 2\omega}\frac{1 - \alpha(1+\gamma)\mathrm{e}^{\jmath\omega} + \gamma\mathrm{e}^{\jmath 2\omega}}{(1 - \alpha(1+\gamma)\mathrm{e}^{-\jmath\omega} + \gamma\mathrm{e}^{-\jmath 2\omega})^*} \end{aligned}$$

Therefore, if we take the argument of the RHS and LHS, we obtain

$$\begin{aligned} -2\psi &= -2\omega + 2\arg\left(1 - \alpha(1+\gamma)\mathrm{e}^{\jmath\omega} + \gamma\mathrm{e}^{\jmath 2\omega}\right) \\ \Leftrightarrow \psi &= \omega - \arg\left(1 - \alpha(1+\gamma)\mathrm{e}^{\jmath\omega} + \gamma\mathrm{e}^{\jmath 2\omega}\right) \\ &= \omega - \arctan\left(\frac{-\alpha(1+\gamma)\sin\omega + \gamma\sin 2\omega}{1 - \alpha(1+\gamma)\cos\omega + \gamma\cos 2\omega}\right) \\ &= \omega - \rho(\omega;\alpha,\gamma) \end{aligned} \tag{3.43}$$

where ψ is our old frequency, ω is the transformed frequency and $\rho(\omega;\alpha,\gamma)$ can be considered as a "correction" term. The fact that we obtained an expression of the old frequency ψ given the new frequency is not an issue *per se*. If the analytic form of a frequency warping is known as

$$\psi = f(\omega;\alpha,\gamma), \tag{3.44}$$

it is possible to implement its inverse

$$\omega = f^{-1}(\psi;\alpha,\gamma). \tag{3.45}$$

by means of a search algorithm (the binary search algorithm is one of them). It is a simple solution to find f^{-1} when only the analytic form f is known as long as f is one-to-one and monotonous which is the case for the BPT.

The resulting frequency warping is controlled by the parameters α and k as seen in Figure (4) and Figure (5). The mark "+" and "×" on each curve correspond respectively to the transform of $\psi = w_{p1} = \frac{\pi}{4}$ and $\psi = w_{p2} = \frac{3\pi}{4}$. The placement of those two points

controls the shape of the transformation entirely. One can notice that the two degrees of freedom enabling complicated shapes for the frequency warping function. However, it is still constrained for the reason that all those functions have a physical sense behind them: they are mapping a BP filter to a BP filter.

3.3.3 Frequency Warping Interpretation

A way to interpret the new frequency warping that we propose for our research is to analyze the "control points" of the mapping. By "control points", we mean the DC-frequency ($\omega = 0$), the Nyquist frequency ($\omega = \pi$) and the two cut-off frequencies $w_{p1} = \frac{\pi}{4}$ and $w_{p1} = \frac{3\pi}{4}$. The DC and Nyquist frequencies are mapped to themselves. The cut-off frequencies are mapped to new frequencies accordingly to Equation (3.43). For a sampling frequency $F_s = 8000\,\text{Hz}$, we saw earlier that the low cut-off frequency is located at $F_{p1} = 1000\,\text{Hz}$ and the high cut-off frequency is located at $F_{p2} = 3000\,\text{Hz}$. These two control points are located near the first and second formant region (for F_{p1}) and near the third formant region (for F_{p2})[18].

The transformations in Figure (4) and Figure (5) demonstrate the way these control points move when α and k vary. If we keep k constant and make α vary, the control points shift up or down the new frequency axis. As a consequence, the formants move up or down the frequency axis. If we keep α to a constant value and make k vary, the cutoff frequencies move closer or further together. The larger the value of k is, the closer together the cutoff frequencies will move; and so will the formants. The combination of this two transformations of the frequency axis can have interesting properties for VTLN.

This new frequency transformation offers two degrees of freedom allowing complex frequency axis transformations. It still needs to be proven that such frequency warping brings some improvement compared to other transformations used in VTLN. Once this is achieved, the issue of finding a parameter estimation procedure based on some criterion needs to be assessed. All those points are developed in the next chapter.

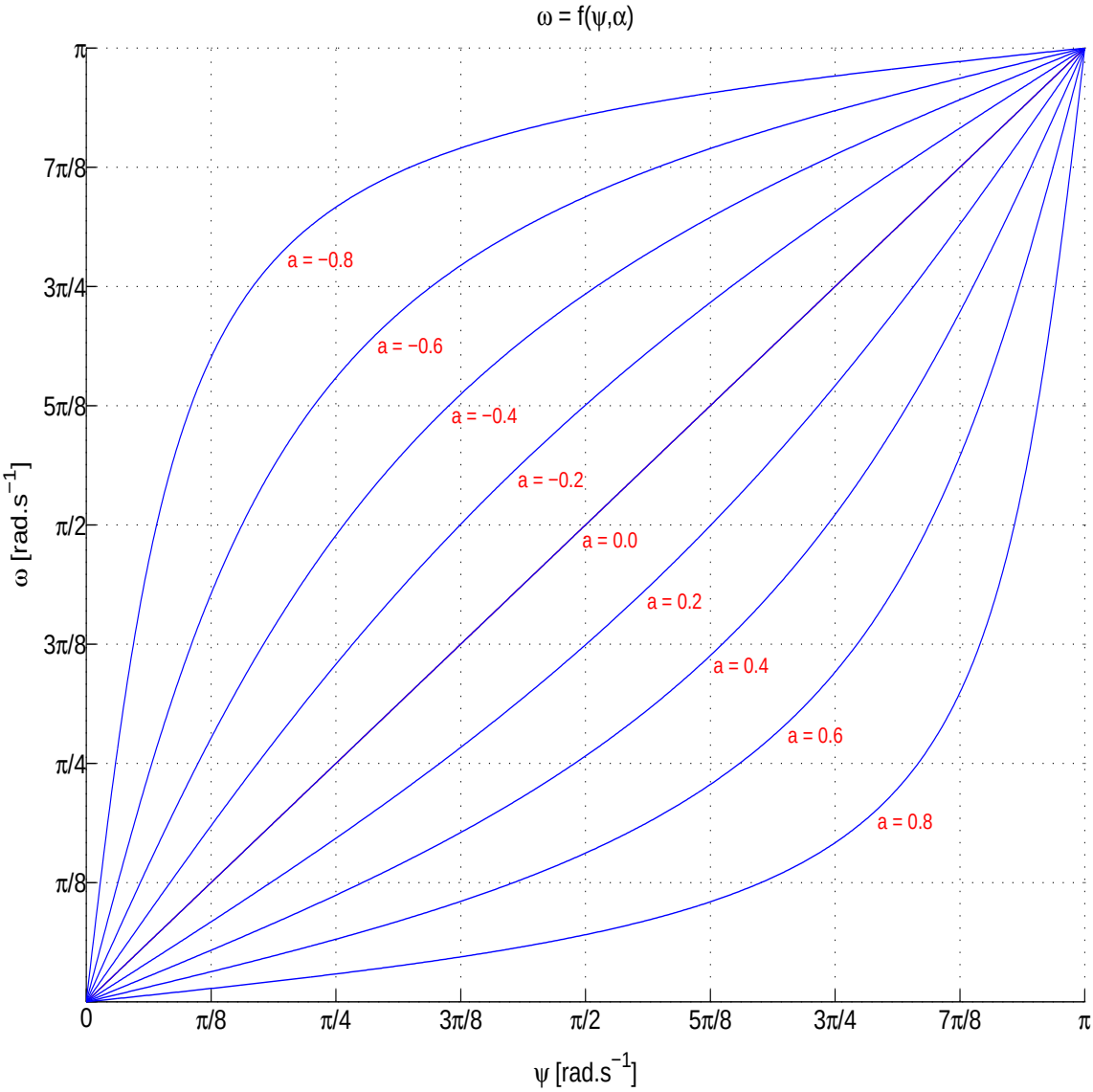

Figure 3: Frequency warping from a First Order Bilinear Transform. Real parameter a varies from -0.80 to 0.80 with 0.2 steps.

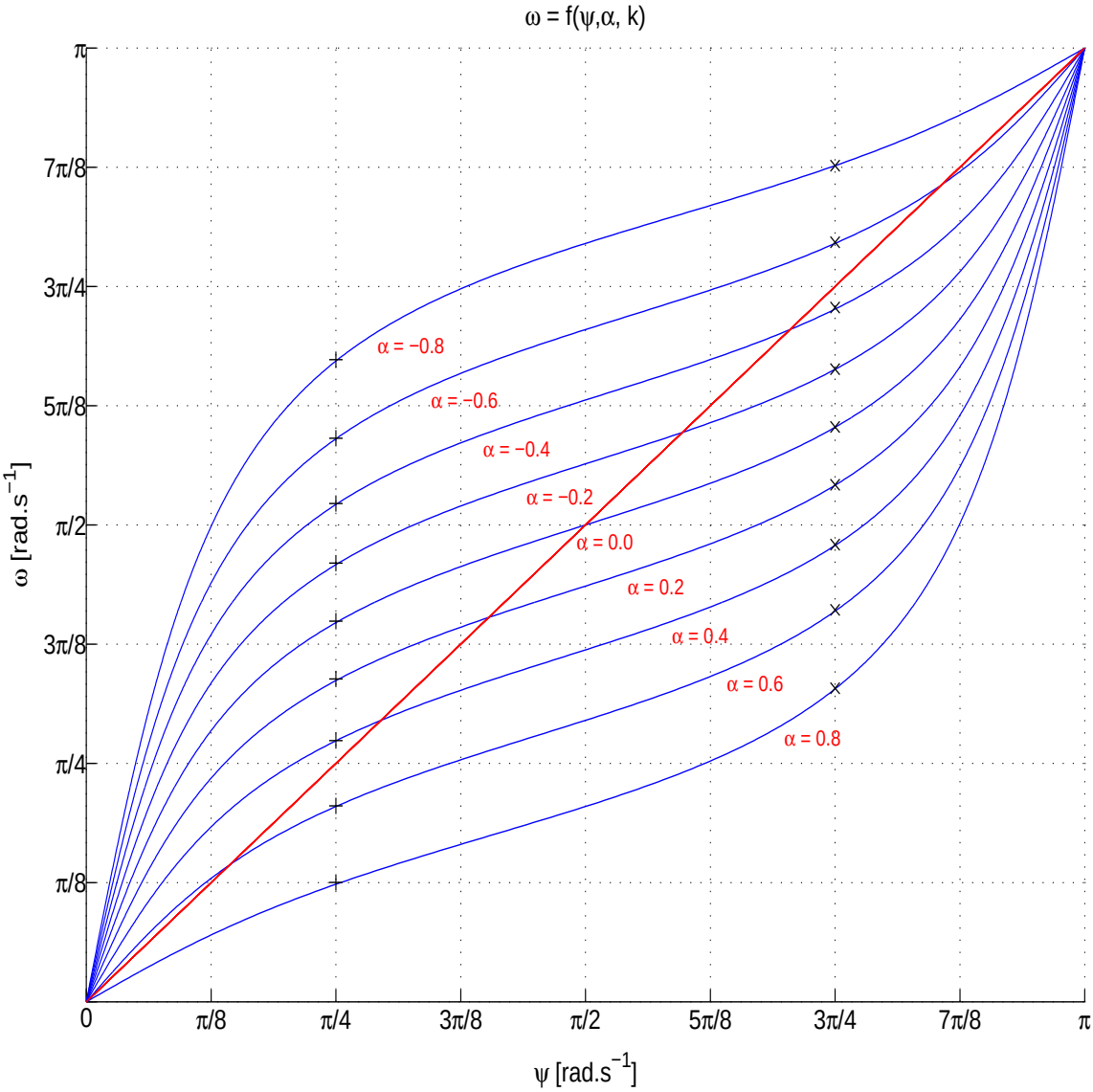

Figure 4: Frequency Warping for the Bandpass Transform for $k = 3$ and $\alpha = -0.8$ to 0.8 with a 0.2 step.

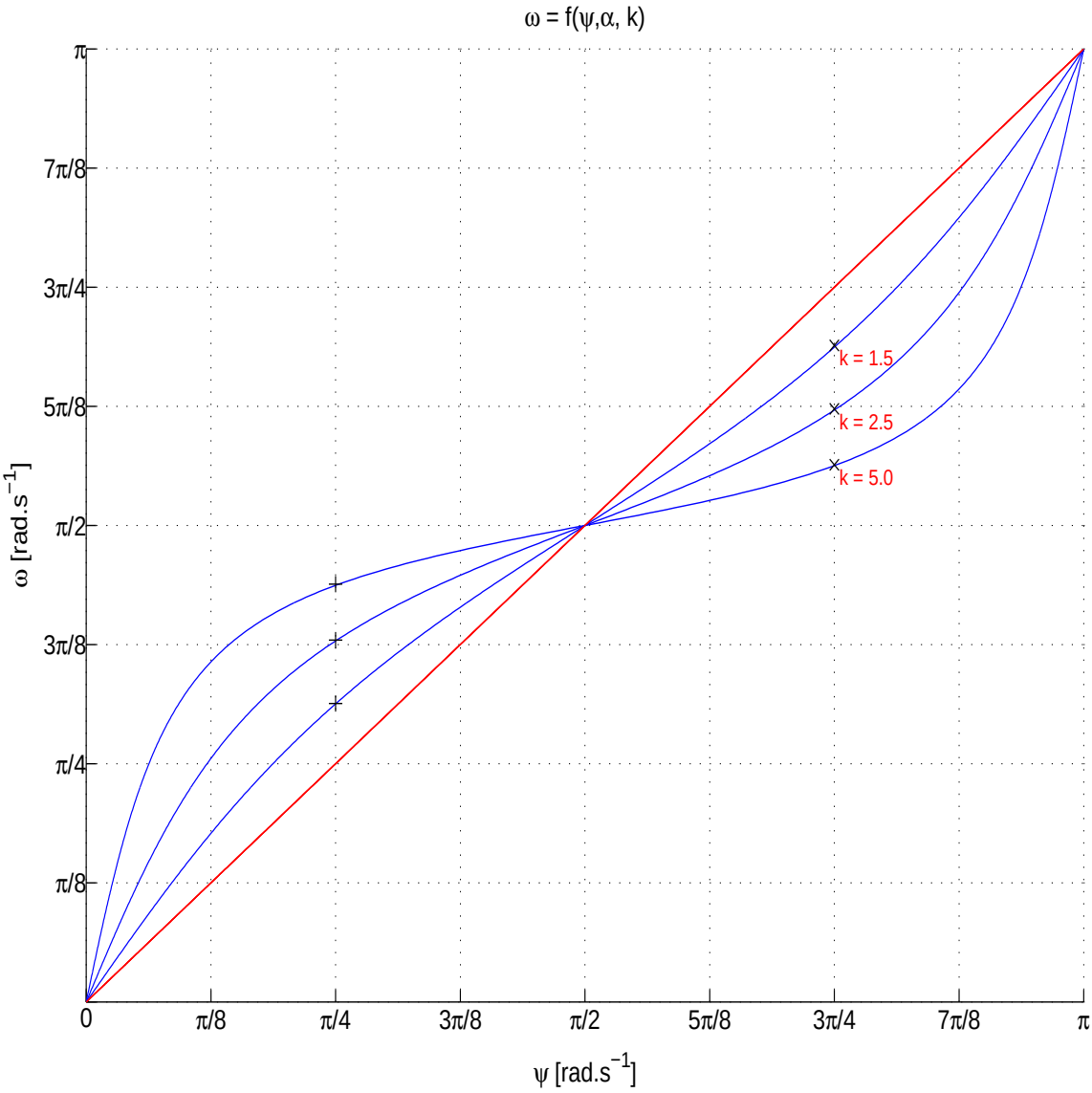

Figure 5: Frequency Warping for the Bandpass Transform for $\alpha = 0$ and $k = \{1.5, 2.5, 5.0\}$.

4.0 SYSTEM DESCRIPTION

In Chapter 3, the Band-Pass Transform was proposed as a candidate for Speaker Normalization and more precisely for VTLN. The validity of this new frequency warping for VTLN still needs to be established. Consequently, there is a need to define a criterion for deciding if this new transformation is an improvement to our ASR system. Only a transformation that improves our system's "performance" can be considered of interest to us. In addition to finding a performance measure, we also need to establish an experimental setup that proves that any observed improvement is due to the newly defined frequency warping.

4.1 PERFORMANCE MEASURE: WORD ERROR RATE

In ASR, it is commonly accepted to measure a system performance in terms of its *Word Error Rate* (WER)[24]. The WER is directly linked to the number of words that were not recognized correctly, which is ultimately the measure that we would like to minimize. Therefore, a modification in a component of an ASR system expected to be an "improvement" should be accompanied by a reduction in WER. The criterion to establish if the new frequency transformation performs better than other VTLN techniques is therefore defined: a significant reduction in WER[1].

A WER however has meaning only when it is associated with an experimental setup. It is, therefore, crucial to define an experimental setup that leaves no doubts (or more realisticly

[1]It is important to acknowledge the fact that the complete dynamic of the building elements of an ASR system is extremely complex. Therefore, an improvement of one of its components does not necessarily *always* bring a general reduction in its WER. However, a large and consistent WER reduction is very likely to come from a system improvement rather than some other noise in the system's dynamic. The whole art of ASR is to distinguish improvements from noise.

few doubts) that any observed improvement comes from modifications in the VTLN part of the system's Front End. In order to be able to appreciate the variations in WER, a reference system is needed. This reference system is called a *Baseline* system or Baseline. All our results will be compared against the Baseline system's WER as it represents the system performance prior to any modifications.

4.2 THE BYBLOS SYSTEM

As mentioned in Section 1.6, the ASR system used for this research is a modified version of the BYBLOS System created by BBN Technologies (BBN). BBN developed this system to participate to the 2001 Hub-5 Evaluation Benchmark[25] from the National Institute of Standards and Technology (NIST). The Hub-5 Evaluation Benchmark was organized by NIST to measure the current performance status of ASR on conversational telephone data. BYBLOS offers a solid software development backbone for research in ASR.

A general experimental setup for ASR research is decomposed in three distinct steps:

- The *Acoustic Training* step, also known as *Training*, is the first step to be run when an ASR system needs to be built. The goal of the Acoustic Training is to create Acoustic Models from observed speech data and its text transcription. From a dictionary, the system can decompose each word seen in the text transcription into a string of phonemes. Each phoneme is modeled with a 5-state HMM. A first step called *Labeling* is performed on the speech data. Labeling consists in associating each observed speech frame to a unique state in a phoneme's HMM. Once each speech frame has been associated to a specific HMM state, the main goal of Training is to update the HMM parameters (means and covariance matrices) for each phoneme so that the likelihood of the observed speech is increased. This is done by means of the *Expectation-Maximization* (EM) algorithm[2, 26]. The EM algorithm is run several times to update the HMMs' parameters. At each step, the likelihood of the observed speech data is increased making our Acoustic Models have a better fit to the observed data.
- A *Language Modeling* step often called *Word Modeling* is then conducted. A Language

Model is built by analyzing the construction of sentences in a series of documents. These documents can be newspaper articles, text transcription of programs broadcasted on radio or television, etc. The goal of Language Modeling is to capture the statistical patterns of the words sequences that exist for a particular language and come up with a statistical description of its grammar. A Language Model will provide the prior probabilities for small subsets of words such like bigrams (2 words), trigrams (3 words), etc.

- A *Decoding* step, also called *Testing*, is the final step of a full ASR system. The Decoding makes use of the Acoustic Models and Language Models previously created in order to transcribe speech data. The goal of a Decoding step is often to evaluate the performance of the system on transcribing test data whose transcription is already known. By doing this, a system's performance can be established before deployment on a real world task.

The three steps that have been described are the building stones of any ASR systems. The definition of each of these steps, especially for the Training and Language Modeling, will determine directly the performance of a system. The two sets of speech data used for Training and Testing are extremely important when building an ASR system. In ASR, a speech data set is referred to as *Corpus* (*Corpora* being its plural). A Corpus is a set of recorded conversations accompanied with a text transcription as well as various information about the speech that was produced. Such information includes the gender of the speakers, time stamps of all beginning and end of sentences, etc. Both Training and Testing sets can be an entire *Corpus* or a subset of it. When selecting sets for Training and Testing, it is crucial to have sets that are distinct to ensure the fairness of any results.

4.2.1 Training and Testing Corpora

The credibility of any experimental result is directly linked to the choice of Training and Testing data sets. This choice depends closely on the type of application the ASR system will work on. For this research, the goal is to improve the BYBLOS system's Speaker Normalization task on telephone data or Conversational Telephone Speech (CTS). In order to create the narrow-band American-English Baseline system, we need to define carefully the Training and Testing sets.

4.2.2 Training Set

Our Training is the Swbd40hrs set. It is a gender-balanced 40hours subset of the Switchboard training Corpus. The choice of this training set is partly motivated by the fact that it is of reasonable size, allowing to train completely a system "from scratch" in a matter of a day or so of computation. Another important point is that this 40hours subset of the Switchboard has been used at BBN for fast system training. The results in our work have the advantage of being comparable to recent research work conducted at BBN. The characteristics of this Training set are summarized in Table 1. In this Table, a *speaker* is defined as a conversation

Table 1: Statistics for the Swbd40hrs Training Corpus.

Swbd40hrs	Gender		
	Female	Male	Both
Number of Speakers	364	386	750
Number of Utterances	20,993	18,478	39,471
Total Time	19h 56min	20h 7min	40h 3min

side. The data for this Corpus is composed of telephone conversations between two speakers. Each conversation side corresponds to one speaker. The audio files are coded in stereo with each channel hosting one side of the conversation. The gender balance is in terms of speech duration with roughly 20hours of speech per gender.

4.2.3 Decoding Sets

For our experimental results, Decodings are performed on the two following Corpora:

- The Hub5.English.Dev01 Corpus (or Dev01 for short)
- The Hub5.English.Eval01 Corpus (or Eval01 for short)

Both Decoding sets (also called *test sets*) are from the 2001 Hub-5 Evaluation. The reason to present experimental results on two different Corpora is to ensure that they are not test set specific. The properties of various Decoding sets can differ significantly, The "difficulty"

of a test set is one of the observable differences. Indeed, the WER of a baseline system on one test set can be quite different from the baseline on another test set. Yet, the Decodings are performed using the same Acoustic and Language Models from the same Training. Some test sets have a baseline with a low WER because the speech is less noisy and closer in some sense to the Acoustic Models from the Training. It is therefore important to be sure that the performance gain observed on one set can be reproduced in the same proportions on another set before it can be considered an improvement. This is one way to ensure that results are less likely to be test-set specific[2]. The fact of having results on bigger test sets (more than 2hour long) is important as a WER decrease is often seen on a smaller decoding set but disappears when the amount of speech data increases.

The Dev01 test set is composed of 48 speakers coming from three different *conditions*: the original Switchboard Corpus (Swbd1), the Switchboard-2 Phase-3 Corpus (Swbd2) and the Switchboard-2 Phase-4 Corpus (Cellphone). The latter consists of conversations over a cellular phone channel. This evaluation set is particularly difficult because it includes conditions not seen in the Acoustic Training, particularly the cell phone channel. Table 2 gives the characteristics of this test set.

Table 2: Statistics for the Hub5.English.Dev01 Corpus.

Hub5.English.Dev01	Gender		
	Female	Male	Both
Number of Speakers	23	25	48
Number of Utterances	1123	1135	2258
Total Time	1h 13min	1h 18min	2h 31min

The Eval01 test set is composed of 120 speakers coming from the same three conditions as for Dev01. There is no overlap between the speakers of Eval01 and Dev01 Corpora. Table 3 gives the characteristics of this test set. It can be noticed that Eval01 is more than twice

[2]An example of a test-set specific improvement is the covariance normalization. It consists in normalizing the covariance matrix of our models so that it becomes a unity matrix, therefore "forcing" the Gaussians in our GMMs to become spheres (in N dimensions). This brings an improvement of 1.0% absolute WER for Dev01 when no improvement is observed on Eval01.

Table 3: Statistics for the Hub5.English.Eval01 Corpus.

Hub5.English.Eval01	Gender		
	Female	Male	Both
Number of Speakers	62	58	120
Number of Utterances	3079	2816	5895
Total Time	3h 11min	2h 59min	6h 10min

the size of Dev01. For each test set, we obtain an average WER by decoding each speaker individually and combining their WERs to get a global performance.

4.3 SPEECH FEATURES

The Front End of the system analyzes speech from waveform audio files and produces a sequence of cepstral feature vectors as presented in Chapter 2. These features known as *Mel-Frequency Cepstrum Coefficients* (MFCCs) are used both in Training to build the Acoustic Models and in Decoding as input features. Frequency Transformation of speech is performed in the *Analysis* part of the Byblos system. In order to implement the BPT, the regular Byblos Analysis tool needed to be modified.

4.3.1 Analysis Tool

The software that analyzes speech waveforms consists of one program called Tiny, acronym of "This Is Not Yaat" referring to the original Byblos program Yaat from which it departed. Tiny utilizes the skeleton of the former implementation but differs from the original Front End as the signal processing steps were entirely revisited and rewritten using newly defined functions. This new implementation is crucial to our work as it allows the use of any new frequency warping function and offers more signal processing options.

4.3.2 Analysis Parameters

The Analysis presented in detail in Chapter 2 corresponds to the sum of all the signal processing steps needed to obtain the cepstral coefficients that are the basis elements of speech feature vectors. Each one of the signal processing steps is controlled by parameters that need to be defined prior to running the Analysis. For instance, in our experimental results, speech was analyzed using a bandwidth of $[125\,\text{Hz}, 3750\,\text{Hz}]$ with a 25 ms frame duration at a rate of 100 frames per second. LPC smoothing is performed using 40 LPC coefficients allowing a rather low amount of smoothing. The FFT size is set to 256 points for each frame for a window length of 200 points. VTLN is performed on a log power spectrum using a simple first order interpolation. The output speech feature vector for each frame is a 15-dimensional vector composed of 14 cepstral coefficients with the energy. The features are normalized using CMS and variance normalization other the length of each conversation side.

4.4 ACOUSTIC AND LANGUAGE MODELS

Acoustic modeling is Speaker Independent (SI). A coarse *Phonetically Tied Mixture* (PTM) within-word triphone model and a finer *State Clustered Tied Mixture* (SCTM) between-word quinphone model are computed during Training. Those models are later adapted using *Speaker Adaptive Training* (SAT) based on *Linear Discriminant Analysis* (LDA). LDA starts with a 60-dimensional feature space and reduces it to a 39-dimensional feature space. LDA provides a 39 by 60 matrix that is used to transform the original features. The original 60 features are composed of a 15-dimensional vector described in the previous section as well as its first, second and third discrete derivatives. The Language Models (LMs) are trained on 3 millions words from the Switchboard and CallHome Corpus data and 140 million words from CNN Broadcast News data. Bigram and trigram LMs are trained to be used in different passes in the Decoding step.

5.0 EXPERIMENTAL SETUP

One of the goals of this work is to determine if a performance gain is achievable by using BPT in Speaker Normalization. First, a performance gain compared to a baseline system needs to be established. Second, this gain will be compared to performances from the standard VTLN method, the Eide VTL, used in the Byblos system. This is achieved by defining an experimental setup that allows a thorough study of the BPT's performances.

5.1 PARAMETER ESTIMATION PROCEDURE

One of the key parts of our experimental setup is the definition of the parameter estimation procedure to use for the BPT. The quality of the parameters choices will completely condition the BPT's performance.

5.1.1 Objective Function Definition

The BPT is controlled by the two parameters α and k. We need to define a procedure to estimate a set of parameters $\boldsymbol{\beta}_s = [\alpha_s, k_s]$ for each speaker s. As described in Chapter 1, the goal for ASR is to find the most likely word sequence $\hat{\boldsymbol{W}}$ that satisfies Equation (1.5) as it is

$$P(\hat{\boldsymbol{W}}|\boldsymbol{X}) = \max_{\boldsymbol{W}} P(\boldsymbol{W})P(\boldsymbol{X}|\boldsymbol{W})$$

where a Language Model associates a *prior* $P(\boldsymbol{W})$ to a word sequence $\boldsymbol{W}$. The probability of the acoustic information $\boldsymbol{X}$ given a specific word sequence $\boldsymbol{W}$ is given by $P(\boldsymbol{X}|\boldsymbol{W})$ which is evaluated from the Acoustic Models created in the Training. The BPT with parameter $\boldsymbol{\beta}_s$ transforms $\boldsymbol{X}$ into the sequence $\boldsymbol{X}^{\boldsymbol{\beta}_s}$. We see that the BPT has obviously no impact on

$P(\boldsymbol{W})$, but transforming the features $\boldsymbol{X}$ will change the probability $P(\boldsymbol{X}|\boldsymbol{W})$ as it becomes $P\big(\boldsymbol{X}^{\boldsymbol{\beta}_s}|\boldsymbol{W}\big)$. The parameter set $\boldsymbol{\beta}_s$ can be chosen to maximize the likelihood of the sequence of observed transformed features $\boldsymbol{X}^{\boldsymbol{\beta}_s}$ given an Acoustic Model $\boldsymbol{\Lambda}$ and the transcripts $\boldsymbol{W}_s$ of all the spoken utterances for speaker s [5].

$$\hat{\boldsymbol{\beta}}_s = \arg\max_{\boldsymbol{\beta}_s} P\big(\boldsymbol{X}^{\boldsymbol{\beta}_s}|\boldsymbol{\Lambda}, \boldsymbol{W}_s\big) \tag{5.1}$$

A closed-form solution is not easy to find for Equation (5.1) due to the fact that the frequency warping performs a non-linear transformation from $\boldsymbol{X}$ to $\boldsymbol{X}^{\boldsymbol{\beta}_s}$, as it has been mentioned in previous works on speaker normalization[5]. Any solution based on an optimization method requires numerical evaluations of the function

$$P\big(\boldsymbol{X}^{\boldsymbol{\beta}_s}|\boldsymbol{\Lambda}, \boldsymbol{W}_s\big) \tag{5.2}$$

in Equation (5.1).

5.1.2 Numerical Evaluation of the Objective Function

The evaluation of Equation (5.2) is done in two distinct steps in the Byblos system. The first step is a forward-backward decoding that provides a n-best list consisting of the 100 best transcript hypotheses for the observed acoustic sequence $\boldsymbol{X}^{\boldsymbol{\beta}_s}$. This forward-backward decoding is based on the forward-backward search algorithm[27] that makes use of the Viterbi algorithm[26]. This search algorithm finds the most likely sequence of HMM states for an observed acoustic features sequence given a PTM Acoustic Model. In the second step, the n-best list is "rescored" with a more detailed between-word SCTM Acoustic Model as described in Section 4.4. This step provides an Acoustic and a Language Model score for each of the 100 hypotheses.

This procedure is known as a 2-pass strategy[28] for decoding. It provides a best hypothesis for each utterance which is the one hypothesis from the n-best list that has the maximum *global* likelihood score. The global likelihood score is a weighted sum of the logarithm of the score from the Language Model and the score from the Acoustic Model. The weights are pre-determined and often chosen to be the ones from the baseline system where they were

optimized to give the best overall WER. In our case, only the acoustic score will be used as we perform a transformation solely on the acoustic features.

The most time consuming task in the procedure is the creation of the n-best list by the forward-backward search algorithm. For a large set of speakers, this task can become computationally overwhelming, especially if the estimation procedure for the BPT parameters requires several evaluations of our objective function in order to converge to an estimate. One way to decrease the computation time is to get the n-best lists from a previous experiment which can be our baseline experiment. In this case, only a rescoring of the n-best lists (referred to as *"n-best rescoring"*) is performed. Using the n-best list of a previous baseline decoding significantly reduces our computation time especially if we need a large number of evaluation of our objective function.

Consequently, the n-best rescoring is bounded by the 100 hypotheses provided by our baseline decoding. This a strong constraint since we then impose on our system to chose only between the 100 hypotheses from the baseline decoding which may be far from the "true" transcription. However, the goal here is not to provide the best transcription of the observed speech yet, but to provide a parameter set β_s that increases the likelihood of the transformed speech features given the Acoustic Models from the Training. In this regard, our parameter estimation procedure can be viewed as a black box that will associate a parameter set β to each speaker. Once the best parameter sets have been defined, a "regular" full-decoding is performed from scratch using the transformed acoustic features. This decoding will generate a completely new n-best list and perform a rescoring on this n-best list to give the best transcription possible.

A more radical step to use if the speed up is not sufficient involves getting the best hypothesis from the baseline decoding. Then, only the best hypothesis for each utterance is rescored (rather than the 100 in the n-best list), speeding the rescoring pass by a factor of 100. This referred to as a *"1-best rescoring"*. This setup is very close to the one used for *Maximum Likelihood Linear Regression* (MLLR) and has been recently used in the context of Speaker Normalization[29]. For MLLR, the goal is to find a linear transformation of the Acoustic Models parameters that increases the likelihood of the observed speech. Both n-best and 1-best rescoring techniques are used in our experimental results.

It is necessary to further define the parameter estimation procedure that will use numerical evaluations of the previously defined objective function.

5.1.3 Nelder-Mead Optimization Procedure

For each speaker in a test set, the BPT parameter estimation procedure finds the best set $\boldsymbol{\beta}_s$ that maximizes the objective function in Equation (5.2). This is at the condition that our objective function displays only few local maxima or, ideally, only one global maxima. For our work, the *Nelder-Mead* (NM) unconstrained optimization method was utilized to find the parameter set $\boldsymbol{\beta}_s$ that maximizes our objective function. Several reasons motivated our choice. First, we do not want to restrict the range of the parameters values, making an unconstrained method a natural choice. Second, since the bottleneck of our procedure is the evaluation of our objective function, an algorithm based on evaluation of first and/or second derivatives is not an option. Indeed, deriving a closed form for the first and second derivatives of our objective function is a difficult task. The BPT performs a frequency transformation that corresponds to a transformation that is highly non-linear on the speech features themselves [5]. Therefore, only the computation of the discrete first and second derivatives is possible. This requires several numerical evaluations of the objective function which will be computationally expensive.

Several methods, such as the Newton method, try to fit the objective function curve with a quadratic curve to find the location of its maximum. By repeating this approximation, this method attempts to converge to a solution. The NM algorithm does not require any derivatives and does not make any assumption about the objective function. Those two properties make the NM algorithm interesting for our application and were the main motivations for selecting it. The convergence properties of the NM algorithm have been subject to thorough analysis[30]. We keep in mind that, like other optimization methods, convergence to a solution can mean convergence to a local maxima. Even though the NM algorithm is usually used to find a function minimum but can be easily modified for the search of a maximum with a few changes in the algorithm's implementation.

The NM algorithm is based on updating an initial *simplex*. In the 2-dimensional case,

a simplex is composed of 3 points evaluated at different values of $\boldsymbol{\beta}_s$. After each algorithm iteration, the simplex point with the worst score is replaced by a point of better score. In the special case where after few function evaluations no better point is found, the algorithm "shrinks" the simplex, keeping only the best score point. The algorithm first tries to find better points based on the assumption that a point geometrically further from the worst point could be an improvement, but modifies its strategy if it is not the case. As a consequence, for a 2-dimensional search, each iteration consists of 1 to 4 function evaluations. After several iterations, it is expected that the simplex would move towards a region with a function's maximum. For our experimental results, we chose the initial simplex to be evaluated at $[0.0, 1.0]$ (no transformation), and $[-0.80, 0.80]$ and $[-0.80, 1.20]$ which correspond to two cases of extreme frequency warping. The algorithm will stop after 50 function evaluations which is approximately 20 iterations. It was demonstrated that the NM algorithm gives valid estimations of the BPT parameters in a previous work[31].

5.1.4 Parameter Estimation Procedure Performance

Before using our NM-based parameter estimation procedure, we need to establish the validity of the estimation of $\boldsymbol{\beta}_s$. It is important to define a reference to which we can compare the parameter selected by our procedure. We opted in this case for the solution of *Oracle* experiments. An Oracle experiment (or simply Oracle) is a grid search of the parameter value that minimizes the system's Word Error Rate (WER). The advantage of an Oracle is to provide the *best* WER attainable for the parameter values *on the grid*. However, since the WER is the objective function, the main disadvantages reside in having to run a decoding for each parameter value and to know the true transcription for the observed speech which makes an Oracle impractical in many cases. For the BPT VTLN, the estimation of two SD parameters is required which can make a grid search problematic in terms of computational needs. While an Oracle on all test speakers is not imaginable, we can reduce the number of speakers dramatically to retain the Oracle solution. This will allow us to investigate thoroughly the BPT properties by allowing a fine sampling for our grid search. More importantly, it will also give a target value for $\boldsymbol{\beta}_s$ and a WER performance to attain

when we estimate our parameter with the NM algorithm. In this case, results from Oracle experiments inform us on what to expect in term of possible performance for the BPT.

Two female speakers SW04537A and SW20316A from the Dev01 set are selected based on several criteria such as number of uttered words, improvement from VTLN methods, etc. Oracle experiments were run for the BPT with a fine grid search consisting of 41 values for α on the $[-0.20, 0.20]$ interval and 41 values for k in the $[0.80, 1.20]$ interval which results in $N_p = 1681$ possible (α, k) pairs.

The WERs for both speakers for all VTLN methods available are presented in Table 4. N_p indicates the number of points in the grid search. In terms of WER, for both speakers, the

Table 4: WERs for all available VTLN methods.

VTLN Method	SW04537A		SW20316A		N_p
	Errors	WER %	Errors	WER %	
no VTLN	162	30.62	232	50.00	1
Eide	128	24.20	221	47.63	41
1st BLT	131	24.76	219	47.20	41
BPT	115	21.74	216	46.55	1681
Reference	529 words		464 words		

BPT offers the best performances. For SW04537A, with a 30.62% baseline, an improvement of a 8.88% absolute gain can be achieved, 2.27% absolute better than the Linear VTLN, second best. Speaker SW20316A, with a 50.0% baseline, sees an improvement of 3.45% absolute accomplished by BPT, 0.65% absolute better than second best 1st-BLT. Table 5 presents the results for using a Maximum Likelihood (ML) approach to find $\boldsymbol{\beta}_s$ for the BPT alone. The N_e column indicates the number of likelihood function evaluations. The results on the "Oracle Grid" row are for a ML selection for $\boldsymbol{\beta}_s$ using the same grid as for the Oracle experiment. These results are our reference for ML selection of $\boldsymbol{\beta}_s$. The following rows present results when the NM approach is utilized. Interestingly, the NM methods (n-best and 1-best rescoring) offer both encouraging results for SW04537A with only a slight increase in WER. For SW20136A, the case is a bit different as the NM n-best offers an improvement

Table 5: Results for ML selection of the BPT parameters.

ML Method	SW04537A		SW20316A		N_e
	Errors	WER %	Errors	WER %	
Oracle Grid	121	22.87	223	48.06	1681
NM n-best	123	23.25	228	49.14	50
NM 1-best	123	23.25	219	47.20	50

compared to the Oracle Grid results. In order to better understand these results, the ML values selected for α_s and k_s are shown in Table 6. From Table 6, it is clear that NM

Table 6: Parameters selection for both speakers.

Selection Method	SW04537A		SW20316A		N_e
	α_s	k_s	α_s	k_s	
Oracle	0.17	1.17	-0.05	1.0	1681
Oracle Grid	0.14	1.20	-0.05	1.01	1681
NM n-best	0.1340	1.2235	-0.0597	0.9904	50
NM 1-best	0.1341	1.2235	-0.0506	1.0058	50

algorithm selected values are quite close to the Oracle Grid values for α_s and k_s. The NM algorithm gives a reasonable selection for parameters α_s and k_s.

5.2 SCORE COMPENSATION

For each speaker, the parameter estimation procedure finds estimates of the BPT parameters that maximize the likelihood of the observations for a given word sequence. This estimation procedure uses the objective function defined in Section 5.1.1 which is the likelihood of our transformed data given our Acoustic Models. Therefore, the goal of the Speaker Normalization procedure is to increase the likelihood of the observed features by transforming them using the BPT.

As discussed in Chapter 1, the Acoustic Models used in current ASR systems are based on HMMs. Each phoneme is modeled by a 5-state HMM and, for each state, the observation is modeled by a GMM. The observation probability of a feature vector $\boldsymbol{x}_t$ for a state s at time t was given by Equation (1.1) that was described as

$$P^s(\boldsymbol{x}_t; \boldsymbol{\Gamma}^s) = \sum_{k=0}^{K-1} w_k^s \mathcal{N}^s(\boldsymbol{x}_t; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$$

where $\mathcal{N}^s(\boldsymbol{x}; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$ is the k^{th} multivariate Gaussian distribution consisting of mean vector $\boldsymbol{\mu}_k^s$ and covariance matrix $\boldsymbol{\Sigma}_k^s$. w_k^s is a normalized weight associated to the k^{th} Gaussian for state s and $\boldsymbol{\Gamma}^s$ is the set of all the model parameters (mean vectors $\boldsymbol{\mu}_k^s$ and covariance matrices $\boldsymbol{\Sigma}_k^s$) for the GMM at state s. The observation probability for each Gaussian for the state s is defined by

$$\mathcal{N}^s(\boldsymbol{x}; \boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_k^s|^{\frac{1}{2}}} \quad \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k^s)^T {\boldsymbol{\Sigma}_k^s}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k^s)}.$$

In order to simplify our equations, we consider the case of an observation modeled by a GMM with only one Gaussian ($K=1$) for each state s. To simplify our notation, references to the state s of the HMM and to the time t a vector is observed are removed. The likelihood of the observed vector $\boldsymbol{x}$ in our simplified notation is

$$\mathcal{L}(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \quad \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \tag{5.3}$$

for $K=1$. Taking the logarithm of the likelihood gives the log-likelihood of $\boldsymbol{x}$ that can be derived easily from Equation (5.3) as

$$\mathcal{LL}(\boldsymbol{x}) = \underbrace{-\frac{n}{2} \log(2\pi)}_{\text{cte}} - \frac{1}{2} \log(|\boldsymbol{\Sigma}|) - \frac{1}{2} (\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}). \tag{5.4}$$

The BPT provides transformed features $\tilde{\boldsymbol{x}}$ such that $\tilde{\boldsymbol{x}} = f_{\boldsymbol{\beta}_s}(\boldsymbol{x})$ where the function $f_{\boldsymbol{\beta}_s}$ corresponds to the transformation of the speech features resulting from the BPT with parameter $\boldsymbol{\beta}_s$. This function is non-linear and assumed to be invertible. The BPT, with

its frequency warping properties, substitutes each feature vector $\boldsymbol{x}$ with its transformed counterpart $\tilde{\boldsymbol{x}}$. The Log-Likelihood in Equation (5.4) after BPT is applied becomes

$$\begin{aligned}\mathcal{LL}(\tilde{\boldsymbol{x}}) &= \mathcal{LL}\left(f_{\boldsymbol{\beta}_s}(\boldsymbol{x})\right) \\ &= \text{cte} - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(f_{\boldsymbol{\beta}_s}(\boldsymbol{x}) - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(f_{\boldsymbol{\beta}_s}(\boldsymbol{x}) - \boldsymbol{\mu}\right)\end{aligned} \tag{5.5}$$

where the constant term called "cte" is the one defined in Equation (5.4).

Now consider the case where the function $f_{\boldsymbol{\beta}_s}(\boldsymbol{x})$ can be approximated by a linear transformation such that $\tilde{\boldsymbol{x}} = f_{\boldsymbol{\beta}_s}(\boldsymbol{x}) = \boldsymbol{A}_{\boldsymbol{\beta}_s}\boldsymbol{x}$. The matrix $\boldsymbol{A}_{\boldsymbol{\beta}_s}$ (or $\boldsymbol{A}$ for short) depends directly on $\boldsymbol{\beta}_s$. The Equation (5.5) then becomes

$$\begin{aligned}\mathcal{LL}(\tilde{\boldsymbol{x}}) &= \mathcal{LL}(\boldsymbol{A}\boldsymbol{x}) \\ &= \text{cte} - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\mu}\right).\end{aligned} \tag{5.6}$$

Equation (5.5) and Equation (5.6) show the transformation of the features in order to increase their likelihood score given the observation model.

Another valid strategy is to modify the Acoustic Models to better account for the observed data. This strategy, called Model Adaptation, has been extensively studied. Its solutions have been mathematically formulated for the case of linear transformations[3, 32] and it is used extensively in ASR. One particular type of model adaptation is called *Constrained Adaptation* (CA). The reason it is called "constrained" is because the linear transformation applied on the means $\boldsymbol{\mu}$ and the covariances $\boldsymbol{\Sigma}$ of the GMMs is the same linear transformation[3]. If we consider the invertible linear transformation $\boldsymbol{B}$ applied on our models parameter, the new models parameters are expressed as

$$\begin{aligned}\boldsymbol{\mu} &\Rightarrow \boldsymbol{B}\boldsymbol{\mu} = \tilde{\boldsymbol{\mu}} \\ \boldsymbol{\Sigma} &\Rightarrow \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T = \tilde{\boldsymbol{\Sigma}}\end{aligned}$$

The log-likelihood of the observed speech feature $\boldsymbol{x}$ given the transformed models is

$$\begin{aligned}
\mathcal{LL}_{\text{CA}}(\boldsymbol{x}) &= \text{cte} - \frac{1}{2}\log\left(|\tilde{\boldsymbol{\Sigma}}|\right) - \frac{1}{2}\left(\boldsymbol{x}-\tilde{\boldsymbol{\mu}}\right)^T \tilde{\boldsymbol{\Sigma}}^{-1}\left(\boldsymbol{x}-\tilde{\boldsymbol{\mu}}\right) \\
&\Leftrightarrow \quad \text{cte} - \frac{1}{2}\log\left(|\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T|\right) - \frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{B}\boldsymbol{\mu}\right)^T \left(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T\right)^{-1}\left(\boldsymbol{x}-\boldsymbol{B}\boldsymbol{\mu}\right) \\
&\Leftrightarrow \quad \text{cte} - \frac{1}{2}\log\left(|\boldsymbol{B}||\boldsymbol{\Sigma}||\boldsymbol{B}^T|\right) - \frac{1}{2}\left(\boldsymbol{B}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)\right)^T \left(\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T\right)^{-1}\left(\boldsymbol{B}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)\right) \\
&\Leftrightarrow \quad \text{cte} - \frac{1}{2}\log\left(|\boldsymbol{B}|^2|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)^T \underbrace{\boldsymbol{B}^T\left(\boldsymbol{B}^T\right)^{-1}}_{\boldsymbol{I}} \boldsymbol{\Sigma}^{-1} \underbrace{\boldsymbol{B}^{-1}\boldsymbol{B}}_{\boldsymbol{I}} \left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right) \\
&\Leftrightarrow \quad \text{cte} + \log\left(|\boldsymbol{B}|^{-1}\right) - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right) \\
&\Leftrightarrow \quad \text{cte} + \log\left(|\boldsymbol{B}^{-1}|\right) - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{B}^{-1}\boldsymbol{x}-\boldsymbol{\mu}\right)
\end{aligned}$$

since $\det(\boldsymbol{B})^{-1} = \det(\boldsymbol{B}^{-1})$. If we define $\boldsymbol{A} = \boldsymbol{B}^{-1}$ then

$$\mathcal{LL}_{\text{CA}}(\boldsymbol{x}) = \text{cte} + \log\left(|\boldsymbol{A}|\right) - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{\mu}\right). \tag{5.7}$$

Interestingly, one can notice that there is a strong resemblance between Equation (5.6) and Equation (5.7). In fact for a feature vector $\boldsymbol{x}$, we can write $\mathcal{LL}_{\text{CA}}(\boldsymbol{x})$ as a function of $\boldsymbol{A}$ and $\mathcal{LL}(\boldsymbol{A}\boldsymbol{x})$. From Equation (5.7) we know that

$$\begin{aligned}
\mathcal{LL}_{\text{CA}}(\boldsymbol{x}) &= \text{cte} + \log\left(|\boldsymbol{A}|\right) - \frac{1}{2}\log\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}\left(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{\mu}\right) \\
&= \log\left(|\boldsymbol{A}|\right) + \mathcal{LL}(\boldsymbol{A}\boldsymbol{x}).
\end{aligned} \tag{5.8}$$

This means that the linear transformation $\boldsymbol{A}$ performed on the feature vector $\boldsymbol{x}$ corresponds to a constrained adaptation of the Acoustic Models if the term $\log\left(|\boldsymbol{A}|\right)$ is added to the log-likelihood score. This is equivalent to a scaling of the likelihood score by $|\boldsymbol{A}|$.

The term $\log\left(|\boldsymbol{A}|\right)$ is viewed here as a compensation factor. By adding the term $\log\left(|\boldsymbol{A}|\right)$ to the log-likelihood score $\mathcal{LL}(\boldsymbol{A}\boldsymbol{x})$, a score compensation is performed so that the overall transformation on the speech features correspond to a constrained adaptation of the Acoustic Models using the linear transformation $\boldsymbol{A}$. The problem is that the transformation on the speech features performed by the BPT is expressed as $\tilde{\boldsymbol{x}} = f_{\boldsymbol{\beta}_s}(\boldsymbol{x})$ and not $\tilde{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$. We need to approximate the function $f_{\boldsymbol{\beta}_s}$ by a linear transformation.

5.2.1 Mean Square Error Matrix

The transformation performed by the BPT on a speech feature vector has the following expression:

$$\tilde{\boldsymbol{x}}_t = \boldsymbol{y}_t = f_{\boldsymbol{\beta}_s}(\boldsymbol{x}_t) \quad \forall t. \tag{5.9}$$

If we assume that Equation (5.9) can be expressed as a linear transformation, we can rewrite it as

$$\tilde{\boldsymbol{x}}_t = \boldsymbol{y}_t = \boldsymbol{F}_{\boldsymbol{\beta}_s}\boldsymbol{x}_t \quad \forall t. \tag{5.10}$$

$\boldsymbol{F}_{\boldsymbol{\beta}_s}$, or $\boldsymbol{F}$ for shorter notation, can be estimated by $\hat{\boldsymbol{F}}$ such that

$$\hat{\boldsymbol{y}}_t = \hat{\boldsymbol{F}}\boldsymbol{x}_t \quad \forall t. \tag{5.11}$$

From Equation (5.10), we can derive the resulting estimation error $\boldsymbol{e}$ and its norm.

$$\begin{aligned} \boldsymbol{e}_t &= \hat{\boldsymbol{y}}_t - \boldsymbol{y}_t \quad \forall t \\ &= \hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t \quad \forall t \\ \Leftrightarrow \|\boldsymbol{e}_t\| &= \|\hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t\| \quad \forall t \end{aligned} \tag{5.12}$$

The Mean Square Error (MSE) for all speech features is

$$\begin{aligned} \frac{1}{T}\sum_t \|\boldsymbol{e}_t\|^2 &= \frac{1}{T}\sum_t \|\hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t\|^2 \quad \forall t \\ &= \frac{1}{T}\sum_t \left(\hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t\right)^T \left(\hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t\right) \quad \forall t, \quad \text{since } \|\boldsymbol{x}_t\|^2 = \boldsymbol{x}_t^T\boldsymbol{x}_t \\ &= \frac{1}{T}\sum_t \left(\boldsymbol{x}_t^T\hat{\boldsymbol{F}}^T - \boldsymbol{y}_t^T\right)\left(\hat{\boldsymbol{F}}\boldsymbol{x}_t - \boldsymbol{y}_t\right) \quad \forall t. \end{aligned} \tag{5.13}$$

In order to simplify our notation even further, the subscript t will be omitted for the rest of this section such that $\boldsymbol{x}$ will in fact represent $\boldsymbol{x}_t$. Therefore, we can rewrite Equation (5.13) as

$$\begin{aligned} \frac{1}{T}\sum_t \|\boldsymbol{e}\|^2 &= \frac{1}{T}\sum_t \left(\boldsymbol{x}^T\hat{\boldsymbol{F}}^T - \boldsymbol{y}^T\right)\left(\hat{\boldsymbol{F}}\boldsymbol{x} - \boldsymbol{y}\right) \\ &= \frac{1}{T}\sum_t \Big(\underbrace{\boldsymbol{x}^T\hat{\boldsymbol{F}}^T\hat{\boldsymbol{F}}\boldsymbol{x}}_{\mathcal{A}} + \boldsymbol{y}^T\boldsymbol{y} - \underbrace{\boldsymbol{x}^T\hat{\boldsymbol{F}}^T\boldsymbol{y}}_{\mathcal{B}} - \boldsymbol{y}^T\hat{\boldsymbol{F}}\boldsymbol{x}\Big) \end{aligned} \tag{5.14}$$

where the terms $\mathcal{A}$ and $\mathcal{B}$ can be further simplified.

- $\mathcal{A} = \boldsymbol{x}^T \hat{\boldsymbol{F}}^T \hat{\boldsymbol{F}} \boldsymbol{x}$. For two vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and a matrix $\boldsymbol{A}$, the expression $\boldsymbol{\alpha}^T \boldsymbol{A} \boldsymbol{\beta}$ is equal to $\text{tr}\{\boldsymbol{A}\boldsymbol{\beta}\boldsymbol{\alpha}^T\}$ where $\text{tr}\{\boldsymbol{A}\}$ is the trace of $\boldsymbol{A}$[33]. Therefore the term $\mathcal{B}$ can be rewritten as $\mathcal{B} = \text{tr}\left\{\hat{\boldsymbol{F}}^T \hat{\boldsymbol{F}} \boldsymbol{x}\boldsymbol{x}^T\right\}$.
- $\mathcal{B} = \boldsymbol{x}^T \hat{\boldsymbol{F}}^T \boldsymbol{y}$. $\boldsymbol{\alpha}^T \boldsymbol{\beta}$ is equal to $\boldsymbol{\beta}^T \boldsymbol{\alpha}$. Therefore, the term $\mathcal{B}$ can be rewritten as $\mathcal{B} = \boldsymbol{y}^T \hat{\boldsymbol{F}} \boldsymbol{x}$.

The two terms $\mathcal{A}$ and $\mathcal{B}$ have been simplified and Equation (5.14) can be rewritten into

$$\begin{aligned}
\frac{1}{T}\sum_t \|\boldsymbol{e}\|^2 &= \frac{1}{T}\sum_t \left[\text{tr}\left\{\hat{\boldsymbol{F}}^T \hat{\boldsymbol{F}} \boldsymbol{x}\boldsymbol{x}^T\right\} + \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T \hat{\boldsymbol{F}} \boldsymbol{x}\right] \\
&= \frac{1}{T}\sum_t \left[\text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{x}\boldsymbol{x}^T \hat{\boldsymbol{F}}^T\right\} + \boldsymbol{y}^T\boldsymbol{y} - 2\text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{x}\boldsymbol{y}^T\right\}\right] \quad \text{since } \text{tr}\{\boldsymbol{AB}\} = \text{tr}\{\boldsymbol{BA}\} \\
&= \text{tr}\left\{\hat{\boldsymbol{F}} \underbrace{\frac{1}{T}\sum_t \boldsymbol{x}\boldsymbol{x}^T}_{\boldsymbol{R}_{\boldsymbol{xx}}} \hat{\boldsymbol{F}}^T\right\} + \frac{1}{T}\sum_t \boldsymbol{y}^T\boldsymbol{y} - \frac{2}{T}\sum_t \text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{x}\boldsymbol{y}^T\right\} \\
&= \text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{R}_{\boldsymbol{xx}} \hat{\boldsymbol{F}}^T\right\} + \frac{1}{T}\sum_t \boldsymbol{y}^T\boldsymbol{y} - 2\text{tr}\left\{\hat{\boldsymbol{F}} \underbrace{\frac{1}{T}\sum_t \boldsymbol{x}\boldsymbol{y}^T}_{\boldsymbol{C}_{\boldsymbol{xy}}}\right\} \\
\frac{1}{T}\sum_t \|\boldsymbol{e}\|^2 &= \underbrace{\text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{R}_{\boldsymbol{xx}} \hat{\boldsymbol{F}}^T\right\}}_{\mathcal{C}} + \underbrace{\frac{1}{T}\sum_t \boldsymbol{y}^T\boldsymbol{y}}_{\mathcal{D}} - 2\underbrace{\text{tr}\left\{\hat{\boldsymbol{F}} \boldsymbol{C}_{\boldsymbol{xy}}\right\}}_{\mathcal{E}}
\end{aligned} \tag{5.15}$$

where $\boldsymbol{R}_{\boldsymbol{xx}}$ is the auto-correlation of $\boldsymbol{x}$ and $\boldsymbol{C}_{xy}$ is the cross-correlation between $\boldsymbol{x}$ and $\boldsymbol{y}$. The three terms $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$ will be used further in this section.

In order to find the matrix $\hat{\boldsymbol{F}}$ that best approximates our feature transformation, we need to find $\hat{\boldsymbol{F}}$ that minimizes Equation (5.15). In other words, if

$$\mathcal{Q} = \frac{1}{T}\sum_t \|\boldsymbol{e}\|^2,$$

then we need to find $\hat{\boldsymbol{F}}$ such that

$$\frac{\text{d}\mathcal{Q}}{\text{d}\hat{\boldsymbol{F}}} = 0. \tag{5.16}$$

The matrix $\hat{\boldsymbol{F}}$ is the matrix that minimizes the Mean Square Error of the transformation. In Equation (5.15), the RHS of the equation is composed of the three terms $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{E}$. Finding their derivatives with respect to $\hat{\boldsymbol{F}}$ will automatically give a solution to Equation (5.16).

- For $\mathcal{C}$, we need to use the fact that for two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{A}^T}\mathrm{tr}\left\{\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A}\right\} = \boldsymbol{A}^T\left\{\boldsymbol{B}+\boldsymbol{B}^T\right\}$[33] . In our case, $\hat{\boldsymbol{F}}=\boldsymbol{A}$ and $\boldsymbol{R_{xx}}=\boldsymbol{B}$. Therefore, the derivative for $\mathcal{C}$ is $\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\hat{\boldsymbol{F}}}=\hat{\boldsymbol{F}}\left(\boldsymbol{R_{xx}}+\boldsymbol{R_{xx}}^T\right)$.
- For $\mathcal{D}$, the solution is straightforward as $\frac{\mathrm{d}\mathcal{D}}{\mathrm{d}\hat{\boldsymbol{F}}}=0$.
- For $\mathcal{E}$, we need to use the fact that for the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{A}}\mathrm{tr}\left\{\boldsymbol{A}\boldsymbol{B}^T\right\}=\boldsymbol{B}$[33]. In our case, $\hat{\boldsymbol{F}}=\boldsymbol{A}$ and $\boldsymbol{B}^T=\boldsymbol{C_{xy}}$. For the derivative of $\mathcal{E}$ is $\frac{\mathrm{d}\mathcal{E}}{\mathrm{d}\hat{\boldsymbol{F}}}=\boldsymbol{C_{xy}}^T$.

In the light of these new results, the derivative of Equation (5.15) with respect to $\hat{\boldsymbol{F}}$ under the condition set by Equation (5.16) can be written as

$$\begin{aligned}\frac{\mathrm{d}\mathcal{Q}}{\mathrm{d}\hat{\boldsymbol{F}}} &= \hat{\boldsymbol{F}}\left(\boldsymbol{R_{xx}}+\boldsymbol{R_{xx}}^T\right)+0-2\boldsymbol{C_{xy}}^T=0 \\ &\Leftrightarrow\ 2\hat{\boldsymbol{F}}\boldsymbol{R_{xx}}-2\boldsymbol{C_{xy}}^T=0, \quad \text{since } \boldsymbol{R_{xx}}^T=\boldsymbol{R_{xx}} \\ &\Rightarrow\ \hat{\boldsymbol{F}}=\boldsymbol{C_{xy}}^T\boldsymbol{R_{xx}}^{-1}\end{aligned} \tag{5.17}$$

$\hat{\boldsymbol{F}}$ is the MSE estimate of $\boldsymbol{F}$. The implementation of Equation (5.17) is straightforward. In order to find $\hat{\boldsymbol{F}}$, the inverse of $\boldsymbol{R_{xx}}$ is required. A *Single Value Decomposition* (SVD) of $\boldsymbol{R_{xx}}$ is used to ensure a better stability for the computation of $\boldsymbol{R_{xx}}^{-1}$. The value needed for score compensation is in fact the determinant of $\hat{\boldsymbol{F}}$ (and more precisely $\log(|\hat{\boldsymbol{F}}|)$) as seen in Equation (5.8). The determinant of $\hat{\boldsymbol{F}}$ is

$$\left|\hat{\boldsymbol{F}}\right| = \left|\boldsymbol{C_{xy}}^T\boldsymbol{R_{xx}}^{-1}\right|. \tag{5.18}$$

Once $\boldsymbol{C_{xy}}^T$ and $\boldsymbol{R_{xx}}^{-1}$ have been computed for a speaker, the determinant of the matrix $\hat{\boldsymbol{F}}$ is computed by means of a SVD. From Equation (5.18), a simplified expression of $\det(\hat{\boldsymbol{F}})$ that uses only auto-correlation matrices can be derived.

5.2.2 Correlation Matrices

From Equation (5.18), it is possible to obtain a simpler expression for $\det(\hat{\boldsymbol{F}})$ that does not involve any matrix inversion. The matrix $\hat{\boldsymbol{F}}$ is the MSE estimate of $\boldsymbol{F}$. If $\boldsymbol{F}=\hat{\boldsymbol{F}}$, then $\boldsymbol{y}=\boldsymbol{F}\boldsymbol{x}=\hat{\boldsymbol{F}}\boldsymbol{x}$. The determinant of $\boldsymbol{F}$ is

$$|\boldsymbol{F}| = \left|\boldsymbol{C_{xy}}^T \boldsymbol{R_{xx}}^{-1}\right| \tag{5.19}$$

with $\boldsymbol{C_{xy}}=\boldsymbol{x}\boldsymbol{y}^T$ and $\boldsymbol{R_{xx}}=\boldsymbol{x}\boldsymbol{x}^T$. The matrix $\boldsymbol{C_{xy}}^T$ can be rewritten as

$$\begin{aligned}
\boldsymbol{C_{xy}}^T &= (\boldsymbol{x}\boldsymbol{y}^T)^T \\
&= \boldsymbol{y}\boldsymbol{x}^T \\
&= \boldsymbol{F}\boldsymbol{x}\boldsymbol{x}^T \\
&= \boldsymbol{F}\boldsymbol{R_{xx}} \\
\Leftrightarrow |\boldsymbol{C_{xy}}^T| &= |\boldsymbol{F}\boldsymbol{R_{xx}}| \\
&= |\boldsymbol{F}||\boldsymbol{R_{xx}}| \\
&= \frac{|\boldsymbol{F}|\,|\boldsymbol{R_{xx}}|\,\left|\boldsymbol{F}^T\right|}{\left|\boldsymbol{F}^T\right|} \\
\Leftrightarrow |\boldsymbol{C_{xy}}^T| &= \frac{|\boldsymbol{F}|\,|\boldsymbol{R_{xx}}|\,\left|\boldsymbol{F}^T\right|}{|\boldsymbol{F}|}
\end{aligned} \tag{5.20}$$

since $|\boldsymbol{F}|=|\boldsymbol{F}^T|$. In the light of Equation (5.20), Equation (5.19) can be rewritten as

$$\begin{aligned}
|\boldsymbol{F}| &= \left|\boldsymbol{C_{xy}}^T \boldsymbol{R_{xx}}^{-1}\right| \\
&= \left|\boldsymbol{C_{xy}}^T\right| \left|\boldsymbol{R_{xx}}^{-1}\right| \\
&= \frac{|\boldsymbol{F}|\,|\boldsymbol{R_{xx}}|\,\left|\boldsymbol{F}^T\right|}{|\boldsymbol{F}|\,|\boldsymbol{R_{xx}}|} \\
&= \frac{\left|\boldsymbol{F}\boldsymbol{R_{xx}}\boldsymbol{F}^T\right|}{|\boldsymbol{F}|\,|\boldsymbol{R_{xx}}|} \\
\Rightarrow |\boldsymbol{F}|^2 &= \frac{\left|\boldsymbol{F}\boldsymbol{R_{xx}}\boldsymbol{F}^T\right|}{|\boldsymbol{R_{xx}}|}.
\end{aligned} \tag{5.21}$$

By definition, $\boldsymbol{R_{yy}}$ is equal to $\boldsymbol{yy}^T$. By rewriting this expression, one can obtain

$$\begin{aligned} \boldsymbol{R_{yy}} &= \boldsymbol{yy}^T \\ &= \boldsymbol{Fx}(\boldsymbol{Fx})^T \\ &= \boldsymbol{Fxx}^T\boldsymbol{F}^T \\ &= \boldsymbol{FR_{xx}F}^T \\ \Rightarrow |\boldsymbol{R_{yy}}| &= \left|\boldsymbol{FR_{xx}F}^T\right|. \end{aligned} \tag{5.22}$$

From Equation (5.22), it is possible to rewrite Equation (5.21) as

$$\begin{aligned} |\boldsymbol{F}|^2 &= \frac{|\boldsymbol{R_{yy}}|}{|\boldsymbol{R_{xx}}|} \\ \Rightarrow |\boldsymbol{F}| &= \sqrt{\frac{|\boldsymbol{R_{yy}}|}{|\boldsymbol{R_{xx}}|}}. \end{aligned} \tag{5.23}$$

The principal advantage of Equation (5.23) is to require no matrix inversion. The two matrices $\boldsymbol{R_{yy}}$ and $\boldsymbol{R_{xx}}$ are straightforward to compute. Their determinants are found by means of a SVD.

5.2.3 Function Approximation

As described in the previous section, it is possible to estimate a linear transformation given the original features $\boldsymbol{x}$ and the transformed features $\boldsymbol{y}$. The matrix computed is a MSE estimate of this linear transformation. Up to now, only the case of $\boldsymbol{y}=\boldsymbol{Fx}$ as been considered. In fact, Equation (5.17) can be used to estimate a matrix $\boldsymbol{L}$ such that $\boldsymbol{y}=\boldsymbol{Fx}+\boldsymbol{b}=\boldsymbol{L}\left[\boldsymbol{x}^T\ 1\right]^T$ where $\boldsymbol{b}$ is a bias vector and $\left[\boldsymbol{x}^T\ 1\right]^T$ is an extended version of vector $\boldsymbol{x}$. The matrix $\boldsymbol{L}$ is in fact $\boldsymbol{L}=[\boldsymbol{F}\ \boldsymbol{b}]$. The matrix $\boldsymbol{F}$ can be easily recovered from $\boldsymbol{L}$.

Approximating a non-linear transformation with only one matrix may be too constrained. Instead of having only one matrix, the function is approximated with several matrices. A non-linear transformation could be approximated by a piecewise linear function. The problem that is now encountered is to divide the function into subparts to be approximated by a linear transformation. One solution to this problem is to cluster the data using a Vector Quantization (VQ) approach. Vectors that are "close" to each others in the original space

are assumed to have their transformed counterparts in the same region in the transformed space. Therefore, a linear transformation that maps a cluster in the original space to a corresponding cluster in the transformed space can be approximated. Each data cluster i is associated to a matrix $\boldsymbol{F}_i$.

The computation of the score compensation for a particular speaker is done in several steps. First, the original speech feature vectors are clustered through a VQ step. From this step, each feature vector is associated to a specific cluster i. Second, by means of the BPT, the features are transformed accordingly to the parameter set $\boldsymbol{\beta}$ associated to the speaker. Third, a matrix $\boldsymbol{F}_i$ is estimated using Equation (5.17) for each cluster i from its associated original and transformed feature vectors. The determinant of $\boldsymbol{F}_i$ is then computed and stored.

The computation of the compensation factor for a sequence of observed feature vectors $\boldsymbol{x}$ is straightforward. From the vector sequence, the VQ step provided a sequence of cluster indices.

$$\begin{aligned} \{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \ldots, \boldsymbol{x}_M\} &\overset{\text{VQ}}{\Longrightarrow} \{i_0, i_1, i_2, i_3, i_4, \ldots, i_M\} \\ &\Longrightarrow \{1, 3, 2, 2, 4, \ldots, 3\} \end{aligned}$$

From this sequence of cluster indices, one can find the sequence of determinants that is needed for the compensation factor computation.

$$\{1, 3, 2, 2, 4, \ldots, 3\} \Leftrightarrow \{|\boldsymbol{F}_1|, |\boldsymbol{F}_3|, |\boldsymbol{F}_2|, |\boldsymbol{F}_2|, |\boldsymbol{F}_4|, \ldots, |\boldsymbol{F}_3|\}$$

Since all computations are done with log-likelihoods, the compensation factor τ is the sum of the logarithm of these determinants.

$$\{|\boldsymbol{F}_1|, |\boldsymbol{F}_3|, |\boldsymbol{F}_2|, |\boldsymbol{F}_2|, |\boldsymbol{F}_4|, \ldots, |\boldsymbol{F}_3|\} \Leftrightarrow \tau = \sum_{m=0}^{M} \log\left(|\boldsymbol{F}_{i_m}|\right).$$

5.3 SPEAKER NORMALIZATION PROCEDURE

The Speaker Normalization Procedure based on the newly defined BPT is achieved in two distinct steps. First, BPT parameters are estimated for each speaker using the NM algorithm. This first step can be viewed as a black box that provides the best BPT parameters for a set of speakers. The second step consists in applying the BPT to the speech of each speaker accordingly to their best BPT parameters. In the Analysis step, the speech of each speaker is transformed by means of the BPT. It is actually this step that performs the Speaker Normalization. Using the transformed features as input features, a regular Decoding is then performed.

6.0 EXPERIMENTAL RESULTS

Our experimental results combine results from both the Dev01 and Eval01 test sets. As mentioned earlier in Chapter 4, the likelihood that observed improvements are test-set specific decreases as they are reproduced on more than one test set. By presenting results on Dev01 and Eval01, we give more credibility to any possible performance gain.

A Gender Dependent (GD) system as well as a Gender Independent (GI) system were used in our experimental setup. The reason for giving results for both GD and GI system is to provide WERs for the two types of system setup which are widely used in ASR. For instance BBN Technologies used a GI system for the 2003 Rich Transcription Evaluation (RT03) held in April 2003.

For the GD system, Acoustic Models are created for each gender, male and female. A direct consequence for clustering the data by gender is to allow only one half of the training data for the Acoustic Models of each gender. By regrouping the data by gender, part of the speech variability between genders does not need to be modeled by the Acoustic Models. For the GI system, Acoustic Models are created on speech from both genders. All the data is available to train the GI Acoustic Models. However, part of the statistical modeling property of the models accommodates for the difference between genders. As a consequence, GD Acoustic Models will always perform better or equally than their GI counterparts.

The first step for our experimental results is to determine the GD and GI baselines for each decoding set.

6.1 BASELINE RESULTS

The GD and GI baseline systems correspond to our reference points in terms of performance. Results for the GD system can be found in Table 7. It is interesting to note that the Eval01 seems to be an "easier" set than Dev01. Indeed, the same Acoustic Models and Word Models were used for the Decodings of both sets. The Dev01 set has a WER of 44.47% while Eval01

Table 7: Baseline WERs for the Gender Dependent system.

GD System	**Dev01**	**Eval01**
Baseline	44.47%	40.05%

displays a 40.05% WER. One must keep in mind that with 120 speakers Eval01 is more than 2 times bigger in duration than Dev01 with 48 speakers. The difference of WERs in Table 7 is a good example of the difference of "difficulty" observable between various test sets.

Results for the GI system can be found in Table 8. Once again, there is a difference of difficulty between Dev01 and Eval01. With 41.56% WER, Eval01 has a better overall performance than Dev01 (46.11% WER). As mentioned earlier, a GD system always performs

Table 8: Baseline WERs for the Gender Independent system.

GI System	**Dev01**	**Eval01**
Baseline	46.11%	41.56%

equally or better than its GI counterpart, as can be observed by comparing results from the two tables. For each test set, the GI system has a higher WER than the GD system.

6.2 BPT EXPERIMENTAL RESULTS

The experimental results for a GD system can be found in Table 9 for Dev01 and Table 10 for Eval01. Each table includes the WERs for the BPT using the two types of parameter

estimation procedure discussed in Chapter 5. The first procedure, using the Nelder-Mead algorithm, performs a n-best rescoring of the n-best list from the baseline experiment. The second one performs a 1-best rescoring of the best hypothesis from the baseline system. The Eide frequency warping defined in Equation (2.38) is used to perform an Eide VTLN. As discussed in Chapter 2, the Eide VTLN requires the estimation of one SD parameter. In order to compare the results of the Eide VTLN and the BPT VTLN, the same approach was used for both techniques for their parameter estimation procedures. For both BPT and Eide VTL, the estimation procedure uses the same objective function, defined in Equation (5.2). For the Eide VTL, only the 1-best rescoring is used. The parameter selection is similar to a grid search. The objective function is evaluated on the same grid used in the VTLN step of the Byblos system at BBN. The grid is on the interval $[0.80, 1.20]$ with a step of 0.02.

Table 9: Results for Dev01 Gender Dependent Decodings.

Speaker Normalization Procedure for Dev01	**WER (in %)**
None (Baseline)	44.47%
BPT VTL (NM n-best rescoring)	43.15%
BPT VTL (NM 1-best rescoring)	43.13%
Eide VTL (Grid 1-best rescoring)	44.18%
Eide VTL (ML with VTL models)	44.17%

From Table 9, the BPT n-best rescoring (43.15% WER) offers a 1.32% absolute gain compared to the baseline (44.47% WER). The BPT 1-best rescoring (43.13% WER) offers the same performance but with a reduced computation time. It is interesting to see that the Eide VTL (44.18% WER) offers a 0.29% absolute gain compared to the baseline system which is smaller than the possible gain from BPT. However, the Eide VTL performance is limited by the grid used in the parameter search. Using a finer grid will not entirely compensate for the difference of WER gain between BPT and Eide VTL. Another approach for Eide VTL is to estimate the parameters not from a 1-best rescoring but by means of a simplified Acoustic Model[34]. This approach requires to build a GMM that models the distribution of the speech features of a "canonical" speaker. Each speaker is assigned the

transformation parameter that maximizes the likelihood of its transformed features given the canonical model. The features are modified using the Eide VTL warping function. This *Maximum Likelihood* (ML) paradigm is the one currently used by BBN for VTLN. The ML Eide VTL (44.17% WER) is also offering a smaller gain than for the BPT.

Table 10: Results for Eval01 Gender Dependent Decodings.

Speaker Normalization Procedure for Eval01	**WER (in %)**
None (Baseline)	40.05%
BPT VTL (NM n-best rescoring)	38.92%
BPT VTL (NM 1-best rescoring)	39.04%
Eide VTL (Grid 1-best rescoring)	39.72%
Eide VTL (ML with VTL models)	39.67%

For Eval01, in Table 10, the BPT n-best rescoring (38.92% WER) offers a 1.13% absolute gain compared to the baseline system (40.05% WER). The BPT 1-best rescoring with 39.04% WER offers a slightly worse performance (0.12% absolute increase) but with a significant reduction in computation time. This becomes important when a decoding set has a large amount of speakers as in Eval01. The 1-best rescoring Eide VTL (39.72% WER) offers a 0.33% absolute gain compared to the baseline system which is smaller than the possible gain from BPT. Once again, the ML Eide VTL (39.67% WER) does improve the 1-best rescoring Eide VTL but brings a smaller gain than for the BPT.

By comparing the results from Table 9 and Table 10, it becomes clear that the BPT can give a steady 1.1% absolute gain or better. For Dev01, the gain for BPT is 3.01% relative while for Eval01, it is 2.82% relative. The Eide VTL, using the same objective function offers a steady 0.3% absolute gain on Dev01 and 0.4% absolute gain on Eval01 which is smaller than what is possible from the BPT. These interesting results demonstrate that the frequency warping from the BPT brings a significant gain as compared to the standard Eide VTL. Another important point is that the gain from BPT is still present for a large test set such as Eval01. This result is particularly encouraging and reinforces the conclusion that BPT offers a steady improvement.

The performance gains from using the BPT on a GI system are presented in Table 11 for Dev01 and Table 12 for Eval01. Only a BPT 1-best rescoring was performed for each set. As described earlier, the performances for a n-best rescoring and a 1-best rescoring are similar while their respective computation times differ significantly. This confers an important advantage to the 1-best rescoring.

Table 11: Results for Dev01 Gender Independent Decodings.

Speaker Normalization Procedure	**WER (in %)**
None (Baseline)	46.11%
BPT VTL (NM 1-best rescoring)	45.07%

In Table 11, the BPT (45.07% WER) offers a 1.04% absolute gain compared to the baseline system (46.11% WER). This corresponds to a relative gain of 2.26% compared to the baseline.

Table 12: Results for Eval01 Gender Independent Decodings.

Speaker Normalization Procedure	**WER (in %)**
None (Baseline)	41.27%
Nelder Mead 1-best rescoring	40.98%

In Table 12, the VTL (40.98% WER) display a 0.29% gain compared to the baseline system (41.27% WER). This corresponds to a relative gain of 0.7% compared to the baseline.

In the case of a GI system, the BPT still offers between 2.26% and 0.7% relative gain compared to the baseline systems. Once again, Dev01 sees a greater improvement from BPT than Eval01.

6.3 EXPERIMENTAL RESULTS WITH SCORE COMPENSATION

Score compensation has been included in the numerical evaluation of the objective function. Results on Dev01 only for a GD system were conducted due to time constraint. These results

are summarized in Table 13. Before applying any score compensation, the best performance is found for the BPT using the NM 1-best rescoring (43.13% WER). This corresponds to a 1.34% absolute gain compared to the baseline system (44.47% WER).

Table 13: Score Compensation Results for Dev01 Gender Dependent Decodings.

Parameters Estimation	**Score Compensation**			**WER in %**
	Active	Cluster	Weight	
Baseline	No	N/A	N/A	44.47%
NM 1-best	No	N/A	N/A	43.13%
NM 1-best	Yes	1	1.0	43.51%
NM 1-best	Yes	1	0.1	43.24%
NM 1-best	Yes	1	0.05	42.96%
NM 1-best	Yes	1	0.025	42.86%
NM 1-best	Yes	1	0.0125	43.14%
NM 1-best	Yes	10	0.025	42.90%
NM 1-best	Yes	200	1.0	43.52%

It is decided to first study the impact of the compensation factor on a system where the feature transformation function is approximated by only one linear transformation. The first experiment using score compensation with one cluster gives a WER of 43.51%. This is better than the baseline but slightly worse than without any score compensation. After analysis of the BPT parameters chosen by the compensated procedure, it becomes clear that the compensation factor allows the BPT parameters to differ only slightly from the no-transformation case. In other words, the compensation factor is too large. A solution to this problem is to have a weighted compensation factor. Table 13 has a column for the weight applied to the compensation factor. By allowing to decrease the weight, one can reduce the influence of the compensation factor until the weight becomes so small that the system goes back to behaving as if no compensation is performed. This implies that an "optimal" weight could be found for our system.

It is this exact pattern that is observed in the Table 13. Decreasing the weight from

1.0 to 0.1 does decrease the WER from 43.51% to 43.24%. The weight is then divided by two for each following experiment. As a matter of fact, the WER decreases as the weight diminishes until it reaches the value of 0.0125 where the WER starts to increase again. The best performance (42.86% WER) is obtained for a weight value of 0.025 which is a 1.61% absolute gain compared to the baseline. This represents a further gain of 0.27% absolute compared to the best NM 1-best rescoring system (43.13% WER). By using a weighted score compensation, the BPT VTLN can now achieve a relative gain of 3.62% on Dev01.

Further experiments are conducted to establish if increasing the number of clusters to approximate the feature transformation can provide further improvement. By keeping the weight at 0.025 and increasing the number of clusters to 10, only a slight increase in WER can be observed (42.90% WER compared to 42.86%). The lack of improvement is even more significant when, for a weight of 1.0, the number of clusters is increased to 200. The WER for this settings is of 43.52% which is identical to the performance for 1 cluster (43.51% WER). This is a clear indication that the approximation of our transformation by several linear transformations is not properly defined. First of all, the clustering that is performed may not cluster data for which the non-linear transformation could be approximated as a linear transform. The VQ approach for such clustering may not be the best one. Second, there is an issue of continuity between one cluster to another. The linear transformation estimated for two contiguous clusters may differ significantly. The transition from one transformation to the other may result in a local discontinuity of the global function. In order to avoid this discontinuity, the estimation of the matrices for all clusters should be a joint estimation with the constraint of continuity from one cluster to another. This is ground for further research and future works.

7.0 CONTRIBUTIONS AND FUTURE RESEARCH

7.1 CONTRIBUTIONS

We have introduced a new frequency transformation for Speaker Normalization. The Bandpass Transform is defined in a Complex Analysis framework. At the same time, the BPT finds an interpretation in signal processing: it is the transformation of a prototype Bandpass filter into a desired Bandpass filter. This allows the BPT to perform complex transformations with two degrees of freedom of the frequency axis of a signal's Fourier spectrum. Those transformations are constrained by the the physical limits associated to a mapping of a Bandpass Filter into another Bandpass Filter. For the case of voiced speech, the allowed transformations can move formants along the frequency axis by sliding all of them up and down the frequency axis or by moving them further or closer from each other. These properties are specific to the BPT.

A Speaker Normalization procedure based on the Nelder-Mead algorithm was designed from the ground up. It enables the use of the BPT for VTL Normalization of Decoding Corpora by estimating the best BPT parameter for each speaker present in a Corpus. This procedure is totally self-contained and can be used to perform a BPT VTLN on any test sets.

The experimental setup that we carefully defined allowed to study the BPT VTLN performances extensively. The experimental results on GD and GI systems demonstrate that the BPT can provide steady improvements on the two test sets Dev01 and Eval01. Improvements of 3.0% relative and higher were observed. In all cases, the BPT offered better performances in all our experimental conditions than the standard Eide VTLN used in the Byblos system.

We proposed a score compensation technique, based on a well-defined theoretical framework from the model adaptation theory, that increases the quality of the BPT parameters selection from our Speaker Normalization procedure. This technique allowed to improve the WER gain from BPT even more. More research needs to be spent on this task to better understand the dynamic of score compensation.

7.2 FUTURE RESEARCH

- The first task for future work would be to establish a better understanding of the Score Compensation. Providing a well-formulated theoretical ground for the use of non-linear transformation of speech feature and non-linear adaptation of Acoustic Models. Such research work could be significant contribution to the field of ASR.
- The Speaker Normalization procedure we proposed works on Decoding only. However, it could be possible to bring even more gain from the BPT by performing this VTL Normalization in Training as well. It has been observed in previous research work that having VTLN on both Training and Decoding provides the maximum gain. Work on speeding up the procedure by trying other optimization procedure and maybe moving away from the Nelder-Mead solution is of course necessary. It is crucial to have a Speaker Normalization Procedure that provides good performance and works fast.
- The BPT is a transformation based on second order polynomials. It is a natural step to consider the use of polynomials of third order (and even more). This will enable the mapping of three (or more) points to other points on the unit circle allowing more degrees of freedom for our frequency transformation. This new frequency transformation still has the advantage of performing a mapping of bands of frequencies to other bands on the frequency axis. By allowing more mapped points, the complexity of the frequency increases, while still keeping a simple physical meaning, and more effective transformations could be defined for VTLN.
- A more general direction of future research would be to evaluate if the gain from BPT VTLN is still present after performing an adaptative Decoding. This is an interesting

subject of research as it opens the broader field of comparing the performance of linear and non-linear transformation for feature transformation as well as model adaptation.

BIBLIOGRAPHY

[1] Huang, X., Ariki, Y., and Jack, M. *Hidden Markov Models for Speech Recognition.* Prentice Hall Publishing Company, 1990.

[2] Jelinek, Frederik. *Statistical Methods for Speech Recognition.* The MIT Press, 1998.

[3] Gales, M.J.F. "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition". *Computer Speech and Language*, 12, 1998.

[4] Stevens, Kenneth N. *Acoustics Phonetics.* the MIT Press, 1999.

[5] Lee, L. and R., Rose. "A Frequency Warping Approach To Speaker Normalization". *IEEE Transactions on Speech and Audio Processing*, Vol. 6 No. 1, pp. 49–60, January 1998.

[6] Lee, L. and Rose, R. "Speaker Normalization using Efficient Frequency Warping Procedures". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 353–356, Atlanta, Georgia, 1996.

[7] Eide, E. and Gish, H. "A Parametric Approach to Vocal Tract Length Normalization". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 346–348, Atlanta, GA., USA, 1996.

[8] Acero, Alejandro. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1990.

[9] McDonough, John W. "Speaker Normalization with All-Pass Transforms". Technical Report 28, Johns Hopkins University, September 1998.

[10] McDonough, John W. *Speaker Compensation with All-Pass Transforms.* PhD thesis, Johns Hopkins University, 2000.

[11] McDonough, John W. "Speaker Compensation with sine-log All-Pass Transforms". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 369–372, 2001.

[12] McDonough, John W. "Transformation of Discrete-Time Sequences with Analytic Functions". Technical Report 35, Johns Hopkins University, September 1998.

[13] Oppenheim, Alan V. and Johnson, D. H. "Discrete Representation of Signals". In *Proc. of IEEE*, volume 60, pp. 681–691, 1972.

[14] Rabiner, Lawrence R. and Schafer, Ronald W. *Digital Processing of Speech Signals*. Prentice Hall Signal Processing Series, 1978.

[15] Cohen, Leon. *Time-Frequency Analysis*. Prentice Hall Publishing Company, 1999.

[16] Makhoul, J. "Spectral Analysis of Speech by Linear Prediction". *IEEE Transactions on Audio Electroacoust.*, 21, pp. 140–148, June 1973.

[17] O'Shaughnessy, Douglas. *Speech Communication*. Addison-Wesley Publishing Company, 1990.

[18] Rabiner, Lawrence and Juang, Biing-Hwang. *Fundamentals of Speech Processing*. Prentice Hall Signal Processing Series, 1993.

[19] Quatieri, Thomas F. *Speech Signal Processing*. Prentice Hall Publishing Company, 2002.

[20] Deller, John R., Hansen, John H.L., and Proakis, John G. *Discrete-Time Processing of Speech Signals*. IEEE Press, 2000.

[21] Oppenheim, Alan V. and Schafer, Ronald W. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series, 1989.

[22] Needham, Tristan. *Visual Complex Analysis*. Oxford University Press, 2001.

[23] Constantinides, A.G. "Spectral Transformations for Digital Filters". In *Proc. of the IEE*, volume 117, pp. 1585–1590, 1970.

[24] Huang, X., Acero, A., and Hon, H.-W. *Spoken Language: A Guide to Theory, Algorithm, and System Development*. Prentice Hall Publishing Company, 2001.

[25] Matsoukas, S., Colthurst, T., O., Kimball, Solomonoff, A., Richardson, F., Quillen, C., Gish, H., and Dognin, P. "The 2001 Byblos English Large Vocabulary Conversational Speech Recognition System". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pp. 721–724, Orlando, Florida, 2002.

[26] Rabiner, Lawrence R. "A Tutorial on Hidden Marok Models and Selected Applications in Speech Recognition". *Proc. of the IEEE*, Vol. 77 No. 2, pp. 257–286, February 1989.

[27] Austin, S., Schwartz, R., and Placeway, P. "The Forward-Backward Search Algorithm". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 697–700, Toronto, Canada, 1991.

[28] Nguyen, L. and Schwartz, R. "Efficient 2-Pass N-Best Decoder". In *Proc. of Eurospeech '97*, pp. 167–170, Rhodes, Greece, 1997.

[29] Welling, L., Ney, R., and Kanthak, S. "Speaker Adaptive Modeling by Vocal Tract Normalization". *IEEE Transactions on Speech and Audio Processing*, Vol. 10 No. 6, pp. 415–426, September 2002.

[30] Lagarias, J. et al. "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions". *SIAM Journal on Optimization*, Vol. 9 No. 1, pp. 112–147, 1998.

[31] Dognin, P. and El-Jaroudi, A. "A New Spectral Transformation for Speaker Normalization". In *Proc. of Eurospeech*, Geneva, Switzerland, 2003.

[32] Gales, M.J.F. "Semi-tied Covariance Matrices for Hidden Markov Models". *IEEE Transactions on Speech and Audio Processing*, Vol. 7 No. 3, May 1999.

[33] Brookes, Mike. *Matrix Reference Manual*. Imperial College, London, UK. http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/.

[34] Dognin, P., Billa, J., and El-Jaroudi, A. "Parameter Optimization for Vocal Tract Length Normalization". In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.