NTNU
Norwegian University of
Science and Technology

Bandwidth Extension of Telephony Speech

Martin Etnestad Johansen

Master of Science in Electronics
Submission date: June 2009
Supervisor: Torbjørn Svendsen, IET
Co-supervisor: Svein Gunnar Pettersen, Tandberg Telecom AS

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

Thanks to advancements in audio coding and communications technology, the audio quality in video-conferencing systems has improved greatly during the last years, allowing for the entire audible frequency band (20 - 20k Hz) to be utilized.
When someone calls into such a system from a unit that restricts the acoustic bandwidth to the traditional telephony band (0.3 - 3.4 kHz) the differences in perceived quality are highly noticeable, and sometimes even distracting.

The objective of this thesis is to investigate and implement systems for bandwith extension of regular telephone quality speech, i.e., synthesize a wideband signal from a narrowband one.

Most bandwidth extension systems are based on the linear source-filter model for speech. According to this model, bandwidth extension is a matter of estimating a spectrally flat excitation signal and a synthesis filter for shaping the spectral envelope.
The main challenge lies in estimation of the wideband synthesis filter, which requires a mapping from a narrowband feature vector to a parameterization of the wideband spectral envelope.
There is a multitude of ways to do this because there is no set method for performing the mapping, nor are the narrowband feature vector or parameterization of the wideband spectral envelope set.

Issues that should be addressed include finding which feature vectors and parameterizations of the spectral envelope are suitable for bandwidth extension, as well as methods for performing the mapping and estimating the excitation signal.

Assignment given: 15. January 2009
Supervisor: Torbjørn Svendsen, IET

Abstract

The public switched telephone network (PSTN) restricts the acoustic bandwidth of telephony speech to less than 4 kHz. For compatibility with analog telephone networks, a $0.3 - 3.4$ kHz pass band is common. This bandwidth reduction has a significant impact on perceived quality, and is especially noticeable and even distracting when PSTN users call into, e.g., video conferencing systems in which the other participants may use wideband ($50 - 7$k Hz) speech codecs. To reduce the gap in quality, one may attempt to resynthesize the missing spectrum. Techniques for this are referred to as bandwidth extension (BWE).

For this thesis, two systems for BWE of speech into the high band ($f \geq 3.4$ kHz) were implemented in Matlab, based on systems proposed in literature. The extension was done according to the linear source-filter model for speech, meaning estimation of the excitation and spectral envelope from the narrowband ($0.3 - 3.4$ kHz) signal were done separately.

BWE System 1 made use of linear prediction (LP) analysis in combination with modulation for extension of the excitation. Its wideband spectral envelope estimation was primarily based on linear prediction cepstral coefficients (LPCC) and artificial neural networks (ANN).

BWE System 2 made use of bandpass-modulation of Gaussian noise (BP-MGN) for extension of the excitation. Its wideband spectral envelope estimation was based on Mel-frequency cepstral coefficients (MFCC) and Gaussian mixture modelling (GMM), which was the most complex estimation method of the two systems.

Objective analysis of the two systems' spectral envelope estimation and informal listening tests were carried out. These analyses showed that BWE System 1 performed best, though both systems improved the perceived quality. BWE systems based on LP analysis therefore seem to be preferrable due to the superior excitation, and efficient computation of the cepstrum.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Aim of this Thesis . . . 1
1.3 Report Structure . . . 2

2 Bandwidth Extension of Speech **3**
2.1 Speech Model . . . 3
2.1.1 Excitation Signal . . . 4
2.1.2 Spectral Envelope . . . 4
2.1.3 Analysis/Synthesis Method . . . 5
2.2 Bandwidth Extension . . . 5
2.3 Extension of Speech Excitation . . . 6
2.3.1 Modulation and Spectral Shifting . . . 7
2.3.2 Bandpass-Modulated Gaussian Noise . . . 8
2.4 Extension of Spectral Envelope . . . 9

3 Regression Methods **11**
3.1 Artificial Neural Networks . . . 11
3.1.1 Computation of Response . . . 11
3.1.2 Activation Function . . . 13
3.1.3 Network Structure . . . 13
3.1.4 Training Basics . . . 14
3.1.5 Weight Space Gradients . . . 14
3.1.6 Backpropagation Training . . . 15
3.1.7 Network Initialization . . . 16
3.1.8 Avoiding Overtraining . . . 16
3.2 Gaussian Mixture Models . . . 16
3.2.1 Training GMMs . . . 17
3.2.2 Mixture Initialization and Splitting . . . 18
3.2.3 GMM-Based Regression . . . 19
3.3 Linear Mapping . . . 20

4 Feature Parameters **21**
4.1 Cepstral Coefficients . . . 21
4.2 Linear Prediction Cepstral Coefficients . . . 22
4.3 Mel Frequency Cepstral Coefficients . . . 23

4.4 Gradient Index . . . 24
4.5 Normed Frame Energy . . . 24
4.6 Related Error Measures and Transforms . . . 25
4.6.1 Cepstral Distance . . . 25
4.6.2 Spectral Distance Measures . . . 25
4.6.3 Cepstral Liftering . . . 26
4.6.4 Discrete Cosine Transform . . . 26
4.6.5 Dynamic Feature Parameters . . . 27

5 Implemented Systems 29
5.1 BWE System 1 . . . 30
5.1.1 Overview . . . 30
5.1.2 Extension of Spectral Envelope . . . 30
5.1.3 Creation of Synthesis Filter . . . 31
5.1.4 Extension of Excitation . . . 31
5.1.5 Implemented Variants . . . 32
5.2 BWE System 2 . . . 32
5.2.1 Overview . . . 32
5.2.2 Extension of Spectral Envelope . . . 32
5.2.3 Creation of Synthesis Filter . . . 33
5.2.4 Extension of Excitation . . . 34
5.2.5 Implemented Variants . . . 34

6 Tools 37
6.1 Speech Database . . . 37
6.2 System Implementation . . . 38
6.2.1 Artificial Neural Networks . . . 38
6.2.2 Gaussian Mixture Models . . . 38

7 Experiments 39
7.1 Performance of Spectral Envelope Estimation . . . 39
7.2 Subjective Evaluation of System Performance . . . 40

8 Results and Discussion 41
8.1 Performance of Spectral Envelope Estimation . . . 41
8.1.1 Benchmark Performance . . . 41
8.1.2 BWE System 1 Performance . . . 42
8.1.3 BWE System 2 Performance . . . 43
8.1.4 Comparing BWE System 1 and 2 . . . 45
8.1.5 Effect of Gain in BWE System 2's GMM . . . 48
8.2 Comparison with Other Systems . . . 48
8.2.1 BWE without Memory . . . 49
8.2.2 BWE with Memory . . . 49
8.3 Subjective Evaluation of System Performance . . . 50
8.3.1 Preferred System . . . 50
8.3.2 Artefacts and Speaker Selectivity . . . 52
8.3.3 Differentiation of Consonant Sounds . . . 52

9 Conclusion 55
9.1 Conclusion . . . 55

9.2 Further Work 56

A Implemented Filters **57**

A.1 POTS Bandpass Filter 57

A.2 Anti-Aliasing Filter 58

A.3 Highpass Filter 58

A.4 Bandpass Filtering for BP-MGN 59

A.5 Mel Filter Bank 59

A.6 Interpolated Filtering 61

B Contents of Enclosed CD **63**

List of Figures

2.1 The linear source-filter model of the human speech apparatus. 3
2.2 Sketch of the human vocal tract. 4
2.3 Example of BWE system structure. 6
2.4 Interpolated, upsampled and modulated POTS excitation. 8

3.1 Example of an multilayer perceptron with four layers, two of whom are hidden. . . 12
3.2 Output of the antisymmetric sigmoid function. 13
3.3 Example of error plot for training and verification 17

4.1 Chain of computations for Mel frequency cepstral coefficients 23
4.2 Sketch of Mel filter distribution in Mel and Hertz 24

5.1 Overview of BWE system based on ANN and CCs. 30
5.2 Overview of BWE system based on GMM and MFCCs. 33

8.1 LSD for different ANN structures in BWE System 1 42
8.2 RMS-LSD for different ANN structures in BWE System 1 43
8.3 LSD for different GMMs in BWE System 2 (no memory) 44
8.4 RMS-LSD for different GMMs in BWE System 2 (no memory) 44
8.5 LSD for different GMMs in BWE System 2 (with memory) 45
8.6 RMS-LSD for different GMMs in BWE System 2 (with memory) 45
8.7 Reference and gain corrected, estimated spectral envelopes from 'i' in 'sier' 46
8.8 Reference and gain corrected, estimated spectral envelopes from 's' in 'sier' 47
8.9 Spectrogram of wideband speech (“vi yngre ser jo”). 50
8.10 Spectrogram of POTS quality speech (“vi yngre ser jo”). 51
8.11 Spectrogram of output from BWE System 1 (“vi yngre ser jo”). 51
8.12 Spectrogram of output from BWE System 2 (“vi yngre ser jo”). 52
8.13 Spectrogram of wideband speech (“energi”). 53
8.14 Spectrogram of output from BWE System 1 (“energi”). 53

A.1 Magnitude response of implemented POTS filter. 57
A.2 Magnitude response of implemented anti-aliasing filter. 58
A.3 Magnitude response of implemented highpass filter, alone and in combination with the POTS filter. 59
A.4 Magnitude response for BP-MGN filtering. 60
A.5 Mel filter bank's coverage of the spectrum. 61
A.6 Interpolation by two of filter coefficients. 61

List of Tables

8.1 LSD and RMS-LSD for benchmark cases. 41
8.2 LSD and RMS-LSD for BWE System 1 42
8.3 LSD and RMS-LSD for BWE System 2 43
8.4 LSD and RMS-LSD for GMM trained on MFCCs exclusively 48

A.1 Center frequencies of the low band Mel filters. 60
A.2 Center frequencies of the high band Mel filters. 61

Abbreviations

ANN	Artificial Neural Network
BP-MGN	Bandpass-Modulated Gaussian Noise
BWE	Bandwidth Extension
CC	Cepstral Coefficient
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
EM	Expectation Maximization
GI	Gradient Index
GMM	Gaussian Mixture Model
GSM	Global System for Mobile communications
IP	Internet Protocol
ISDN	Integrated Services Digital Network
LP	Linear Prediction
LPC	Linear Prediction Coefficient
LPCC	Linear Prediction Cepstral Coefficient
LSD	Log Spectral Distance
MFCC	Mel Frequency Cepstral Coefficient
MLP	Multilayer Perceptron
NFE	Normed Frame Energy
POTS	Plain Old Telephone System
PSTN	Public Switched Telephone Network
RMS-LSD	Root Mean Square Log Spectral Distance

Chapter 1

Introduction

1.1 Motivation

In the modern, digital PSTN (Public Switched Telephone Network) the sample rate for telephony is restricted to 8 kHz with 8 bit resolution, giving a total bitrate of 64 kb/ sec. This is a legacy from analog telephone networks in which the acoustic bandwidth is restricted to the band $0.3 - 3.4$ kHz, commonly referred to as the POTS (Plain Old Telephone System) band. The main reasons for this restriction are reduction of crosstalk between users, and maximization of channel capacity at toll quality. Digital telephone networks such as ISDN and GSM do not require this bandwidth restriction, but are restricted to the same sample rate and resolution for compatibility with the PSTN. For both analog and digital telephony, the degradation in speech quality is highly noticeable and may cause consonant sounds like 's' and 'f', 'p' and 't' to be confused with eachother.

Technologies such as IP telephony and video conferencing, however, are not restricted with respect to complexity nor the regular 64 kb/ sec bitrate. This allows for any desired level of quality, such as wideband telephony with a frequency range of $50 - 7$k Hz. However, this is not of any use for calls in which PSTN users are involved. This is especially noticeable in conferencing situations, and may even be distracting since the different quality levels demand different levels of concentration. To bridge this quality gap, one can attempt to synthesize the missing spectral content. Techniques for this are collectively referred to as bandwidth extension (BWE).

1.2 Aim of this Thesis

BWE of telephony speech has been an area of research for quite some time without culminating in any standards, de facto or formal, nor any freely available implementations. The aim of this thesis was therefore to study methods and proposed systems for BWE, and suggest a speaker-independent system for approximately doubling the bandwidth of POTS speech with reasonable quality. The effects of noise, or non-speech signals in general were not considered.

1.3 Report Structure

In chapter 2, the theoretical foundation and main principles for BWE of speech are introduced, as well as some extension methods relevant to this thesis.

Three methods for regression are explained in detail in chapter 3, due to their use in the implemented systems and experiments. A selection of signal feature parameters with related transforms and distance measures are then presented in chapter 4.

In chapter 5, two implemented BWE systems are described in detail. Then follows a description of the tools and data used for their implementation in chapter 6.

The experiments that were carried out on the two systems are described in chapter 7. The results are then presented and discussed in chapter 8.

Finally, in chapter 9, the conclusions and ideas for further work are presented.

Details regarding filters implemented in the BWE systems, and a description of the accompanying CD are found in the appendices. All relevant source code and test sound clips are available on the CD.

Chapter 2

Bandwidth Extension of Speech

In this chapter, the underlying theory of BWE of speech is presented. The speech model is described first to give some insight into the physical foundation. Then, the principles behind BWE as well as some implemented methods are described.

The reader is assumed to be familiar with linear prediction and multirate signal processing [1], as well as basic information theory [2].

2.1 Speech Model

For analysis and synthesis of speech signals to be practical, a parametric signal representation is needed. The most common model in this context is the linear *source-filter model* [3], which consists of two independent components:

1. excitation source: gives the fine structure of the spectrum
2. filter: shapes the spectral envelope of the signal

A time-discrete version of the model with the relevant parameters is shown in figure 2.1. The reasoning behind the model is outlined in the following subsections.

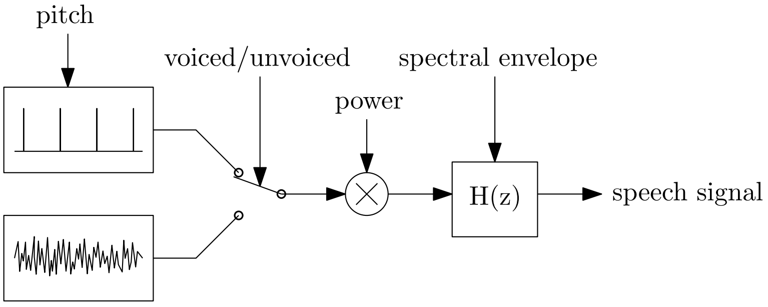

Figure 2.1: The linear source-filter model of the human speech apparatus.

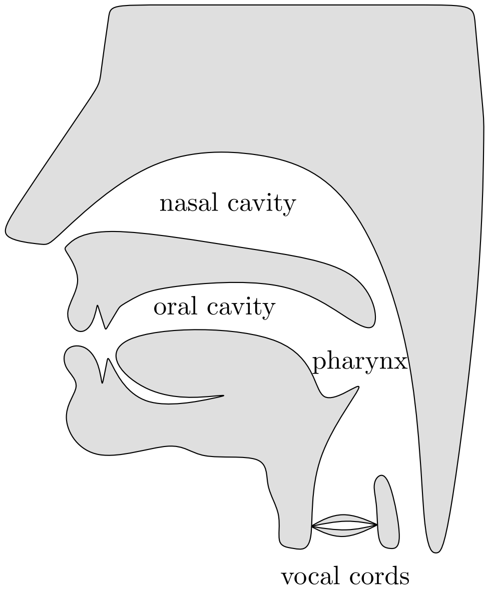

Figure 2.2: Sketch of the human vocal tract.

2.1.1 Excitation Signal

The excitation arises when air is forced out of the lungs and through constrictions in mainly the throat or oral cavity.

In the throat, only the *glottis* (vocal cords and space between them) is considered as a constriction. Depending on the tension of the vocal cords, the air will pass through either as a series of pulses, or as a steady, turbulent stream. The resulting excitation is a harmonic signal in the first case, and noise in the other. These two cases define the main categories of speech: *voiced* and *unvoiced*, respectively.

Constrictions in the oral cavity are generally a result of the shape and position of the tongue. On their own, these will only give rise to unvoiced speech. However, in combination with the vocal cords, so-called *voiced fricatives* may be generated.

Since the model filter is supposed to shape the spectral envelope, the excitation should be spectrally flat. The excitations are therefore modelled as either a pulse train or white noise. Other defining parameters of the excitation are its power and the pitch, or repetition frequency of the pulse train.

It should be noted that voiced and unvoiced excitations actually have a low- and usually high-pass characteristic. But in the source-filter model, any spectral tilt is attributed to the filter. Mixed excitations are therefore somewhat more problematic to model than simply mixing the two models, but such an approximation is still used.

2.1.2 Spectral Envelope

The spectral envelope is a consequence of the shape of the vocal tract, shown in figure 2.2.

By modelling the vocal tract as a cascade of lossless tubes, one finds that it may be represented as an all-pole filter. The pronounced resonance frequencies of the filter are referred to as formants and are vital to the classification of *phonemes*, which is the smallest linguistic distinctive unit of sound.

The main simplification of this model is that it doesn't take branching of the vocal tract into account, something that occurs for nasal speech sounds. Such branches allow for parallell resonances, causing dips in the spectral envelope and requiring zeroes in the filter for proper representation. This fact is largely ignored for two reasons:

1. zeroes may be expressed by infinitely many poles
2. spectral dips may be encoded in the representation of the excitation

Hence, with enough poles and a good representation of the excitation, the all-pole model is a reasonable choice.

2.1.3 Analysis/Synthesis Method

Based on this speech model, linear prediction (LP) is a prime candidate for both analysis and synthesis of speech. As the name implies, LP deals with the prediction of the future samples from a process as a linear combination of past samples, as shown in (2.1).

$$\begin{aligned} y(n) &: \text{output signal from a process} \\ e(n) &: \text{prediction error or excitation signal} \\ a_k &: \text{prediction coefficients} \end{aligned}$$

$$y(n) + e(n) = \sum_{k=1}^{N} a_k \cdot y(n-k) \tag{2.1}$$

Performing LP is equivalent to filtering with an all-pole filter, which corresponds to the filter in the linear source-filter model. By filtering $y(n)$ with the inverse of the LP filter, the excitation $e(n)$ is extracted. Hence, estimation of the LP coefficients (LPC) a_k for a given signal allows for *source-filter separation*.

A common method for this estimation, guaranteed to produce stable filters, is the *autocorrelation method*. Regarding the number of prediction coefficients, a rule of thumb is to use 1 per kHz of samplerate, plus an extra $2 - 4$ coefficients [3, 4].

One should note that LP is based on the assumption that the signal is stationary. This is obviously not true for speech, but the changes are slow enough for the assumption to hold fairly well in frames of length 30 ms or less. The assumption of *short time stationarity* is essential to all areas of speech processing.

2.2 Bandwidth Extension

BWE is the process of synthesizing a wideband signal from a narrowband one. For correct, or near correct extension to be possible, knowledge about the underlying process is required, preferrably as a mathematical model. Further, there must also be some dependence between the known and unknown parameters of the model.

For speech, BWE is usually based on the described linear source-filter model [4, 5]. This allows for the extension to be done separately for the source and filter, more commonly referred to as the excitation and spectral envelope. It should be noted that the words *extension* and *estimation* are used interchangeably in literature.

Figure 2.3 shows an example BWE system structure for extension of POTS quality speech. POTS quality entails a bandwidth of 3.1 kHz in the range 300 − 3.4k Hz and a sample rate of 8 kHz. Synthesis of a wideband signal (50 − 7k Hz) is most common, and therefore requires an interpolated samplerate of 16 kHz.

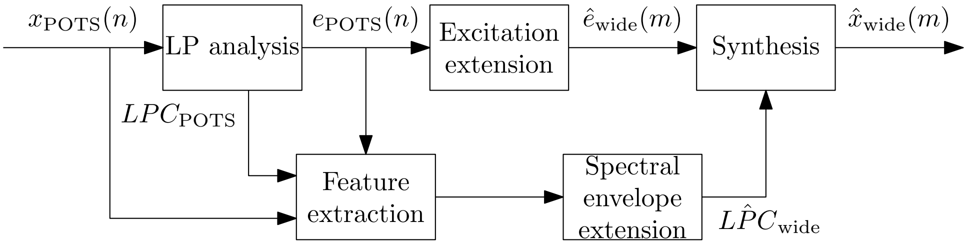

Figure 2.3: Example of BWE system structure.

Extension to the high band is assumed to be the most important since it has the greatest bandwidth and few formants appear below 300 Hz [3]. Because of this, and the poor bass reproduction in existing telephony equipment, most proposed BWE systems deal only with the high band.

Though there are no set answers as to how BWE should be done, a common principle is that the system should be transparent to the original POTS signal. Spectral discontinuities and overlaps between the synthesized and original signals therefore need to be minimized. This is commonly done by applying a correction gain to the synthesized part of the signal.

For fairly comprehensive and updated overviews of research on BWE, the reader is referred to [4, 5], on which this and most of the following sections are based.

2.3 Extension of Speech Excitation

From the described source-filter model, one expects extension of the excitation to be the easiest part of BWE. That is, the difference between harmonic and noisy excitations should be fairly obvious even in a narrowband signal. Creating a wideband excitation should then almost be trivial. However, since it is a model, it is a simplifcation of the real process of speech generation: excitations are not perfectly harmonic or noisy, and mixed excitations are not properly modelled.

For the purpose of excitation extension, some inherent properties of speech may make up for the model's inadequacies:

- low band excitation ($f < 300$ Hz) is most prominent in voiced speech [4]
- correctness of high band excitation ($f > 4$ kHz) is not crucial [6, 7]
- correct pitch structure is only crucial in the low band [4]

One may therefore assume that using different methods for extension in the two bands may give better results than relying on just one method for both.

It should be noted that delay compensation may be necessary if the extension method causes a large delay relative to the pass-through POTS signal due to, e.g., filtering.

In the following subsections, two extension methods relevant for this thesis are presented. Other extension methods may be split in mainly two families [4, 5]:

- nonlinearities
- source modelling

Nonlinearities are well known for creating sub- and superharmonics, and are therefore particularily useful for extension of voiced speech. Some drawbacks are that they require pre/post-processing due to coloration of the spectrum and the unpredictable effects they have on signal power. In addition, they are prone to cause aliasing if sufficient interpolation and decimation is not used.

Source modelling deals with implementations of the source in the linear source-filter model. Consequently, there is some overlap with methods used in speech coding.

2.3.1 Modulation and Spectral Shifting

Modulation and spectral shifting are extension methods based on moving the excitation signal extracted in LP analysis, or parts of it, in the frequency domain. The motivation behind these methods is to preserve the characteristics of the original excitation. In [5] the distinction made between the two is that modulation is done in the time domain, while spectral shifting is done in the frequency domain.

Spectral shifting requires the use of the discrete Fourier transform (DFT) and overlap-add synthesis for realtime extension. Though this method gives absolute control over the spectrum, it is also more computationally expensive and prone to cause discontinuities in phase. It does not appear to be much used.

Modulation, on the other hand, is simply done by multiplying the excitation signal with a modulation carrier as shown in (2.2). The factor k is a used to counter any changes in amplitude or power from the modulation, and may usually be set to 2.

$$e_{\mathrm{mod}}(n) = k \cdot e(n) \cdot \cos(2\pi f_{\mathrm{mod}} n) \tag{2.2}$$

There are two approaches to both of these methods for extension:

- pitch adaptive
- fixed

The difference is whether the algorithm is meant to keep the pitch structure intact or not. As noted in [4, 5], good pitch estimation is crucial for the pitch adaptive approach. Jitter in the pitch estimates in particular will introduce noticeable artefacts. Few examples of its use could be found, indicating that the improvement in quality does not make up for the required complexity. Fixed modulation is therefore the most common approach, although it can only be used for extension to the high band.

A special case of fixed modulation is called *spectral folding*. This is done by either upsampling the excitation without an anti-aliasing filter, or by modulating with the Nyquist frequency. Though this method is simple and free of *interfering* aliasing, it tends to cause a gap between the POTS and extended frequency band. The difference between proper modulation and upsampling without anti-aliasing is shown in figure 2.4.

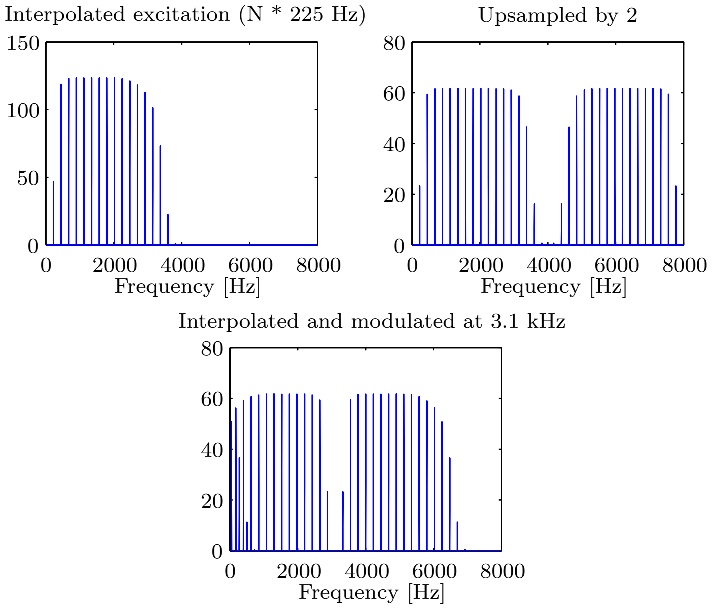

Figure 2.4: Interpolated, upsampled and modulated POTS excitation.

Besides its simplicity, extension by modulation has two main advantages:

1. spectrum is not colored
2. power changes predictably

Modulation therefore entails a minimum of post-processing. Note that aliasing will occur if the signal is not properly bandlimited before or after modulation. Since this method is only used for high band extension anyway, aliasing in the low band is of no consequence.

2.3.2 Bandpass-Modulated Gaussian Noise

Bandpass-modulated Gaussian noise (BP-MGN) was used in [7, 8, 9] for generation of high band excitation. It is most likely only useful in systems which are not based on LP analysis of the POTS signal.

The method is simply to filter out a band of the speech signal, and modulate Gaussian noise (with a constant variance) with the envelope of the resulting bandpass signal, as shown in (2.3).

$$e_{\mathrm{BP-MGN}}(n) = e_{\mathrm{Gauss}}(n) \cdot |x_{\mathrm{speech}}(n) * h_{\mathrm{bandpass}}(n)| \tag{2.3}$$

The modulation is done in an attempt to imprint any pitch structure in the speech signal onto the white noise. Suggested bands for the bandpass filter lie in the $2-4$ kHz range, with roughly 1 kHz bandwidth. This is because the characteristics of the excitation can be assumed to be similar in the upper POTS band and the high band one wishes to synthesize. Since a POTS signal is already bandlimited upwards, the bandpass filtering may simply be done by applying a highpass filter.

This extension method does, however, have the drawback that the power of the synthesized excitation depends on the bandpass filtered signal. To avoid spectral discontinuities in the BWE signal, a correction gain is needed to compensate for this side effect, as well as the synthesis filter. This gain is usually computed using analysis-by-synthesis when creating training data, and then estimated from the narrowband spectral envelope during BWE.

2.4 Extension of Spectral Envelope

Extension of the spectral envelope is a matter of mapping from a set of narrowband feature parameters to a set of wideband feature parameters that facilitate the creation of a wideband LP filter. There is no set way to perform the mapping, nor what feature parameters to use, meaning there is a multitude of possible solutions.

The mapping is both the most difficult and the most important task in BWE, particularily with respect to high band extension. Since the true mapping is not known, regression analysis of training data must be used to create estimators. The effect of estimation errors should also be taken into account when choosing wideband feature parameters, since some parameterizations may be more robust than others.

The extension relies on a dependence, or mutual information between the narrow- and wideband feature parameters. This dependence is usually assumed to be strong, but studies show that this is not the case, meaning there is a great deal of uncertainty about the missing spectrum. In [10], it is therefore concluded that most BWE systems perform reasonably not because they accurately estimate the spectral envelope, but because the extended signal sounds pleasant.

An information theoretic approach to estimating an upper bound for the performance of memoryless extension was presented by Jax in [4, 11]. Extension performance was represented by the root mean square log spectral distance (RMS-LSD, see 4.6.2) as a function of mutual information between the POTS band and each of the missing bands, i.e. the low (0 - 300 Hz) or high (3.4 - 7 kHz) band. Based on the mutual information for various feature parameters, Jax estimated that an optimal selection can yield 3 bits of mutual information for both bands[1]. This translated to a minimum RMS-LSD of approximately 3 dB in both the low and the high band. As a sidenote, Jax concluded that memoryless BWE cannot compete with wideband speech codecs due to this performance bound.

Another study of BWE performance, dealing with the effect of memory (temporal information) on extension, was presented by Kabal et al. in [12, 13]. Two scenarios of memory were studied:

1. memory of the POTS band only
2. memory of both POTS and high bands

Memory was incorporated by the use of linear regression analysis on a set of neighbouring analysis frames, resulting in Δ-coefficients (see 4.6.5).

The most interesting conclusion of this study was that only memory of both bands increases the inter-band mutual information. That is, adding memory of only the low band did not improve the spectral envelope extension. Further, it was found that the most significant improvement

[1] The high and low band entropies were not stated.

was achieved for $t \leq 160$ ms worth of memory. This corresponds to $t \leq 80$ ms of prebuffering, introducing an additional delay in the system. According to the ITU-T standard G.114 [14], end-to-end delays below 150 ms are considered acceptable in a telephone network. Introducing additional delay may therefore make communication cumbersome. Hence, one may assume that there is a trade-off between BWE performance and introduced delay.

A selection of feature parameters for use in spectral envelope extension are presented in chapter 4. For approximation of the mapping, three regression methods relevant to this thesis are described in chapter 3. Other, commonly used methods are mainly:

- vector quantization [15, 16, 17]
- hidden Markov modelling [18, 19]

Extension by use of vector quantization requires two jointly trained codebooks of feature vectors (one narrow- and one wideband), with a one-to-one mapping between them. Vector quantization may also be used in combination with other estimation methods, such as linear mapping. This allows for a classification of the narrowband feature vectors and the use of estimators that are specialized on each class.

Hidden Markov modelling is the most advanced method, and is used extensively in speech recognition [3]. It is a method for modelling the probability of sequences, and is well suited for modelling linguistic units. Though one might argue that this gives a language-dependent system, research has shown that speech quality is not lost if the system trained and tested on related languages [19].

Chapter 3

Regression Methods

In this chapter, three methods for minimum mean square error regression are described: artificial neural networks, Gaussian mixture models and linear mapping. These methods are explained in detail because they are used for spectral envelope estimation in the two implemented BWE systems, as well as benchmarks in experiments.

3.1 Artificial Neural Networks

Artificial neural networks (ANN) are mathematical models based on biological neural networks. They consist of interconnected neurons, where each neuron is defined by a set of weights and an activation function. The weights dictate how the input value to the activation function is computed from an input vector, while the neuron's output is simply the resulting value of the activation function.

ANNs come with various topologies and are used for applications such as pattern recognition, classification and function approximation. In this thesis, focus is on multilayer perceptrons (MLP) used for function approximation. MLPs are so-called feed forward networks, meaning the neurons are organized in layers and the signals only propagate forward. An example network is shown in figure 3.1.

For further information about ANNs, the reader is referred to [20], on which most of this section is based.

3.1.1 Computation of Response

Borrowing the notation used by Haykin [20], one may express the computation of a layer's input and output values as in (3.1) through (3.2). It is here assumed that all neurons in each layer use the same activation function.

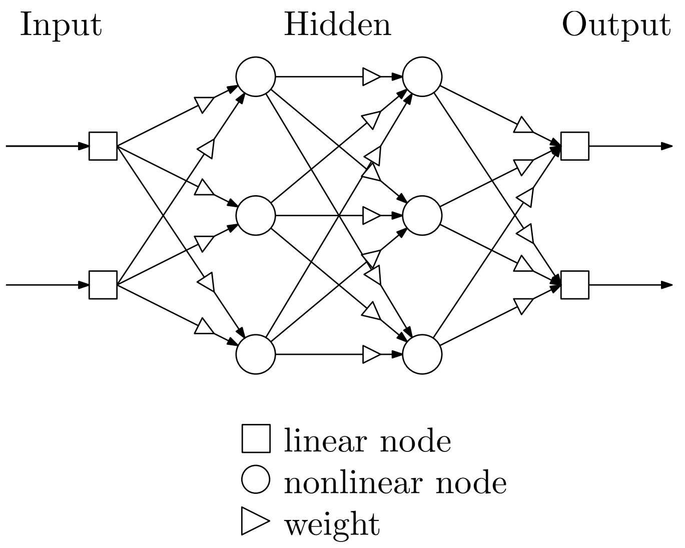

Figure 3.1: Example of an multilayer perceptron with four layers, two of whom are hidden.

$$
\begin{aligned}
\mathbf{v}^i &: \text{input vector for neurons in layer } i \\
\mathbf{W}^i &: \text{bias and weighting matrix for layer } i \\
\mathbf{y}^{i-1} &: \text{output vector from neurons in layer } i-1 \\
\varphi^i\left(...\right) &: \text{activation function of layer } i
\end{aligned}
$$

$$
\begin{gathered}
\mathbf{v}^i = \mathbf{W}^i \cdot \begin{bmatrix} 1 \\ \mathbf{y}^{i-1} \end{bmatrix} \\
\begin{bmatrix} v_1^i \\ v_2^i \\ \dots \\ v_m^i \end{bmatrix} = \left[\begin{array}{c|ccc} b_1^i & w_{1,1}^i & \dots & w_{1,n}^i \\ \dots & & & \\ \text{bias} & \dots & \text{input weights} & \dots \\ \dots & & & \\ b_m^i & w_{m,1}^i & \dots & w_{m,n}^i \end{array} \right] \cdot \left[\begin{array}{c} 1 \\ \hline y_1^{i-1} \\ y_2^{i-1} \\ \dots \\ y_n^{i-1} \end{array} \right]
\end{gathered}
\tag{3.1}
$$

$$
\begin{gathered}
\mathbf{y}^i = \varphi^i\left(\mathbf{v}^i\right) \\
\begin{bmatrix} y_1^i \\ y_2^i \\ \dots \\ y_m^i \end{bmatrix} = \varphi^i \left(\begin{bmatrix} v_1^i \\ v_2^i \\ \dots \\ v_m^i \end{bmatrix} \right)
\end{gathered}
\tag{3.2}
$$

These computations are done recursively to propagate an input vector from the input to the output layer. The first layer is, of course, fed with an input vector and not the output from a previous layer.

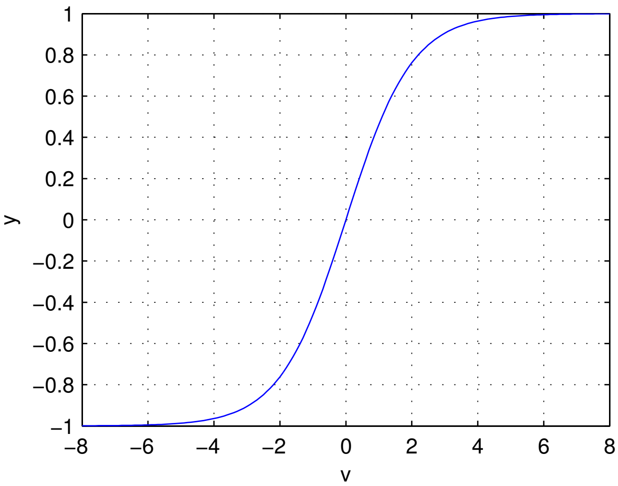

Figure 3.2: Output of the antisymmetric sigmoid function.

3.1.2 Activation Function

The activation function is what enables the ANN to learn complex tasks, and usually means a nonlinear function since layers using linear functions may be replaced with matrices.

Somewhat motivated by the response of biological neurons, the most common activation functions for ANNs with continuous output are sigmoidal nonlinearities. Examples of these are the logistic function in (3.3) and variants of the antisymmetric sigmoid in (3.4).

$$\varphi(v) = \frac{1}{1 + \mathrm{e}^{-v}} \qquad \in (0, 1) \tag{3.3}$$

$$\varphi(v) = \frac{2}{1 + \mathrm{e}^{-v}} - 1 \qquad \in (-1, 1) \tag{3.4}$$

These functions are bounded, differentiable and monotonic, all useful properties for training. The antisymmetric sigmoid function is shown in figure 3.2.

3.1.3 Network Structure

The use of MLPs for function approximation is motivated by the *universal approximation theorem*. This is an existence theorem, essentially stating that a single hidden layer can give an arbitrarily close approximation of any continuous function. But, as Haykin [20] points out, it does not say that a single hidden layer is optimal in the sense of learning time, ease of implementation or generalization. In addition, for tasks of appreciable complexity, it is not obvious how many neurons per layer are necessary or optimal. These parameters may therefore best be selected based on experimentation.

One must also take the size of the training set into account, since larger networks also mean an increased number of parameters to train. Though there is no set rule for how much training data is needed for a given neural network, Haykin suggests having more training examples than the ratio of parameters to the wanted mean square error. Note that the wanted mean square error might not actually be achievable due to, e.g., the statistical properties of the training data or a suboptimal ANN structure.

3.1.4 Training Basics

To use an ANN for function approximation, supervised training must be used. This means that for every input vector, the corresponding output vector is known. A pair of such vectors are referred to as a *training vector*, and the collection of them is the *training set.* Training an ANN means optimizing its weights (including bias) with respect to some measure of error. For a nonlinear network there is no analytic solution to this problem, so the training must be done iteratively.

The training may be done per training vector, called *sequential* training, or once per complete processing of the entire training set, called *batch* training. The term *epoch* is used to refer to the completed processing of the entire training set. In the case of sequential training, the order of the training vectors should be randomized for every epoch to explore more of the weight space.

3.1.5 Weight Space Gradients

Training algorithms for ANNs are variants of gradient searches, aiming to minimize the mean squared error of the output, shown in (3.6).

$\mathbf{y}_i$: computed response vector of neural net for input vector $\mathbf{x}_i$

$\mathbf{d}_i$: desired response vector of neural net for input vector $\mathbf{x}_i$

$$\mathcal{E}_i = \frac{1}{2} \cdot (\mathbf{d}_i - \mathbf{y}_i)^{\mathrm{T}} \cdot (\mathbf{d}_i - \mathbf{y}_i) \tag{3.5}$$

$$\mathcal{E}_{\mathrm{avg}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{E}_i \tag{3.6}$$

The error gradients in weight space are found by differentiating the squared error with respect to the individual weights (including bias) in all the layers. For brevity, only the resulting two-pass algorithm is repeated here.

First, the neurons' local gradients must be computed, and this is basically done by backwards propagation of the output error. For the last layer, the local gradients are initialized as shown in (3.7).

$g_j^i = \dfrac{\delta \mathcal{E}}{\delta v_j^i}$: local gradient for neuron j in layer i

$\varphi\prime(v)$: derivative of $\varphi(\ldots)$ with respect to v

$$\begin{bmatrix} g_1^N \\ g_2^N \\ \dots \\ g_n^N \end{bmatrix} = - \begin{bmatrix} \varphi\prime(v_1^N) & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \varphi\prime(v_n^N) \end{bmatrix} \cdot \begin{bmatrix} d_1 - y_1 \\ d_2 - y_2 \\ \dots \\ d_n - y_n \end{bmatrix} \tag{3.7}$$

The local gradients for neurons in the "earlier" layers are then recursively computed as shown in (3.8). Note that bias does not affect local gradients since the error only propagates through the neurons.

$$\begin{bmatrix} g_1^i \\ g_2^i \\ \dots \\ g_n^i \end{bmatrix} = \begin{bmatrix} \varphi\prime(v_1^i) & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \varphi\prime(v_n^i) \end{bmatrix} \cdot \begin{bmatrix} w_{1,1}^{i+1} & \dots & w_{1,n}^{i+1} \\ \dots & & \dots \\ w_{m,1}^{i+1} & \dots & w_{m,n}^{i+1} \end{bmatrix}^{\mathrm{T}} \cdot \begin{bmatrix} g_1^{i+1} \\ g_2^{i+1} \\ \dots \\ g_m^{i+1} \end{bmatrix} \tag{3.8}$$

Finally, the weight space gradients, including bias, may be computed as shown in (3.9).

$$\mathbf{G}^i = \left[\begin{array}{c|ccc} \delta b_1^i & \delta w_{1,1}^i & \dots & \delta w_{1,n}^i \\ \dots & \dots & & \dots \\ \delta b_m^i & \delta w_{m,1}^i & \dots & \delta w_{m,n}^i \end{array} \right] = \begin{bmatrix} g_1^i \\ g_2^i \\ \dots \\ g_m^i \end{bmatrix} \cdot \left[\begin{array}{c} 1 \\ \hline y_1^{i-1} \\ y_2^{i-1} \\ \dots \\ y_1^{i-1} \end{array} \right]^{\mathrm{T}} \tag{3.9}$$

3.1.6 Backpropagation Training

Backpropagation is perhaps the most common training algorithm for ANNs. It is a simple steepest descent search, meaning the network weights are updated by subtracting the weight space error gradients multiplied with a nonnegative scalar, as shown in (3.10) and (3.11).

$$\begin{aligned} \eta &: \text{ training coefficient} \\ \mathbf{W}(n) &: \text{ arbitrary weight matrix at epoch } n \\ \mathbf{G}(n) &: \text{ gradients for the same matrix at epoch } n \end{aligned}$$

$$\Delta\mathbf{W}(n) = -\eta \cdot \mathbf{G}(n) \tag{3.10}$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \Delta\mathbf{W}(n) \tag{3.11}$$

Some drawbacks of backpropagation is that it tends to train the last layers faster than the first ones, and convergence is rather slow for batch training. A very small or decreasing η should also be used for stable convergence, i.e., to avoid "overshoots" in weight updates.

Other algorithms, like quickprop [21] and conjugate gradient searches [20], attempt to speed up the training by making certain assumptions about the error surface. Yet another popular algorithm is R-prop [22], short for resilient propagation, which attempts to train all the weights at the same pace.

3.1.7 Network Initialization

Prior to training, the network weights must be initialized. This should be done in a random fashion since the optimum weights are not known, and a deterministic algorithm might consistently cause the training to converge on a local error minimum rather than a global. However, a truly random initialization of the weights may actually slow down the training. This may occur due to mainly two reasons:

- saturated neurons
- small weights

Both of these will tend to cause very small weight space gradients, which can be unfortunate for initial training.

To avoid saturated neurons, one should take the mean values and ranges of the training vector elements into consideration when setting the weights. By, e.g., subtracting the mean values and dividing by the standard deviations[1] in the input layer weights, such saturation can be reduced to a great extent. In the output layer the opposite must be done. This essentially means that the input and output layers are initialized to perform a normalization and de-normalization of the input and output vectors, respectively. Some randomness may of course still be applied to these weights.

For the hidden layers, one should also take care to avoid saturation, but still randomize the weights. Haykin [20] therefore suggests drawing these weights from a zero-mean, uniform distribution with a variance that is reciprocal to the number of connected neurons.

3.1.8 Avoiding Overtraining

When training a neural network, care must be taken to avoid *overtraining*. Overtraining means that the network becomes specialized on the training set and therefore *generalizes* poorly.

A common way of avoiding overtraining is the so called *early stopping method*. With this method, the training set is split in two subsets: one set for training, and one set for verification. The verification set is simply used to measure how well the ANN generalizes after each training epoch, and is not used for the training itself. Assuming the two error curves evolve as in figure 3.3, the network that generalizes best is indicated by the minima of the verification error.

3.2 Gaussian Mixture Models

Data from complex real world processes rarely have probability distributions that are known in advance or are easily describable. Some form of modelling is therefore necessary to approximate the distributions.

A common approach to this is probability mixture modelling. This may be thought of as a missing data problem in which the missing data is a classification of each training vector. The idea behind mixture modelling is to divide the probability mass into clusters (or classes), each with their own probability distribution and apriori probability. The perhaps most common variant of this modelling technique is Gaussian mixture modelling (GMM) [3].

[1] The necessary scaling actually depends on both the activation function and the distribution of the values.

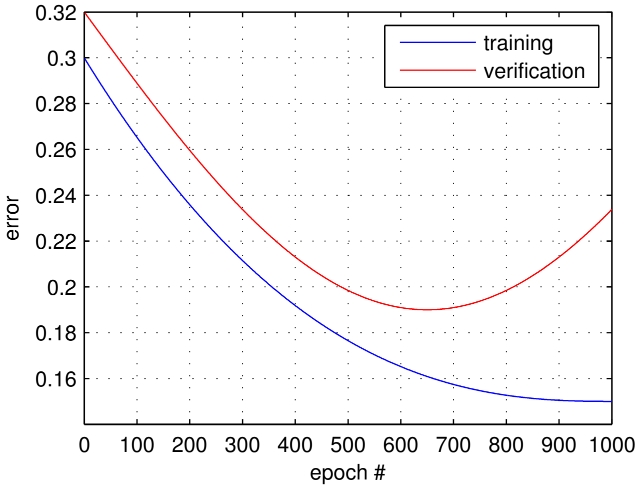

Figure 3.3: Example of error plot for training and verification

GMM gets its name from the fact that the clusters are modelled as Gaussian distributions. Each such cluster, with its distribution and apriori probability is called a *mixture component*. The result of this modelling is a probability density function as shown in (3.12), consisting of N mixture components. Note that the apriori probabilities must sum up to 1.

$$\begin{aligned} p(...) &: \text{probability density function for } \mathbf{x} \text{ vector} \\ \alpha_i &: \text{apriori probability for component } i \\ \Theta_i &: \text{Gaussian distribution parameters for component } i \end{aligned}$$

$$p(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \cdot \mathcal{N}(\mathbf{x}|\Theta_i) \tag{3.12}$$

For the use of GMM intended in this thesis, both joint and marginal probabilities are of interest. This means that $\mathbf{x}$ actually consists of two vectors, say $\mathbf{a}$ and $\mathbf{b}$. This is not of any consequence before the models are to be used in regression, which is presented in 3.2.3.

3.2.1 Training GMMs

The goal of mixture modelling is to estimate the mixture components that best describe the probability distribution of the training set. A common approach to this is the iterative Expectation Maximization (EM) algorithm [3]. This algorithm consists of two steps, from which its name stems.

The first step, expectation, is a computation of weights that indicate the "membership probability" of each training vector to the individual mixture components. This is shown in (3.13), where

i specifies a mixture component and j specifies a training vector.

$$w_{i,j} = \frac{\alpha_i \cdot \mathcal{N}(\mathbf{x}_j|\Theta_i)}{\sum\limits_{k=1}^{N} \alpha_k \cdot \mathcal{N}(\mathbf{x}_j|\Theta_k)} \tag{3.13}$$

The second step, maximization, then aims to maximize the accumulated probability density of the training set. To this end, a new apriori probability for each mixture component is first estimated as the average probability of membership, as shown in (3.14).

$$\alpha_i = \frac{1}{M} \sum_{m=1}^{M} w_{i,m} \tag{3.14}$$

Next, the parameters of the mixture components' Gaussian distributions, Θ_i, are updated. New mean vectors are first estimated according to a relative membership weighting, as shown in (3.15).

$$\mu_i = \frac{\sum\limits_{j=1}^{M} \mathbf{x}_j \cdot w_{i,j}}{\sum\limits_{k=1}^{M} w_{i,k}} = \frac{\sum\limits_{j=1}^{M} \mathbf{x}_j \cdot w_{i,j}}{M \cdot \alpha_i} \tag{3.15}$$

Finally, using these mean vectors, new covariance matrices for the mixture components are estimated. This is also done using the relative membership weighting method, as shown in (3.16). Note that for use in regression, full covariance matrices should be used to exploit the correlation between vector elements.

$$\mathbf{\Sigma}_i = \frac{\sum\limits_{j=1}^{M} w_{i,j} \cdot (\mathbf{x}_j - \mu_i) \cdot (\mathbf{x}_j - \mu_i)^{\mathrm{T}}}{M \cdot \alpha_i} \tag{3.16}$$

These two steps are then repeated until a convergence criteria is fulfilled. For example, the rate of change in total accumulated probability density for the training set may be used as an indicator of convergence. It should be noted that the EM algorithm is only guaranteed to converge on a local maximum.

3.2.2 Mixture Initialization and Splitting

The EM algorithm requires prior estimates of the mixture components. For this reason, some form of initialization is necessary. Assuming an N mixture model is wanted, one has the choice between initializing all, part or just one of the N mixture components before training.

In the two latter cases, the model must be expanded between training sessions by splitting one or more of the existing mixture components. Such *mixture splitting* is usually done by copying the selected mixture component(s), halving the apriori probabilities of both copy and original, and applying small, random deviations to the mean vectors. There is no set strategy for selecting candidates for mixture splitting, but one example is comparison of apriori probabilities and number of earlier splits [23]. Starting with a single mixture component and adding one at a time is assumed to be the best strategy, though it can also be the most time consuming.

For initialization of several mixture components, clustering of the dataset by means of vector quantization is common. K-means and LBG, short for Linde–Buzo–Grey, are the perhaps most popular algorithms for this [3]. The resulting clusters are then used for estimating the mixture components.

Regarding the number of mixtures to use, an upper bound was suggested in [10]. They found that a ratio of at least 100 of the number of training vectors to the number of parameters in the GMM worked well. For N_{vec} training vectors of dimension N_{dim}, the maximum number of mixture components with full covariance matrices may then be computed as

$$N_{\text{mix}} = \left\lfloor \frac{N_{\text{vec}}}{100 \cdot \left(1 + N_{\text{dim}} + \frac{N_{\text{dim}} \cdot (N_{\text{dim}}+1)}{2}\right)} \right\rfloor \tag{3.17}$$

3.2.3 GMM-Based Regression

Regression based on GMM means computation of the most probable output vector $\mathbf{b}$, given an input vector $\mathbf{a}$ as shown in (3.18) [24]. This is essentially a soft decision classification of $\mathbf{b}$, with interpolation of all the $\mathbf{a}$ estimated for the individual classes.

For regression to be possible, the GMM must have been trained on a combination of these vectors, i.e., $\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$, to give a joint probability distribution.

$$\mathrm{E}\left[\mathbf{b}|\mathbf{a}\right] = \frac{\sum\limits_{i=1}^{N} \alpha_i \cdot \mathcal{N}(\mathbf{a}|\Theta_i) \cdot \underset{\mathbf{b}}{\arg\max}\left(\mathcal{N}(\mathbf{a},\mathbf{b}|\Theta_i)\right)}{\sum\limits_{j=1}^{N} \alpha_j \cdot \mathcal{N}(\mathbf{a}|\Theta_j)} \tag{3.18}$$

Regression relies on both the marginal and joint probability distributions of the model. The simplest way to get the marginal distribution is by utilizing the fact that the covariance matrices, $\boldsymbol{\Sigma}_i$, and mean vectors, μ_i may be split according to the subvectors of $\mathbf{x}$. Assuming $\mathbf{x}$ is composed of two subvectors as mentioned above, the covariance matrices and mean vectors are as shown in (3.20) and (3.19).

$$\mu_i = \begin{bmatrix} \mu_a^i \\ \mu_b^i \end{bmatrix} \tag{3.19}$$

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_{aa}^i & \boldsymbol{\Sigma}_{ab}^i \\ \boldsymbol{\Sigma}_{ba}^i & \boldsymbol{\Sigma}_{bb}^i \end{bmatrix} \tag{3.20}$$

By only using the relevant parts of the mean vector and covariance matrix, the probability density of subvector $\mathbf{a}$ for a given mixture component may be computed as in (3.21). The 2π factor is not strictly necessary in regression due to the normalization in (3.18).

$$\mathcal{N}(\mathbf{a}|\Theta_i) = \frac{1}{\sqrt{(2\pi)^n \cdot |\boldsymbol{\Sigma}_{aa}^i|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu_i)^{\mathrm{T}} \cdot (\boldsymbol{\Sigma}_{\mathbf{aa}}^{\mathbf{i}})^{\mathbf{-1}} \cdot (\mathbf{x}-\mu_i)} \tag{3.21}$$

Computation of the most probable subvector **b** for a given **a** is done by translation and linear mapping, as shown in (3.22).

$$\arg\max_{\mathbf{b}} \mathcal{N}(\mathbf{a}, \mathbf{b}|\Theta_i) = \mu_b^i + \mathbf{\Sigma}_{ba}^i \cdot \left(\mathbf{\Sigma}_{aa}^i\right)^{-1} \cdot \left(\mathbf{a} - \mu_a^i\right) \tag{3.22}$$

Note that if diagonal covariance matrices are used, this computation will always result in the mixture components' respective mean vectors.

3.3 Linear Mapping

Linear regression is probably the simplest estimation method. Regression analysis is used to compute the linear mapping between input and output variables which minimizes the mean square error.

Linear mappings are commonly represented as matrices. Given two training matrices of input and output row vectors, **X** and **Y**, the linear map **A** is derived as shown in (3.23) through (3.25) [25].

$$\mathbf{X} \cdot \mathbf{A} = \mathbf{Y} \tag{3.23}$$

$$\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X} \cdot \mathbf{A} = \mathbf{X}^{\mathrm{T}} \cdot \mathbf{Y} \tag{3.24}$$

$$\mathbf{A} = \left(\mathbf{X}^{\mathrm{T}} \cdot \mathbf{X}\right)^{-1} \cdot \mathbf{X}^{\mathrm{T}} \cdot \mathbf{Y} \tag{3.25}$$

Naturally, this method is best suited for cases where there is a linear, or near-linear relationship between the input and output variables.

Chapter 4

Feature Parameters

In this chapter, a selection of signal feature parameters and spectral envelope parameterizations are introduced. As described in 2.4, these are meant for use in the spectral envelope estimation.

Distance measures and transforms related to the feature parameteres are presented last.

4.1 Cepstral Coefficients

In addition to LP analysis, source-filter separation of a linear system may be performed by means of the *cepstrum* [3, 4, 5]. The cepstrum consists of *cepstral coefficients* (CC) which are suitable for use in estimation mainly due to three properties:

1. CCs have proven to be more robust against distortion than LPCs
2. CCs are well uncorrelated
3. cepstral distance (see 4.6.1) may directly be used as an error measure

The complex[1] cepstrum of a signal is obtained by transforming it to a frequency domain, taking the logarithm of the spectrum, then applying the inverse transform. Source-filter separation is achieved thanks to the logarithm, which turns multiplication (convolution in time domain) into addition. This is shown in (4.1) through (4.4), where the Fourier transform is used.

$$\begin{aligned} x(n) &= e(n) * h(n) && (4.1)\\ &\Downarrow \mathcal{F} \\ X(f) &= E(f) \cdot H(f) && (4.2)\\ \ln(X(f)) &= \ln(E(f)) + \ln(H(f)) && (4.3)\\ &\Downarrow \mathcal{F}^{-1} \\ c_x(n) &= c_e(n) + c_h(n) && (4.4) \end{aligned}$$

The cepstrum is essentially the spectrum of a signal's log-spectrum. Since the filter represents the spectral envelope, a slow changing feature, its CCs will be concentrated around $c_x(0)$. A

[1] By discarding phase information, the real cepstrum is obtained.

common method for extracting the envelope is therefore to simply truncate the cepstrum and transform the remaining CCs back to the frequency domain.

To obtain the impulse response $h(n)$ from a given $c_h(n)$ using the Fourier transform, one can perform the inverse operations in reverse as shown in (4.5).

$$h(n) = \mathcal{F}^{-1}\left\{\mathrm{e}^{\mathcal{F}\{c_h(n)\}}\right\} \tag{4.5}$$

4.2 Linear Prediction Cepstral Coefficients

Though CCs are preferrable over LPCs for use in estimation, transforming a signal back and forth from the cepstral domain is computationally expensive. If the filter and its structure is known, a simpler transform may be found. In the case of LP filters, the equality in (4.6) is of help.

$$\ln(1-x) = -\sum_{n=1}^{\infty}\frac{x^n}{n} \tag{4.6}$$

Using the $\mathcal{Z}$-domain, it is apparent from (4.7) and (4.8) that the cepstrum is computed by recursively filtering the LPCs with themselves and summing the results, resulting in *linear prediction cepstral coefficients* (LPCC). Note that σ, the gain of the LP filter, is actually the power of the excitation extracted in LP analysis. In most cases, this gain is ignored since mainly the spectral envelope's shape is of interest.

$$H(z) = \frac{\sigma}{1-\sum_{i=1}^{p} a_i z^{-i}} = \frac{\sigma}{1-A(z)} \tag{4.7}$$

$$\ln(H(z)) = \ln(\sigma) - \ln(1-A(z)) = \ln(\sigma) + \sum_{n=1}^{\infty}\frac{A^n(z)}{n} \tag{4.8}$$

This may be simplified into the transforms shown in (4.9) through (4.11) [4].

$$c_0 = \ln(\sigma) \qquad \text{for} \quad p = 0 \tag{4.9}$$

$$c_m = a_m + \frac{1}{m}\sum_{k=1}^{m-1}(m-k)\cdot a_k \cdot c_{m-k} \qquad \text{for} \quad 1 \le m \le p \tag{4.10}$$

$$c_m = \frac{1}{m}\sum_{k=1}^{p}(m-k)\cdot a_k \cdot c_{m-k} \qquad \text{for} \quad p < m < n \tag{4.11}$$

Reconstruction of LPCs may be done by inverting (4.10), resulting in (4.12). Note that it is only necessary to have just as many CCs as there were LPCs originally.

$$a_m = c_m + \frac{1}{m}\sum_{k=1}^{m-1}(m-k)\cdot c_{m-k}\cdot a_k \tag{4.12}$$

This inverse transform is based on the assumption that the CCs stem from a specific filter type. However, this assumption may not hold if the CCs are distorted or noisy due to, e.g., estimation. To avoid stability issues in such cases, the frequency domain may be used to obtain the autocorrelation function. LPCs can then be computed according to the autocorrelation method by use of, e.g., the Levinson–Durbin algorithm [1].

The autocorrelation function can be obtained as shown in (4.13) through (4.15).

$$H(f) = \mathrm{e}^{\mathcal{F}\{c_h(n)\}} \tag{4.13}$$

$$R_h(f) = H(f) \cdot H^*(f) \tag{4.14}$$

$$r_h(n) = \mathcal{F}^{-1}\left\{R_h(f)\right\} \tag{4.15}$$

Note that a sufficiently large DFT must be used to avoid aliasing in the time domain. Assuming the cepstrum was obtained from an analysis frame of length N, the autocorrelation function will in theory have at most $2 \cdot N - 1$ non-zero elements. Hence, a DFT of length $2 \cdot N - 1$ must be used to avoid aliasing.

4.3 Mel Frequency Cepstral Coefficients

Another form of CC, based on a more perceptual approach, is the Mel frequency cepstral coefficient (MFCC) [3, 8, 9]. MFCCs are used extensively in speech recognition as they have proven to be well decorrelated with respect to classes of speech sounds.

MFCCs are based on a signal's power spectrum, computed by use of a DFT. The power spectrum is passed through a special filterbank, called a *Mel filter bank*, in which the dimensionality is reduced. The logarithm of the filters' accumulated powers are then passed through a discrete cosine transform (DCT, see 4.6.4) to produce MFCCs. An overview of the entire MFCC generation process is shown in figure 4.1.

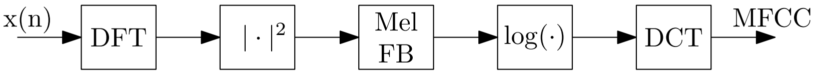

Figure 4.1: Chain of computations for Mel frequency cepstral coefficients

The Mel filter bank consists of overlapping, triangular bandpass filters that are spaced uniformly in Mel frequency to approximate the frequency resolution of the human ear. The transformation from Hertz to Mel is given by

$$f_{\mathrm{Mel}}\left(f_{\mathrm{Hz}}\right) = 2595 \cdot \log_{10}\left(1 + \frac{f_{\mathrm{Hz}}}{700\ \mathrm{Hz}}\right) \tag{4.16}$$

Assuming N filters with 50% overlap are to be placed in the frequency range $F_{\min} - F_{\max}$ Hz, the distance between their center frequencies is

$$\Delta M = \frac{f_{\mathrm{Mel}}(F_{\max}) - f_{\mathrm{Mel}}(F_{\min})}{N+1} \tag{4.17}$$

The N triangular filters are then centered at $f_{\text{Mel}}(F_{\min}) + k \cdot \Delta M$, where $k \in [1...N]$. Converting these frequencies back to Hertz is then a trivial matter.

If the filter bank is meant to function as a sampling of the power spectrum, the filters must be normalized so that they each have an “area” of 1. This is necessary if reconstruction of the power spectrum from the Mel filter powers is wanted. Figure 4.2 illustrates a Mel filter bank in Mel and linear frequency, with normalization in the latter case.

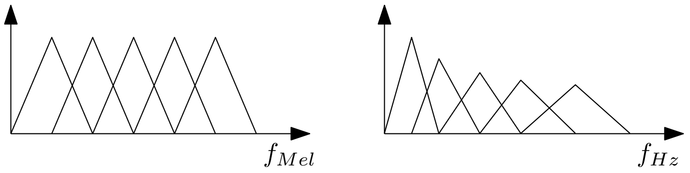

Figure 4.2: Sketch of Mel filter distribution in Mel and Hertz

It should be noted that truncation of the MFCCs to reduce dimensionality is common. Due to the dimensionality reduction in the filter bank and possible coefficient truncation, perfect recreation of the power spectrum is not possible.

4.4 Gradient Index

The gradient index (GI) is a measure of an analysis frame’s voicedness [4, 11]. It is computed by counting all the gradient sign changes, weighted by the absolute value of the corresponding gradients, and normalizing by the square root of the frame’s energy as shown in (4.18) and (4.19). This efficiently captures the differences in “smoothness” for voiced and unvoiced speech signals, without any dependence on amplitude (disregarding noise).

$$x_{\text{GI}} = \frac{\sum\limits_{n=2}^{N} \Psi(n) \cdot |x(n) - x(n-1)|}{\sqrt{\sum\limits_{n=1}^{N} x(n)^2}} \tag{4.18}$$

$$\Psi(n) = \frac{1}{2} \cdot |\text{sgn}\,(x(n) - x(n-1)) - \text{sgn}\,(x(n-1) - x(n-2))| \tag{4.19}$$

4.5 Normed Frame Energy

Measures of energy are a good way to distinguish speech from silence, as well as between different speech sounds. To avoid errors due to varying gains and dynamic ranges between speakers, a normalized energy measure can be used. Normed frame energy (NFE) is such a measure [4, 11, 5], and is defined as shown in equations (4.20) through (4.22).

$$E(n): \text{ energy of frame } n$$

$$NFE(n) = \frac{\log E(n) - \log E_{\text{min}}(n)}{\log \overline{E}(n) - \log E_{\text{min}}(n)} \tag{4.20}$$

$$E_{\text{min}}(n) = \min_{m=0}^{N_{\text{min}}} E(n-m) \tag{4.21}$$

$$\overline{E}(n) = \alpha \cdot \overline{E}(n-1) + (1-\alpha) \cdot E(n) \tag{4.22}$$

Suggested values for N_{min} and the forgetting factor α are 4 seconds worth of frame energies and 0.96, respectively. For initialization of the energy memory $\overline{E}$, a predetermined constant or the mean frame energy (in training and test) are reasonable choices.

4.6 Related Error Measures and Transforms

In this section, a selection of distance measures and transforms related to the feature parameters are described.

4.6.1 Cepstral Distance

Cepstral distance is defined as shown in (4.23).

$\mathbf{c}$: desired vector of cepstral coefficients

$\hat{\mathbf{c}}$: estimated vector of cepstral coefficients

$$d_{\text{ceps}}(\mathbf{c}, \hat{\mathbf{c}}) = \sqrt{\sum_{k=-\infty}^{\infty} (c_k - \hat{c}_k)^2} \tag{4.23}$$

The use of this distance measure is interesting because it can be shown to be proportional to RMS-LSD [5], which is presented in 4.6.2.

Consequently, training an estimator to minimize cepstral distance is analogous to minimizing the error of the estimated spectral envelopes, which is precisely what is wanted.

4.6.2 Spectral Distance Measures

For measurement of distance between spectral envelopes, several methods exist. Two distance measures that are much used are the log spectral distance (LSD), shown in (4.24), and the root mean square log spectral distance (RMS-LSD), shown in (4.25) [26, 5]. It should be noted that the precise definitions seem to either vary or are frequently mistyped in literature.

$$d_{\text{LSD}} = \frac{1}{f_{\text{high}} - f_{\text{low}}} \int_{f_{\text{low}}}^{f_{\text{high}}} \left| 20 \cdot \log_{10} \left| \frac{\hat{H}(f)}{H(f)} \right| \right| \mathrm{d}f \tag{4.24}$$

$$d_{\mathrm{RMS-LSD}} = \sqrt{\frac{1}{f_{\mathrm{high}} - f_{\mathrm{low}}} \int\limits_{f_{\mathrm{low}}}^{f_{\mathrm{high}}} \left| 20 \cdot \log_{10} \left| \frac{\hat{H}(f)}{H(f)} \right| \right|^2 \mathrm{d}f} \tag{4.25}$$

In BWE, the main use of spectral distance measures is in measuring the performance of the spectral envelope extension. Given an estimated and desired spectral envelope, the distance between them can be measured in relevant parts of the spectrum to give an indication of the estimator's performance.

For the spectral distance measures implemented in this thesis, the integrals are approximated by summations with a frequency resolution of 50 Hz. Normalization is therefore done using the number of summands, resulting in the sample mean.

4.6.3 Cepstral Liftering

The dynamic range and variance of the CCs tend to decrease for higher coefficients [27]. To counteract this effect, it is common to perform *liftering*, which is simply a weighting of the coefficients. One such popular lifter is shown in (4.26).

$$\begin{aligned} N_{\mathrm{CC}} &: \text{ number of cepstral coefficients} \\ k &: \text{ multiplication factor, usually} \geq 2 \end{aligned}$$

$$l(n) = 1 + \frac{k \cdot N_{\mathrm{CC}}}{2} \cdot \sin\left(\frac{n \cdot \pi}{k \cdot N_{\mathrm{CC}}}\right) \tag{4.26}$$

In the context of estimator training, liftering of the desired output CCs has the effect of increasing the sensitivity to errors in higher coefficients during training. De-liftering must then be used on the estimated CCs. This way, one can avoid that the estimator "neglects" the higher coefficients, since they are also important for the spectral envelope.

4.6.4 Discrete Cosine Transform

A discrete cosine transform (DCT) is used in the computation of MFCCs. There are several variants of DCTs in existence, but presented here are only the type-II DCT [3]

$$c(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{(2 \cdot n + 1) \cdot k\pi}{2N}\right) \quad \text{for } k \in [0, 1, ..., N-1] \tag{4.27}$$

and its inverse, the type-III DCT

$$x(n) = \sqrt{\frac{2}{N}} \cdot \left[\frac{c(0)}{2} + \sum_{k=1}^{N-1} c(k) \cdot \cos\left(\frac{(2 \cdot n + 1) \cdot k\pi}{2N}\right)\right] \quad \text{for } n \in [0, 1, ..., N-1] \tag{4.28}$$

Interpolation may be performed with the inverse DCT by allowing n to take fractional values. This is particularily useful for recreating power spectra from MFCCs. In this case n must also be warped to translate from Mel to linear frequency as shown in (4.29).

$$\begin{aligned} f_{\mathrm{Hz}} &: \text{frequency of arbitrary DFT bin} \\ M_{\min} &: \text{center Mel frequency of lowest filter} \\ M_{\max} &: \text{center Mel frequency of highest filter} \\ \Delta M &: \text{spacing of filters in Mel frequency} \end{aligned}$$

$$n(f_{\mathrm{Hz}}) = \frac{f_{\mathrm{Mel}}(f_{\mathrm{Hz}}) - M_{\min}}{\Delta M} \tag{4.29}$$

To avoid aliasing, $f_{\mathrm{Mel}}(f_{\mathrm{Hz}})$ must be clamped to the interval $[M_{\min}, M_{\max}]$. This is the same as clamping n to the interval $[0...N-1]$.

4.6.5 Dynamic Feature Parameters

Temporal information about a signal is often represented with dynamic feature parameters, which are estimates of the rates of change for the static[2] feature parameters. These are usually computed for an analysis frame by performing regression analysis on a symmetric set of neighbouring frames. The derivative of the regression function is then an estimate of the rate of change.

In [12], linear regression was used. The transform for this case is shown in equation (4.30) for a given parameter $p(n)$ and a maximum, symmetric frame offset N.

$$\delta p(n) = \frac{\sum\limits_{\theta=1}^{N} \theta \cdot (p(n+\theta) - p(n-\theta))}{2 \cdot \sum\limits_{\theta=1}^{N} \theta^2} \tag{4.30}$$

Note that the resulting parameters are usually referred to as delta or Δ-coefficients.

[2] Computed from a single analysis frame.

Chapter 5

Implemented Systems

For this thesis, it was desired to implement one or more state-of-the-art systems for speaker-independent BWE to test, compare and possibly improve upon. Literature studies revealed that a multitude of systems had been proposed, without any particular approach seeming to be the most prevalent.

System performance was in most cases reported in the form of objective measurements of spectral envelope estimation error. Finding the "best" systems by comparing their performance proved to be difficult, however, because the measurement methods also differed. In addition, one should be wary about drawing conclusions based on objective analysis of systems that are ultimately meant for human senses. This is especially important when comparing fundamentally different systems, since their errors and artefacts may differ in perceptual importance.

Subjective comparisons would perhaps have been the best way to select systems for further study, but since listening examples were not available[1], this could not be done. It was therefore decided to implement systems based on two in particular, mainly because their approaches to spectral envelope extension were found to be interesting.

The two implemented BWE systems are described in detail in the following sections, but some common properties should first be mentioned:

- focus is on extension of the high band
- analysis is done on Hamming-windowed frames of length 20 ms and 50% overlap
- interpolated LP filters (see A.6) with 18 coefficients are used for synthesis
- synthesis is done continuously (as opposed to overlap-add)

Extension to the low band was also attempted, but was eventually abandoned since high band extension proved to be problematic enough.

[1] In only one paper/article out of almost 50, had listening examples at one point been available for download.

5.1 BWE System 1

5.1.1 Overview

This system is a continuation of the work presented by Shahina and Yegnanarayana in [28]. They reported good results in informal listening tests for a speaker-dependent system. It is expected that a speaker-independent version performs worse since it entails more generalisation. However, no objective performance measurements were presented, so no comparisons could be made.

A schematic overview of the implemented system is shown in figure 5.1.

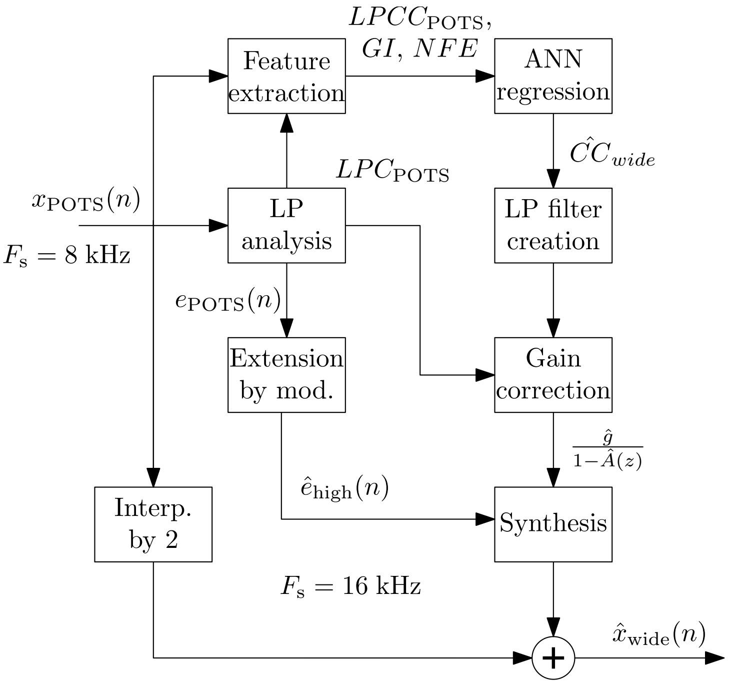

Figure 5.1: Overview of BWE system based on ANN and CCs.

To simulate a real telephone system, the input speech signal is filtered with a POTS bandpass filter (see A.1) and downsampled to 8 kHz sample rate. This allowed for some initial experimentation with different methods of excitation extension. The original POTS band signal is passed through the system unchanged, thus requiring an interpolation by 2.

5.1.2 Extension of Spectral Envelope

This system is based on the use of ANNs (see 3.1) for estimation of the wideband spectral envelope. Since ANNs are trained to minimize the mean square error, it makes sense to use cepstral representation of the spectral envelopes. This is due to properties of the cepstral distance measure (see 4.6.1) and the cepstral coefficients themselves (see 4.1).

To reduce the computational load and simplify the generation of an excitation signal, LP analysis is used. This allows for the cepstrum to be efficiently computed as LPCCs (see 4.2). For representation of the POTS signal's spectral envelope, 12 LPCCs computed from the same number of LPCs are used. Liftering (see 4.6.3) is performed, using a multiplication factor of 2, to equalize the variances prior to estimation.

As recommended by Jax in [11], estimation should not solely be based on the narrowband spectral envelope parameters. Measures of signal power and "voicedness" can be helpful, and are included by use of NFE (see 4.5) and GI (see 4.4). Early experiments showed that this improved the ANN's estimation performance.

The output of the ANN is 18 CCs, which in the training data stem from 18 liftered wideband LPCCs. Liftering was early on found to improve filter stability[2] for estimated cepstra, and was therefore used further. De-liftering must therefore be performed before the estimated cepstrum is processed.

It should be mentioned that Shahina and Yegnanarayana used more LPCCs than LPCs for ANN input and output, with a somewhat hazy explanation. Experiments indicated that filter stability was improved if the simple inverse LPCC transform (see 4.2) was used to compute LPCs from the estimated cepstrum in this case. This does not mean that the filter response was more correct since the extra LPCCs do not add more information (see 4.2), but still complicate the estimation. The improved filter stability was therefore assumed to be a side effect of the spectral envelope extension performing worse, but was not investigated further. The autocorrelation method was used to compute the LPCs in this system, guaranteeing stable filters, so extra LPCCs were deemed unnecessary.

5.1.3 Creation of Synthesis Filter

For computation of wideband LPCs, the simple inverse transform for LPCCs (see 4.2) is not used. This is because stability of the resulting filters cannot be guaranteed for estimated cepstra. Instead, the method based on recreating the autocorrelation function from the power spectrum (also in 4.2) is used to obtain LPCs.

To minimize spectral discontinuities between the pass-through POTS signal and the resynthesized high band, a correction gain is needed. Due to the way the excitation is extended, this gain may simply be computed as the ratio of the average gains of the POTS and estimated wideband LP filters in a part of the POTS band, as shown in (5.1).

$$\hat{g} = \frac{\int_{500\text{ Hz}}^{3.2\text{ kHz}} \left|H_{\text{POTS}}(f)\right| df}{\int_{500\text{ Hz}}^{3.2\text{ kHz}} \left|\hat{H}_{\text{wide}}(f)\right| df} \tag{5.1}$$

In the implemented system, these integrals are approximated by sums with a frequency resolution of 50 Hz.

5.1.4 Extension of Excitation

The POTS excitation signal is extracted during LPC analysis. Based on initial experiments, fixed modulation was chosen for its extension to the high band. The use of nonlinearities, code excited linear prediction[3] and spectral folding were also investigated, but did not result in satisfactory quality.

[2] More specifically, attenuating formants that were perceived as unnaturally loud.

[3] This was tested with a simple Matlab-implementation. A proper CELP-codec could give better results, but is suspected to be a needlessly complex approach.

Since the input signal has a sample rate of 8 kHz, the original POTS excitation must first be interpolated. The interpolated excitation is then modulated with a carrier at 3.4 kHz to extend it into the high band, and multiplied with 2 to correct the amplitude. To avoid spectral overlap with the POTS band, the excitation is highpass filtered before resynthesis.

For details about the filters involved in this process, see appendix A.

5.1.5 Implemented Variants

Since there is no set answer for what the optimal ANN structure is, a selection of different networks were trained to study the effect on BWE performance.

The input and output layers were linear, with just as many neurons as there were input and output variables. Hence, the input layer consisted of 14 neurons (12 LPCCs, GI and NFE), while the output layer consisted of 18 neurons (18 CCs).

As for the number of hidden, nonlinear layers, the common choice seemed to be either one or two [15, 29, 30, 28, 26]. A reasonable lower bound for the number of hidden neurons is the number of output neurons. It was therefore decided to train networks with one and two hidden layers, and integer multiples (up to 4) of 18 neurons per layer. All in all, this meant a selection of eight different ANN structures.

Also, due to the inherent randomness in initializing and training ANNs, 5 “iterations” of each network structure were trained. Only the ones which performed best in the objective analysis experiments (see 7) were used further.

5.2 BWE System 2

5.2.1 Overview

This system is a variant of those proposed by Kabal and Nour-Eldin in [8, 9]. These were motivated by studies of the mutual information between the POTS and high band spectral envelope for different parametric representations [12, 13]. The studies had also shown that memory of both the low and high band had a beneficial effect, warranting the use of an estimation method capable of utilizing this. They reported results for what is assumed to have been speaker-independent BWE mainly in the form of objective performance measures such as RMS-LSD (see 4.6.2).

A schematic overview of the implemented system is shown in figure 5.2. Note that although this system looks simpler than the previously presented system, this is the most complex of the two.

The input to this system is POTS filtered speech with a sample rate of 16 kHz, thus representing an already interpolated POTS signal. In this system, too, the original POTS band is passed through without any change.

5.2.2 Extension of Spectral Envelope

This system uses GMM (see 3.2) for statistical estimation of the spectral envelope. MFCCs (see 4.3) are used for its representation.

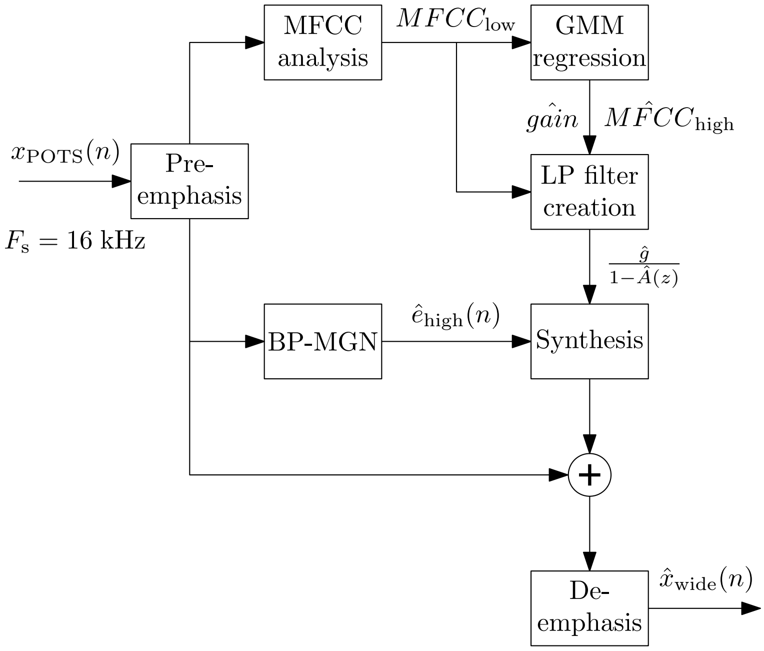

Figure 5.2: Overview of BWE system based on GMM and MFCCs.

Prior to analysis, the interpolated POTS signal is passed through a *pre-emphasis* filter, consisting of a single pole at $z = -0.97$. This is done to flatten the spectrum by countering the -6 dB per octave spectral slope which is common for speech. Since all analysis is done on pre-emphasized signals, *de-emphasis* must be applied on the bandwidth extended signal prior to output.

The Mel filter bank in this system consists of 21 filters and is described in detail in A.5. It is divided in low and high band sections, consisting of 15 and 6 filters, respectively. From this filterbank, 15 MFCCs are computed for the low band and 6 for the high band.

For the optional incorporation of memory, ΔMFCCs (see 4.6.5) in both bands are used. This is done to improve the GMM by increasing the inter-band mutual information during training.

During BWE, only the low band MFCCs are known. The high band MFCCs are then statistically estimated using GMM-based regression. In the system with memory, the estimation is based on both static and ΔMFCCs of the low band. The high band ΔMFCCs are also estimated in this case, though they only serve a purpose during training of the GMM.

5.2.3 Creation of Synthesis Filter

From the estimated high band MFCCs, the corresponding band of the power spectrum can be recreated by use of an inverse DCT (see 4.6.4) and subsequent exponentiation. By applying an inverse DFT to the power spectrum, one gets the autocorrelation function from which the LPCs can be obtained just as in BWE System 1.

To avoid aliasing in the autocorrelation function, an inverse DFT of length $2 \cdot N - 1$ must in theory be used for the power spectrum, where N is the length of the DFT used in MFCC analysis. The inverse DCT must therefore interpolate in addition to warp the frequency axis from Mel to Hertz.

Initial experiments with the correct highband MFCCs indicated that the best results were obtained if LP filters for the entire wideband spectral envelope were estimated, rather than for just the high band. The reason for this was that the LP filters otherwise were found to cause unpleasant resonances near the discontinuous edges in the power spectrum. Since the POTS band power spectrum is known from the MFCC analysis, recreating the power spectrum for the combined POTS and high band is unproblematic.

To simplify the interpolation of the POTS band power spectrum, another inverse DCT is applied to the 15 low band MFCCs. Note that this method gives a reduced spectral resolution for increasing frequency, but is used anyway since mainly the high band is of interest.

The entire power spectrum is thus recreated from both the known and the estimated MFCCs, with the transition frequency between the low and high band recreation defined as right between the highest and lowest Mel-filter in the two bands.

5.2.4 Extension of Excitation

LP analysis is not used in this system, so the excitation must be created by other means. This is done using BP-MGN (see 2.3.2), which also requires the estimation of a correction gain. For the training data, this gain is computed from the ratio of energy in the high band of the original wideband and BWE signals. The gain is thus also used to correct for any spectral discontinuity due to the synthesis filter.

Since both the power of the excitation and estimated synthesis filter depend on the spectral envelope of the POTS signal, this gain is also estimated from the low band MFCCs. It is therefore another parameter that may be statistically estimated by use of GMM-based regression.

5.2.5 Implemented Variants

To study the difference between the case of memoryless estimation and estimation with memory, two system variants were implemented. These relied on a single GMM for estimation of both high band MFCCs and the excitation gain. The dimensionalities of the feature vectors chosen for the two variants were:

- 17 for memoryless system:
  - 10 low band MFCC
  - 6 high band MFCC
  - correction gain
- 27 for system with memory:
  - 10 low band MFCC
  - 6 low band ΔMFCC
  - 6 high band MFCC
  - 4 high band ΔMFCC
  - correction gain

Regarding the span of memory, 100 ms was chosen since it was reported to give the highest certainty about the high band in [9]. This corresponds to a symmetric window covering 10 frames (5 past and 5 future frames), which translates to 50 ms worth of prebuffering delay.

All variants were trained with up to 64 mixture component GMMs to study the effect of model complexity on estimation performance. This upper limit was chosen due to the time required to perform the training (almost 4 days for both GMMs) and the size of the training set (see 6.1). Note that according to the upper bound given in 3.2.2, no more than 54 mixture components should actually be used for the memory-based GMM for the chosen training set.

Chapter 6

Tools

In this chapter, the tools and data used to implement, train and test the systems are presented. Accordingly, the training methods for the estimators are also explained here.

6.1 Speech Database

The speech database of Nordisk Språkteknologi (NST) was used for this thesis. This database contains recordings of Norwegian speech with 16 kHz sample rate at 16 bit resolution, and is split in a training and test part.

The training part consists of approximately 311 recordings of speech times 893 speakers, while the test part consists of 987 recordings of speech times 78 speakers. In addition, a file containing only recording setup noise is included for each speaker.

For the creation of training data, 50 random speakers from the training part were selected. A selection of 100 sentences, all differing across the speakers, were then extracted from each speaker. This gave a total of 5000 sound files.

To trim away leading and trailing silence in the sound files, a simple speech detection method based on the recording setup noise was used:

- For each speaker, the average frame power in the noise recording was first computed to estimate the noise level. This was done using Hamming-windowed frames of length 20 ms and 50% overlap.
- Then, for each speech recording for the current speaker, the first and last frame with a level 6 dB above that of the noise were used as delimiters for a *contiguous* segment of speech.
- To ensure that some low energy frames were available for computation of the NFE feature parameter used in BWE System 1, an extra buffer of 10 frames (100 ms) was included in its training sets.
- After analysis of a POTS and wideband version of this speech segment, the resulting training vectors were sifted to avoid training much on silence. This was done by discarding training vectors that stemmed from frames with a level less than 6 dB above that of the noise.

Without the final sifting, over 2.7 million training vectors were obtained. With the sifting, the number of training vectors was reduced to slightly more than 2.2 million.

6.2 System Implementation

Both BWE systems were implemented in Matlab®[31], which was also used to create the training data.

Apart from the Mel filter bank, all filters were created using the Signal Processing Toolbox™in Matlab. The freeware Voicebox Speech Processing Toolbox [32] was used to create the Mel filter banks, for transforming LPCs to LPCCs, and converting between Mel and linear frequencies.

6.2.1 Artificial Neural Networks

Due to a lack of freely available, versatile Matlab toolboxes for ANNs, a proprietary system for training neural nets was first implemented. Four training algorithms were implemented in this system: batch and sequential backpropagation, quickprop and R-prop. Sequential backpropagation proved to be the fastest training method during this phase.

As the amount of training data was increased, Matlab was found to be too inefficient and memory consuming for ANN training. Consequently, the freeware C library FANN [33] was adopted instead, and a command line tool for training ANNs created. Most of the necessary Matlab bindings were available [34] for this library to be used with Matlab, requiring only the addition of a binding to load pre-trained networks from file.

Training of ANNs was done using sequential backpropagation with a training coefficient $\eta = 0.0001$, which was divided by 1.2 per epoch[1]. The ANNs' weights were initialized before training with "Widrow & Nguyen's algorithm"[2], implemented in the FANN library.

This training method does not necessarily give a monotonic decrease in verification error, so a variant of the early stopping method was used:

- $\frac{2}{3}$ of the training vectors were used for training, and the rest for verification
- verification was performed for every training epoch
- training was done for a maximum of 1000 epochs and halted if the verification error did not decrease during the last 100 epochs

6.2.2 Gaussian Mixture Models

For the training of GMMs, the Hidden Markov Model Toolkit [23] was used.

The GMMs were grown incrementally from a single mixture, by adding 1 mixture component per training session. This method was chosen because it was assumed to result in the best modelling, though it may also be the most time consuming. The training process was automated by use of a Python script borrowed from supervisor Svein Gunnar Pettersen's doctoral thesis [35].

[1] These values were selected based on experimentation

[2] Detailed information about this algorithm was not found.

Chapter 7

Experiments

In this chapter, the experiments done on the implemented BWE systems are described. These deal with objective and subjective evaluation of their performance.

Regarding objective measurements of BWE performance, analysis of the spectral envelope estimation is the most common. There is no set way to do this, which was apparent from the different kinds of results presented in literature on BWE systems. The details of the measurement procedures are also usually unclear. In this thesis, a detailed method for measurement of envelope estimation performance is therefore suggested.

However, since BWE systems are meant to improve the *perceived* quality of telephony speech, the most meaningful analysis of their performance is done using subjective testing. To get reliable, empirical data, formal listening tests should be done. In the simplest case, the test subjects could be presented with a selection of speech clips in both POTS and bandwidth extended versions, and asked to rate which sounds better.

Formal listening tests were not carried out due to the amount of time and resources these require. The subjective analysis of the systems is therefore based on informal listening tests.

7.1 Performance of Spectral Envelope Estimation

For objective evaluation of the BWE systems, the error of their estimated LP filters was measured. This method was used because it does not consider any effect that the excitation extension or gain estimation during BWE may have. Better performance in this respect is therefore an indication of, but not synonymous with higher quality.

To serve as benchmarks, three additional cases were analyzed:

1. extension by linear mapping of liftered CCs
2. extension by linear mapping of liftered CCs, GI and NFE
3. flat filter

The linear maps were created from the same training data as was used for the ANNs, and were meant to represent the absolutely simplest method of extension.

Informal listening tests during implementation indicated that the implemented systems performed well in estimation of the POTS band. Using the same method for gain correction as in BWE System 1, this was utilized to ensure that the response in the POTS band of the estimated LP filters were as similar to that of the wideband reference as possible. For the reference wideband LP filters, 18 LPCs were used, which is the same order as the estimated ones.

Though both BWE System 1 and 2 estimated the wideband (or POTS + high band) envelope, performance in the high band was mainly of interest. The frequency band $4 - 7$ kHz was therefore chosen for analysis, partially because it allowed for some comparison with results reported in [9], on which BWE System 2 was based.

Further, since mainly estimation during speech is interesting, only the performance during frames with a level 12 dB above the estimated noise level (see 6.1) was considered. This simple speech detection was performed on the original wideband signal, and used to sift out the relevant LP filters after extension. The sample means and standard deviations of the LSD and RMS-LSD (see 4.6.2) were then computed.

For test data, 50 random speakers from the test part of the NST database were selected. From each speaker, 15 unique sentences were chosen, resulting in a total of 750 test sentences. With the above mentioned level limit for detection of speech, 280497 spectral envelopes from the bandwidth extended sentences were used in the analysis. In the case of estimation with memory, this number was reduced to 280133 frames due to the buffering required for computation of ΔMFCCs.

7.2 Subjective Evaluation of System Performance

Informal listening tests were also carried out to evaluate the systems. The immediate goal of this was to find out if the systems actually improved the subjective quality, and if so, which system was the preferred one.

In addition, it was desirable to answer questions such as whether the systems:

- consistently introduced artefacts
- helped differentiate consonant sounds (e.g., ’s’ and ’f’, ’p’ and ’t’)
- performed better for some speakers

This was done using a set of 10 files drawn from the objective testing of the spectral envelope estimation. The variants of BWE System 1 and 2 that performed best according to the previously performed objective measurements were used to generate these files.

Chapter 8

Results and Discussion

In this chapter, the objective performance measurements are first presented and discussed. The two systems are then compared with each other, before both are compared with the results reported in the paper on which BWE System 2 was mainly based.

Results of the subjective evaluation of the implemented BWE systems are presented last. No listening examples could be found for systems proposed in literature, so no subjective comparisons with other systems were made.

8.1 Performance of Spectral Envelope Estimation

8.1.1 Benchmark Performance

The sample means and standard deviations of LSD and RMS-LSD for the benchmark cases are shown in table 8.1.

Table 8.1: LSD and RMS-LSD for benchmark cases.

Variant	μ_{LSD} [dB]	σ_{LSD} [dB]	$\mu_{\text{RMS-LSD}}$ [dB]	$\sigma_{\text{RMS-LSD}}$ [dB]
Lin. map of CC	6.973	4.398	7.765	4.444
+ GI & NFE	5.881	3.621	6.681	3.701
Flat filter	20.957	9.732	21.692	9.715

Initial comparisons showed that both BWE System 1 and 2 always performed better than the benchmark cases. Though a larger difference was somehow expected, this agrees with the notions that an estimator's performance and complexity are related, and that the mapping between the POTS and wide/high band envelope is not a linear one.

The difference in performance for the two linear maps also indicated that the use of GI and NFE can improve estimation, though a possible additional cause for the difference could be that GI and NFE allowed for the addition of a "mean vector".

8.1.2 BWE System 1 Performance

Table 8.2 shows the sample means and standard deviations of LSD and RMS-LSD for BWE System 1 with the various ANN structures.

Table 8.2: LSD and RMS-LSD for BWE System 1

Variant	μ_{LSD} [dB]	σ_{LSD} [dB]	$\mu_{\text{RMS-LSD}}$ [dB]	$\sigma_{\text{RMS-LSD}}$ [dB]
1x18	5.522	3.312	6.328	3.416
1x36	5.396	3.210	6.204	3.320
1x54	5.320	3.118	6.126	3.233
1x72	5.294	3.096	6.099	3.212
2x18	5.370	3.192	6.180	3.299
2x36	5.226	3.041	6.033	3.159
2x54	5.169	2.999	5.974	3.125
2x72	5.125	2.977	5.926	3.105

These results indicated that an ANN consisting of two layers with N neurons each performs better than one consisting of a single layer with $2 \cdot N$ neurons. Figures 8.1 and 8.2 help illustrate this.

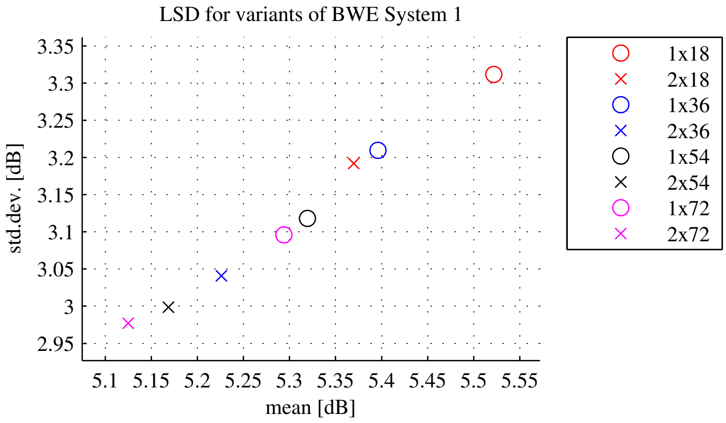

Figure 8.1: LSD for different ANN structures in BWE System 1

Since the former network structure also entails less computations due to the smaller weighting matrices, using two layers seems to be preferrable overall.

Further, adding more neurons always improved the performance. The improvement decreased with the size of the network, however, an effect which was expected due to the bound on performance [4].

As a sidenote, in [7], it is claimed that an RMS-LSD[1] of up to approximately 6.0 dB in the synthesized band allows for "high quality" bandwidth extension. Taking the increase in computational complexity into account, one might therefore not need nor want to use a network much larger than two layers with 36 neurons.

[1] The definition of RMS-LSD in said paper differs from the one used in this thesis, but is assumed to be a misprint.

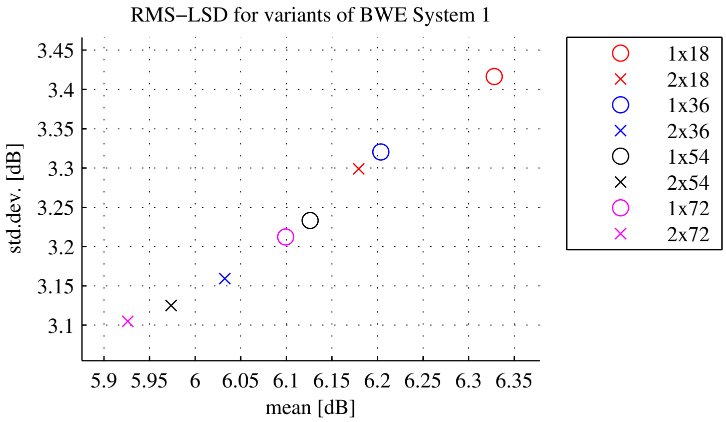

Figure 8.2: RMS-LSD for different ANN structures in BWE System 1

8.1.3 BWE System 2 Performance

Table 8.3 shows the sample means and standard deviations of LSD and RMS-LSD for the two variants of BWE System 2, for GMMs with $N \cdot 16$ mixture components. Note that 54 was the upper bound on mixture components (see 3.2.2) for the second variant and is therefore included as an extra data point.

Table 8.3: LSD and RMS-LSD for BWE System 2

Variant	μ_{LSD} [dB]	σ_{LSD} [dB]	$\mu_{\mathrm{RMS-LSD}}$ [dB]	$\sigma_{\mathrm{RMS-LSD}}$ [dB]
No mem., 16 mix	5.485	3.206	6.307	3.303
32 mix	5.430	3.135	6.250	3.236
48 mix	5.430	3.137	6.251	3.237
54 mix	5.425	3.130	6.247	3.231
64 mix	5.422	3.124	6.243	3.225
With mem., 16 mix	5.376	3.103	6.198	3.206
32 mix	5.297	3.059	6.120	3.169
48 mix	5.266	3.025	6.087	3.136
54 mix (bound)	5.262	3.015	6.084	3.128
64 mix	5.261	2.999	6.083	3.113

For both variants, the results showed minute improvement for each doubling of the number of GMM mixture components.

Figures 8.3 and 8.4 illustrate the test results for the first variant (no memory) of BWE System 2. As can be seen, the greatest gain in performance was achieved when doubling from 16 to 32 mixture components. Adding further mixture components had little effect on performance.

Figures 8.5 and 8.6 illustrate the test results for the second variant (with memory) of BWE System 2, with the same limits of the axes as for the first variant. The greatest gains in performance for the second variant were achieved for up to 48 mixture components in the GMM, compared to the 32 mixture components for the first variant. This was expected since the use of larger feature vectors is likely to result in more clusters, warranting more detailed modelling.

Comparison of the two variants reveals that memory of both bands helped in training of the GMMs, since LSD and RMS-LSD were reduced by respectively 3.0% and 2.6% for GMMs with

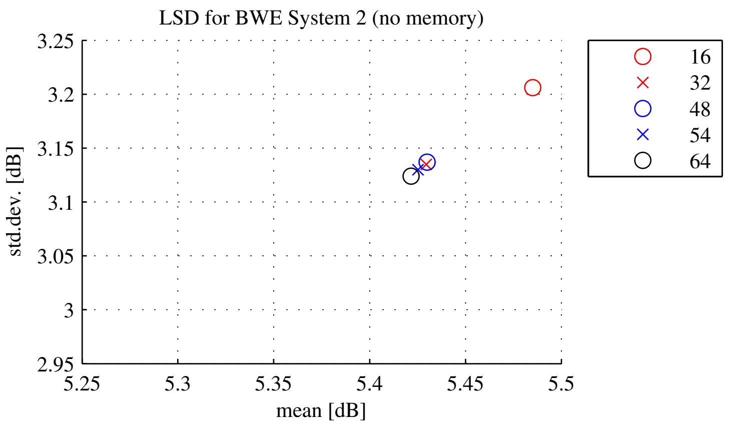

Figure 8.3: LSD for different GMMs in BWE System 2 (no memory)

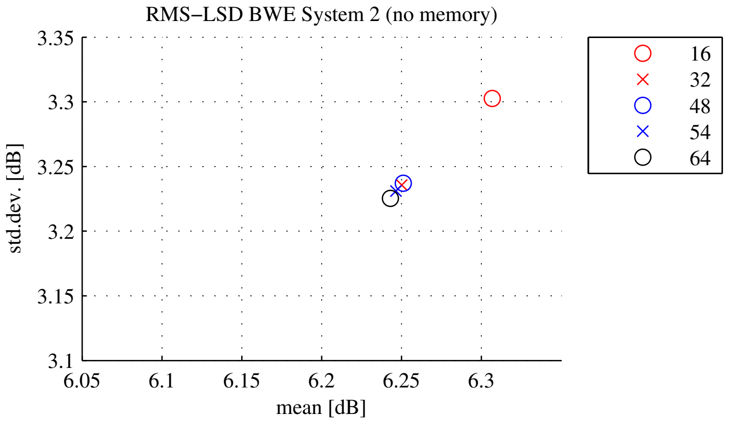

Figure 8.4: RMS-LSD for different GMMs in BWE System 2 (no memory)

more than 48 mixture components. In fact, the smallest GMM in the second variant even outperformed the largest GMM in the first variant of this system. This was unexpected, but is further proof that memory can be very helpful for estimation.

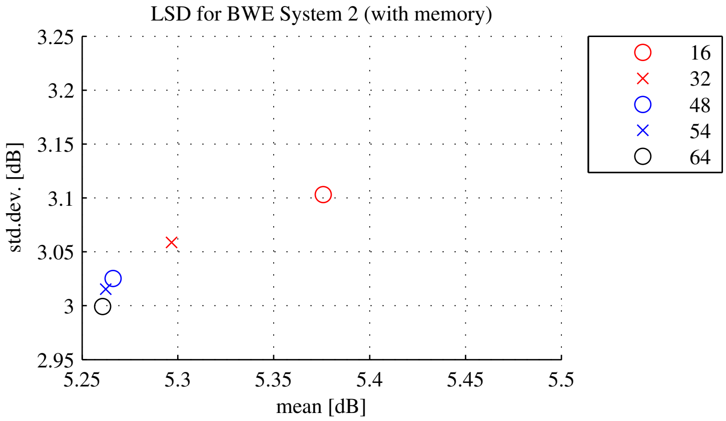

Figure 8.5: LSD for different GMMs in BWE System 2 (with memory)

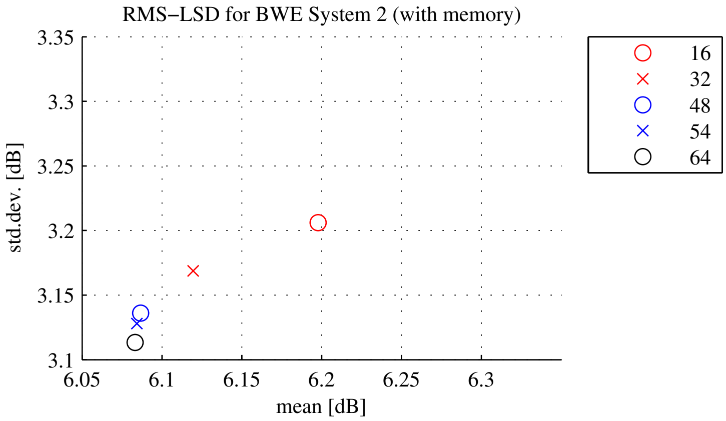

Figure 8.6: RMS-LSD for different GMMs in BWE System 2 (with memory)

8.1.4 Comparing BWE System 1 and 2

For both BWE systems, the most complex estimators always resulted in best performance. But BWE System 2 showed no improvement over BWE System 1 with an ANN consisting of 2 layers with 36 or more neurons. This was surprising because it was assumed that a GMM with many mixtures could better model the feature mapping from POTS to wide/high band than a relatively simple ANN was capable of, even though the two systems were based on slightly different feature vectors.

Disregarding possible implementation errors, two possible causes for BWE System 2's inferior performance were initially suspected:

1. poor recreation of the POTS-band power spectrum in combination with the applied gain correction
2. de-emphasis applied to the estimated spectral envelope

A random selection of estimated and gain corrected LP filters were visually inspected and compared with the reference to investigate these possibilities. The best performing variants of BWE System 1 and 2 were used for this.

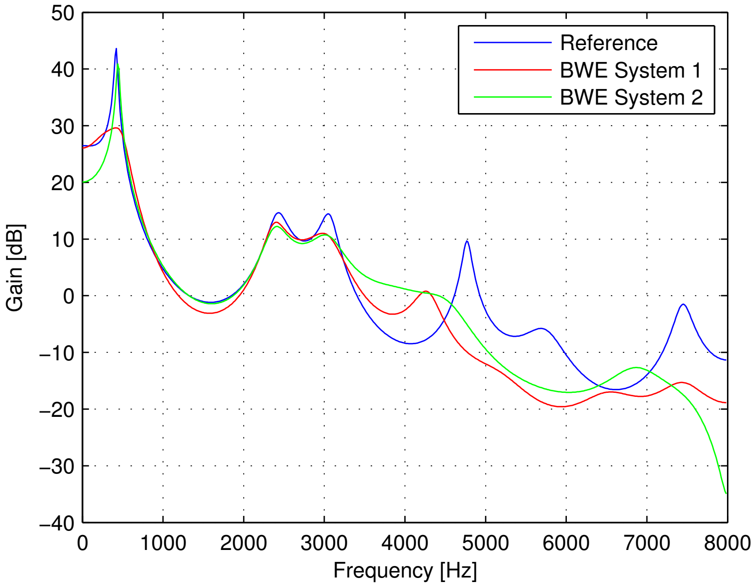

Figure 8.7: Reference and gain corrected, estimated spectral envelopes from 'i' in 'sier'

Figure 8.7 shows a typical case in which the systems seemed to perform equally well (or poor). Since the spectral envelope for frequencies above 7 kHz were not analyzed, de-emphasis was ruled out as a major cause for the inferior performance. And although the POTS band could be seen to suffer from the reduced spectral resolution due to its recreation from the lowband MFCCs[2], it seemed no worse than for BWE System 1. Therefore, the use of gain correction did not seem like a cause for the inferior performance.

Figure 8.8, however, shows an interesting case. The curvature of the estimated high band for BWE System 2 corresponded better with that of the reference, than for BWE System 1. But due to an apparent offset and reduced dynamic range of the high band for BWE System 2, it still had a greater spectral distance than BWE System 1. This indicated that the relatively poor performance could be due to the recreation of the power spectrum, or rather in estimation errors of the mean power in the high band.

In an attempt to avoid this problem, both translation and scaling of the recreated high band power spectrum were tried to minimize discontinuity between the two bands. Though these methods could reduce the spectral distance in some cases, they could just as well increase it in others. The latter proved to be the predominant case, because both methods caused an overall deterioration of performance.

[2] Less pronounced formants, and formants at higher frequencies fused together.

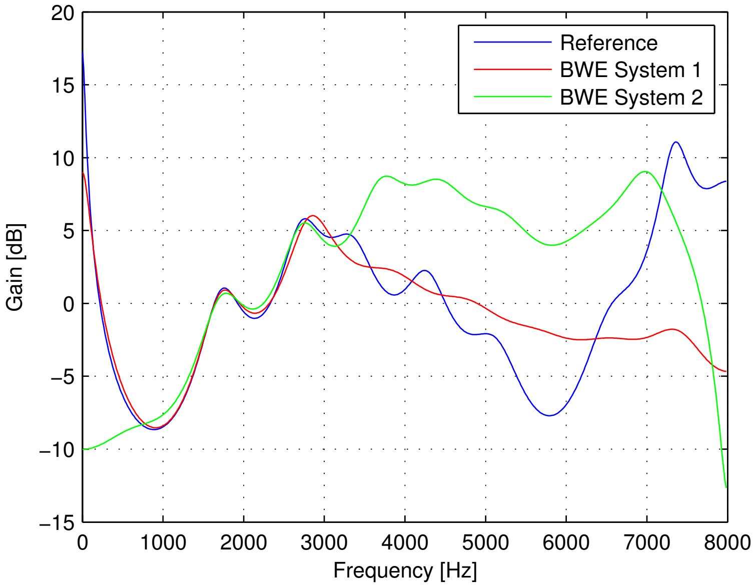

Figure 8.8: Reference and gain corrected, estimated spectral envelopes from 's' in 'sier'

8.1.5 Effect of Gain in BWE System 2's GMM

Since the problem seemed to be with the MFCC estimation, mainly one suspected reason for the relatively poor performance of BWE System 2 was left: the inclusion of correction gain estimates in the GMM. This could possibly affect the mixture splitting (see 3.2.2) in such a way that the added mixture components did not help model the phonemic feature space very well.

To investigate the gain's possible detrimental effect on estimation, a final GMM was trained on MFCCs and ΔMFCCs only. The same number of coefficients as in the second variant of BWE System 2 were used. Table 8.4 shows the estimation performance of this GMM.

Table 8.4: LSD and RMS-LSD for GMM trained on MFCCs exclusively

Variant	μ_{LSD} [dB]	σ_{LSD} [dB]	$\mu_{\text{RMS}-\text{LSD}}$ [dB]	$\sigma_{\text{RMS}-\text{LSD}}$ [dB]
With mem., 16 mix	5.362	3.085	6.177	3.190
32 mix	5.275	3.028	6.089	3.137
48 mix	5.251	3.000	6.066	3.113
54 mix	5.239	2.978	6.054	3.093
58 mix (bound)	5.234	2.977	6.049	3.093
64 mix	5.229	2.971	6.045	3.087

Relative to the second variant of BWE System 2, exclusion of the correction gain caused a reduction of both LSD and RMS-LSD by at most 0.6%, for 64 mixture components. This improvement was not enough to outperform BWE System 1.

In summation, this may indicate that MFCCs are not as suited for BWE as LPCCs or/and that the implemented method for reconstructing LP filters is flawed. The former agrees with results presented in [5], where the use of different spectral envelope parameterizations and vector quantization in extension was evaluated.

8.2 Comparison with Other Systems

Before comparing these results with those reported in literature, some possible issues should be mentioned:

- the definitions of LSD and RMS-LSD either vary or are simply mistyped.
- it is not known if only estimation performance during speech is analysed, or what effect silence may have had on the result.
- the order (or use) of the LP filters is rarely stated.

This, in addition to the use of different training and test data means that differences in measured performance should not be seen as the "absolute truth" regarding one system's superiority over another. It is therefore hard to draw any real conclusions based on these comparisons.

8.2.1 BWE without Memory

In [9], Nour-Eldin and Kabal presented measurements of RMS-LSD[3] for the system on which BWE System 2 was mainly based. Though not all details of the system were presented in this paper, it was inferred that their system:

- used a GMM with 107 mixture components for high band MFCC estimation [12]
- used separate GMMs for estimation of high band MFCCs and correction gain [12]
- was based on a $0 - 4$ kHz signal as narrowband input [9]
- only recreated LPCs for the high band [9]

Especially the first three points can be assumed to improve performance, though it is not known by how much.

In their system, the same number of static low and high band MFCCs (10 and 6) as for BWE System 2 were used. An RMS-LSD of approximately 5.18 dB was reported for this system, which is roughly 0.75 dB better than BWE System 1 with the largest ANN structure. Compared to BWE System 2, it is 1.06 dB and 0.90 dB better than the first and second variants, respectively, with the largest GMMs.

Considering the minute reduction of RMS-LSD for increased GMM complexity in BWE System 2, it seemed unlikely that such performance was achievable. And for BWE System 1, linear regression of RMS-LSD with respect to the number of neurons indicated that roughly 250 extra neurons per layer would be needed to match the performance. This also seemed like a highly unlikely possibility.

A perhaps more interesting comparison would have been between the performance improvements for increasing GMM size, but Nour-Eldin and Kabal did not present such data.

8.2.2 BWE with Memory

In Nour-Eldin and Kabal's BWE system with memory, half of the low and high band static MFCCs were simply replaced with ΔMFCCs, thus maintaining the dimensionality of the feature vectors. For this system, a minimum RMS-LSD of approximately 4.79 dB was reported for a memory span of ±1 frame, which is an improvement of 7.5% from their memoryless system.

This relative improvement is almost three times what was found for BWE System 2 (2.6%). It should also be mentioned that the reported RMS-LSD varied with less than 0.05 dB as the span of the memory was increased up to ±30 frames. These results were curious for two reasons:

1. truncation of highband MFCCs assumedly leads to poorer recreation of the power spectrum
2. the inter-band mutual information is higher for memory spans larger than ±1 frame [9]

It therefore seemed somewhat odd that BWE System 2 did not show roughly the same relative improvement with the introduction of memory. However, it is hard to specify why this was not the case.

[3] They actually state it is LSD, but the formula indicates RMS-LSD with a normalization that is most likely mistyped. The results are assumed to be for RMS-LSD as defined in 4.6.2.

8.3 Subjective Evaluation of System Performance

8.3.1 Preferred System

Informal listening tests showed that both implemented systems improved the perceived quality of POTS speech. BWE System 1 was preferred over BWE System 2, however, because the latter was perceived to be noisier. This could be attributed to the differing methods for excitation extension.

Figures 8.9 through 8.12 show spectrograms of a wideband signal, the corresponding POTS version and output from the two BWE systems, respectively. The uttered phrase is “vi yngre ser jo”[4].

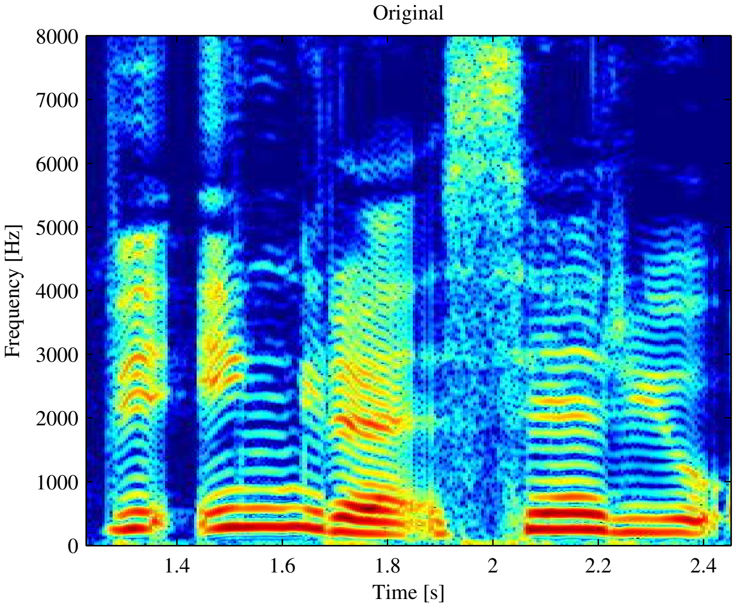

Figure 8.9: Spectrogram of wideband speech (“vi yngre ser jo”).

The horizontal lines in the spectrograms show the pitch structure. In the spectrogram for BWE System 2 (figure 8.12) the pitch structure in the high band is not well defined, which agrees with the perceived noisiness. In addition, the disruptive effect of fixed modulation on pitch structure can be seen in the spectrogram for BWE System 1 (figure 8.11), though this effect was not noticed in the listening tests.

Also, in comparison with the wideband spectrogram (figure 8.9), both BWE systems’ high band spectral envelopes seem smoothed. This is a natural consequence of high band uncertainty and that the systems were trained to be speaker-independent. The use of liftering in BWE System 1 was also a likely cause.

[4] Files 161_*.wav in the test set on the accompanying CD.

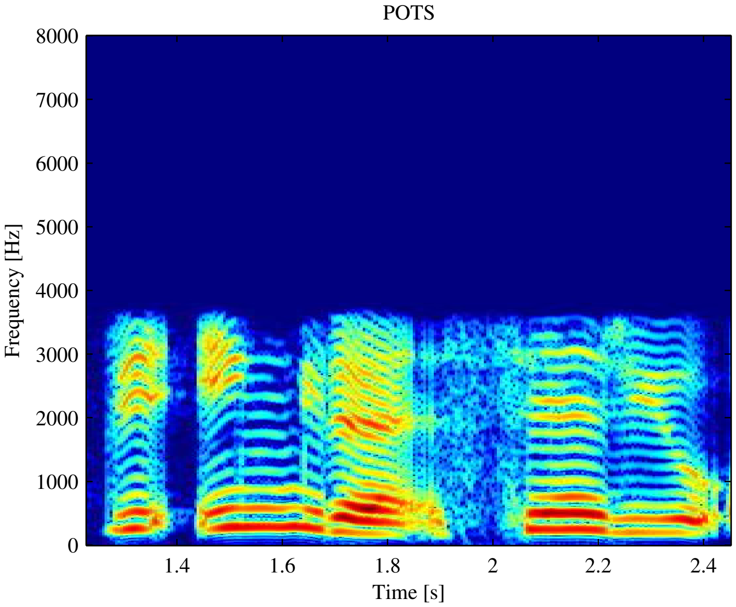

Figure 8.10: Spectrogram of POTS quality speech ("vi yngre ser jo").

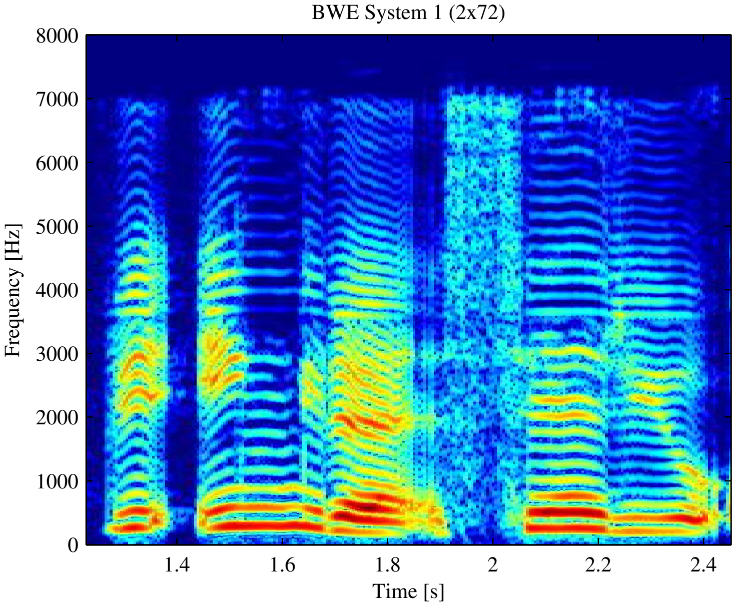

Figure 8.11: Spectrogram of output from BWE System 1 ("vi yngre ser jo").

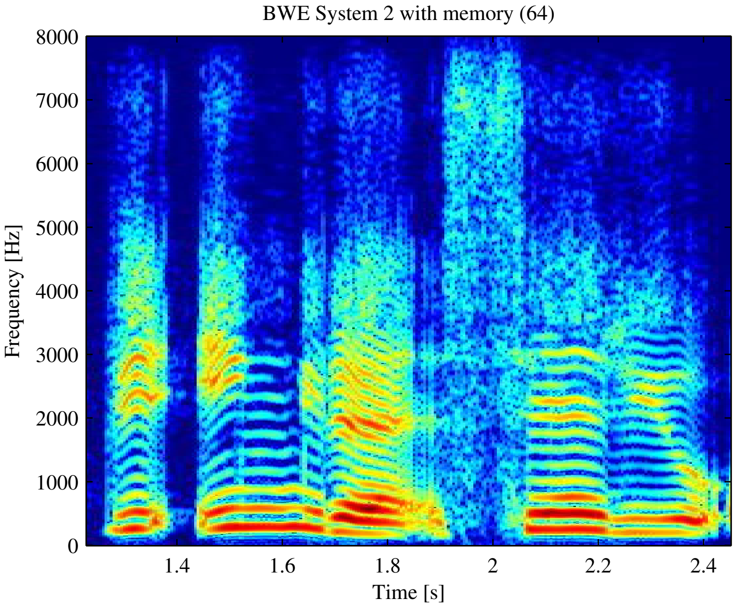

Figure 8.12: Spectrogram of output from BWE System 2 ("vi yngre ser jo").

8.3.2 Artefacts and Speaker Selectivity

No obvious, consistent artefacts or varying sound quality across the ten speakers were heard in the test set. But for a few speakers[5] it was found that BWE System 1 occasionally introduced some high frequency ($f \approx 7$ kHz) noise. Though audible, this was not found to be particularily unpleasant or annoying.

Figures 8.13 and 8.14 show spectrograms of the wideband signal and BWE System 1 output for one example of this. The uttered word is "energi", and the artefact is especially visible at $t = 3.2$ sec.

These artefacts were found to stem from the excitation extraction in LP analysis, and seemed to be related to the filter coefficient updates (see A.6) because more noise was introduced when interpolation was disabled. Further interpolation of the filter coefficients is therefore expected to help, though the computational load will also increase. It is not clear to the writer why these artefacts were only noticeable for certain speakers.

8.3.3 Differentiation of Consonant Sounds

It proved to be difficult to find examples where consonant sounds were confused with others. Since the test set mainly contained familiar words, any confusion was countered by the knowledge of the rest of the word.

Simple tests based on extracted sounds from the test sound clips indicated that the BWE systems did not worsen the situation. That is, consonant sounds which could be correctly classified in POTS quality speech, could be correctly classified after BWE was performed. A better experiment for this purpose should perhaps be based on recordings of nonsensical words.

[5] 41_*.wav and 361_*.wav in test set.

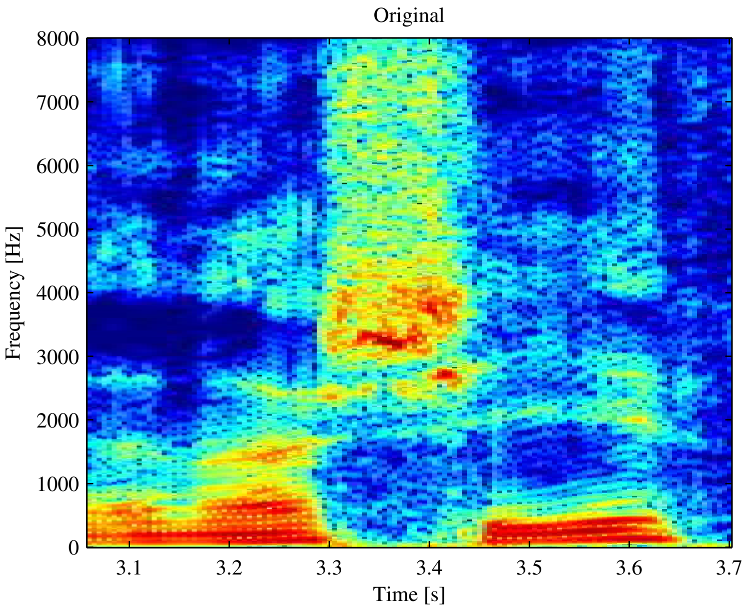

Figure 8.13: Spectrogram of wideband speech ("energi").

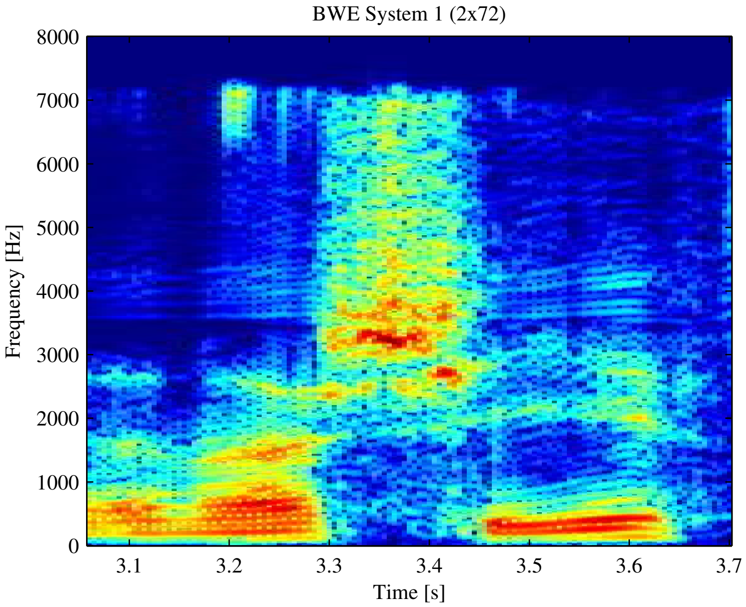

Figure 8.14: Spectrogram of output from BWE System 1 ("energi").

Chapter 9

Conclusion

9.1 Conclusion

Based on systems proposed in literature, two systems for BWE of telephony speech were implemented. Both systems focused on extension into the high band ($f \geq 3.4$ kHz) and were found to improve the perceived quality.

For extension of the spectral envelope, BWE System 1 was based on the use of ANN and mainly LPCCs, while BWE System 2 was based on the use of GMM and MFCCs. A detailed method for objective analysis of extension performance was suggested, since literature studies had shown that there is no standard method for this, and the methods used are rarely explained in detail. Analysis according to this method showed that BWE System 1 outperformed BWE System 2, even though it relied on the simplest estimation method of the two. This was thought to indicate that MFCCs are not better suited for extension of the spectral envelope than LPCCs, which agrees with results presented in [5]. The introduction of memory via dynamic feature parameters was found to slightly increase BWE System 2's performance.

It was also found that ANNs with two hidden layers performed better than single-layer networks in estimation, even though the same total number of neurons was used. This meant that improved performance was achieved even though the computational complexity was reduced.

Subjective analysis by use of informal listening tests showed that BWE System 1 gave the best sound quality. This preference was mainly due to the differing methods for excitation extension. LP analysis and modulation was used in BWE System 1, while BP-MGN was used in BWE System 2 for this extension. The latter method resulted in a noticeably more noisy excitation, while the former was found to cause artefacts for a few speakers in the listening tests. These artefacts were most likely due to insufficient interpolation of filter coefficients during the LP analysis and should therefore be simple to reduce. Neither system seemed to improve nor worsen differentiation between consonant sounds.

Based on these analyses and literature studies, BWE systems based on LP analysis seem to be a good choice because it allows for:

- simple, yet good quality extension of the excitation
- efficient computation of the cepstrum via LPCCs

Cepstral representation of the spectral envelope seems to be a good choice due to properties of the cepstral distance measure, and the simple method by which stable synthesis filters can be computed.

And lastly, it was found that highly accurate extension of the spectral envelope does not seem to be necessary for an improvement in perceived quality, as was also concluded in [10].

9.2 Further Work

For further work, it is suggested to look into the use of classification and specialized estimators. The idea is to perform a classification of the narrowband feature vector, and then use an estimator that has been trained on that specific class for spectral envelope extension. This approach is similar to GMM-based regression, where the mixture components represent classes. However, GMM-based regression is computationally expensive because it uses soft decision classification, thus giving an estimate that is interpolated from all the mixture components. Hard or soft classification by use of vector quantization or ANNs, and estimation by use of a linear mapping for each class could be an interesting low-complexity alternative.

The effect of dynamic feature parameters could also be investigated further. As an example, the use of a higher order regression function might improve estimation performance relative to the first order (linear) function used in this thesis. Note that if these are to be used in BWE System 1, it only makes sense to include dynamic feature parameters for the narrowband signal.

A database with wideband and real PSTN/POTS quality speech can be helpful for future work. The combination of TIMIT and NTIMIT [36] is an example of such a database, though they are both over 15 years old and might not be representative for channels[1] found in today's PSTN. A system that properly models relevant PSTN channels, possibly including different microphone characteristics, could therefore be useful. It would also be a good idea to include different languages in the database to avoid language-dependent BWE.

Extension to the low band was not implemented in the final systems in this thesis, but should be considered. Early experiments indicated that it is best done by use of nonlinearities, since it is most prominent for voiced speech. The added complexity required for this extension method might be considered high in comparison to the increase in perceived quality, since a whitening of the excitation and power control is needed. The possibility of using equalization instead, as proposed in [7], should therefore be investigated.

Finally, the effects of noise (including quantization) and non-speech signals or ways to deal with them were not studied in this thesis. These issues should of course be addressed before deployment of a BWE system.

[1] With *channel* it is meant the telephone and PSTN combined.

Appendix A

Implemented Filters

The filters and their use in the two BWE systems is explained in detail in the following sections.

A.1 POTS Bandpass Filter

To simulate a POTS quality telephone line, a 100th order bandpass FIR filter was used. Quantization noise was not simulated.

The FIR filter was created by use of the window method, with an upper corner frequency of 3.465 kHz and a lower corner frequency of 240 Hz. These corner frequencies were manually tuned to get an approximate attenuation of 3 dB at the edges of the defined POTS band, i.e. 0.3 – 3.4 kHz. The magnitude response is shown in figure A.1.

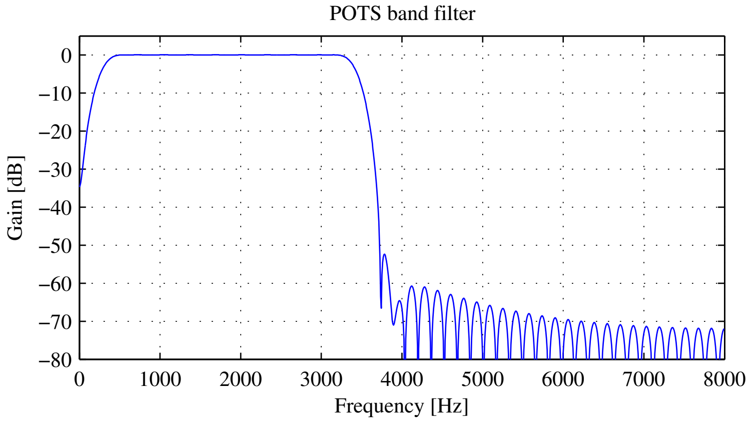

Figure A.1: Magnitude response of implemented POTS filter.

The group delay of this filter is 50 samples across the spectrum, meaning synchronization of the wide- and POTS band signal can easily be achieved by simply discarding the first 50 samples of the filtered signal. Further, it has at least 60 dB attenuation above 4 kHz, meaning downsampling by 2 does not introduce significant aliasing. This way, a POTS band signal with 8 kHz sample rate signal can easily be simulated.

A.2 Anti-Aliasing Filter

After upsampling of the excitation or POTS band signal in BWE System 1, filtering is needed to remove aliased frequency components. This filter was implemented as a 9^{th} order elliptical filter to get a sharp cutoff so that its effect on the POTS band could be minimized. The filter was specified with a maximum of 0.05 dB ripple in the passband and at least 60 dB attenuation in the stopband. The corner frequencies of the pass- and stopband were set to 3.7 kHz and 4.0 kHz, respectively. Its magnitude response is shown in figure A.2.

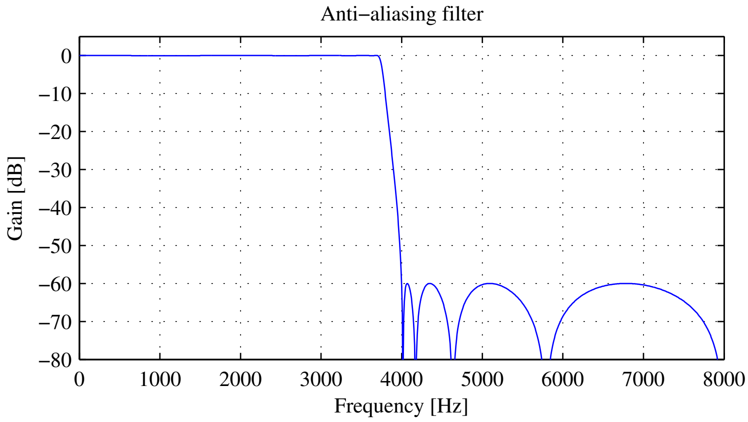

Figure A.2: Magnitude response of implemented anti-aliasing filter.

The average group delay of this filter was 7 samples for $f < 4$ kHz, which is of no consequence since both the extended excitation and the POTS signal are subjected to it.

A.3 Highpass Filter

To reduce the spectral overlap between the POTS band and the synthesized high band, a high pass filter was required. This was implemented as a 6^{th} order elliptical filter to get a sharp cutoff without too much group delay.

The filter was specified with a maximum of 0.05 dB ripple in the passband and at least 40 dB attenuation in the stopband. The corner frequencies of the pass- and stopband were selected to 3.7 kHz and 3.0 kHz, respectively. Though this gives some overlap, it was found to minimize the gap between the POTS and high band. Naturally, this filter is meant specifically for the POTS bandpass filter in this thesis.

Figure A.3 shows the magnitude response of the highpass filter alone and in combination with the POTS bandpass filter (summed absolute amplitude responses).

The filter had a mean group delay of approximately 4 samples for $f > 3.4$ kHz. This was deemed insignificant with respect to synchronization of the synthesized highband and original POTS band signal.

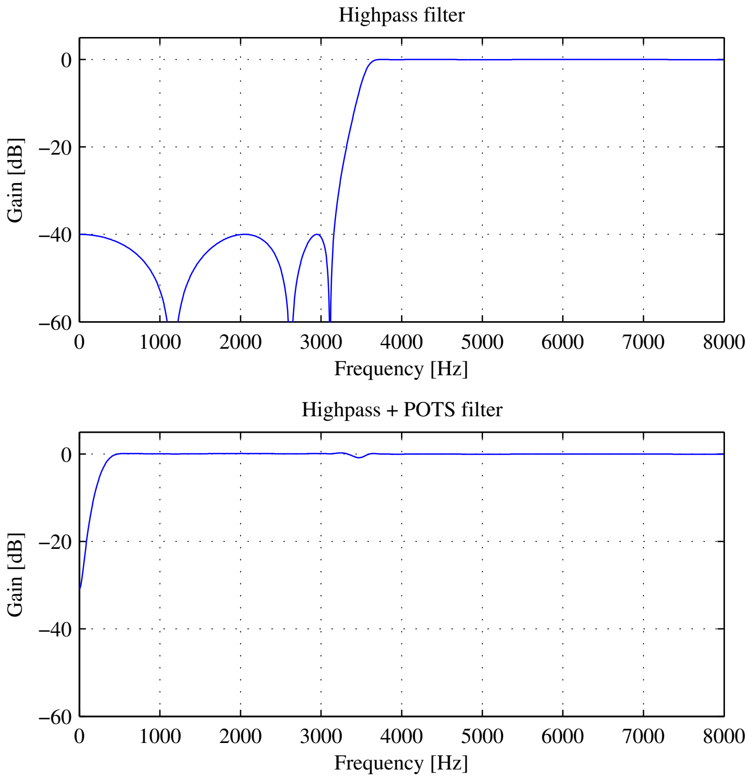

Figure A.3: Magnitude response of implemented highpass filter, alone and in combination with the POTS filter.

A.4 Bandpass Filtering for BP-MGN

To get a bandpass signal from the high POTS band for use in BP-MGN, only a highpass filter was needed. The filter was created using the window method, resulting in a linear phase FIR filter. Its order was chosen to 40, resulting in a 20 sample delay across the spectrum, with a corner frequency of 2.5 kHz.

Figure A.4 shows the magnitude response of this filter alone, and in combination with the POTS filter (summed absolute response). Note that the red line indicates −6 dB, giving a passband bandwidth of roughly 960 Hz (2.50 − 3.46 kHz).

A.5 Mel Filter Bank

The Mel filter bank in BWE System 2 consisted of 21 filters. The POTS band was thus covered by the 15 lowest filters, while the upper 6 covered the remaining high band. Tables A.1 and A.2 show the center frequencies of the high and low band filters.

Figure A.5 illustrates the filter responses on a linear frequency scale. The decreasing filter "heights" are due to the normalization needed for computation of average power.

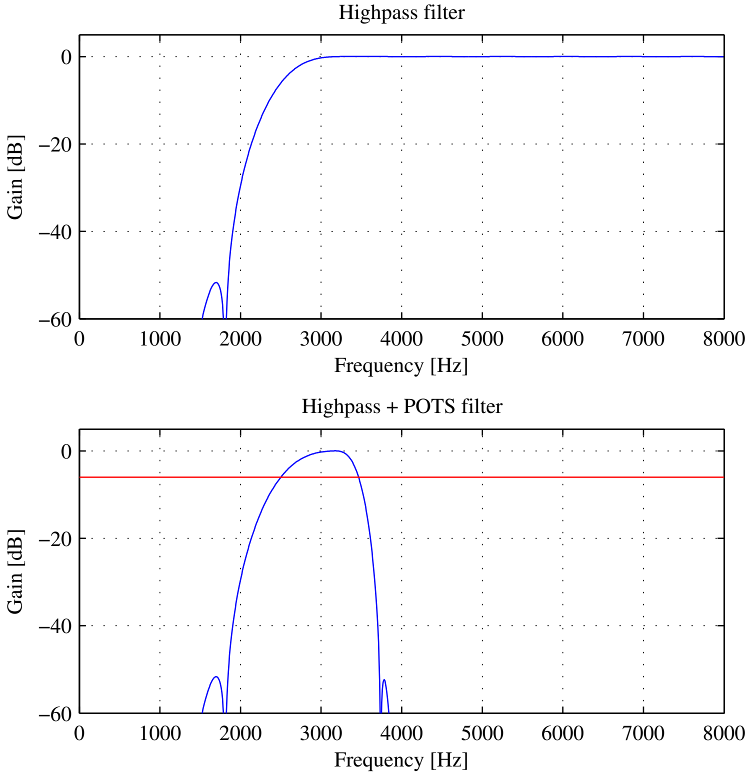

Figure A.4: Magnitude response for BP-MGN filtering.

Filter #	Frequency (Mel)	Frequency (Hz)
1	129	85
2	258	180
3	387	287
4	516	407
5	646	541
6	775	692
7	904	861
8	1033	1050
9	1162	1263
10	1291	1501
11	1420	1768
12	1549	2067
13	1678	2403
14	1807	2780
15	1936	3202

Table A.1: Center frequencies of the low band Mel filters.

Filter #	Frequency (Mel)	Frequency (Hz)
1	2066	3676
2	2195	4207
3	2324	4802
4	2453	5470
5	2582	6219
6	2711	7058

Table A.2: Center frequencies of the high band Mel filters.

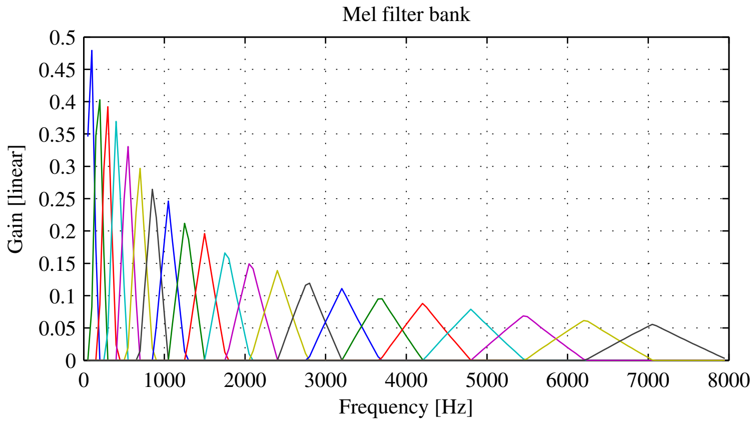

Figure A.5: Mel filter bank's coverage of the spectrum.

A.6 Interpolated Filtering

To reduce discontinuities due to filter updates in synthesis, interpolation of gain and coefficients should be used.

For the systems implemented in this project, an interpolation factor of 2 was used. The filtering is thus performed in subframes, with an averaging of filter parameters in the subframes that are inbetween analysis frames. This is illustrated in figure A.6 for 50% overlap of analysis windows. For analysis windows of length 20 ms, the subframes will be 5 ms long.

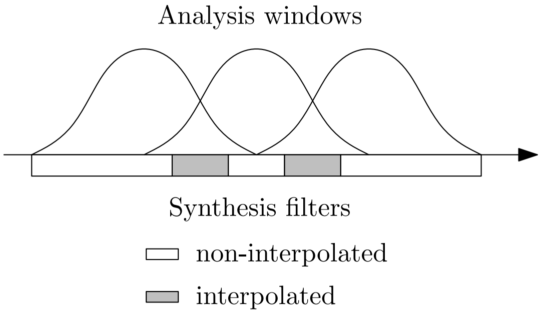

Figure A.6: Interpolation by two of filter coefficients.

Appendix B

Contents of Enclosed CD

The enclosed CD contains all the listening examples for the different variants of BWE System 1 and 2, as well as most of the C and Matlab code needed to train and test the systems. A .PDF version of this report is also included.

HTK [23] is not included since only distribution via its official webpage is allowed. The FANN [33] library and binaries of the related Matlab-bindings [34] are not included since these must be compiled on the target system. The NST database is not distributed on the CD, nor are the training data used for the two systems.

A summary of the most important CD contents:

/**Test_set**/**orig**/ - The 10 original wideband speech clips.

/**Test_set**/**pots**/ - POTS-band filtered speech clips.

/**Test_set**/**sys1 (YxZ)**/ - Output from BWE System 1, using ANN with Y hidden layers with Z neurons each

/**Test_set**/**sys2 (X)**/ - Output from BWE System 2 without memory, using GMM with X mixture components.

/**Test_set**/**sys2mem (X)**/ - Output from BWE System 2 with memory, using GMM with X mixture components.

/**C**/**mex**/ - Matlab bindings for the FANN library.

/**C**/**neurotrain**/ - Commandline tool for training ANNs, based on the FANN library.

/**C**/**savetrain**/ - Commandline tool for converting training data into a format that HTK can read. This is based on a library created by supervisor Svein Gunnar Pettersen.

/**Matlab**/ - The main .m-files for the BWE systems, as well as for creating training data and testing them.

/**Matlab**/**misc**/ - Miscellaneous support functions (with self-explanatory names) for computing feature parameters, creating filters, interpolated filtering, loading and regressing with

GMMs from HTK, computing spectral distance, creating DCTs, saving and loading binary training data, ++

In most directories there is also a `00readme.txt` which may be of further help.

Bibliography

[1] J. G. Proakis and D. Manolakis, *Digital Signal Processing - Principles, Algorithms and Applications.* Pearson Prentice Hall, 4 ed., 2007.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley, 2006.

[3] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing.* Prentice Hall, Inc., 2001.

[4] E. Larsen and R. Aarts, *Audio Bandwidth Extension.* John Wiley & Sons Ltd., 2004.

[5] B. Iser, G. Schmidt, and W. Minker, *Bandwidth Extension of Speech Signals.* Springer Science+Business Media, LLC, 2008.

[6] H. Pulakka, P. Alku, L. Laaksonen, and P. Valve, "The effect of highband harmonic structure in the artificial bandwidth expansion of telephone speech," in *Proceedings of Interspeech (Antwerp, Belgium)*, 2007.

[7] Y. Qian and P. Kabal, "Combining equalization and estimation for bandwidth extension of narrowband speech," in *Proceedings of ICASSP (Montreal, Canada)*, 2004.

[8] A. Nour-Eldin and P. Kabal, "Mel-frequency cepstral coefficient-based bandwidth extension of narrowband speech," in *Proceedings of Interspeech (Brisbane, Australia)*, pp. 53–56, 2008.

[9] A. H. Nour-Eldin and P. Kabal, "Combining frontend-based memory with MFCC features for bandwidth extension of narrowband speech," in *Proceedings of ICASSP (Taipei, Taiwan)*, 2009.

[10] M. Nilsson, H. Gustafsson, S. V. Andersen, and W. B. Kleijn, "Gaussian mixture model based mutual information estimation between frequency bands in speech," in *Proceedings of ICASSP (Orlando, USA)*, 2002.

[11] P. Jax and P. Vary, "Feature selection for improved bandwidth extension of speech signals," in *Proceedings of ICASSP (Montreal, Canada)*, 2004.

[12] A. H. Nour-Eldin and P. Kabal, "Objective analysis of the effect of memory inclusion on bandwidth extension of narrowband speech," in *Proceedings of Interspeech (Antwerp, Belgium)*, 2007.

[13] A. H. Nour-Eldin, T. Z. Shabestary, and P. Kabal, "The effect of memory inclusion on mutual information between speech frequency bands," in *Proceedings of ICASSP (Toulouse, France)*, 2006.

[14] "G.114 - one way transmission time."
`http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.114-200305-I!!PDF-E&type=items`
last retrieved: 11. May, 2009.

[15] B. Iser and G. Schmidt, "Neural networks versus codebooks in an application for bandwidth extension of speech signals," in *Proceedings of Eurospeech (Geneva, Switzerland)*, 2003.

[16] I. Y. Soon and C. K. Yeo, "Bandwidth extension of narrowband speech using soft-decision vector quantization," in *Proceedings of ICICS (Bangkok, Thailand)*, 2005.

[17] U. Kornagel, "Techniques for artificial bandwidth extension of telephone speech," *Signal Processing (Elsevier)*, vol. 86, pp. 1296–1306, 2006.

[18] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing (Elsevier)*, vol. 83, pp. 1707 – 1719, 2003.

[19] P. Bauer and T. Fingscheidt, "An HMM-based artificial bandwidth extension evaluated by cross-language training and test," in *Proceedings of ICASSP (Las Vegas, USA)*, pp. 4589–4592, 2008.

[20] S. Haykin, *Neural Networks - A Comprehensive Foundation.* Prentice Hall, Inc., 2 ed., 1999.

[21] S. E. Fahlman, "An empirical study of learning speed in back-propagation networks," in *Proceedings of Connectionist Models Summer School (Los Altos, USA)*, 1988.

[22] M. Riedmiller, "Rprop - description and implementation details," tech. rep., Institut für Logik, Komplexität und Deduktionssysteme University of Karlsruhe, 1994.

[23] "Hidden Markov Model Toolkit (HTK)."
`http://htk.eng.cam.ac.uk/`
last retrieved: 29. Apr, 2009.

[24] K.-Y. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proceedings of ICASSP (Istanbul, Turkey)*, vol. 3, pp. 1843–1846, 2000.

[25] G. Strang, *Linear algebra and its applications.* Thomson Brooks/Cole, 4 ed., 2006.

[26] D. Zaykovskiy and B. Iser, "Comparison of neural networks and linear mapping in an application for bandwidth extension," in *Proceedings of SPECOM (Patras, Greece)*, 2005.

[27] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, "On the use of bandpass liftering in speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, pp. 947–954, 1987.

[28] A. Shahina and B. Yegnanarayana, "Mapping neural networks for bandwidth extension of narrowband speech," in *Proceedings of Interspeech (Pittsburgh, USA)*, 2006.

[29] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 873–881, March 2007.

[30] A. Shahina and B. Yegnanarayana, "Mapping speech spectra from throat microphone to close-speaking microphone: A neural network approach," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 10 – 10, 2007.

[31] “Matlab - high-level programming tool for technical computing.”
`http://www.mathworks.com/products/matlab/`.

[32] “Voicebox - a speech processing toolbox for Matlab.”
`http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`
last retrieved: 1 Feb, 2009.

[33] “Fast artificial neural network library (FANN).”
`http://leenissen.dk/fann/`
last retrieved: 5. March, 2009.

[34] “Matlab bindings for FANN.”
`http://sumowiki.intec.ugent.be/index.php/FANN_Bindings`
last retrieved: 10. March, 2009.

[35] S. G. S. Pettersen, *Robust Speech Recognition in the Presence of Additive Noise.* PhD thesis, Norwegian University of Science and Technology, 2009.

[36] “TIMIT and NTIMIT - wideband and telephone quality speech corpora.”
`http://www.ldc.upenn.edu/Catalog/topten.jsp`
last retrieved: 19 Jun, 2009.