Artificial Bandwidth Extension of Narrowband Speech

Signal and Information Processing for Communications

Department of Electronic Systems

Group 1092

June 7 2007

Aalborg University

Aalborg University

Department of Electronic Systems

Fredrik Bajers Vej 7 ▪ DK-9220 Aalborg Telephone +45 96 35 86 00

Master's Thesis

Title: Artificial Bandwidth Extension of Narrowband Speech

Project period: September 4 2006 - June 7 2007

Project group:

1092

Group members:

Nels Rohde

Svend Aage Vedstesen

Supervisors:

Søren Holdt Jensen

Jesper Jensen

Thomas Østergaard*

Bo Arden Kristensen*

*External supervisors, ETI A/S

Publications: 7

Pages in

Main Report: 103

Appendix: 11

Finished: June 7 2007

Abstract

This thesis addresses the challenges of estimating wideband speech (0-8000 Hz) from narrowband speech (0-3400 Hz). This is done by estimating the missing upper spectral components from the narrowband speech using statistical approaches. Utilizing the Source-Filter model, the estimation problem is divided into estimating a wideband envelope and a wideband excitation signal. These two estimates are then combined to obtain an artificially extended wideband speech signal. Three methods based on Vector Quantization, Gaussian Mixture Models and Hidden Markov Models respectively, have been developed for estimation of the wideband envelope. Results show that the two later outperforms the method based on vector quantization, in both objective and audible results. Estimation of excitation is done by simple spectral replication. A new perceptual training procedure which utilizes Mel Frequency Cepstral Coefficients for estimation of the envelope is proposed. A formal listening test conclude, that the proposed method of extending the wideband speech, is preferred over bandlimited narrow band speech with a level of significance of more than 99%.

Preface

This master thesis is written at the Department of Electronic Systems at Aalborg University, Denmark. It is written by Group 1092 in the period from September 4 2006 to June 7 2007.

Nomenclature

All references to source material are specified with square brackets after the part of the text, where they are used, e.g. [30]. The references are listed in the bibliography on page 110. The Matlab® code for the bandwidth extension algorithm can be found on the attached CD-ROM together with an electronic version of the report and the available literature. This thesis is written in LaTeX. Formulas, figures and tables are numbered in succession for each chapter. References to equations are made in two ways: equation 4.74 or (4.74), i.e. equation is implied, when the number referred to is in brackets.

Through the report abbreviations have been used frequently. They are explained in the list of abbreviations after the conclusion on page 105. The frequently used symbols are listed in a symbol list with explanations on page 107. Estimates are marked with a hat, e.g. $\widehat{\Phi}_w$ which denotes an estimated wideband power spectrum. Throughout this thesis the term power spectrum will be used extensively. By this term, we mean short term power spectrum as all processing is done frame wise. References are made to speech files (wav-files) in the thesis. An explanation of these speech files which originated from the TIMIT speech database is given in appendix A.

Acknowledgments

We would like to thank O.A. Niamut for providing us with the Benchmark program used in the formal listening test. Furthermore we would like to thank the persons which participated in the formal listening for making such at test possible. In acknowledgement for the donations, which made our study trip to ICASSP 2007 possible and unforgettable, we would like to thank (listed randomly): KIRK Telecom A/S, Marie & M.B. Richters Fond, Danphone A/S, C.W Obels Fond, Texas Instruments Denmark A/S, Terma A/S, E-Studienævnet. A special thanks is given to the two external supervisors from ETI A/S for constructive proposals and dissents.

Aalborg University, June 7, 2007

Nels Rohde

Svend Aage Vedstesen

Contents

Chapter 1 Introduction

1.1 Motivation

In todays telephone networks the bandwidth is typically limited to a frequency range of 300 Hz to 3400 Hz. Even though almost every public telephone exchange today is digital, this limitation still applies because of the old analogue telephone networks and standards, see e.g. appendix A. The intelligibility for such bandlimited speech is around 99% when whole sentences are pronounced [16]. However studies have shown that the acoustic bandwidth has significant impact on the perceived quality of speech [35]. A higher quality of speech would also result in the speech being more pleasant to listen to, thus reducing the listening effort.

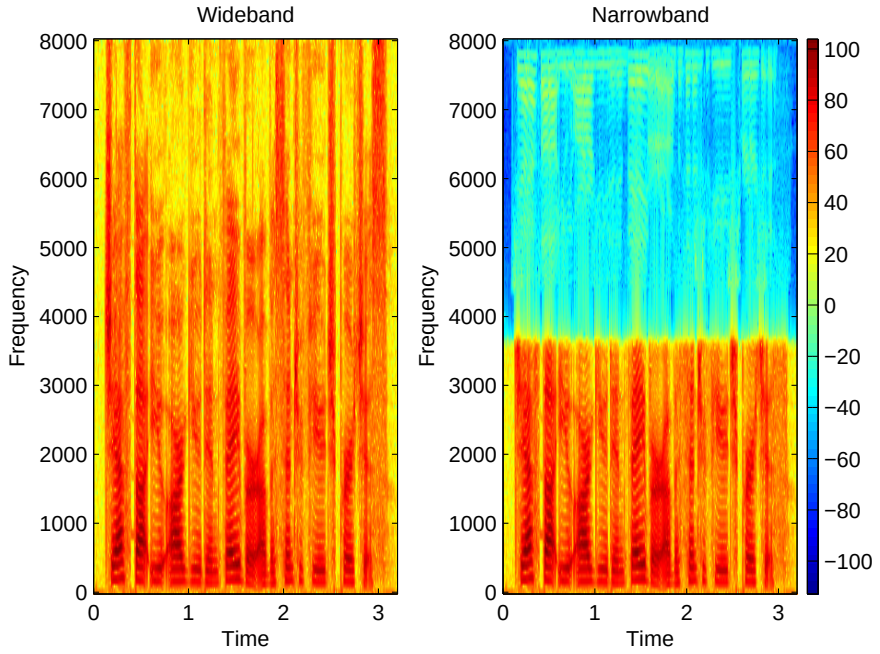
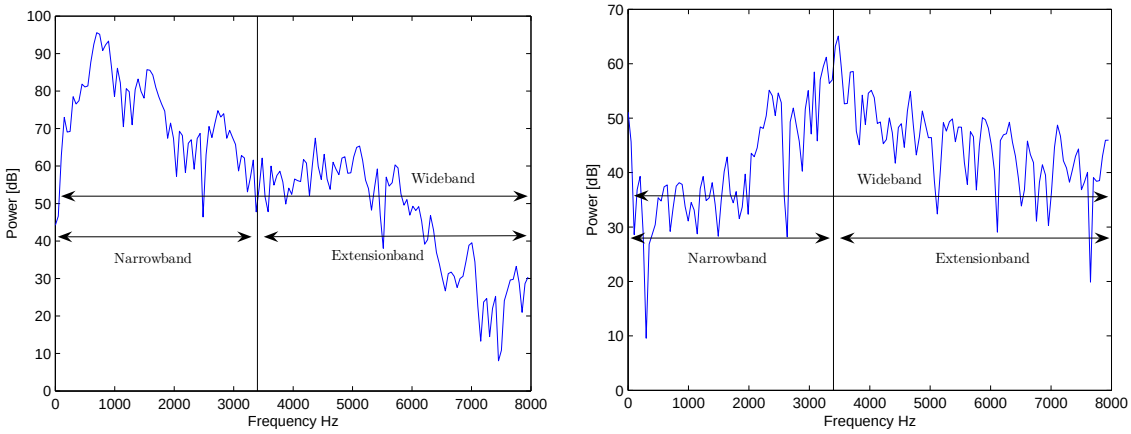


Figure 1.1: Spectrograms of a wideband and narrowband speech signal respectively. As can be seen in the figure, the upper spectral components are missing in the narrowband spectrogram. The signal is created using the filter from appendix A.2. The 8 kHz (narrowband) sampled signal has been upsampled to 16 kHz for comparison. Note that the range from 0-300 Hz is available, compared to true "telephone" speech.

Figure 1.1 shows a spectrogram of respectively a 16 kHz (wideband) and a 8 kHz (narrowband) sampled speech signal. The 8 kHz signal has been upsampled for better comparison. As it can be seen the narrowband is missing the upper spectral components from 3400 Hz to 8000 Hz. Our perception as humans is therefore, that narrowband is not as

broad and pleasant to listen to as wideband. Figures 1.2(a) and 1.2(b) show a section of the spectrogram for an unvoiced and a voiced sound respectively. The range from 300-3400 Hz is called the narrowband. The complete range is referred to as wideband. The "missing" range in narrowband compared to the wideband is referred to as the extensionband. Throughout the thesis the bands will frequently be referred to by their abbreviations; nb, wb and eb respectively. As it can be seen in both cases, there are a considerable amount of energy in the extensionband, which would result in a significantly improved version of the speech, if it was available.



(a) Voiced speech frame.

(b) Unvoiced speech frame.

Figure 1.2: Power spectrum showing the distribution of power within one frame.

A lot of todays terminals are capable of reproducing wideband speech. In order to exploit this wideband capability when receiving a call from e.g. the public service telephone network (PSTN), new codecs and better telephones have to be introduced together with a modification of the network/transmission link. Of economical reasons, true wideband is not likely to be supported in the old telephone networks in the near future. This challenge can be overcomed by doing some processing on a terminal capable of playing back wideband audio. The objective is then to artificially add some "missing" spectral components to the received signal. Such processing is referred to BandWidth Extension (BWE), which results in a higher perceived quality of speech. One great advantage of this approach is that no modifications to the existing system is required.

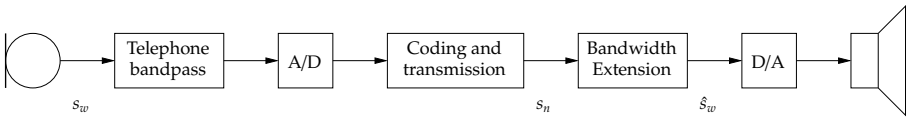


Figure 1.3: Speech is recorded using a microphone, resulting in a wideband speech signal s_w. This signal is bandpass filtered and transmitted across a telephone network. At the receiving end an estimate $\hat{s}_w$ is obtained by the BWE algorithm, having only a narrowband version s_n of the signal at disposal.

Figure 1.3 illustrates how BWE can be included into the existing network. The wideband signal s_w from the microphone is bandpass filtered before digital conversion, compressed with e.g. μ- or A-law and transmitted across the telephone network. At the receiver only the narrowband signal s_n is available. From this signal the upper and/or lower spectral components are estimated by the BWE algorithm. By estimating and synthesizing the

upper spectral components an estimate $\hat{s}_w$ is obtained, which is played back on the loudspeaker of the receiving terminal.

1.2 Previous work

The first known experiment in the field of BWE was carried out in 1933 and tried to extend the bandwidth by non linear processing. In 1972 BBC introduced a method to enhance telephone speech during broadcasting [11]. All of these early methods of BWE used some simple signal processing techniques, without taking the model of speech production into account.

In the literature these techniques are all referred to as non-model based algorithms for bandwidth extension. Other proposals have been made in which a source-filter model of speech production is taken into considerations [14, 26, 17]. As a result of this, the problem of BWE can be divided into two separate tasks. One is to extend the spectral envelope, and the other is to extend the excitation signal. To extend the envelope, several proposals have been made. Some approaches use conventional codebook implementations (vector quantization), where a codebook is trained with a corpus of bandlimited speech signals. This codebook $\mathbb{C}_n$ has a counterpart, called a shadow codebook $\mathbb{C}_w$, containing the wideband counterpart of the narrowband envelope. Another implementation is to have a stacked vector of features representing the narrowband envelope and their associated wideband features. Approaches based on codebooks have been proposed by e.g. [6, 5].

Park and Kim proposed a method [26], where the spectral transformation of the narrowband envelope is based on a Gaussian Mixture Model (GMM) with joint density estimation. This study showed that the method based on GMM outperforms the conventional codebook mapping. One drawback of both the codebook and the GMM implementation is that temporal dependencies of the speech are not modeled. To incorporate the dependency of speech from one frame to another, methods using a Hidden Markov Model (HMM) have been proposed by e.g. [16, 17, 18]. This estimation technique assumes that one frame of speech is dependent on the previous (and/or the next) one. Jax [16] found that estimating the missing frequency bands, using an HMM significantly reduced unwanted artifacts in the bandwidth extended signal. One of the great challenges when doing estimation like this is to find a set of features which describes the envelope the best. At the same time they should have good discriminative properties. Jax did an intensive study of both the features; $\mathbf{x}$ used to represent the narrowband and $\mathbf{y}$ representing the extensionband. This was motivated by the problem of finding the most "suitable" set of features for the estimation problem. In order to do this, he looked at mutual information and separability. The mutual information $\mathbf{I}(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ can be seen as a measure of how much information is available about $\mathbf{y}$ from $\mathbf{x}$. A high mutual information is therefore desirable. However a high mutual information is not enough when doing bandwidth extension. Of course we want $\mathbf{x}$ to give as much information about $\mathbf{y}$ as possible, but another requirement is that the features can be properly separated. The separability is a measure of how well a given feature set can be discriminated, and this is important when doing classification. BWE algorithms both rely on estimation and classification. Therefore both high separability and high mutual information is desired. A compromise between the two, is often what constitutes the feature vector.

As indicated earlier, a dependance between the narrowband and the extensionbands are required in order for the estimation of the spectral envelope to succeed. A study by

Nilsson et al.[25] showed that the mutual information was only a small fraction of the perceived entropy of the high band. Therefore if bandwidth extension of speech are to succeed it should not solely rely on mutual information between the bands. It should rely on methods which make use of perceptual properties, to make the extended speech signal sound pleasant [25]. This study made use of a GMM to jointly model the envelopes. The envelopes were represented by Mel Frequency Cepstrum Coefficients (MFCC) and the log energy of the disjoint frequency bands. A study like this, as when doing estimation, however highly relies on the features chosen to represent the envelopes.

As the mutual information is limited when representing the envelopes in conventional ways, efforts have been made to transmit some auxiliary information together with the narrowband speech signal, see e.g. [1, 8]. The result of doing this has shown to improve speech quality. These approaches however require a modification of the network. That is, a better terminal which is capable of recording true wideband is required and new codecs have to be introduced. As BWE is a solution for the transition phase until true wideband is available, the auxiliary approach is most likely not going to gain acceptance. If too many modifications are required, a true wideband codec might as well be introduced. Extending the signal artificially on the other hand using advanced signal processing, only require minor (physical) changes at the receiving side. This would therefore be more tractable.

1.3 Scope of thesis

In this Thesis we address the problem of bandwidth extension on speech. That is, we try to increase the perceived quality of narrowband speech. This is done by estimating the upper and/or lower spectral components from a narrowband signal. These components are then artificially added the narrowband, to obtain an estimated wideband speech signal. By using the source-filter model, estimation can be divided into two subtasks, one extending the excitation signal and one extending the spectral envelope of the narrowband signal. Different types of frameworks which realize estimation of the excitation and the spectral envelope are proposed and evaluated. One framework is selected based on optimization criteria and flexibility. A new method of extending the excitation signal is proposed, based on observations of the wideband excitation signal. The idea is to shape the estimated excitation such that it has the same spectral characteristics as the observed excitation.

Estimation of the spectral envelope of the narrowband is done and analyzed using three methods. The methods we examine are estimation of the spectral envelope using codebooks, a GMM and finally Hidden Markov Models. In most BWE algorithms linear prediction (LP)-derived features such as Auto Regressive (AR)-coefficients, Cepstral Coefficients (CC) etc. are used as features to represent the extension- or wideband band envelope. In this work we propose a perceptual method by using MFCC to represent the extensionband envelope during training of the models. The MFCC feature has been used as a narrowband feature by Enbom and Kleijn [5]. To our best knowledge the MFCC has never been used, as in our proposal.

Different estimation methods are compared and evaluated. Both objective and subjective measures are used to evaluate the performance of the individual methods. Objective measures are calculated from estimated and original wideband envelopes from speech provided by the TIMIT database. A formal listening test is carried out to evaluate the

subjective performance of the estimation methods and whether or not the estimated speech is preferred over narrowband speech.
If a BWE system was to be implemented in real telephone networks, noise from transmitting the signal, coding etc. should be taken into account, as it affects the signal at the receiving part. Both interference and coding scheme vary from network to network. In this thesis we focus on the principles of bandwidth extension and therefore influences from external factors are omitted. That is, we assume the input signal to be optimal in such a sense that it contains no noise from transmission and coding. Furthermore we assume the lower band from 0-300 Hz to be available during estimation. That is, only the upper spectral components are estimated, but we show how the lower can easily be estimated using the same framework. The narrowband speech from which estimation is performed, is obtained by lowpass filtering and downsampling wideband speech. See e.g. appendix A.2 for further details. In this thesis our main concern is not complexity but proof of concept. The BWE algorithm is therefore developed and based on the following assumptions:

- Errorfree transmission and coding
- The band from 0-300 Hz is available
- No computational restrictions

1.4 Structure of thesis

The following gives a brief description of each chapter. The structure of the thesis is as follows:

Chapter 2 introduces the source filter model. Introducing this linear model, speech can be represented in a convenient way. In this way estimation can be divided into estimation of the wideband envelope and estimation of the wideband excitation. Estimation of excitation is described in chapter 5 and estimation of the wideband envelope is described in chapter 4. In order to estimate the wideband envelope a compact representation is required. This is done representing the envelope using features, representing both the narrowband speech and an extensionband envelope, which are described in this chapter in section 2.2.

Chapter 3 is a review of possible frameworks which can be used to implement the bandwidth extension algorithm. One framework is chosen in view of different criteria. This chapter is concluded by describing a required method needed for implementing the chosen framework.

Chapter 4 describes the central part of the algorithm, which is to estimate the wideband envelope. Three methods based on Vector Quantization, GMM and HMM are implemented and evaluated separately. First a brief overview is given of how features are extracted and used to train the different models. For each of the methods training is explained in details and an estimator is derived. In this chapter a new method of training an HMM is also proposed.

Chapter 5 presents conventional methods of extending the excitation together with a proposed observation based method. All methods however are all based on spectral replication.

Chapter 6 verifies implementation of both the EM-algorithm and the Baum-Welch algorithm used to estimate model parameters. Furthermore visual plots of original and extended speech are presented to the reader.

Chapter 7 evaluates the performance of the BWE algorithm using both objective and subjective measures (a formal listening test). In this chapter the PESQ measure is also included to support the findings in the listening tests.

Chapter 8 summarizes and concludes the thesis. Furthermore it discusses the future use of bandwidth extension.

Finally abbreviations, symbols, bibliography and appendices ends the thesis.

Chapter 2

Speech Modeling and Analysis

In this chapter we show how speech is modeled by an envelope and an excitation signal. The envelope and speech characteristics are then characterized by different features. Features representing narrowband are afterwards used to estimate the extensionband. Objective measures are introduced to evaluate the goodness of the estimation.

To make bandwidth extension possible at all, some prior knowledge (e.g. certain characteristics) need to be known about the signal. Otherwise it would be impossible to make any reasonable estimates. Imagine that all that was known about the signal was that it was sampled at 8 kHz. Then it would be impossible to make any qualified estimates about the signal. If e.g. it was known that it was either music or speech or some other signal, then it would be much more feasible. A requirement is that the signal somehow can be described/parameterized by a mathematical model. Studies have shown that the exact shape of musical spectra are determined by both the instrument and the genre being played [22]. Strictly speaking, if music should be bandwidth extended, using a parametric approach, it would require that a model for each instrument and each genre of music were available.

2.1 Source-filter model

If speech is to be extended, as in this thesis, a very recognized model describing the production of speech is available. Namely the source-filter model [31]. This model is motivated by studies of the human speech production system and makes a decomposition of a given speech signal into two parts. One part describing the excitation signal from the source and another part describing the filters, which are driven by this excitation. The result of driving these filters with the excitation is then speech.

To understand the source-filter model a closer look at the anatomy of the speech production system is required. Figure 2.1 shows a diagram of the human speech production system. All the important organs used when producing speech are depicted in the figure. The major components of the system are the lungs, trachea (windpipe), larynx (organ of voice production), pharyngal cavity (throat), oral cavity (mouth) and nasal cavity (nose). Usually the Pharyngeal- and oral cavity are referred to as the vocal tract and the nasal cavity, the nasal tract [20].

The lungs provide the air flow used in the speech production. At the larynx, which among others include the vocal folds/chords and the glottis, this air flow is shaped into certain characteristics. Dependent of the shape of the cavities and position of tongue, lips, jaw

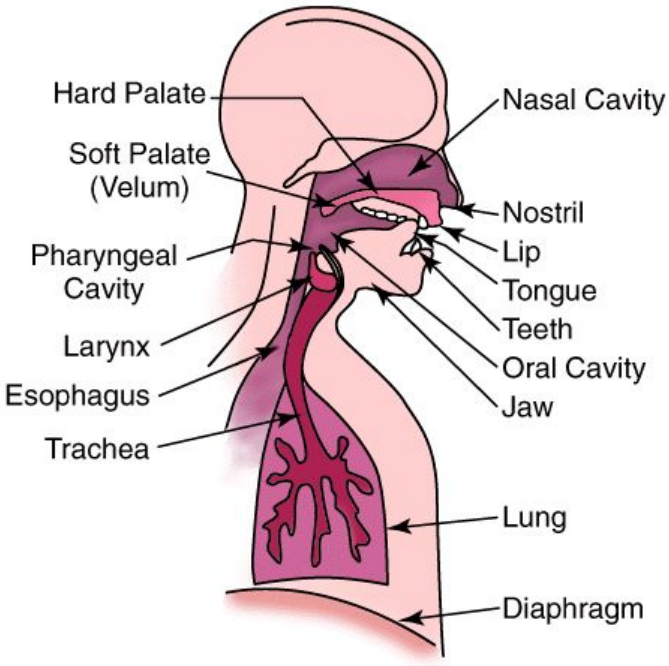


Figure 2.1: Diagram of the human speech production system. From [15].

etc. a certain speech sound is then produced.

A simplified model describing these ideas is shown in figure 2.2. It is now possible to describe the speech production as an acoustic filtering process with an input (excitation) to the filter and an output (speech). The airflow from the lungs goes through the Trachea and passes the Vocal folds. The resulting signal is shaped such that it can either be noise-like, pulse-like (or silence). This constitutes the excitation signal. The last part from the vocal chords to the lips, i.e the cavities, tongue, jaw etc. can now be modeled by some filters as mentioned earlier. The excitation and the filters are examined further in the following. This is used to derive the final source-filter model as depicted in figure 2.3

2.1.1 The excitation signal

The excitation describes the air flow from the lungs passing the vocal chords at the larynx. When the air passes these vocal chords, it results in either a pulse-like excitation caused by vibration of the vocal chords, or a noise like excitation when the vocal chords contract and create only a small passage for the air to pass. In the case of pulse-like excitation the period of the pulses depends on both mass and tension of the vocal chords [20]. The average fundamental frequency, i.e the time in between each occurrence of the before mentioned pulses, is for example larger for men than women and children. For men this pitch range is typically lying between 50-250 Hz, women 125-500 Hz and children a bit higher. In the noise-like case the excitation is much more simple. It is a source of random noise with some gain [31]. Such a signal has the characteristics that it is spectrally flat [11].

Pulse-like excitation occurs typically when a voiced sound is pronounced, and noise-like excitation when an unvoiced sound is pronounced. It can be seen from the model in figure 2.3 that a switch decides whether it is unvoiced or voiced sounds being pronounced. Because this is a model, it is an ideal case. In reality, however the excitation is typically a

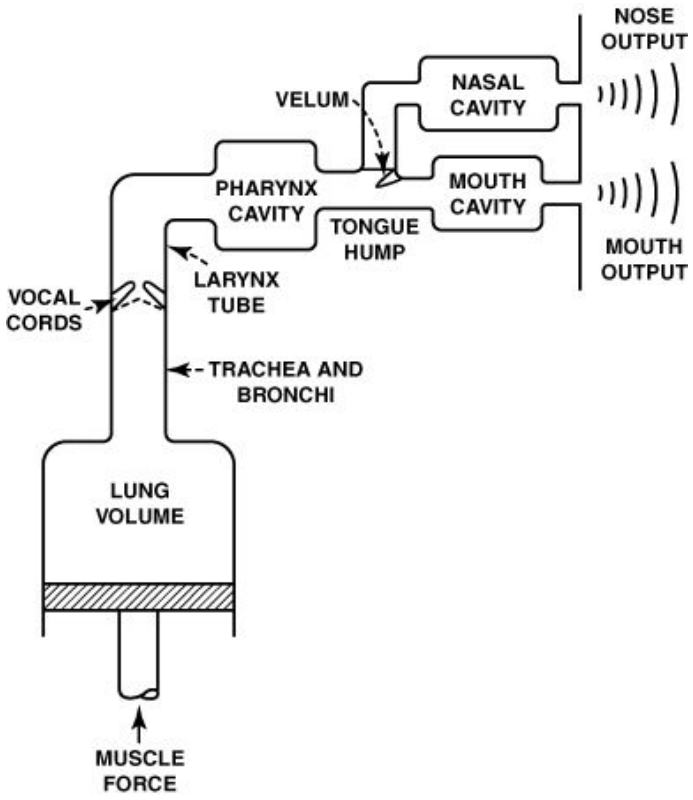


Figure 2.2: A simplified block diagram describing the human speech production. From [15]

mix of the two with one of them dominating.

2.1.2 Filter models

Typically three filters are used to model the speech production[31]: The Glottal Pulse Model $G(z)$, The Vocal Tract Model $V(z)$ and The Radiation Model $R(z)$. These models can all be thought of as linear time invariant systems, because a frame based approach is utilized in which we can assume the speech to be stationary. The glottal pulse model is only used to model speech in the voiced case. It shapes the pulse train before it is used as input to $V(z)$. The vocal tract model $V(z)$ is modeling the region from the vocal chords and the glottis to the lips where the sound is finally pronounced. The radiation model accounts for the radiation which occurs at the lips. Sometimes it is convenient to represent all of these models together in one single transfer function $H(z)$, i.e:

$$H(z) = G(z)V(z)R(z) \tag{2.1}$$

$H(z)$ is referred to as the synthesis filter and is depicted on figure 2.3. When LP analysis is performed on a speech signal, an excitation signal and an analysis filter $A(z)$ is obtained [31]. The synthesis filter is the inverse of the analysis filter, i.e:

$$H(z) = 1/A(z) \tag{2.2}$$

This way of parameterizing a speech signal is a convenient and well defined method, which is also being exploited in several speech coders as e.g. the CELP codec. The reader of interest is referred to e.g [31] and [20] for more about the acoustic theory behind the source-filter model.

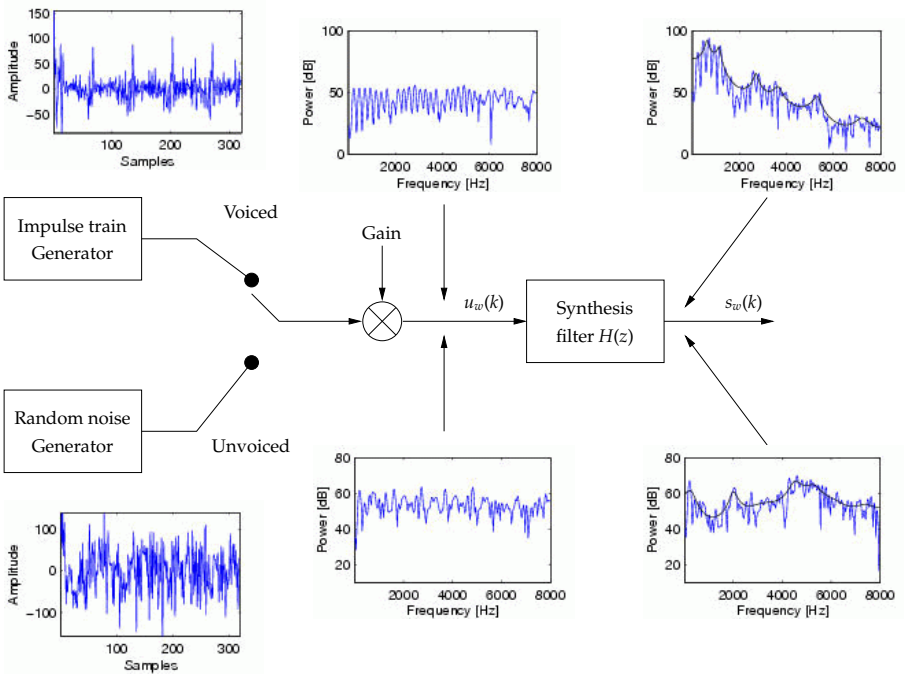


Figure 2.3: Source-filter model describing speech production. Speech can either be generated by driving the synthesis filter $H(z)$ by either an impulse train(voiced) or random noise(unvoiced), resulting in speech being produced.

2.1.3 AR modeling of speech in BWE systems

As mentioned before the presence of a good mathematical model to describe speech is of great importance in order to make a good estimate of the missing spectral components. A very good characteristic of this model is that the vocal tract transfer function $H(z)$ is much alike for persons pronouncing the same sound. That is the position of the formants only varies slightly within a certain region, which makes clustering of envelopes producing a certain wideband sound feasible. If this was not the case, as for e.g. power spectra, then a requirement would be that an estimation rule/model had to be developed individually for each person. This would not be very feasible. Based on these assumptions, one statistical model based on a priori knowledge would be sufficient.

Because the period of the pulses in the excitation signal changes from person to person, this signal is not speaker independent. Because of this changing pitch from person to person, it would not be very feasible to develop a model, to estimate the higher frequencies of the excitation. It would require different models for different pitch periods. Simple and straightforward ways to bandwidth extend such signals have been implemented with success [16].

The bandwidth extension problem, when applying the source-filter model, can therefore be solved by extending the spectral envelope and the excitation signal separately. Because of the properties of the envelope, extension of the spectral envelope can be done by applying a priori knowledge when estimating. This is e.g. done by simple classification using codebooks or other estimation models based on e.g. a GMM or Hidden Markov Models (HMM).

By exploiting the acoustic theory behind speech production, the source filter model has been derived. This model can now be utilized to extend both the excitation and the envelope of narrowband speech signals. Figure 2.4 shows how it would be possible to

obtain a wideband speech signal by bandwidth extension.

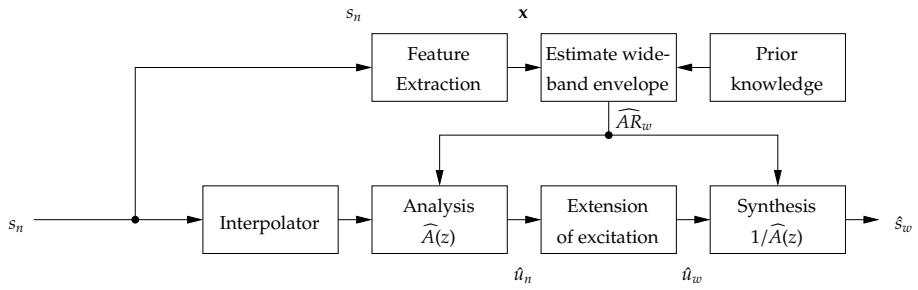


Figure 2.4: System overview of how a possible BWE system might work.

Feature vectors $\mathbf{x}$ are extracted from the narrowband signal s_n and from these a set of wideband AR-coefficients are estimated. To obtain a wideband estimate $\hat{s}_w$, the narrowband speech is first interpolated and then fed to an analysis filter $\widehat{A}(z)$ obtained from estimation of the envelope. The excitation $\hat{u}_n$ is then extended and given to the synthesis filter $1/\widehat{A}(z)$. A wideband estimate $\hat{s}_w$ of the speech is obtained by this process.

2.2 Features

The estimation of the spectral envelope of the wideband signal highly relies on the quality of the features which are used to estimate the coefficients describing the spectral envelope of the wideband speech. The speech is a slowly time varying signal in the sense that, when examined over a sufficiently short period of time (between 5 and 100 ms), its characteristics are fairly stationary [29, p.17]. In order for the features to be descriptive, they should be extracted in intervals in which we assume the speech, and thereby the envelope, to fulfill this requirement. Typically a framelength of 20-30 ms is used [10, p.280]. Throughout this thesis a frame length of 20 ms (320 samples at 16 kHz and 160 samples at 8 kHz) is used. The best possible set of features would of course be every single sample in each frame. This is computationally untractable, and in order to make a BWE system realizable a much smaller set is required. Jax has made a study [16] which examines the suitability of different features for such a system. The features used in our work is inspired from the results of that study. Only the features which are used in this project will be described in the following, the reader of interest is therefore referred to [16].
The section is divided into two subsections. The first section describes the features which have been used by Jax to characterize the narrowband. The features are described in the second section and can represent: A narrowband envelope, an envelope found by Selective Linear Prediction (SLP) (discussed in section 3.2.1) or a wideband envelope.

2.2.1 Features characterizing narrowband

A feature vector consisting of the following has been used in the BWE algorithm [16]:

- A vector $\mathbf{x}_{acf}$ containing the ten first coefficients of the autocorrelation function.
- The zero crossing rate x_{zcr}.

- The gradient index x_{gi}.
- The normalized relative frame energy x_{nrp}
- The local kurtosis x_k
- The spectral centroid x_{sc}.

These features could be used to represent a wideband signal, but in our case they are used to represent narrowband. In the calculation of the features, $s_n(k)$ refers to the k^{th} sample in the narrowband speech signal. N_k is the number of samples within a single frame. In the following the features are presented.

Auto-Correlation coefficients $\mathbf{x}_{acf}$

The autocorrelation coefficients are calculated for lag η and normalized with respect to the energy as follows:

$$\mathbf{x}_{acf}(\eta) = \frac{\sum_{k=\eta}^{N_k-1} s_n(k-\eta)s_n(k)}{\sum_{k=0}^{N_k-1} (s_n(k))^2} \tag{2.3}$$

The zero crossing rate x_{zcr}

This feature contains information about how many times the signal crosses the zero level within a frame. Unvoiced speech and noise result in many crossings, voiced speech fewer. The feature is normalized by the maximum number of times it is possible to cross the zero level.

$$x_{zcr} = \frac{1}{N_k - 1} \sum_{k=1}^{N_k-1} \frac{1}{2} |sign(s_n(k-1)) - sign(s_n(k))| \tag{2.4}$$

where

$$sign(x) = \begin{cases} +1, & \text{for } x > 0 \\ 0, & \text{for } x = 0 \\ -1, & \text{for } x < 0 \end{cases} \tag{2.5}$$

The gradient index x_{gi}

This feature tells something about changes of direction of a signal. During voiced sounds which do not have many fluctuations, this measure gives a low value, whereas in unvoiced sounds where a lot of changes occur, the gradient index yields a high value [16]. The measure is defined in the following way:

$$x_{gi} = \frac{1}{10} \frac{\sum_{k=2}^{N_k-1} \Delta\Psi(k)|s_n(k) - s_n(k-1)|}{\sqrt{\sum_{k=0}^{N_k-1} (s_n(k))^2}} \tag{2.6}$$

with

$$\Delta\Psi(k) = \frac{|\psi(k) - \psi(k-1)|}{2}, \quad \in \{0, 1\} \tag{2.7}$$

$$\psi(k) = \frac{s_n(k) - s_n(k-1)}{|s_n(k) - s_n(k-1)|}, \quad \in \{-1, 1\} \tag{2.8}$$

The constant in front of the expression in (2.6) is a normalization factor to make the features lie in approximately the same range.

The normalized relative frame energy x_{nrp}

The normalized relative frame energy feature is a robust indication for voice activity. The power for voiced sounds are in general higher than for unvoiced sounds [22, p. 212]. Therefore x_{nrp} makes it possible to distinguish voiced/unvoiced sounds.
The normalized relative frame energy is calculated as [16, p.86]:

$$x_{nrp}(m) = \frac{\log_{10} E(m) - \log_{10} E_{min}(m)}{\log_{10} \bar{E}(m) - \log_{10} E_{min}(m)} \tag{2.9}$$

$E(m)$ is the energy for one frame:

$$E(m) = \sum_{k=0}^{N_k-1} (s_n(k))^2 \tag{2.10}$$

$E_{min}(m)$ is the minimum energy in the previous N_{min} frames:

$$E_{min}(m) = \min_{\eta \in \{0 \ldots N_{min}\}} E(m - \eta) \tag{2.11}$$

$\bar{E}(m)$ is the average frame energy calculated recursively:

$$\bar{E}(m) = \alpha \bar{E}(m-1) + (1 - \alpha) E(m) \tag{2.12}$$

The forgetting factor $\alpha = 0.96$ and the length of the search window $N_{min} = 200$ have been found to be reasonable [22].
Equation 2.11 requires that the previous N_{min} frames are available, which is not the case when the feature is extracted for the first N_{min} frames. The implementation is made such that as many as possible frames within the last N_{min} frames are used. The average calculation is performed recursively therefore it is necessary to initialize it. One possibility is to increase alpha from zero to 0.96 during startup. If the BWE should be used in a telephone it would be possible to store the average value from the last call. In our implementation we set the average value $E(1)$ to a specific value, that is the average of the first 200 frames.

The local kurtosis x_k

The local kurtosis is the measure of gaussianity of a signal and is computed using fourth and second moments[21].

$$x_k = log_{10} \frac{\frac{1}{N_k} \sum_{k=0}^{N_k-1} (s_n(k))^4}{\left(\sum_{k=0}^{N_k-1} (s_n(k))^2 \right)^2} \tag{2.13}$$

There are substantial peaks in the local kurtosis measure at the onsets of plosives and strong vowels [16].

The spectral centroid x_{sc}

The spectral centroid is defined as the center of gravity of the magnitude spectrum of the bandlimited speech [16], i.e the point in the magnitude spectrum where the power is equally distributed around:

$$x_{sc} = \frac{\sum_{i=0}^{N_i} i \cdot |S_n(e^{j\Omega_i})|}{\left(\frac{N_i}{2} + 1\right) \sum_{i=0}^{N_i/2} |S_n(e^{j\Omega_i})|} \tag{2.14}$$

Where $S_n(e^{j\Omega_i})$ denotes the i^{th} coefficient of a Discrete Fourier Transform (DFT) of the bandlimited speech frame, and N_i is the length of the DFT. The length of the DFT is specified to be greater than the number of samples in a speech frame, i.e $N_i > N_k$. This can be achieved by zero padding.

Figure 2.5 shows the features for a specific utterances. It is seen that the zero crossing rate x_{zcr} increases during unvoiced sounds, e.g. the "s-sound" at frame 30 and 160. The gradient index x_{gi} shows the same trend, but it is not entirely equal to x_{zcr} therefore we hope it will bring valuable information. The normalized frame energy x_{nrp} seems to be a good indication of voice activity and it also distinguish the voiced/unvoiced section, see e.g. the differences between "s" (frame 30) and "ae" (frame 20). The local kurtosis x_k have peaks at about frame: 65 (k), 110 (t), 170 (p), i.e. when a plosive consonant occur. A plosive consonant is a consonant which involves a complete blockage of the oral cavity [10]. The spectral centroid x_{sc} increase during unvoiced frames and plosives, e.g. frame 30 (s), frame 110 (t). During voiced frames it is about 0.25.

2.2.2 Feature vectors representing an envelope

This section describes the features which will be used to represent envelopes:

- AR coefficients
- Cepstral coefficients derived from AR coefficients
- Mel-frequency cepstral coefficients

The features will be described along with the advantages and disadvantages in the following three sections.

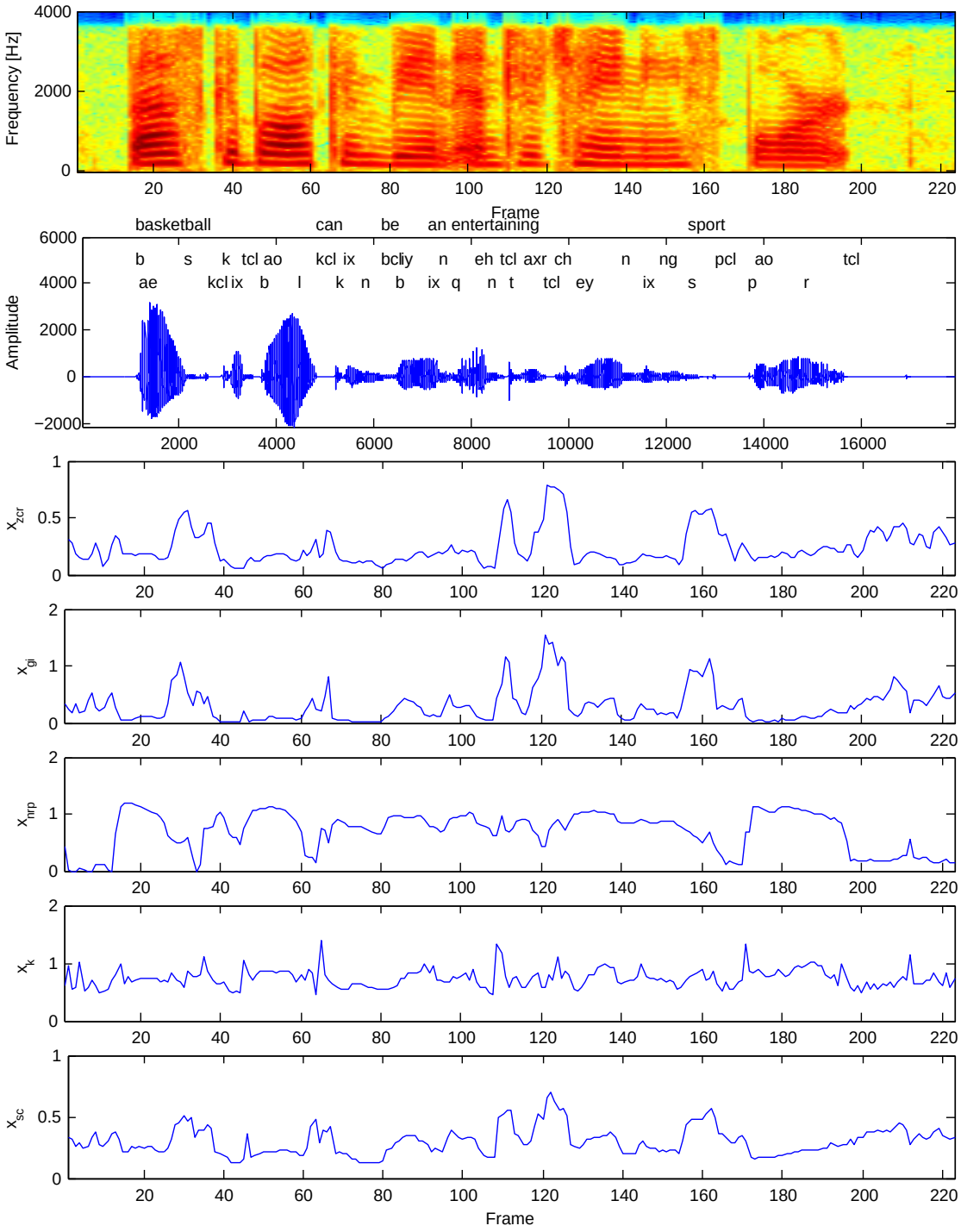


Figure 2.5: The plot shows all the scalar features in the file: TEST_CORE/DR2/FPAS0/SX44.WAV. A woman pronouncing "basketball can be an entertaining sport". The text is displayed above the time plot. The phonetic labels for the sounds pronounced are shown along with the utterance. The labels are shown in two lines to increase readability.

AR coefficients

The envelope can be represented by the AR coefficients, which describe the all-zero filter $A(z)$ (the analysis filter). The filter is defined as [24]:

$$A(z) = \sum_{i=0}^{P} a_i z^{-i} \tag{2.15}$$

The filter coefficient $(a_0 \dots a_P)$ defines the P^{th} order analysis filter. The coefficients can be found by performing linear prediction coding (LPC) on the speech samples.
Linear prediction is done by minimizing the prediction error between the actual sample $s(n)$ and the predicted sample $\hat{s}(n)$, based on a optimization criteria.
The prediction error is defined as [24]:

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{i=1}^{P} a_i s(n-i) \tag{2.16}$$

Usually the total squared error (α) is minimized, which is:

$$\alpha = \sum e(n)^2 \tag{2.17}$$

This can be done by using e.g. the Autocorrelation method. For more particulars on this topic see e.g. [24, 29].
The AR coefficients are a simple representation of the envelope and are computational attractive. AR coefficients in between each other are highly correlated. If a single AR coefficient are alternated it has high impact on the shape of the complete envelope. Two sets of AR coefficient with only one coefficient being different have a small euclidian distance. The shape of the envelopes however may differ a lot. Quantizing the AR coefficients from an Euclidian distance would therefore not be appropriate if similar envelopes should be clustered together.

Cepstral coefficients derived from AR coefficients

The common method to calculate the real cepstral coefficients is to perform a discrete time fourier transform of a signal, apply the log operator to the magnitudes of the output of the transformation, and then do an inverse discrete time fourier transform.
In this thesis we take advantage of the possibility to derive the cepstral coefficients from the AR coefficients. Markel and Gray [24, p. 230] have an equation from which both the CC $c(n)$ and AR coefficients a_i can be derived recursively:

$$-n \cdot c(n) - n \cdot a_n = \sum_{k=1}^{n-1} (n-k) \cdot c(n-k) \cdot a_k \qquad \text{for} \quad n > 0 \tag{2.18}$$

where

$$a_0 = 1 \quad \text{and} \quad a_k = 0 \quad for \quad k \geq P \quad \text{and} \quad c(0) = \ln(\alpha) \tag{2.19}$$

P is the order of the autoregressive model. α is the gain coefficient.

In order to find the cepstral coefficients equation 2.18 is rewritten to:

$$c(n) = -\frac{1}{n}\left(n \cdot a_n + \sum_{k=1}^{n-1}(n-k)\cdot c(n-k)\cdot a_k\right) = -a_n - \sum_{k=1}^{n-1}\frac{(n-k)}{n}\cdot c(n-k)\cdot a_k \quad n > 0 \quad (2.20)$$

The cepstral coefficients $c(0)$ and $c(1)$ can be found from (2.19) and (2.20) as:

$$c(0) = \ln(\alpha) \quad c(1) = -a_1 \quad (2.21)$$

The derived cepstral coefficients can be converted back into AR coefficients and gain again by applying the following ($n > 0$):

$$a_n = -\frac{1}{n}\left(n \cdot c(n) + \sum_{k=1}^{n-1}(n-k)\cdot c(n-k)\cdot a_k\right) = -c(n) - \sum_{k=1}^{n-1}\frac{(n-k)}{n}\cdot c(n-k)\cdot a_k \quad (2.22)$$

The conversion between AR coefficients and CC is relatively cheap and easy to perform. The conversion has the advantage that it can be reversed without any loss in information. It is an advantages to transform the AR coefficients into cepstral coefficients, because the cepstral representation has the advantage of a good decorrelation of the coefficients [22, p. 212]. This is advantageous for modeling CC with a probability density function (pdf), because the parameters of the pdf, can be estimated (optimized) for one element in the feature independently of the other elements. Furthermore minimizing the Euclidian distance between two sets of CC, is closely related to minimizing the Log Spectral Distortion (LSD) between two spectra [24].

Mel-frequency cepstral coefficients

The mel-frequency scale is a perceptual scale found by performing listening experiments. Listeners were asked to adjust the frequency of a tone, such that the frequency of which the tone was perceived was doubled, compared to the reference frequency 1000 Hz = 1000 mels. The adjusted frequency was then labeled 2000 mels and it corresponded approximately to 3.4 kHz. The listeners were then asked to adjust the frequency of the tone, such that it was perceived as various multiples of the reference. A mel frequency can be approximated from the normal frequency scale. The mapping is approximately linear below 1 kHz and logarithmic above [20].

The ability of humans to perceive a particular frequency is influenced by the energy in the critical band around this frequency [20]. The width of the critical band is linear below 1000 Hz and increases logarithmic above, as the mapping from frequency to mel-frequency. A plot of 29 critical bands is depicted in figure 2.6. There exist various different filterbanks (different center frequencies and number of filters), which are used for implementing MFCC, see e.g. [34] where four different filterbanks are presented. In our case we choose a filterbank with 29 filters, constant amplitude and logarithmic spacing.

To obtain the MFCCs, the power spectrum of the speech is calculated and the log-operator is applied, see figure 2.7. This is similar to calculating the normal cepstral coefficients.

The output is filtered with the critical bands from figure 2.6. That is, the DFT-bins are weighted according to the i^{th} triangular band and summed to form the total log energy ($Y_{TLE}(i)$) for the i^{th} band. Performing Inverse DFT (IDFT) results in the MFCCs. The

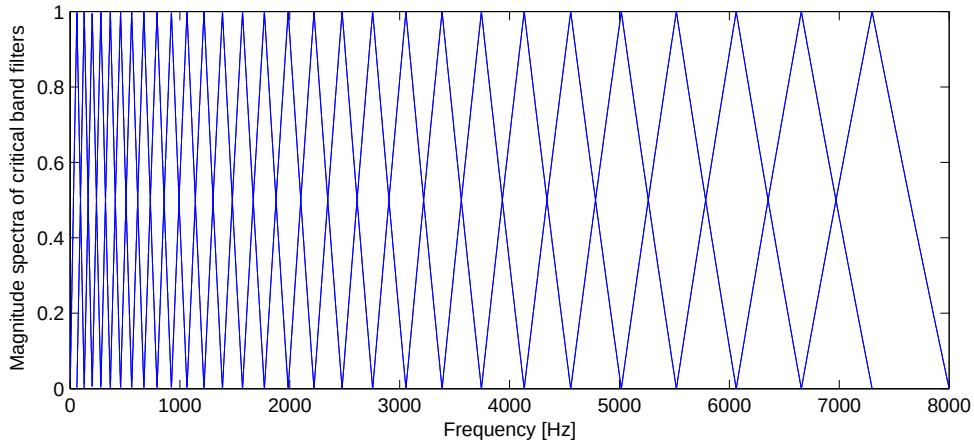


Figure 2.6: The magnitude spectra of 29 critical bands.

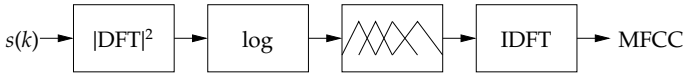


Figure 2.7: Block diagram showing the principle for calculating the MFCCs.

MFCC feature is popular for speech recognition applications. Studies have shown that a 13 dimensional MFCC feature vector achieve low word error recognition rates and an increase in dimension did not lead to an improvement [10]. Even though the MFCC feature is used for a different application in our case, 13 coefficients are used for representing the wideband envelope.

From the cepstral coefficients it is possible to calculate the AR coefficients again (2.22). A similar reconstruction of the log power spectrum is not possible for the MFCC, because the total log energy is calculated for each band as a sum of the underlaying bins. One particular bin can therefore not be reconstructed again. The main advantages of MFCC is that it includes perceptual properties.

The above mentioned method calculates the MFCC for the entire frequency band (0-8 kHz). Modeling only a subband, e.g. the band from 3.4 kHz to 8 kHz, is done by setting the lower critical bands to zero. On figure 2.6 it corresponds to keeping the 9 upper filters and setting the others to zero. When only 9 filters are used the number of cepstral coefficients can also be reduced. The modeling of a subband will be utilized in section 4.3.6.

2.3 Objective measures

A method for doing bandwidth extension was discussed in section 2.1. The bandwidth extension is performed by estimating a wideband envelope from features of the narrowband signal. The most optimal estimate would be the envelope from the original wideband speech signal. It would be useful to have some measure describing how "close" estimation is to the original envelope. One measure is to conduct a formal listening test which compares the original wideband signal and the bandwidth extended signal. Unfortunately this will be very time consuming whenever a new estimation is performed. Therefore an objective measure is introduced, which is less time consuming. This will give an

identification of how well the estimation is conducted.
Three different measures of how close the estimated envelope is to the original wideband envelope will be presented. It should be noted, that these objective measures will never be able to make it out for a subjective listening test.
The three objective measures to be presented are:

- Log Spectral Distortion (LSD)
- Itakura distance
- Itakura-Saito distance

Investigations have shown that these measures correlate reasonable well with subjective quality assessments [28, 16]. These measures are therefore a good alternative to listening tests. These measures are included in the evaluation in chapter 7.

2.3.1 Log Spectral Distortion

The log spectral distortion is a measure of the spectral differences between the original wb envelope and the estimated envelope. It is calculated as [22, eq. 6.19]:

$$d_{LSD}^2(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(20\log_{10}\frac{\sigma_w}{|A_w(e^{j\Omega};m)|} - 20\log_{10}\frac{\hat{\sigma}_w}{|\widehat{A}_w(e^{j\Omega};m)|}\right)^2 d\Omega \tag{2.23}$$

where $1/A_w(e^{j\Omega};m)$ denotes the original wideband frequency spectrum of the AR-model, and $\widehat{A}_w(e^{j\Omega},m)$ is the estimated frequency spectrum of the AR-model. σ denotes the gain for the envelopes and m denotes the frame number.
The measure can be implemented with a discrete fourier transform with N points as:

$$d_{LSD}^2(m) = \frac{1}{N}\sum_{k=0}^{N-1}\left(20\log_{10}\frac{\sigma_w}{|A_w(e^{j2\pi k/N};m)|} - 20\log_{10}\frac{\hat{\sigma}_w}{|\widehat{A}_w(e^{j2\pi k/N};m)|}\right)^2 \tag{2.24}$$

Equation 2.24 gives the LSD for one frame. The root mean square (RMS) LSD can be used as a result for several frames:

$$\bar{d}_{LSD} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} d_{LSD}^2(m)} \tag{2.25}$$

The log spectral distortion measure correlates reasonably well with the subjective speech quality [16].

2.3.2 Itakura and Itakura-Saito distance

Even though the Itakura distance is not a real distance measure, since it is not symmetric, it is widely used as a similarity measure between AR coefficients. The Itakura distance is heavily influenced by spectral dissimilarity due to mismatch in formant locations, which is desirable since the auditory system is sensitive to these errors [20]. The idea is to measure the log of the ratio between the total energy of the residual signal for two set of AR coefficients. Denoting this energy for α_a and α_b for two set of AR coefficients

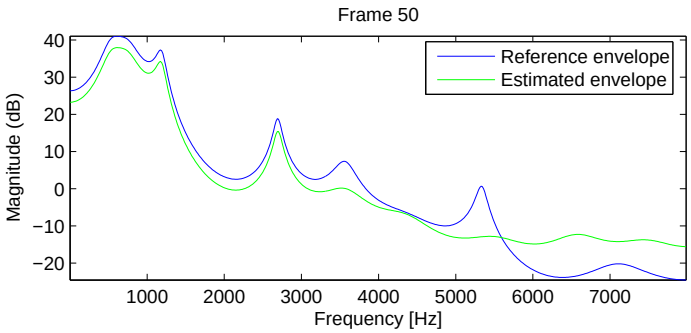


Figure 2.8: Two envelopes compared, which gave the following three measures; $d_{LSD} = 5.25$, $d_I = 0.64$ and $d_{I-S} = 0.59$.

$\mathbf{a} = [1, a_1, \ldots a_P]^{\mathrm{T}}$ and $\mathbf{b} = [1, b_1, \ldots b_P]^{\mathrm{T}}$ the Itakura distance can be calculated as [20, p. 329]:

$$d_I(\mathbf{a}, \mathbf{b}) = \log \frac{\alpha_b}{\alpha_a} = \log \frac{\mathbf{b}^{\mathrm{T}}\mathbf{R}\mathbf{b}}{\mathbf{a}^{\mathrm{T}}\mathbf{R}\mathbf{a}} \tag{2.26}$$

where $\mathbf{R}$ is the enlarged autocorrelation matrix with Toeplitz structure:

$$\mathbf{R} = \begin{bmatrix} r(0) & \cdots & r(P) \\ \vdots & \ddots & \vdots \\ r(P) & \cdots & r(0) \end{bmatrix} \tag{2.27}$$

The AR coefficients $\mathbf{a}$ are found from a signal with auto correlation matrix $\mathbf{R}$. $\mathbf{b}$ can be a completely different set of AR coefficients, e.g. from a different frame or an estimate from a codebook. The Itakura measure will always be non-negative, because α_a is the lowest possible achievable error. If the measure is zero, then the envelopes are identical and the more it increase, the more they differ.

A slight modification of the Itakura distance results in the Itakura-Saito distance, defined as [20]:

$$d_{I-S}(\mathbf{a}, \mathbf{b}) = \frac{(\mathbf{a} - \mathbf{b})^{\mathrm{T}}\mathbf{R}(\mathbf{a} - \mathbf{b})}{\mathbf{a}^{\mathrm{T}}\mathbf{R}\mathbf{a}} \tag{2.28}$$

For similar $\mathbf{a}$ and $\mathbf{b}$ the Itakura and the Itakura-Saito distance are almost equal. Throughout this report the distances will be reported as RMS values.

Figure 2.8 gives an impression of the three measures between two envelopes. To give a feeling of the correlation between the measures, figure 2.9 depicts the measures for several frames along with their RMS values. The measures seem to correlate, but differ also in some frames. Therefore we expect the measures will supplement each other.

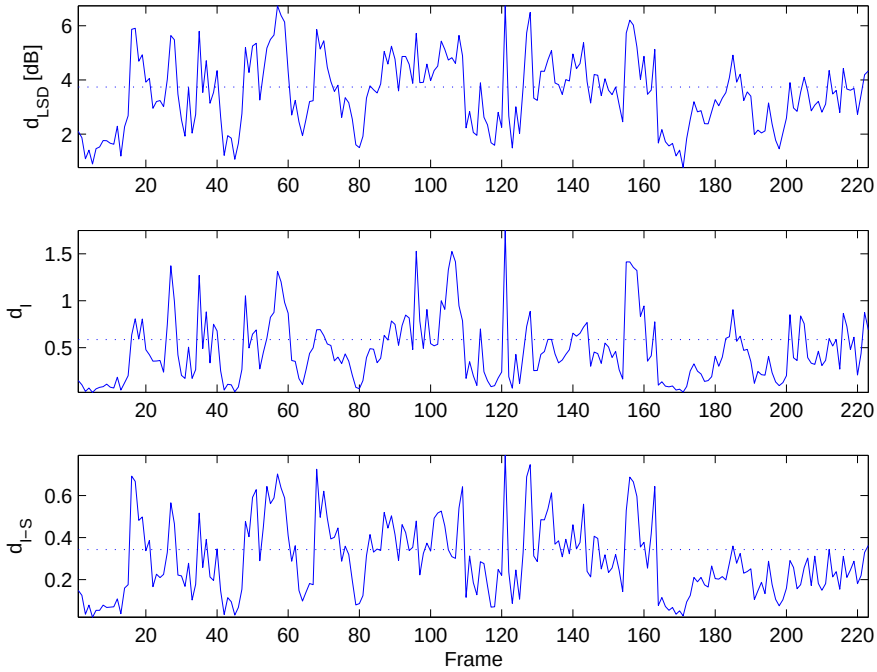


Figure 2.9: Comparison between envelopes for wideband speech and band width extended speech. The three measures a plotted along with their RMS values (dashed line). The speech file used is TEST_CORE/DR2/FPAS0/SX44.WAV. The reader is referred to figure 2.5 to see a time plot and a spectrogram of the speech.

Chapter 3

Bandwidth Extension Algorithm

Bandwidth extension can be performed in several ways. Choosing a parametric approach as the source-filter model reduces the number of different methods. Four of these are discussed and analyzed, and one is chosen for the realization of the algorithm.

A possible system for bandwidth extension was presented in figure 2.4. This section will give a more detailed discussion of possible frameworks for the bandwidth extension. Several ways of implementing a BWE algorithm using the source-filter model have been proposed in literature [22, 26, 5]. The framework is the system which encloses the estimators. It should be possible to choose different estimators for the envelope and excitation signal without modifying the framework. The framework shall be able to generate an estimate of the wide-band speech, from the available narrowband speech. The most optimal estimate of the narrowband speech available, is the narrowband speech itself, therefore the narrowband speech should not be modified.

The above can be summarized into two main concerns when choosing framework:

- Transparency in the narrowband region.
- Replacement of estimators should be flexible.

3.1 Feasible frameworks

Two fundamentally different frameworks will be presented which fulfil the transparency requirement. The first is similar to the framework presented in the introduction and is characterized by its mutual inverse analysis and synthesis filter. The second framework does not have inverse filters. To maintain the narrowband speech the estimated wideband speech is band-stop filtered and added to the narrowband speech.

A further division of the two fundamental frameworks is done during the description of the frameworks, such that there are four frameworks, which are:

- Frameworks based on mutual inverse filters
  - 1) Estimation of entire wb envelope
  - 2) Estimation of extensionband envelope and subband assembly

- Frameworks based on LP analysis and addition in time domain
  - 3) Estimate wideband envelope
  - 4) Estimate extensionband envelope

The advantages and disadvantages for the four different frameworks will be clarified, which leads to the choice of the framework. The main concern for this comparison is not computational complexity, but proof of concept.

3.1.1 Mutual inverse filters

This framework makes use of two mutual inverse filters - the analysis and synthesis filters. Figure 3.1 depicts a block diagram of the system. The narrowband speech signal, $s_n(k')$, is split into two branches. The purpose of the upper branch is analyze the signal in order to estimate the AR coefficients for the wideband envelope. Before analysis a Hanning window is applied to reduce sidelobes [20]. The window has a length which corresponds to 20 ms, and an overlap of 50% is used. Estimation of the wideband envelope can be done in several different ways, as discussed in section 1.2. For now we consider it as a black-box. Two ways of realizing it will be presented after describing the lower branch.

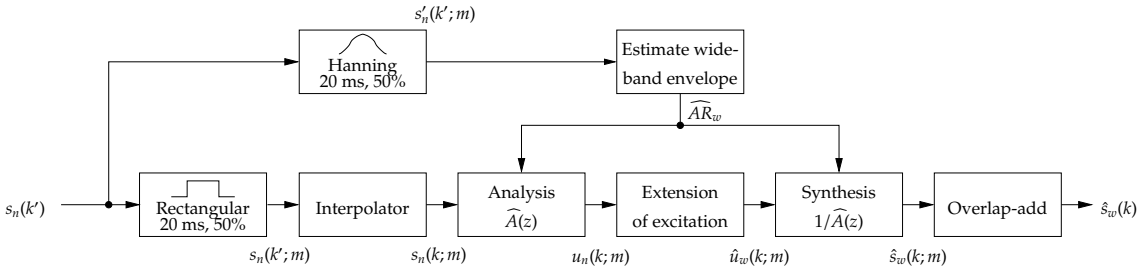


Figure 3.1: Block diagram of the BWE system. The estimated wideband envelope is used in both the analysis filter and the inverse synthesis filter. The frames applied a Hanning window is marked with a prime. Those in the signal path is left unchanged.

The lower branch is the signal path. Since no transformations are performed in the signal path a rectangular window is sufficient for dividing the signal into frames. The blocks which follow the window blocks are all processed on frame basis. The signal path consists of an interpolator, such that Nyquists sample theorem is kept after the extension. The discrete indices k'/k correspond to the signal having a sampling frequency of 8 kHz or 16 kHz respectively. This notation is used in the following. The interpolated frames are passed through the analysis filter, and the excitation signal is obtained. The narrowband excitation, denoted $u_n(k;m)$, is extended in the block Extension of excitation, which yields $\hat{u}_w(k;m)$ - the estimated wideband excitation. It is important $\hat{u}_w(k;m)$ only consists of the unmodified $u_n(k;m)$ in the narrowband frequency interval, such that the system remains transparent in the narrowband region. A modification of the signal will result in a change of the narrowband speech, which was required to be left unmodified.

The speech is reproduced by driving the synthesis filter with the wideband excitation signal. Since the analysis and synthesis filters are each others inverse, the narrowband signal is not modified. Finally the estimated wideband frames $\hat{s}_w(k;m)$ can be reassembled in the overlap-add block, to form the estimated wideband signal $\hat{s}_w(k)$.

The framework for the estimation of the AR coefficient for the wideband envelope will be discussed in the following.

Estimation of entire wb envelope

One approach for estimating the wideband envelope is to estimate the entire envelope over the frequency band from 0 to $f_s/2$. A block diagram for this approach is given in figure 3.2. First the feature vector $\mathbf{x}_m$ is extracted for every frame to reduce the complexity of the estimator. The wideband envelope is afterwards estimated based on some prior knowledge. It is important to notice that estimation of $\widehat{AR}_w$ is performed in a single step, i.e. a nb feature vector maps directly to a wb feature vector.

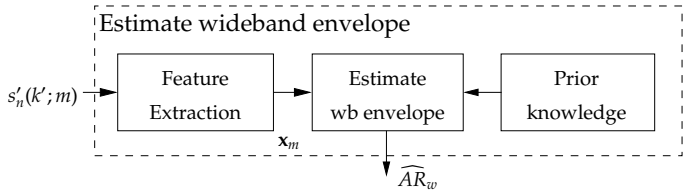


Figure 3.2: Block diagram of the framework for estimating the wideband envelope. Based on some prior knowledge, e.g. a codebook, the AR coefficients for the entire wideband envelope is estimated.

Estimation of extensionband envelope and subband assembly

The procedure in the following is depicted in figure 3.3. From the available narrowband signal a short term power spectrum can be obtained. An estimated extensionband power spectrum is approximated from an extensionband envelope. The two spectra are then assembled to form an estimate of the wideband power spectrum. This wideband power spectrum can be converted to a set of AR-coefficients $\widehat{AR}_w$, representing an wb envelope. As a result of using the narrowband power spectrum the estimated wideband and narrowband envelope are identical in the narrowband region.

The idea is to find the wideband power spectrum, $\Phi_w(\omega;m)$, and then transform it into the auto correlation function, $r(\eta;m)$, by using the inverse Fourier transform:

$$r(\eta;m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_w(\omega;m)\cdot e^{j\omega\eta}\,d\omega \tag{3.1}$$

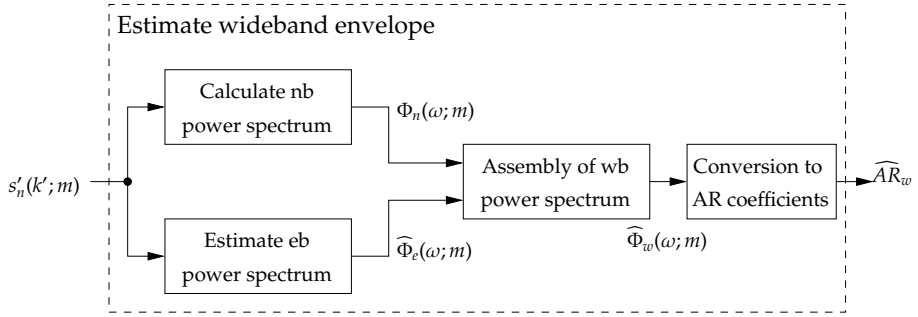


Figure 3.3: Block diagram for estimation of the extensionband power spectrum and subband assembly.

Where η is the sample lag. When the auto correlation function is available, the AR coefficients representing the wideband envelope can be calculated using the Levinson-Durbin algorithm.

The challenge is to find the wb power spectrum. It can be assembled from several subbands. If the narrowband speech consist of frequencies between 0 Hz and 3.4 kHz and an extension between 3.4 kHz and 8 kHz is desired, the estimation of the wideband power spectrum is done by calculating the nb power spectrum and concatenating it with the extensionband spectrum.

The power spectrum of $s'_n(k;m)$ is calculated as:

$$\Phi_n(\omega;m) = |S_n(\omega;m)|^2 \tag{3.2}$$

where $S_n(\omega;m)$ is the short-term Fourier transform of $s'_n(k;m)$:

$$S_n(\omega;m) = \sum_{k=0}^{N_k-1} s'_n(k;m)e^{-j\omega k} \tag{3.3}$$

is the k^{th} sample in the m^{th} frame. N_k is the number of samples within one frame. It should be noted that the nb signal $s'_n(k';m)$ is upsampled, thus k' is replaced by k in 3.3.

From (3.2) the nb power spectrum is obtained. The power spectrum outside this band is estimated since it is not available. In the following we assume only one extensionband, unless else is noted. The extensionband is defined to be within the frequency range: $\omega = \{\omega_{e,l} \dots \omega_{e,u}\}$. The subscripts denote Lower and Upper frequencies in the Extensionband respectively.

The extensionband power spectrum can be approximated with its extensionband envelope and a gain factor [24, eq. 6.1]:

$$\widehat{\Phi}_e(\omega;m) \approx \frac{\sigma_e^2}{|\widehat{A}_e(\omega;m)|^2} \tag{3.4}$$

$|\widehat{A}_e(\omega;m)|$ is the magnitude response of the estimated extensionband envelope. σ_e^2 is the power in the extensionband.

For convenience the relative variable σ_{rel}^2 is introduced as:

$$\sigma_{rel}^2 = \frac{\sigma_e^2}{\bar{\Phi}_n^2} \tag{3.5}$$

$\bar{\Phi}_n^2$ is the mean value of the power spectrum in the narrowband region. For more details about the gain factors see section 3.2.1. Equation 3.4 can then be rewritten as:

$$\widehat{\Phi}_e(\omega;m) \approx \frac{\sigma_{rel}^2 \cdot \bar{\Phi}_n^2}{|\widehat{A}_e(\omega;m)|^2} \tag{3.6}$$

The power spectrum for the extensionband can estimated, by estimating an eb envelope and the relative gain. The eb envelope is represented by AR coefficients. In order to only model the extensionband, selective linear prediction can be used advantageously. SLP is explained in details in section 3.2.1. The wideband spectrum is estimated as a

concatenation of the power spectra for nb and eb:

$$\widehat{\Phi}_w(\omega;m) = \begin{cases} \widehat{\Phi}_e(\omega;m) & \text{if extensionband } (\omega_{e,l} \le \omega \le \omega_{e,u}) \\ \Phi_n(\omega;m) & \text{otherwise} \end{cases} \tag{3.7}$$

The wideband spectrum is obtained and inserted in (3.1) which results in the auto correlation function. The AR coefficients $\widehat{AR}_w$ is calculated by applying Levinson-Durbin. A more detailed block diagram for estimating the wideband envelope utilizing this framework, is shown in figure 3.4.

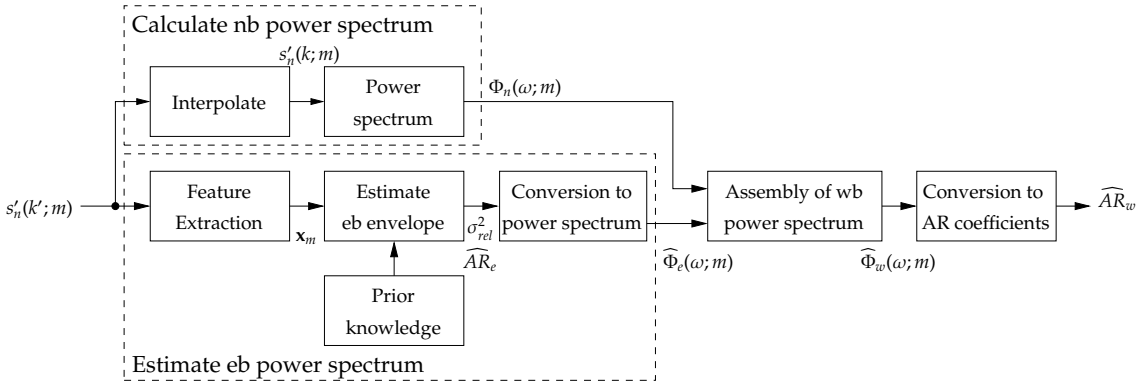


Figure 3.4: Block diagram of the framework for estimating the AR coefficients representing a wideband envelope. The nb and eb power spectra are assembled to form a wb power spectrum. The Levinson-Durbin algorithm is then applied to obtain $\widehat{AR}_w$.

It has been assumed that there is only one extensionband, but this framework can easily be extended to contain two or more extension bands. If two extensionbands are present the extension can be made by modifying (3.7) into:

$$\widehat{\Phi}_w(\omega;m) = \begin{cases} \widehat{\Phi}_{e1}(\omega;m) & \text{if extensionband } (\omega_{e1,l} \le \omega \le \omega_{e1,u}) \\ \widehat{\Phi}_{e2}(\omega;m) & \text{if extensionband } (\omega_{e2,l} \le \omega \le \omega_{e2,u}) \\ \Phi_n(\omega;m) & \text{otherwise} \end{cases} \tag{3.8}$$

The power spectra for the two extensionband can be approximated by their envelopes and gain as in (3.6).

3.1.2 LP analysis and addition in time domain

This section describes the other fundamental different method of how to extend a narrowband signal. In this approach, the excitation is obtained by applying LP analysis to the original narrowband signal, and then using the obtained coefficients in the analysis filter. The speech can be synthesized again in two different ways. Either by estimating a wideband envelope or an extensionband envelope. In the following the two methods are described.

Estimate wideband envelope

The description of this framework is supported by figure 3.5. A set of wb AR coefficients is estimated from narrowband speech. They represent the estimated wb synthesis filter.

The extended excitation is driven through this synthesis filter, which gives a wideband signal. A gain correction is applied after the synthesis filter. Even though the algorithm works reasonably well without estimating the gain, Park and Kim have shown that if gain is estimated, better results are achieved [26]. This signal is then band stop filtered resulting in a signal $\hat{s}_e(k)$ with only contributions outside the narrowband. This signal is then added to the interpolated signal $s_n(k)$ resulting in the extended speech signal $\hat{s}_w(k)$.

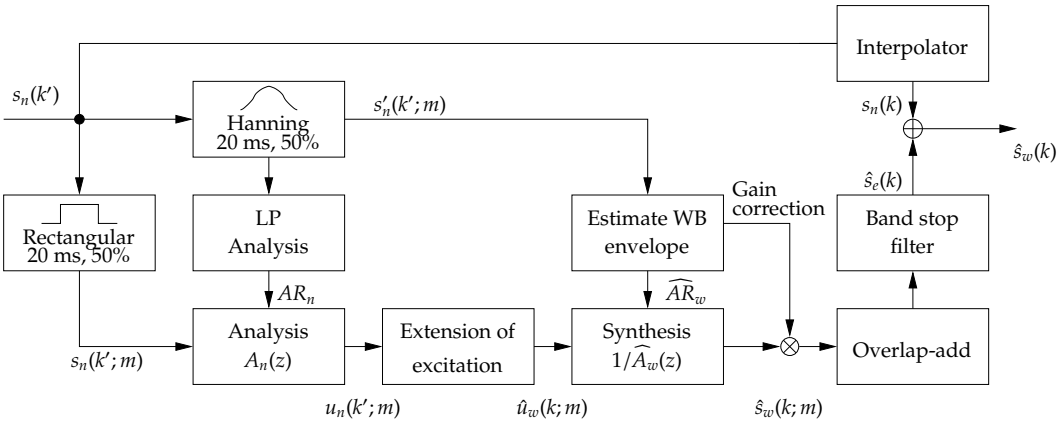


Figure 3.5: Bandwidth extension algorithm.

If only a band limited version of speech is available, say from 300-3400 Hz as in normal telephony, the band stop filter should be designed such that it attenuates this band. In this way an estimate of both frequency components below 300 Hz and above 3400 Hz is obtained.

Estimate extensionband envelope

Another approach is to estimate only a set of extensionband coefficients $\widehat{AR}_e$ instead of a set of wideband coefficients $\widehat{AR}_w$ as in the prior algorithm. The synthesized speech would then be an estimate of the speech in the extensionband. In this way the band stop filter could be omitted and the signals could be added directly, giving an estimate of the wideband speech.
If we wanted to estimate the lower frequencies in the signal (because the signal is band limited say from 300-3400 Hz), one more branch could simply be added which estimated a set of lower band coefficients $\widehat{AR}_{e1}$. These coefficients should then be used to obtain a speech signal, containing the lower frequencies. In fact this approach could be used to estimate different subbands in the missing frequencies, if that was desired.

3.1.3 Choice of framework

Of the four presented frameworks one framework should be chosen to realize the algorithm. Without having implemented the four frameworks completely it is difficult to determine the best one based on computational complexity. Therefore we have to base our choice on other parameters.

Estimation of gain: If the gain between the extensionband and narrowband is estimated wrong, the effect would be, that too much or too little is added the extensionband.

If too much is added, the extensionband will be dominating. On the other hand if too little is added it will at its worst sound as nb speech. The estimation of gain is seen as a weakness.

Parallel processing: The estimation of eb envelopes can be carried out in parallel, if there is more than one extensionband. Parallel should be understood as the subbands can be estimated independently of each other, e.g. each subband in the power spectrum can be calculated individually in (3.6). Estimating the entire wideband envelope without dividing it into subbands makes it impossible to do parallel processing. The ability to do parallel processing is seen as an advantage.

Subband optimization: Estimating subbands makes it possible to optimize the bandwidth extension algorithm individually for each extension band. The ability to analyze and optimize in a subband without influence from other subbands is a major strength.

LP derived analysis filter: All the four frameworks include the EXTENSION OF EXCITATION block. This block extends the excitation signal. In order for this block to have low complexity, it should have an input signal with certain characteristics. According to the source-filter model the excitation signal consist of basic signals such as a pulse train and/or white noise with a certain gain. This knowledge is exploited in this block, such that the block obtain low complexity. To get a signal with these characteristics the most optimal excitation signal is obtained by doing a LP analysis on $s_n(k;m)$. If $u_n(k;m)$ is found using another analysis filter it may result in a non-spectral flat excitation signal in the extensionband, as assumed in the source-filter model. This would result in an error being replicated in the extensionband, if conventional methods to extend the excitation are used. Therefore it is an advantage if the analysis filter for the framework matches an analysis filter with coefficients found by linear prediction.

The numbering of the four frameworks are listed here for convenience:

- Frameworks based on mutual inverse filters
  - 1) Estimation of entire wb envelope
  - 2) Estimation of extensionband envelope and subband assembly
- Frameworks based on LP analysis and addition in time domain
  - 3) Estimate wideband envelope
  - 4) Estimate extensionband envelope

Table 3.1 gives an overview of the above mentioned parameters for the frameworks.

Parameter	Framework			
	1	2	3	4
No estimation of gain	√	%	%	%
Subband optimization:	%	√	%	√
Parallel processing	%	√	%	√
LP derived analysis filter	%	√	√	√

Table 3.1: Overview over different parameters for the four frameworks. The frameworks are numbered in the order they were presented. √ denotes a strength and % denotes a weakness.

Only advantage of framework 1 is no gain estimation. The LP derived analysis filter is seen as a more important parameter, hence framework 1 is discarded. Framework 3 is not capable of parallel processing and subband optimization. Estimation of subbands instead of complete wb envelope furthermore reduces the number of coefficients. This is an advantage during both training and estimation, since the number of features (coefficients) which are to be estimated is lower. Framework 3 is therefore discarded.

Due to time constrains only one framework is fully implemented. Of the remaining two frameworks, we choose to base the algorithm on framework 2 (Estimation of extension-band envelope and subband assembly). This framework seems more elegant from an implementation point of view.

3.2 Realization of BWE algorithm

In order to implement the chosen framework, a set of AR coefficients has to be estimated for the extensionband. In the following we describe how selective linear prediction is applied to efficiently represent an extensionband envelope.

3.2.1 Selective linear prediction

Selective linear prediction is linear prediction conducted on a limited frequency band of a signal. In our case it is used to model the extensionband from the lower limit $\omega_{e,l}$ to the upper limit $\omega_{e,u}$.

One method for conducting linear prediction on a limited frequency interval could be to band pass filter the input signal and then do regular LPC. The band pass filter should have the above mentioned frequencies as cut-off frequencies. This method has one major weakness; the LPC will try to model the slopes of the band pass filter, which result in wasted model capabilities.

To avoid this, another method known as Selective Linear Prediction (SLP) is used. The SLP algorithm is made up from the following four steps [24, p. 148]:

Compute the power spectrum: compute the Fourier transform of the signal and square its magnitudes:

$$\Phi_w(\omega) = |\mathcal{F}\{s_w(k)\}|^2 \tag{3.9}$$

Where $\mathcal{F}\{\bullet\}$ is the Fourier Transform. See e.g. the power spectrum in figure 3.7.

Create a translated spectrum: Calculate discrete frequency bins indices l_1 and l_2 from respectively $\omega_{e,l}$ and $\omega_{e,u}$. Form a translated power spectrum $\Phi_t(s)$ as shown in figure 3.6 , by letting $\Phi_t(s) = \Phi_w(l)$ where
$l = [l_1, l_1 + 1, \ldots, l_2, I - l_2 + 1, I - l_2 + 2, \ldots, I - l_1 - 1]$ and $s = [0, \ldots, 2(l_2 - l_1) - 1]$
Note that (3.9) is implemented as a DFT, with I points - therefore ω is substituted with the discrete indices s and l. An example of the translated power spectrum can be seen in figure 3.8 ($\omega_{e,l} = 0.44$, $\omega_{e,u} = 1$)

Compute the auto correlation sequence: Do a inverse DFT to obtain the auto correlation function of the translated spectrum.

$$r_t(\eta) = \mathcal{F}^{-1}\{\Phi_t(s)\} \tag{3.10}$$

Compute the SLP model parameters: Based on the auto correlation function calculate the AR coefficients e.g. by using the Levinson Durbin algorithm. A plot of the envelope used in the example is given in figure 3.9, (an order of 10 has been used).

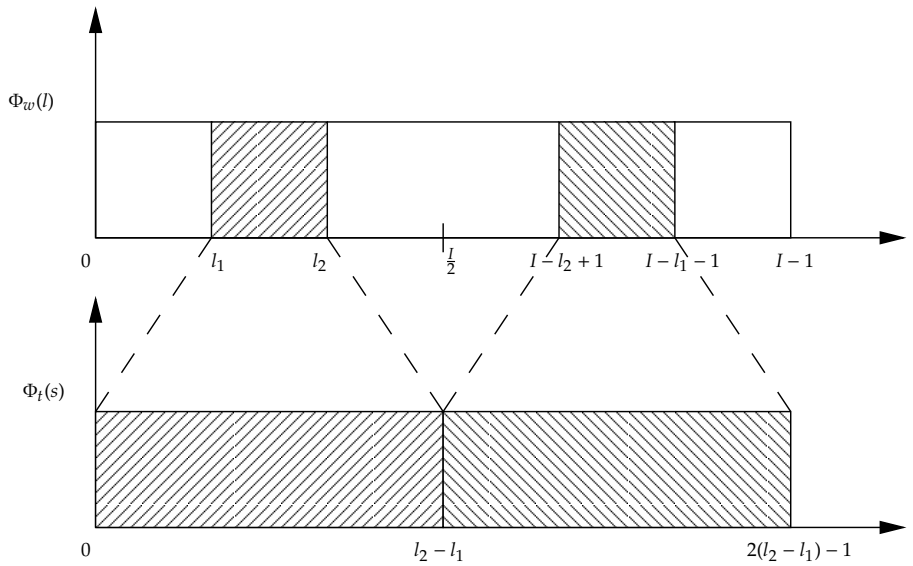


Figure 3.6: The wideband power spectrum ($\Phi_w(l)$) and the translated spectrum ($\Phi_t(s)$). It should be noted that the spectra are double-sided.

3.2.2 Relative gain

The SLP algorithm is used to model an extensionband. The goal in the chosen framework is to estimate the extensionband envelope and perform subband assembly (described in section 3.1.1). In order to perform this assembly, it is necessary to have a gain relation between the available narrowband and the estimated extensionband power spectrum. This gain is referred to as the relative gain and is a estimated value.
Previous work of Jax [16] has defined a relative gain as:

$$\sigma^2_{rel} = \frac{\sigma^2_e}{\sigma^2_n} \tag{3.11}$$

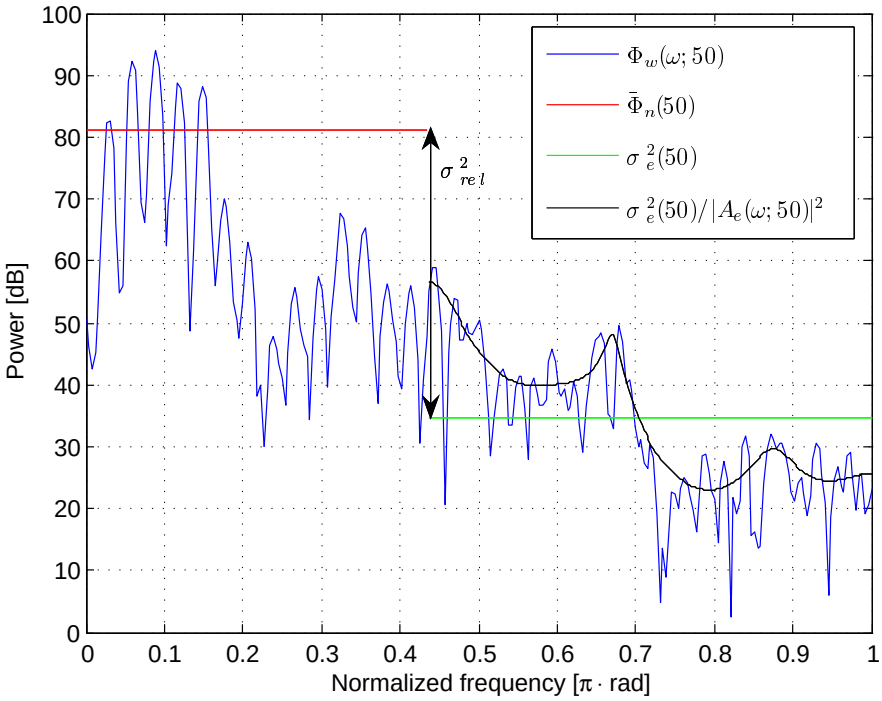


Figure 3.7: Power spectrum. The envelope found in the extensionband with SLP is depicted to ease comparison. The gain factors are plotted as well. For this frame the relative gain factor (σ^2_{rel}) is $2.25 \cdot 10^{-5}$ or -46.5 dB.

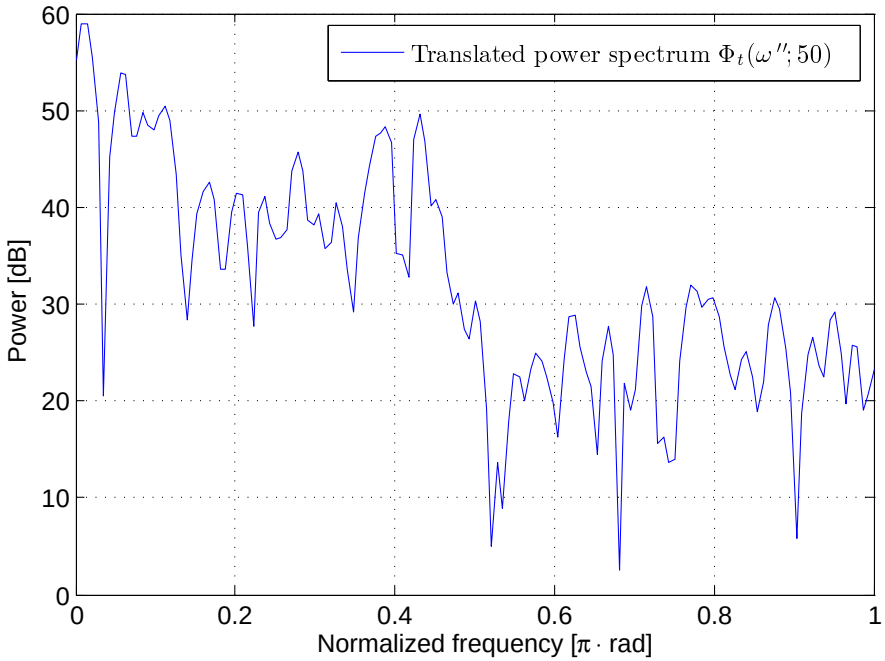


Figure 3.8: Translated power spectrum formed from the wideband power spectrum in figure 3.7 with the extensionband frequencies: $\omega_{e,l} = 0.44$, $\omega_{e,u} = 1$. It should be noted that ω'' denotes change in the frequency range after translation.

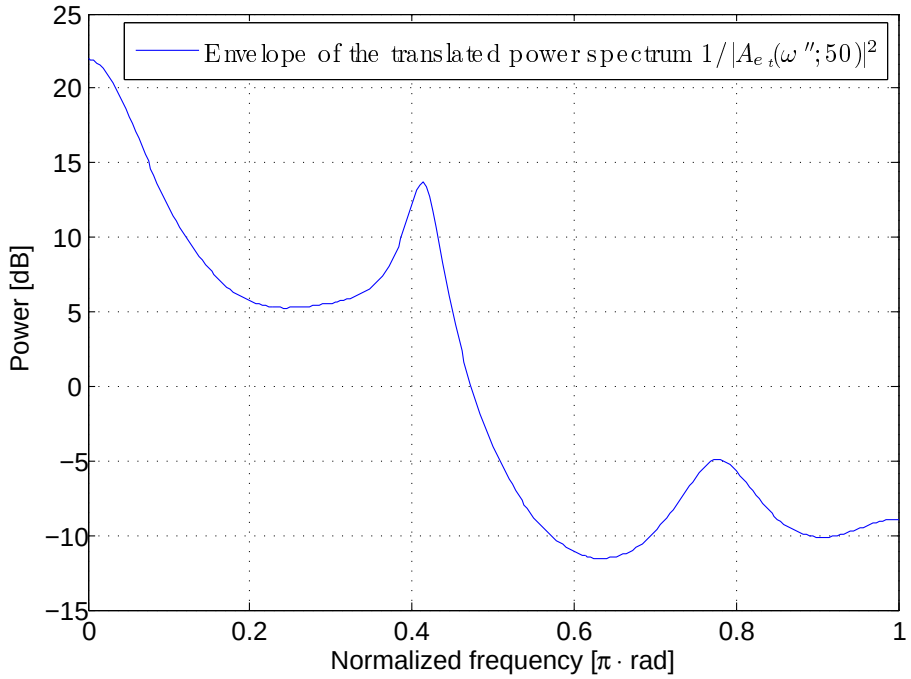


Figure 3.9: Envelope of the SLP. Calculated from the translated spectrum in figure 3.8. The model order was set to 10. Note the envelope is not multiplied by the gain σ_e^2 and therefore it is not in the same power interval as figure 3.8.

Where σ_n^2 and σ_e^2 were found as the prediction error for SLP performed on narrowband and extensionband respectively. The prediction error is equal to the gain factor σ_e^2 in equation 3.4.

In this work we slightly modify the relative gain. SLP is performed on the extensionband, the gain factor σ_e^2 is therefore obtained from the Levinson-Durbin algorithm. The gain factor for the extensionband is plotted in figure 3.7. It is also seen that multiplying the extensionband envelope $1/|A_e(\omega;m)|^2$ by the gain factor makes a good approximation for the power spectrum, which is also indicated by equation 3.4.

The narrowband gain factor in this work is found directly from the available narrowband power spectrum. This makes it unnecessary to compute the SLP in the narrowband region, on the contrary to Jax. The narrowband gain factor is simply the mean value of the power spectrum in the narrowband interval $(\omega_{n,l} \ldots \omega_{n,u})$:

$$\bar{\Phi}_n(m) = E[\Phi_w(\omega;m)] = E[\Phi_n(\omega;m)] \quad \omega \in \{\omega_{n,l} \ldots \omega_{n,u}\} \tag{3.12}$$

Since the wideband and the narrowband spectrum are equal in the narrowband frequency interval, $\bar{\Phi}_n(m)$ can be calculated from both spectra.

The relative gain from (3.11) is modified into:

$$\sigma_{rel}^2(m) = \frac{\sigma_e^2(m)}{\bar{\Phi}_n(m)} \tag{3.13}$$

The narrowband gain factor and the resulting relative gain are depicted in figure 3.7. As the relative gain and the SLP extensionband envelope can be estimated in the bandwidth extension algorithm it is possible to form an estimate of the wideband envelope as described in section 3.1.1. The SLP parameters represent a frequency translated envelope $(1/A_{e_t}(\omega'';m))$. To remove this translation the following mapping is applied to achieve the non-translated eb envelope and thereby the estimated eb power spectrum:

$$\Phi_e(\omega; m) \approx \frac{\sigma^2_{rel}(m) \cdot \bar{\Phi}_n(m)}{|A_{e_t}(\pi \cdot \frac{\omega - \omega_{e,l}}{\omega_{e,u} - \omega_{e,l}}; m)|^2} \quad \omega \in \{\omega_{e,l} \ldots \omega_{e,u}\} \tag{3.14}$$

After obtaining this, the extension spectrum can be concatenated with the narrowband spectrum as given in equation 3.7.

3.2.3 Model order of extensionband

Here we address the problem of choosing the number of coefficients used for modeling the extensionband (3.4-8 kHz). These coefficients will be integrated in the prior knowledge block in figure 3.4. It will be an advantage to keep the number of coefficients on a minimum, such that the prior knowledge is compact. On the other hand the number of coefficients must not be too few such that the envelope can not be represented properly. Markel and Gray [24, p.154] have, based on the physics of speech production system and the velocity of sound, found a relation between the model-order and the sampling frequency as: P = f_s (sampling frequency in kHz.). For voiced sounds it can be necessary to increase the model-order by 4 or 5 to model the glottal and lip radiation characteristics. In our case with a sampling frequency of 16 kHz it will lead to an order between 16 and 21 for a wideband signal. Since several of the coefficients are used for modeling the narrowband, we must expect that the order for the extensionbands is less.

We will investigate the RMS LSD for different model-orders for the extensionband. The estimated $\widehat{AR}_w$ is compared by means of RMS LSD to AR-coefficients found from doing LP-analysis on the wideband signal. The procedure is explained from figure 3.4; the block Estimate eb envelope is bypassed and substituted by a SLP analysis of the wideband signal. Different model orders of the SLP analysis are investigated. The RMS LSD are calculated for the test core in the TIMIT database (192 files, 58,269 frames) and yield the following result:

SLP model-order	7	8	9	10	11	12	13	14	15
RMS LSD [dB]	1.04	0.72	0.36	0.17	0.13	0.11	0.10	0.10	0.09

From the table we see that increasing the model order from eight to nine and from nine to ten, results in a reduction of approximately 50 percent in RMS LSD. Increasing the order to more than ten coefficients only results in a minor improvement of the RMS LSD. Therefore the model-order is chosen to ten. A tenth-order model yields 0.17 dB RMS LSD. This value can be viewed as an average lower bound for the RMS LSD during estimation, and can only be achieved if the estimation is completely correct.

3.2.4 Discussion

The model order for the eb has been found to be ten, based on non-estimates. However a more intensive study of the number of coefficients should be carried out by varying the number of coefficients which has to be estimated. That is, reducing the feature vector representing the extensionband used in training. It is likely that it would result in a better estimate, because fewer coefficients has to be estimated, i.e the degrees of freedom reduces. A better estimate would result in a lower RMS LSD. This would likely compensate for the loss introduced by using lower model orders in the SLP.

Chapter 4
Extension of Envelope

The framework for the algorithm has been established in the previous chapter. A further investigation of the estimation of envelope is given. We will elaborate on estimation of the extensionband envelope and the prior knowledge required for the estimation. This corresponds to going into details about the blocks Estimate eb envelope *and* Prior knowledge *on figure 3.4 on page 27. Three different methods for the estimation approach are presented. For each method an estimator is derived. Aside from estimation, the training of these models are discussed and implemented from a common feature set.*

The chosen framework requires a set of narrowband features **x** for estimating another set of features **y** representing the extensionband envelope. This estimate is based on prior knowledge obtained from training different models. The quality of the estimator, which yields the estimate, is highly dependent on the training of prior knowledge. Therefore training and estimation is given the primary focus.

Three methods for generating the primary knowledge are presented and implemented. The methods will be presented in the order given below:

Codebook: The codebook contains mean values obtained, by clustering the features. The estimate is then obtained by quantizing the narrowband feature vector and using the mean value of the cluster.

Gaussian Mixture Model: The GMM is able to model the probability density function of the data and thereby avoid quantization of the data. The quantization of the narrowband feature, **x**, can result in valuable information about the extensionband being lost. The GMM method therefore achieves better results than the codebook solution.

Hidden Markov Model: The HMM is able to model hidden information, e.g. how a speech sequence evolves over time. Therefore it utilizes information about previous frames to estimate the extensionband. The HMM is the most sophisticated method of the three methods.

Before the estimation methods are discussed, the general setup for extracting the features will be presented. The TIMIT database provides wideband speech files, which are used to train the models. For each frame of these files both a narrowband feature vector **x** and a feature vector **y** describing the extensionband envelope are extracted.

The features which are included in the narrowband vector, are described in section 2.2.1 on page 11. The specific features are the:

- Auto correlation function: $\mathbf{x}_{acf} = [x_{acf}(1), x_{acf}(2), \ldots, x_{acf}(10)]^{\mathrm{T}}$
- Zero crossing rate: x_{zcr}
- Gradient index: x_{gi}
- Normalized relative frame energy: x_{nrp}
- Local kurtosis: x_k
- Spectral centroid: x_{sc}

The features are concatenated to form the feature vector $\mathbf{x} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \end{bmatrix}$, where $\mathbf{x}_{scl}$ is the scalar features: $\mathbf{x}_{scl} = [x_{zcr}, x_{gi}, x_{nrp}, x_k, x_{sc}]^{\mathrm{T}}$ giving $\mathbf{x}$ a dimension of 15.
The AR coefficients which are used to model the eb envelope is found using selective linear prediction (SLP). This means that only the extensionband from $\omega_{e,l}$ to $\omega_{e,u}$ is modeled; resulting in the coefficients AR_e and the relative gain σ^2_{rel}. The model order for the AR model is set to 10 when modeling the extensionband from 3.4-8 kHz (found in section 3.2.3 on page 34).
The AR coefficients could be utilized during training, but since the cepstral coefficients have better quantization properties and good decorrelation, the AR coefficients are transformed into cepstral coefficients. In the transformation, the gain coefficient α (2.19) is converted into the cepstral coefficient $c(0)$. To include the relative gain (σ^2_{rel}) from the SLP analysis it replaces the gain coefficient in the conversion. The conversion results in the following vector for representing the extensionband: $\mathbf{y} = [c(0), c(1), \ldots, c(10)]^{\mathrm{T}}$, with a dimension of 11.
The feature vectors $\mathbf{x}$ and $\mathbf{y}$ can be extracted for each frame of the available speech to form several observations, which will be utilized in the training. The procedure for extracting the features from the TIMIT database is depicted in figure 4.1.

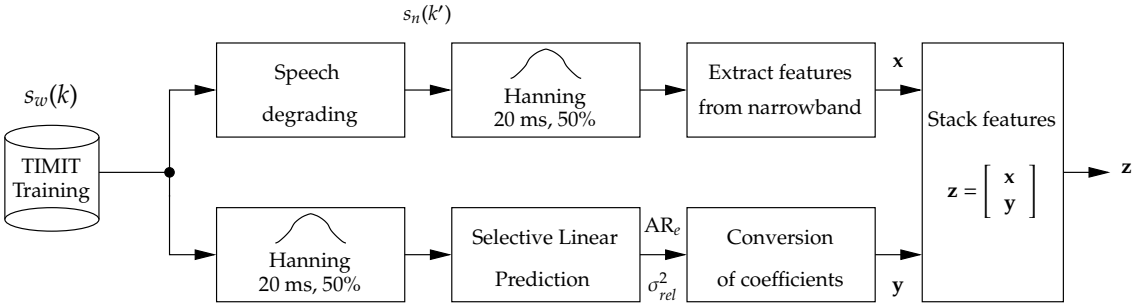


Figure 4.1: Block diagram illustrating features being extracted from the training data in the TIMIT database.

Wideband speech signals s_w are available from the training part of the TIMIT database. The size of the database including the training part is described in appendix A.1. A narrowband version $s_n(k')$ of the speech is obtained by applying a lowpass filter and then subsequently downsample the signal (see e.g. appendix A.2 for particulars).
The narrowband signal $s_n(k')$ goes through the upper part and the wideband signal $s_w(k)$ goes through the lower part on the figure. Both signals are split into frames and a Hanning window is applied. Features from both signals are then extracted, without preprocessing, e.g; noise suppression, excluding silence frames and no manual selections. The AR coefficients are converted into cepstral coefficients. The last block stack the nb and eb feature vectors, which results in the concatenated feature vector $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$. The stacking

makes it possible to describe the joint distribution of $\mathbf{x}$ and $\mathbf{y}$, i.e. the distribution of $\mathbf{z}$. The joint distribution is utilized in the estimation, were the prior knowledge about $\mathbf{z}$ is used to estimate $\mathbf{y}$ from $\mathbf{x}$, i.e estimate the extensionband from narrowband. Storing of the joint distribution is different for the three methods. This also result in the estimation being performed differently for each of the three methods.
In this thesis the number of extensionbands to be estimated is one. If more than one extensionband were to be estimated, the number of joint distributions can e.g. be increased, such that the following are modeled separately as; $\mathbf{z}_1 = \left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{y}_1\end{smallmatrix}\right]$, $\mathbf{z}_2 = \left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{y}_2\end{smallmatrix}\right], \ldots, \mathbf{z}_N = \left[\begin{smallmatrix}\mathbf{x}\\ \mathbf{y}_N\end{smallmatrix}\right]$, where $\mathbf{y}_i$ denotes the feature vector from the i^{th} extensionband. Another solution is to model $\mathbf{z}$ as:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}$$

Since only one band is estimated ($N = 1$) $\mathbf{z}$ is:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

Training of the joint distribution $\mathbf{z}$ will now be presented for the three methods together with estimation of $\mathbf{y}$ given $\mathbf{x}$.

4.1 Codebook-based method

One of the simplest way to obtain some a priori knowledge about the dependencies between the narrowband and the wideband envelope would be to utilize a codebook as the statistical model. In the beginning of the project a codebook approach was chosen to get some hands on feeling about the estimation problem. The codebook denoted $\mathbb{C}_z$, is trained using the stacked feature vector $\mathbf{z}$ consisting of both narrowband features and extensionband features. After training has completed, the codebook $\mathbb{C}_z$ is obtained. We now define an entry $\mathbb{C}_z(i) = [\mathbb{C}_x(i)\ \mathbb{C}_y(i)]$, where $\mathbb{C}_x(i)$ is the part of the entry describing the narrowband features and $\mathbb{C}_y(i)$ describing the extensionband features. By training using the stacked vector $\mathbf{z}$, an estimate of $\mathbf{y}$ can be obtained from $\mathbf{x}$. This can be done by comparing $\mathbf{x}$ and $\mathbb{C}_x(i)$ for all i, and minimizing a distortion measure. The i yielding the lowest distortion, will then be the estimate $\mathbb{C}_y(i)$ of the extensionband envelope. In our implementation the Squared Euclidean distance is chosen as the distortion measure, due to computational tractability during both training and estimation, e.g. the update of centroids during training is straight forward and computational fast.

4.1.1 The LBG algorithm

The distribution of the features for the database is unknown. Therefore the training of the codebook will be based on an unknown distribution. Numerous kinds of vector quantizers exist, which can solve this problem. Furthermore different approaches to design the vector quantizers can be taken, e.g. tree structured vector quantizer, supervised

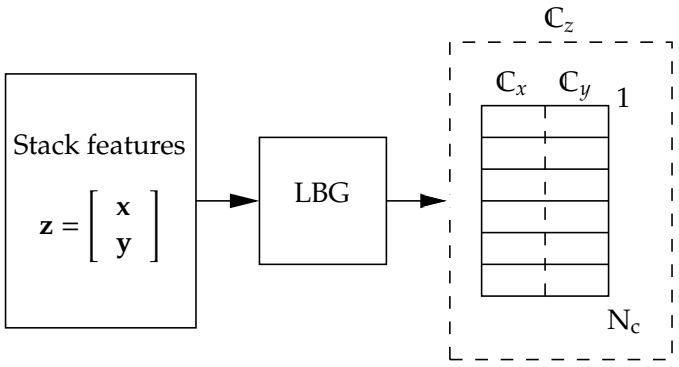


Figure 4.2: Training procedure for obtaining a codebook $\mathbb{C}_z$, including both narrowband features and extensionband features.

learning, and various distortions measures. The Linde, Buzo and Gray (LGB) is a simple and well-known procedure and it is therefore chosen to vector quantize the features $\mathbf{z}_t, t \in \{1, 2 \ldots, T\}$. The algorithm consist of these steps [23] [20, p. 72]:

1. Initialization. Chose the following:

   - N_c - the number of vectors in the codebook
   - $M = N_c \cdot \beta$ arbitrary training features $\mathbf{z}_t$, $t \in \{1, 2....T\}$. $\mathbf{z}_t$ is a feature vector for the t^{th} observation. β is the training ratio [9, p. 191]
   - $\epsilon \geq 0$ - the distortion threshold.
   - N_c initial codewords to form the initial codebook $\mathbb{C}_z^l(i)$, $i \in \{1, 2, \ldots, N_c\}$. Arbitrarily chosen from the T training features.

   Set iteration number $l = 0$ and the initial distortion $D_{-1} = \infty$.

2. Quantization: Quantize each training vector $\mathbf{z}_t$ using the codebook $\mathbb{C}_z^l$:

$$i^* = \underset{i}{\operatorname{argmin}}\, d(\mathbf{z}, \mathbb{C}_z^l(i)) \tag{4.1}$$

3. Distortion: Calculate the average distortion between $\mathbf{z}$ and its quantized value, $Q(\mathbf{z})$:

$$D^l = \frac{1}{T}\sum_{t=1}^{T} d(\mathbf{z}_t, Q(\mathbf{z}_t)) \tag{4.2}$$

4. Check if the distortion have decreased enough. If

$$\frac{D^{l-1} - D^l}{D^l} \leq \epsilon \tag{4.3}$$

   use $\mathbb{C}_z^l$ as the final codebook else continue with 5.

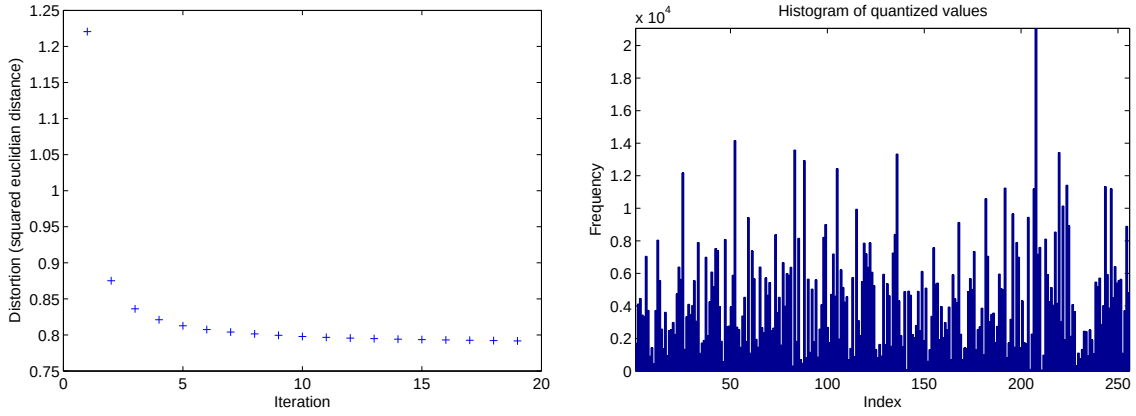
5. Calculate new centroids for the codebook $\mathbb{C}_z^{l+1}$: For all i calculate the mean of all the $\mathbf{z}_t$ quantized into the i^{th} codebook entry.

$$\mathbb{C}_z^l(i) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{z}_t \tag{4.4}$$

   Increase l to $l = l + 1$ and go to 2.

The training ration β has been chosen, such that training is conducted using all training data to form the joint codebook $\mathbb{C}_z$. The convergence criteria is set by letting $\epsilon = 0.0005$. After convergence has been achieved the joint codebook $\mathbb{C}_z$ is obtained. This is illustrated in figure 4.2.

A plot of the average distortion during training as a function of iterations, can be seen in figure 4.3(a). A codebook with a size of 256 clusters has been used in this example. The distortion measure d which is used during training is squared Euclidean. The average distortion can then be calculated from (4.2). As can be seen, the average distortion decreases as expected for each iteration.



(a) Plot of average distortion as a function of iterations. The distortion can be observed to decrease steadily during iterations.

(b) Histogram showing the number of vectors going into each cluster during training.

Figure 4.3: Distortion as a function of iterations and histogram for a codebook of size $N_c = 256$.

Figure 4.3(b) shows a histogram of the number of vectors going into to each cluster. From the figure we see that the variation is quite large. Some clusters have a high number of vectors going into them, whereas others have low numbers. Ideally this distribution should be uniform.

4.1.2 Initialization

In order to get a good initialization, the clusters should "span" or cover the whole distribution of the training data. If some prior knowledge were available about the distribution of the training data, the clusters could then be designed to cover the whole distribution. This is seldom the case, therefore different approaches can be taken in order to initialize the codebook. A simple method is to choose N_c vectors from the training set randomly. These vectors then constitute the initial codebook $\mathbb{C}_z^0$. Other approaches could e.g. be to do splitting. In this implementation N_c samples are drawn randomly from the training data, and used as the initial codebook.

4.1.3 Estimation using nearest neighbor

Estimation is done using nearest neighbor with Euclidean distance. Because only $\mathbf{x}$ is available this is both an estimation and a classification problem. First we need to classify

to which cluster $\mathbb{C}_x(i)$ $\mathbf{x}$ belongs. Then we need to estimate the corresponding $\mathbf{y}$, which would be $\mathbb{C}_y(i)$. An estimator can now be formulated as:

$$i^* = \underset{i \in \{1...N_c\}}{\operatorname{argmin}} d(\mathbf{x}, \mathbb{C}_x(i)) \tag{4.5}$$

Equation 4.5 is the classifier. Having the index i^*, the corresponding estimate of $\mathbf{y}$ can be found as:

$$\hat{\mathbf{y}} = \mathbb{C}_y(i^*) \tag{4.6}$$

Only the i^* which results in the lowest distortion is used in the estimator. Estimation is therefore done by hard-decision. A soft-decision estimator could have been derived by choosing the n nearest i's and then forming $\hat{\mathbf{y}}$ as a linear combination of the corresponding mean values, weighted by the relative distortion.

4.1.4 Discussion

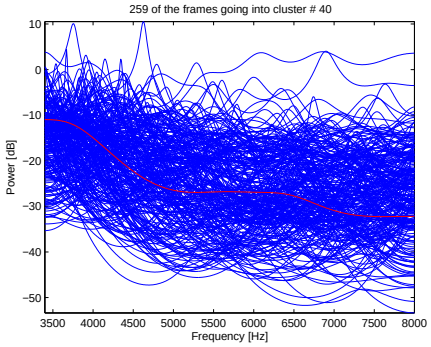
In this section it has been shown how a possible implementation of an estimator using a codebook can be done. The estimator both relies on how well the narrowband features are classified and how well the extensionband features are estimated from this classification. In order to achieve this, good separability in $\mathbf{x}$ is desired together with a high correlation between $\mathbf{x}$ and $\mathbf{y}$. Because this is a codebook implementation, it only looks at snapshots of the speech, and not how it evolve over time (a supervector could of course be created in which more frames are stacked. Most people tends to use other models for this problem, e.g. HMMs). Because no interframe modeling is done, one could imagine that it would result in valuable information being lost. Furthermore the result of the quantization could also result in degraded correlation between $\mathbf{x}$ and $\mathbf{y}$.
To see things from an engineering point of view, the clusters and the corresponding vectors which made up the cluster, have been examined. Figure 4.4 illustrates this small study. A codebook using all training vectors has been trained (approx. 1.1 million vectors). Codebook size is $N_c = 256$. From these training data 4% (45267 vectors) have been chosen randomly and quantized using the codebook. The envelope representing the extensionband part $\mathbf{y}$ of the training vector $\mathbf{z}$ is plotted together with the cluster $\mathbb{C}_y(i)$, which the training vector is quantized to. In this case $i = 40, 110$. A remark needs to be given here. As recalled the relative gain is :

$$\sigma^2_{rel} = \frac{\sigma^2_e}{\bar{\Phi}_{s_n}}$$

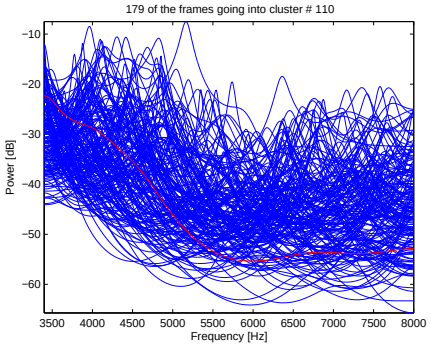
From the estimated features $\mathbf{y}$ the extensionband envelope $1/|A_e(z)|^2$ and σ^2_{rel} can be obtained. This is plotted in figure 4.4 as:

$$\frac{\sigma^2_{rel}}{|A_e(z)|^2}$$

The envelopes used in training are the blue ones and the resulting cluster is the red colored envelope. If estimation is to be carried out having only the corresponding narrowband envelope to one of the blue extensionband envelopes in figure 4.4(a), the estimate would



(a) Envelopes representing extensionband. Blue envelopes are those used in training. The red envelope is obtained from codebook entry $\mathbb{C}_y(40)$.



(b) Envelopes representing extensionband. Blue envelopes are those used in training. The red envelope is obtained from codebook entry $\mathbb{C}_y(110)$.

Figure 4.4: Envelopes from clusters $\mathbb{C}_y(40)$ and $\mathbb{C}_y(110)$ respectively in a codebook of size 256. It is observed that the resulting clusters is somehow averaged out and the extensionband envelopes are not separated properly. It should be noted that only 4% of all the training vectors have been quantized. The codebook however has been trained using all data.

therefore be the red envelope. This envelope is then assembled as described in section 3.1.1.

What is observed from figure 4.4 is that the shapes of the envelopes being clustered is mutually very different. This is not very fortunate. What is desired is clusters made up from envelopes having almost the same shape. With, of course, some variability. The less the better. Because all the envelopes contributing to the clusters are so differently in shape, the "cluster" envelope does not contain the interesting peaks, which we would like to estimate. Instead it gets averaged out, and in a way only represent the relative gain σ^2_{rel} for the extension band.

To see if this trend apply for all clusters in the codebook, all the clusters from a single codebook is plotted in figure 4.5. To make it more comprehensible and foreseeable only a codebook of size $N_c = 32$ is plotted. Both training and plots for this figure has been done in the same way as figure 4.4. It is clearly seen that the shape of the envelopes for some clusters are very much alike. From the plot it may seem like, when doing the clustering, the biggest impact on the clustering is the gain. It seems like the envelopes have been clustered primarily according to relative gain, and afterwards by shape of the envelope. This would mean that a lot of differently shaped envelopes would go to the same cluster. When the quantized values of these envelopes are calculated, it shows that the clusters of the codebook, when converted to envelopes, get averaged out. This indicates low correlation between narrowband and the extensionband, at least when representing the envelopes using these features. Since the feature vector $\mathbf{z}$ consists of different features, a better clustering would most likely be obtained if another distance measure was utilized and/or other features. One obvious solution could be to do a weighting of each of the elements in the feature vector or e.g. normalize each element such that they would get unit variance. One other idea could be to use Euclidean distance on the cepstrum part, i.e $\mathbf{y}$, and some other distance measures on $\mathbf{x}$, and then define a weighting of the two as a total measure. Instead of using a hard decision in the estimator, a soft decision approach could be taken as e.g. in [6], where a weighting of the clusters are utilized.

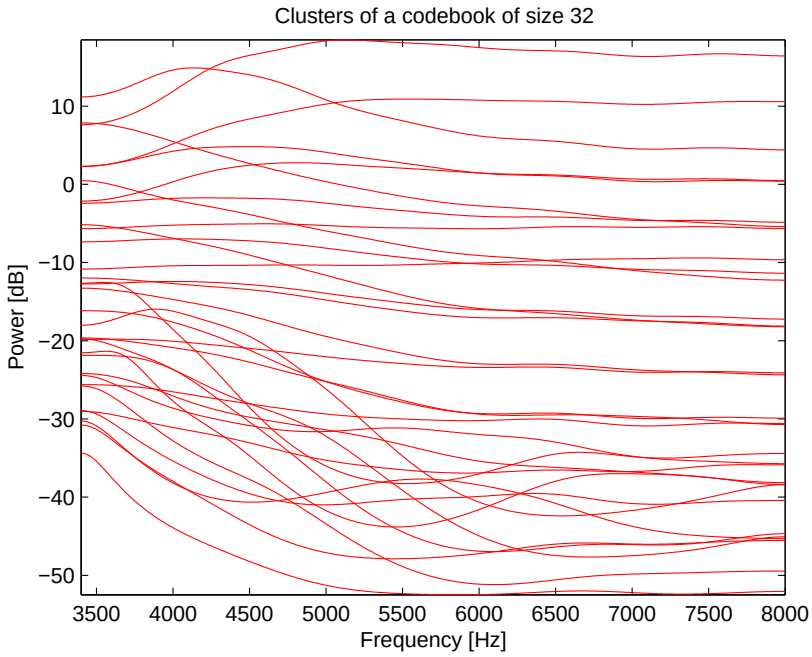


Figure 4.5: Clusters of a codebook of size 32.

One general disadvantage of the codebook implementation is that quantization of both $\mathbf{x}$ and $\mathbf{y}$ could result in valuable information being lost. Information which could result in the estimate being more accurate. By representing the features by a GMM, the information which is lost due to quantization using codebooks, will not occur. This is because a GMM is capable of doing a continuously modeling of the training data. This has shown to reduce the estimation error.

4.2 GMM-based method

Compared to the codebook approach the GMM avoids the quantization, because it models the data continuously. In a previously study a GMM has been utilized to estimate a full wideband envelope from narrowband features. This method was compared to a similar codebook approach (mapping from nb to wb) and the former method outperformed the codebook approach both in objective and subjective tests [26].

It is possible to model the distribution as a histogram of the data, but this would require a large amount of storage. The GMM is a parametric approach and is capable of modeling relatively complex distributions with only few parameters. The GMM is able to model any arbitrary pdf, due to the central limit theorem [10]. The stacked feature vector $\mathbf{z}$ is modeled by a GMM, corresponding to modeling the joint pdf between $\mathbf{x}$ and $\mathbf{y}$: $f(\mathbf{x}, \mathbf{y})$. When this pdf is available, $\mathbf{y}$ can be estimated, based on the observation of $\mathbf{x}$.

A GMM consists, as the name indicate, of a mixture of Gaussian densities. The density function for the stacked vector $\mathbf{z}$ can be modeled by the mixture of M Gaussian densities as [2]:

$$f(\mathbf{z}) = \sum_{m=1}^{M} c_m \cdot g(\mathbf{z}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{4.7}$$

In order for $f(\mathbf{z})$ to be a probability density function, the following requirements needs to

be fulfilled:

$$\int f(\mathbf{z})d\mathbf{z} = 1$$

$$f(\mathbf{z}) \geq 0, \forall\ \mathbf{z}$$

This is done by constraining the weighting coefficients c_m as:

$$\sum_{m=1}^{M} c_m = 1, \quad c_m \geq 0, \forall\ m \tag{4.8}$$

These coefficients are also known as the mixture coefficient or the mixture gains. Fixing $M = 1$ in (4.7) leads to a regular multivariate Gaussian distribution.

$g(\mathbf{z}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$ is the d-dimensional multivariate Gaussian distribution for the mth mixture component and is defined as:

$$g(\mathbf{z}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_m|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_m)^{\mathrm{T}}\boldsymbol{\Sigma}_m^{-1}(\mathbf{z} - \boldsymbol{\mu}_m)\right) \tag{4.9}$$

The shape of the m^{th} distribution is determined by the mean vector ($\boldsymbol{\mu}_m$) and the covariance matrix ($\boldsymbol{\Sigma}_m$). To shorten the notation, the complete set of parameters for a GMM is defined as:

$$\Theta = \{c_1, c_2, \ldots, c_M, \theta_1, \theta_2, \ldots, \theta_M\}$$

where $\theta_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$.

The plots in figure 4.6 depict the pdf of five GMMs with various numbers of mixture components. The pdfs model the distribution of two arbitrary chosen features: the spectral centroid (x_{sc}) and the first cepstral coefficient $c(1)$. It is seen that increasing the number of mixture components leads to a better fit with the histogram. The correlation between the two parameters are limited, but $x_{sc} \approx 0.35$ seems to yield a high probability for $c(1)$ taking a value close to zero. The model with 8 mixture components seems to model the two peaks from the histogram. The GMM with 32 mixture components follows the histogram very well. The number of mixture components can be increased, but this will perhaps lead to an over-fitting of the model. Too few mixture components will result in the model not being capable of modeling the data properly. The number of mixture components is therefore a tradeoff between; not being able to model the data, or over-fitting the model and having unnecessary complexity.

In order to be able to estimate using a GMM as prior knowledge, its parameters need to be know. No analytic solution exist for finding the parameters of a GMM. Therefore the parameters have to be estimated.

4.2.1 EM Algorithm

The Expectation-Maximization (EM) algorithm is a widely used method for estimating parameters of a GMM given a set of observations. That is, it maximizes the probability of a certain set of observations being generated from a distribution with a given set of parameters. This is done by adjusting the parameters such that the likelihood for these parameters is maximized. The EM algorithm performs this estimation iteratively, and for each iteration it guarantees an increase in likelihood.

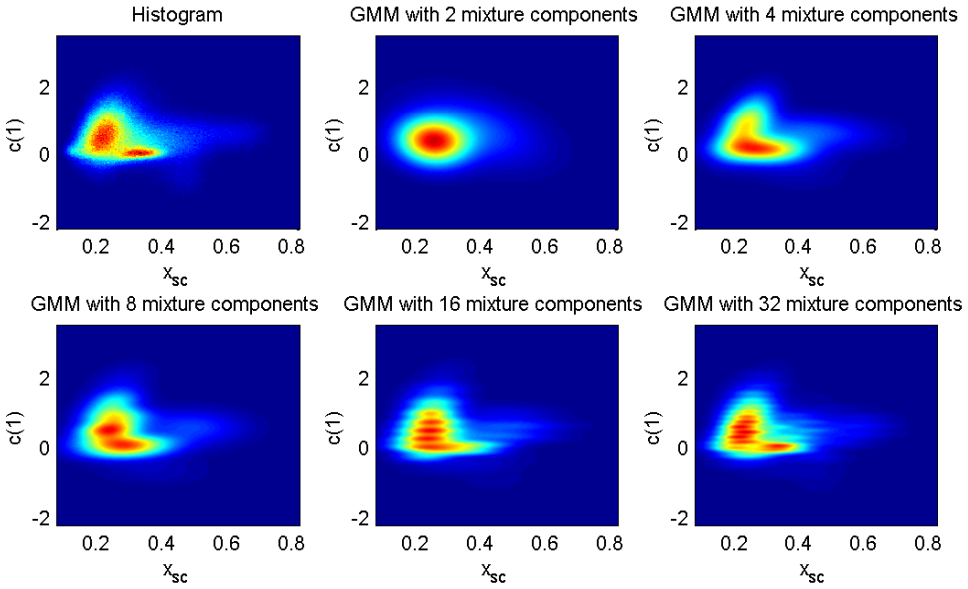


Figure 4.6: GMM densities for different number of mixture components (2, 4, 8, 16 and 32). The GMMs model the joint distribution between the spectral centroid x_{sc} and the first cepstral coefficient c(1). A histogram of the data is plotted in the top left corner. 1 million observations are used for training of the GMMs and for the histogram. The color corresponds to the density $f(x_{sc}, c(1))$. The figures are normalized and warm colors correspond to higher density.

The observed data is denoted as $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$. It is assumed that the observations are independent and identical distributed with the distribution $f(\mathbf{z})$. From these data, a set of parameters Θ, which maximizes the likelihood, is estimated. The likelihood function for the observations can be written as [2]:

$$f(\mathcal{Z}|\Theta) = \prod_{t=1}^{T} f(\mathbf{z}_t|\Theta) = L(\Theta|\mathcal{Z}) \tag{4.10}$$

The in-dependency assumption is exploited to divide the likelihood function into a product of each individual observation. The objective is to maximize the likelihood by adjusting the parameter Θ, i.e. we would like to solve:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, L(\Theta|\mathcal{Z}) = \underset{\Theta}{\operatorname{argmax}} \prod_{t=1}^{T} f(\mathbf{z}_t|\Theta) = \underset{\Theta}{\operatorname{argmax}} \prod_{t=1}^{T} \sum_{m=1}^{M} c_m \cdot g(\mathbf{z}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{4.11}$$

Since $g \geq 0$ and $c_m \geq$ the likelihood function will always be non-negative. The logarithm is a monotonically increasing function and therefore maximizing the log-likelihood is proportional to maximizing likelihood. If the log-likelihood is maximized (4.11) is changed into:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, \log\left[L(\Theta|\mathcal{Z})\right] = \underset{\Theta}{\operatorname{argmax}} \sum_{t=1}^{T} \log\left[\sum_{m=1}^{M} c_m \cdot g(\mathbf{z}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\right] \tag{4.12}$$

The problem is divided into two parts; the Expectation and the Maximization steps, which also gave the algorithm its name.

The expectation step (E-step) is performed by calculating the posterior probability based on the observations $\mathcal{Z}$ and the multivariate Gaussian distributions [2]:

$$P(m|\mathbf{z}_t, \Theta^l) = \frac{c_m^l g(\mathbf{z}_t|\boldsymbol{\mu}_m^l, \boldsymbol{\Sigma}_m^l)}{\sum_{n=1}^{M} c_n^l g(\mathbf{z}_t|\boldsymbol{\mu}_n^l, \boldsymbol{\Sigma}_n^l)} \tag{4.13}$$

where $P(m|\mathbf{z}_t, \Theta^l)$ is the probability that the mth mixture component with parameters Θ^l generated the observation $\mathbf{z}_t$. The superscript l denotes iteration. An initialization of Θ is necessary and will be discussed in the next section. Equation 4.13 is normalized by the sum of likelihoods for observing $\mathbf{z}_t$ from all the mixture components, i.e. $\sum_m P(m|\mathbf{z}_t, \Theta^l) = 1$. Summing the log of denominator in (4.13) yields the log-likelihood for the entire set of observations:

$$\log[L(\Theta|\mathcal{Z})] = \sum_{t=1}^{T} \log\left[\sum_{m=1}^{M} c_m \cdot g(\mathbf{z}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\right] \tag{4.14}$$

The E-step is calculated for all observations and all mixture components. The posterior probability is utilized during the maximization step. An analytic expression for finding an update for Θ^l can be found when the mixture densities are Gaussian distributions. This results in the maximization step (M-step) which is made up from the following steps for updating the parameters for each mixture component [2]:

$$c_m^{l+1} = \frac{1}{T}\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l) \tag{4.15}$$

$$\boldsymbol{\mu}_m^{l+1} = \frac{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)\mathbf{z}_t}{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)} \tag{4.16}$$

$$\boldsymbol{\Sigma}_m^{l+1} = \frac{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)(\mathbf{z}_t - \boldsymbol{\mu}_m^{l+1})(\mathbf{z}_t - \boldsymbol{\mu}_m^{l+1})^{\mathrm{T}}}{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)} \tag{4.17}$$

Some comments need to given for these equations. Equation 4.15 can be interpreted as the more the m^{th} mixture component is expected, the more weight should be given to this component. c_m^{l+1} is normalized with the number of observations, such the constrain on the sum is kept. Equation 4.16 weights the observations $\mathbf{z}_t$ with the posterior probability, so those observations which lead to a high probability for the m^{th} mixture component are weighted most in the calculation of the m^{th} mean vector. Updating of the covariance matrix is conducted similar to as for updating the mean vectors. It should be noted that the updated mean vector $\boldsymbol{\mu}_m^{l+1}$ is utilized for updating the covariance matrix.

The E-step and M-step are conducted iteratively followed by each other until the algorithm reaches convergence. A diagram depicting the training is shown in figure 4.7.

Convergence is achieved, when the absolute increase in log-likelihood between two iterations is below the threshold ϵ. The algorithm then stops and the final parameters

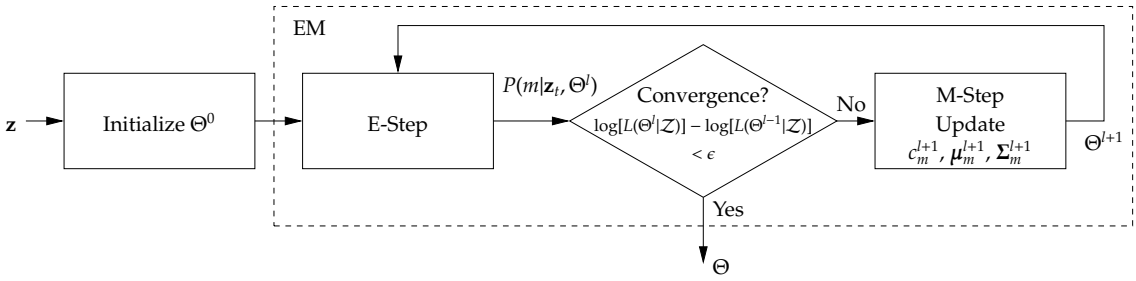


Figure 4.7: Diagram of the training for the GMM approach. The E-step is (4.13). The M-step is (4.15), (4.16) and (4.17). The equation for updating the covariance matrix can be replaced by (4.18) if the covariance matrix is diagonal.

Θ obtained. Each iteration makes the log-likelihood increase and it is ensured that the algorithm converges to a local maximum of the log-likelihood function (4.14) [2].

If low cross-correlation is assumed, the covariance matrix in (4.17) can be written as a diagonal matrix. By writing it as a diagonal matrix, fewer parameters are required to be estimated/updated. As a consequence fewer data is also required in order to get a reliable estimate of the model parameters, because cross correlation is not included as a parameter. Furthermore a reduction in parameters makes estimation of parameters faster, concerning both the amount of data which is processed and calculation of equation 4.9.

Even though the cross-covariance (off-diagonal elements) is removed from the covariance matrixes, it has been noted that the overall mixture model is able to model covariance between the feature vector elements [32, p. 75]. In order to achieve the same effect for diagonal matrixes as for full covariance matrixes the number of mixture components should however be increased. If diagonal covariance matrixes are used, equation 4.17 simplifies to:

$$\boldsymbol{\Sigma}_{m,j}^{l+1} = \frac{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)(\mathbf{z}_{t,j} - \boldsymbol{\mu}_{m,j}^{l+1})(\mathbf{z}_{t,j} - \boldsymbol{\mu}_{m,j}^{l+1})^{\mathrm{T}}}{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)} = \frac{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)(\mathbf{z}_{t,j} - \boldsymbol{\mu}_{m,j}^{(l+1)})^2}{\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l)} \tag{4.18}$$

where $\boldsymbol{\Sigma}_{m,j}^{l+1}$ is the j^{th} diagonal element in the covariance matrix for the m^{th} mixture component, $\mathbf{z}_{t,j}$ is the j^{th} element/dimension in the t^{th} sample and $\boldsymbol{\mu}_{m,j}$ is the j^{th} element in the mean vector for the m^{th} mixture component. All the off-diagonal elements of the covariance matrix should be set to zero if equation 4.18 is used instead of equation 4.17.

4.2.2 Initialization

The EM-algorithm need to be initialized properly, in order to reach a "good" local or global maximum. The initialization of the algorithm is chosen to be done by clustering. The observed data $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T\}$ is split into m clusters and for each cluster $c_m, \boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ is calculated.

$$c_m = \frac{T_m}{T} \tag{4.19}$$

Where T_m is the number of samples going into the m^{th} cluster, and T is the total number of samples. The mean is calculated as follows:

$$\boldsymbol{\mu}_m = \frac{1}{T_m}\sum_{t=1}^{T_m}\mathbf{z}_{t,m} \tag{4.20}$$

Where $\mathbf{z}_{t,m}$ is all the samples going into the m^{th} cluster. $\boldsymbol{\Sigma}_m$ can then be calculated in the following way:

$$\boldsymbol{\Sigma}_m = \frac{1}{T_m}\sum_{t=1}^{T_m}(\mathbf{z}_{t,m} - \boldsymbol{\mu}_m)(\mathbf{z}_{t,m} - \boldsymbol{\mu}_m)^{\mathrm{T}} \tag{4.21}$$

In our implementation 20% of the observations have been used for the clustering and calculating the initial Θ. To increase convergence speed of the EM-algorithm, the EM-algorithm is run for 2 iterations with this reduced data set. This results in a modified Θ which is used for another two iterations using the next 20% of the data. This procedure can be summarized in the following pseudo-code:

1. Calculate initial Θ based on clustering of 20% the observations
2. Run the EM-algorithm for two iterations with 20% the observations.
3. If more data left; Use the updated Θ and the next 20% in point 2, else return the last updated Θ and use it for the EM algorithm with all the observations.

The number of iterations is set to two, because the first iterations changes the log-likelihood most and since it is an initialization the EM algorithm is not required to converge completely.

4.2.3 Estimation

Two estimators for $\mathbf{y}$ are derived - one for mixture models with full covariance matrixes and one for diagonal. The estimators have been chosen such that they minimizes the mean squared error. The estimator minimizes the mean squared error between the estimated extension-band feature $\hat{\mathbf{y}}$ and the real extension band feature $\mathbf{y}$.
The Minimum Mean Squared Error (MMSE) estimator can then be written as:

$$\hat{\mathbf{y}}_{MMSE} = \underset{\hat{\mathbf{y}}}{\operatorname{argmin}}\, E[(\hat{\mathbf{y}} - \mathbf{y})^{\mathrm{T}}(\hat{\mathbf{y}} - \mathbf{y})|\mathbf{x}, \Theta] \tag{4.22}$$

$\mathbf{x}$ denotes the observation which is observed and $\hat{\mathbf{y}}$ is the parameter which are changed. The estimator is conditioned on $\mathbf{x}$ and the GMM model (Θ). The mean squared error is:

$$E[(\hat{\mathbf{y}} - \mathbf{y})^{\mathrm{T}}(\hat{\mathbf{y}} - \mathbf{y})|\mathbf{x}, \Theta] = \hat{\mathbf{y}}^{\mathrm{T}}\hat{\mathbf{y}} + E[\mathbf{y}^{\mathrm{T}}\mathbf{y}|\mathbf{x}, \Theta] - 2E[\hat{\mathbf{y}}^{\mathrm{T}}\mathbf{y}|\mathbf{x}, \Theta] \tag{4.23}$$

Differentiating the right part of equation 4.23 with respect to $\hat{\mathbf{y}}$ and setting it equal to zero in order to find the minimum yields:

$$2\hat{\mathbf{y}} - 2E[\mathbf{y}|\mathbf{x}, \Theta] = 0 \Leftrightarrow \hat{\mathbf{y}}_{MMSE} = E[\mathbf{y}|\mathbf{x}, \Theta] \tag{4.24}$$

A weighting function for each component m is now introduced as:

$$P(m|\mathbf{x}, \Theta) = \frac{c_m g(\mathbf{x}|\boldsymbol{\mu}_m^x, \boldsymbol{\Sigma}_m^{xx})}{\sum_{n=1}^{M} c_n g(\mathbf{x}|\boldsymbol{\mu}_n^x, \boldsymbol{\Sigma}_n^{xx})} \tag{4.25}$$

where g is the Gaussian distribution (4.9). $\boldsymbol{\mu}_m^x$ and $\boldsymbol{\Sigma}_m^{xx}$ are a part of the mean vector and a part of the covariance matrix for the m^{th} component. These arises from a decomposition of $\boldsymbol{\mu}_m^z$ and $\boldsymbol{\Sigma}_m^z$:

$$\boldsymbol{\mu}_m^z = \begin{bmatrix} \boldsymbol{\mu}_m^x \\ \boldsymbol{\mu}_m^y \end{bmatrix} \quad \boldsymbol{\Sigma}_m^z = \begin{bmatrix} \boldsymbol{\Sigma}_m^{xx} & \boldsymbol{\Sigma}_m^{xy} \\ \boldsymbol{\Sigma}_m^{yx} & \boldsymbol{\Sigma}_m^{yy} \end{bmatrix} \tag{4.26}$$

Equation 4.25 is a expression for how probable it is that the observation $\mathbf{x}$ is observed from the m^{th} mixture component. It is assumed that the mixture components in the GMM are independent. This information can be used in the estimator (4.24). Since the distribution of $\mathbf{y}$ is modeled as GMM and the individual mixture components are independent we can write the estimator as a weighted sum:

$$\hat{\mathbf{y}}_{MMSE} = \sum_{m=1}^{M} P(m|\mathbf{x}, \Theta) E[\mathbf{y}|\mathbf{x}, \theta_m] \tag{4.27}$$

The E-operator only operates on a single component. The conditional expectation $E[\mathbf{y}|\mathbf{x}, \theta_m]$ is equal to [33, p. 51]:

$$E[\mathbf{y}|\mathbf{x}, \theta_m] = \left[\boldsymbol{\mu}_m^y + \boldsymbol{\Sigma}_m^{yx}(\boldsymbol{\Sigma}_m^{xx})^{-1}(\mathbf{x} - \boldsymbol{\mu}_m^x)\right] \tag{4.28}$$

Inserting (4.28) in (4.27) leads to the final expression for the estimator:

$$\hat{\mathbf{y}}_{MMSE_F} = \sum_{m=1}^{M} P(m|\mathbf{x}, \Theta) \left[\boldsymbol{\mu}_m^y + \boldsymbol{\Sigma}_m^{yx}(\boldsymbol{\Sigma}_m^{xx})^{-1}(\mathbf{x} - \boldsymbol{\mu}_m^x)\right] \tag{4.29}$$

If the covariance matrix $\boldsymbol{\Sigma}_m^z$ is diagonal, equation 4.29 is reduced, because no cross-variance between $\mathbf{x}$ and $\mathbf{y}$ exist. This result in the reduced equation:

$$\hat{\mathbf{y}}_{MMSE_D} = \sum_{m=1}^{M} P(m|\mathbf{x}, \Theta) \boldsymbol{\mu}_m^y \tag{4.30}$$

i.e. a linear combination of the mean vectors for $\mathbf{y}$. If there is no correlation between $\mathbf{x}$ and $\mathbf{y}$ ($\boldsymbol{\Sigma}_m^{yx}$ is the all zero matrix) the most optimal, in a mean squared error sense, is to choose the mean vector as estimator.

4.2.4 Numerical issues

Rabiner [29] has shown that is necessary to limit the parameter estimates during training of the parameters, to prevent them in becoming to small [29]. It is important to limit the mixture gains and the coefficients on the diagonal of the covariance matrix to be greater or equal to some minimum value. The lower limits on the covariance matrix ensures that the matrix can be inverted and that the determinant is larger than zero. The inversion is necessary for performing the E-step (4.13). If the mixture gains are not limited it is possible for the values calculated in the E-step to yield zero for certain mixture gains. This will cause the updates in the M-step for these mixture components to become unreliable. Rabiner uses a lower limit of 10^{-4} for both the mixture gains and the diagonal elements of the covariance matrix [29]. We will use the same lower limits. The lower limit on the mixture gains is implemented in the M-step as:

$$c_m^{l+1} = \max\left[\frac{1}{T}\sum_{t=1}^{T} P(m|\mathbf{z}_t, \Theta^l), c_{LL}\right] \tag{4.31}$$

In order to keep the constrain on the mixture gains they are normalized after introducing the lower limit c_{LL}.

The lower limit for the covariance matrix could also be included by flooring, but another approach is tried out. The lower limit for the diagonal elements of the covariance matrix is implemented by adding an identity matrix scaled by the lower limit. Thereby it is achieved, that a difference between the entries, with variances below the lower limit, is kept. For example if the two lowest entries on the diagonal is $0.5 \cdot 10^{-4}$ and $0.7 \cdot 10^{-4}$, after addition these variances will become $1.5 \cdot 10^{-4}$ and $1.7 \cdot 10^{-4}$, thereby the entries are still different, on the contrary to flooring.

4.2.5 Discussion

The limiting of the parameters for the covariance matrix was done by adding a scaled identity matrix. It was argued that this would keep differences between two variances, which were below the lower limit. This addition also results in, the two variances becoming more similar and just above the lower limit. For example variances of $1 \cdot 10^{-4}$ and $2 \cdot 10^{-4}$ become $2 \cdot 10^{-4}$ and $3 \cdot 10^{-4}$. The method of adding has drawbacks and a flooring method might have been better.

The convergence criteria for the EM-algorithm was defined to be the absolute change in log-likelihood. This change should be below the threshold ϵ for convergence. The criteria was unfortunately implemented differently. The implemented likelihood was calculated as:

$$L_{imp}(\Theta|\mathcal{Z}) = \sum_{t=1}^{T}\left[\sum_{m=1}^{M} c_m \cdot g(\mathbf{z}_t|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)\right]$$

This is equation 4.12 for calculating the log-likelihood, but without log applied. It can be seen that it is a sum over likelihoods for each observation and not a product as it should have been. Even though this is the likelihood calculated it only influences the convergence criteria and not the other parts of the EM-algorithm.

This likelihood is used to calculate the convergence criteria, which is implemented as:

$$\frac{L_{imp}(\Theta^l|\mathcal{Z}) - L_{imp}(\Theta^{l-1}|\mathcal{Z})}{L_{imp}(\Theta^l|\mathcal{Z})} < \epsilon$$

The algorithm stops when the inequality is true. The numerator is not guaranteed to increase for each iteration. However the EM-algorithm performed on average 12 iterations for the trained models with this convergence criteria. Figure 4.8 depicts the result of the of $L_{imp}(\Theta|\mathcal{Z})$ as a function of iterations for the training of the GMM with 64 mixture components. It is seen that the L_{imp} increases for each iteration. This was the case for all the conducted trainings except for the one with 256 mixture components. Even though the convergence criteria is not implemented correctly it is found, in the verification (chapter 6), that the implemented EM-algorithm gives a very reasonable estimate of the pdf for the underlying data. It is therefore expected, that as long a sufficient amount of iterations are conducted, the convergence criteria is less critical.

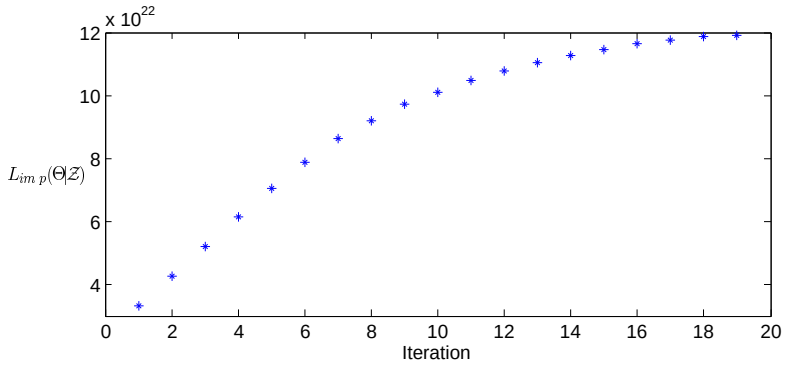


Figure 4.8: The result of implemented likelihood as a function of iterations. From the training of the GMM with 64 mixture components.

4.3 HMM-based method

The hidden Markov model is a strong statistical tool used to describe some process, e.g. real world signals, measurements or some other process producing an observable output. By applying an HMM to a process, the process can be parameterized. The observations can now be assumed to come from such a process with a given set of parameters. This is very useful, as synthetic observations can be generated from this parameterized version of the process. Furthermore by doing simulations, a better understanding of the underlying model, can be obtained. The most important thing is that signal models can often be used to solve complex problems such as recognition, estimation, identification etc. in a very efficient way.

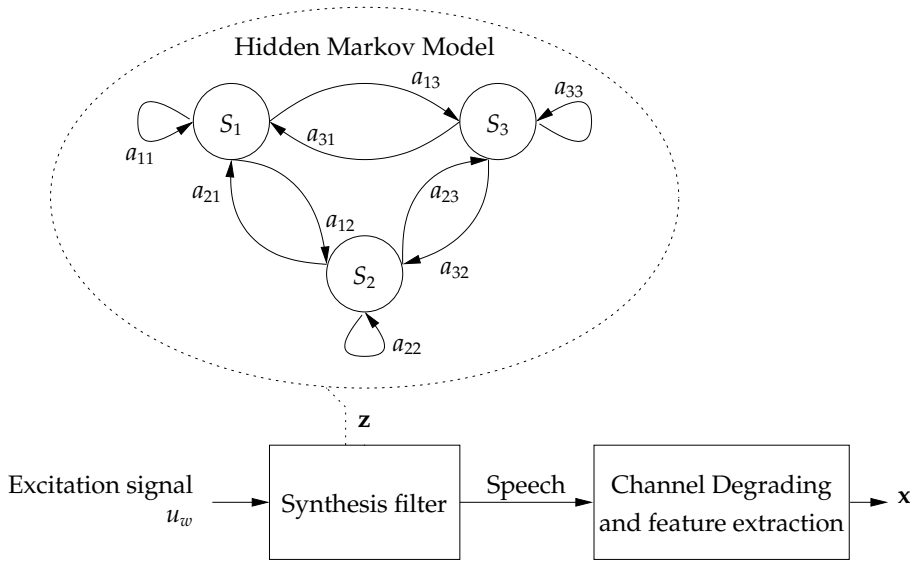


Figure 4.9: Model of speech production utilizing an HMM. Wideband excitation u_w is fed into a synthesis filter. This filter varies over time and changes as defined by an HMM. The result of the filtering is speech. This speech is transmitted over a telephone network, degrading it. After transmission only the feature **x** can be observed. Only a 3 state ergodic HMM is plotted in this example.

Figure 4.9 shows a state diagram of an HMM. Let us focus on only the HMM part. In this example there are 3 states denoted S_1, S_2 and S_3. Each of these states have an associated probability mass (or density) function, which defines the probability of

emitting an observation. This probability is only dependent of the current state. Over time the system changes from one state to another (or itself) defined by a set of probabilities associated with the state. That is at time t the system can be in one state and another at time $t + 1$. The actual state at time t is denoted q_t. It is assumed that the probability of being in state S_i at time t is only dependant of the previous state, that is

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots) = P(q_t = S_j | q_{t-1} = S_i) \tag{4.32}$$

which is the Markov assumption [30]. Furthermore we assume the system to be independent of time, which means that the transition probabilities can be written as:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{4.33}$$

where a_{ij} is the probability of making a transition from state S_i to state S_j.
Compared to both a codebook and a GMM based approach, the properties of an HMM can be used to describe a time varying process. Speech production is such a process, where the position of the organs of the speech production system has a great impact on the speech being produced. Even though speech changes over time, only small variations occur from frame to frame. Figure 4.10 shows how the envelope derived from an LPC changes over time. Because a frame based approach is taken where we assume that

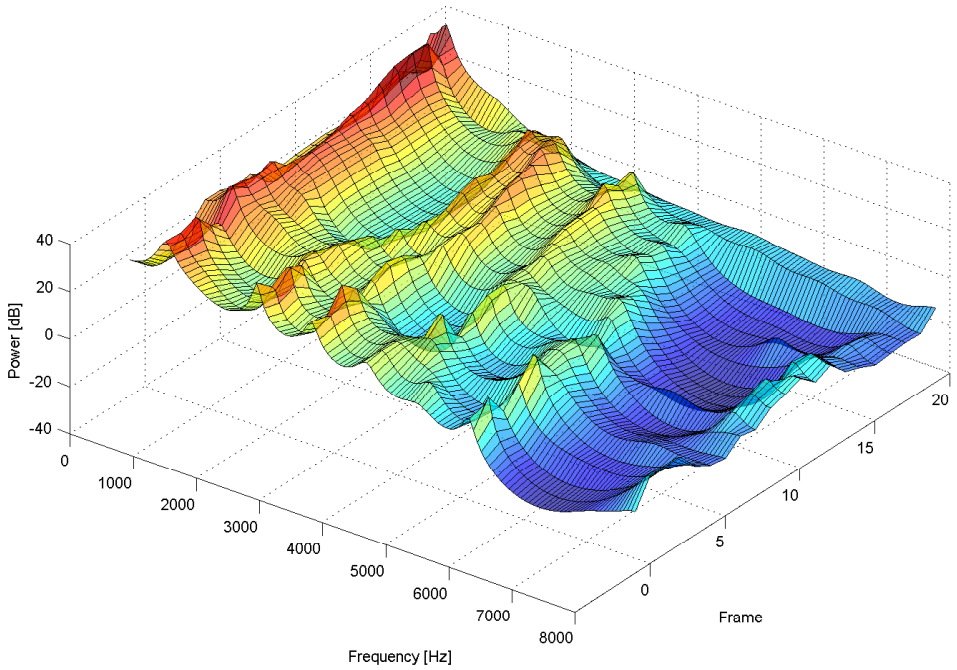


Figure 4.10: Spectral plot of how envelopes changes over time. In this figure the speech is divided into frames of 20 ms using 50% overlap. It can be seen how these envelopes changes gradually over time. This is the process which is modeled using an HMM

speech is stationary within intervals of 20 ms, this change is only gradually between frames. By exploiting this knowledge, speech production can be conceived as coming from a process described by an HMM. By decomposing the speech by the source-filter

model it can be represented by an excitation signal and a synthesis filter. Figure 4.9 shows how an HMM is used to describe the change in envelope of a speech signal over time. The wideband excitation signal u_w is driving a synthesis filter, resulting in wideband speech being produced. The synthesis filter is modeled by an HMM from which features describing the envelope/synthesis filter can be observed. What can be observed is $\mathbf{z}$. The underlying states from which $\mathbf{z}$ is generated, with some probability, can however not be observed. They are said to be hidden. The hidden states S_j can e.g. be seen as the physics from which the observed envelope is generated. As time evolves, the state changes with a certain probability a_{ij}. That is the probability of entering state S_j, given that the previous state was S_i. This change of state results in a change of envelope, used in the speech production. By introducing these states and their associated probabilities, time is now being included into the model. The system used to model the change of envelope is fully ergodic, i.e. it is possible to go to any state from any other state.

Assuming speech can be modeled this way, the only thing we actually observe is the feature vector $\mathbf{x}$ after transmission over the telephone channel. That is, the features describing characteristics of the narrowband speech signal. An HMM model, described by the parameter set λ, can be trained using wideband speech, or more exactly; from the wideband speech, the narrowband feature vector $\mathbf{x}$ and the corresponding extensionband feature vector $\mathbf{y}$ are extracted and they will be used during the training of the model. From the current observation $\mathbf{x}_t$ and the previous observations $\mathbf{x}_1^{t-1}$ the probability for being in S_j at time t can be calculated for all j. This probability for being in state S_j is utilized to obtain a MMSE estimate of $\mathbf{y}_t$. We will use $\mathbf{x}_{t_1}^{t_2}$ to denote the sequence of observations from time t_1 to t_2, where t_2 is greater than t_1. If t_2 is unspecified, it denotes one observation at time t_1. The same notation is also used for $\mathbf{z}$ and $\mathbf{y}$.

4.3.1 Parameters of the HMM

In the previous section the overall idea to estimate $\mathbf{y}$ using an HMM has been presented. In this section the preliminaries such as parameters and notation will be defined.

In the following this notation is used:

- N: Number of states.
- M: Number of mixture components in each state.
- T: Number of observations.

The complete set of model parameters for an HMM is denoted λ and it consist of:

$$\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\Theta}\} \tag{4.34}$$

A state transition probability matrix $\mathbf{A}$ is now defined as:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & \dots & a_{1N} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \dots & a_{ij} & \vdots \\ a_{N1} & \dots & \dots & a_{NN} \end{bmatrix} \tag{4.35}$$

where a_{ij} is the state transition probability defined in (4.33). $\boldsymbol{\pi} = [\pi_1 \dots \pi_N]^{\mathrm{T}}$ is a vector denoting the initial state distribution, i.e. $\pi_j = P(q_1 = S_j)$.

The distribution of observations from each state is modeled as a GMM. $\mathbf{\Theta}$ denotes the parameters for the components of these GMMs, and they are organized as:

$$\mathbf{\Theta} = \begin{bmatrix} \Theta_{11} & \dots & \dots & \Theta_{1M} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & \dots & \Theta_{jm} & \vdots \\ \Theta_{N1} & \dots & \dots & \Theta_{NM} \end{bmatrix} \tag{4.36}$$

where Θ_{jm} is the parameters for component m and state S_j:

$$\Theta_{jm} = \{c_{jm}, \boldsymbol{\mu}_{jm}, \mathbf{\Sigma}_{jm}\} \tag{4.37}$$

4.3.2 Baum-Welch Algorithm

No analytical expressions exists for finding the set of parameters λ of an HMM, such that the available observations denoted $\mathcal{Z} = \{\mathbf{z_1 z_2} \dots \mathbf{z_T}\}$, "fit" the model the best. Therefore these parameters need to be estimated. That is, the problem we want to solve to maximize the likelihood for the parameter set:

$$\lambda = \underset{\lambda}{\operatorname{argmax}} \left[L(\lambda|\mathcal{Z})\right] \tag{4.38}$$

This can be done by training the model using $\mathcal{Z}$. To solve this problem, the Baum-Welch algorithm, also known as the forward-backward algorithm, can be utilized [10]. This algorithm is an extension of the EM-algorithm, and it also insure an increase in likelihood for each iteration. The procedure for estimating the parameters of an HMM using the Baum-Welch algorithm is shown in figure 4.11.

In the following each of the boxes will be described in a section corresponding to the title of the box in figure 4.11. First step is to initialize the algorithm by the initial state distribution $\boldsymbol{\pi}$, the transition probabilities $\mathbf{A}$ and the GMM parameters $\mathbf{\Theta}$ used to calculate emission likelihoods. This can e.g. be done by clustering. Iteration is continued until convergence is achieved.

Emission Likelihoods

The first part of the E-step is to calculate the likelihood that the observation $\mathbf{z}_t$ was generated by the m^{th} mixture component in the j^{th} state. Recalling section 4.2, this can be written as:

$$b_{jm}(\mathbf{z}_t|\mathbf{\Theta}_{jm}) = c_{jm} g(\mathbf{z}_t|\boldsymbol{\mu}_{jm}, \mathbf{\Sigma}_{jm}) \tag{4.39}$$

Where c_{jm} is the mixture coefficient for the m^{th} mixture component in the j^{th} state and $g(\cdot)$ is a Gaussian distribution. $\boldsymbol{\mu}_{jm}$ is the mean value of the m^{th} component in the j^{th} state and $\mathbf{\Sigma}_{jm}$ is the covariance matrix for the m^{th} component in the j^{th} state. Equation 4.39 is calculated for all T observations. For each t, $N \times M$ values are obtained. If it were to be stored it could be done using a three-dimensional matrix. By summing over m, the likelihood that observation $\mathbf{z}_t$ was generated in the j^{th} state is obtained. This can be calculated by equation 4.40:

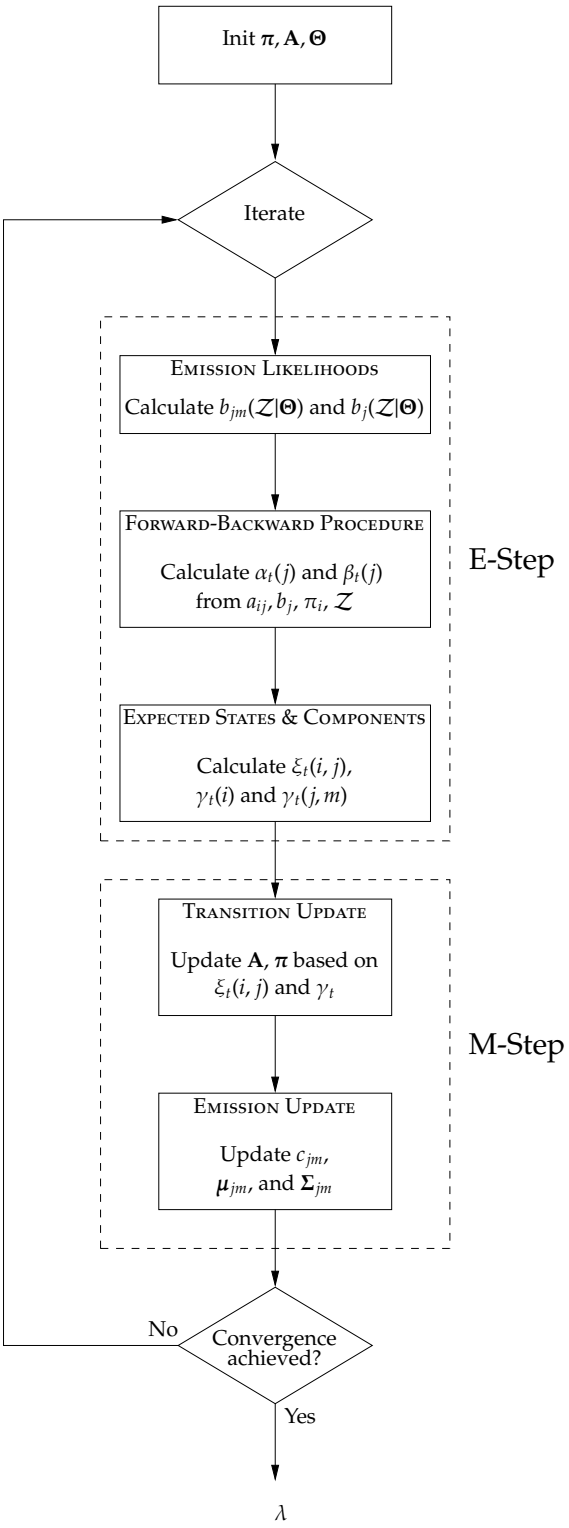


Figure 4.11: Flowchart of the Baum-Welch algorithm.

$$b_j(\mathbf{z}_t|\mathbf{\Theta}) = \sum_{m=1}^{M} b_{jm}(\mathbf{z}_t|\mathbf{\Theta}_{jm}) = \sum_{m=1}^{M} c_{jm} g(\mathbf{z}_t, \boldsymbol{\mu}_{jm}, \mathbf{\Sigma}_{jm}), \quad 1 \leq j \leq N \tag{4.40}$$

Having the emission likelihoods for each state, both the expected times the HMM was in state S_i and that it made a transition to state S_j is now calculated. This can be done by calculating the likelihood that $\mathcal{Z}$ is observed/generated given the model λ. A brute force approach can be taken, by finding every possible sequence, i.e for each sequence calculate the likelihood that the observation $\mathcal{Z}$ was generated by this particular sequence. Sum all of these possible sequences to obtain $p(\mathcal{Z}|\lambda)$. This however can be very computationally intractable, therefore another approach utilizing the Markov assumption is taken [30]. This procedure is called the forward-backward procedure.

Forward -Backward Procedure

One variable is defined as the forward variable:

$$\alpha_t(i) = p(\mathbf{z}_1^t, q_t = S_i|\lambda) \tag{4.41}$$

That is the likelihood of the partial observation $\mathbf{z}_1^t$ being generated and being in state S_i at time t. Letting $t = T$ and sum alpha for all states will give the likelihood, i.e:

$$\sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} p(\mathbf{z}_1^T, q_T = S_i|\lambda) = p(\mathcal{Z}|\lambda) \tag{4.42}$$

The forward variable $\alpha_t(i)$ can be calculated inductively as shown in figure 4.12.

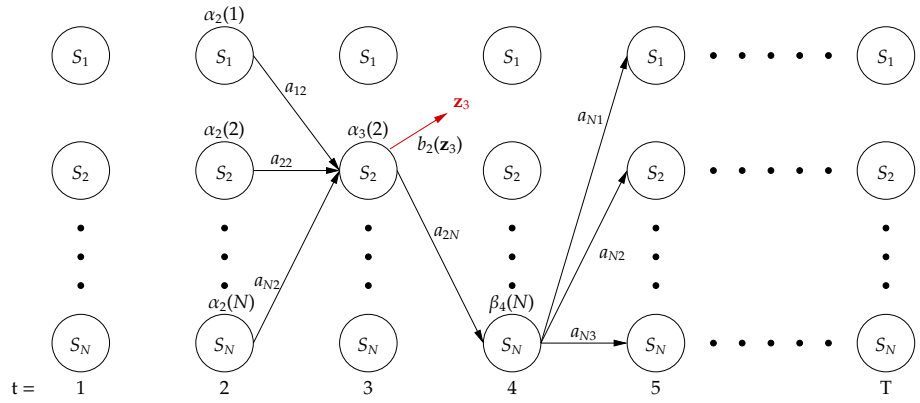


Figure 4.12: Forward-backward procedure. The left part of the figure is showing how to calculate the likelihood of the partial observation $\mathbf{z}_1^t$ and being in state S_i at time t given the model λ. This is the forward procedure. The right part of the figure shows how to calculate the likelihood of the "last" part of the observation sequence $\mathbf{z}_{t+1}^T$ given the state S_i at time t and the model λ.

Let us say we wanted to calculate the likelihood $p(\mathbf{z}_1^3, q_3 = S_2|\lambda)$, i.e. that the HMM was in state S_2 at time $t = 3$ and generated the partial observation $\mathbf{z}_1^3$ given the model λ. This can be found by calculating $\alpha_3(2)$. Because of the Markov assumption, $\alpha_2(j)$ is available for all j. Calculating the term $\alpha_2(1)a_{12} + \alpha_2(2)a_{22} \ldots \alpha_2(N)a_{N2}$ and multiply the likelihood that, state S_2 emitted $\mathbf{z}_3$, gives $\alpha_3(2)$. A general formula can be derived as:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{z}_{t+1}), \quad 1 \leq t \leq T-1 \tag{4.43}$$

As can be seen the procedure has to be initialized. This is done by multiplying the initial state probability for each state by the likelihood that state S_i emitted the first observation $\mathbf{z}_1$. That is:

$$\alpha_1(i) = \pi_i b_i(\mathbf{z}_1), \tag{4.44}$$

In the same way a backward variable $\beta_t(i) = p(\mathbf{z}_{t+1}^T | q_t = S_i, \lambda)$ is defined. Here it is the likelihood of the "last" part of the observation sequence from $t+1$ to T given the state S_i at time t and the model λ. Only the transition probability and not the observation likelihood is used in this calculation for time t, therefore state S_i is given compared to calculation of $\alpha_t(j)$. Calculation of $\beta_4(N)$ is similar to calculation of the forward variable. This calculation can also be explained by figure 4.12 and is calculated as:

$$\beta_4(N) = a_{N1} b_1(\mathbf{z}_5)\beta_5(1) + a_{N2} b_2(\mathbf{z}_5)\beta_5(2) + \ldots + a_{NN} b_N(\mathbf{z}_5)\beta_5(N)$$

Hence the general formula for calculating $\beta_t(i)$ can be derived as follows:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{z}_{t+1})\beta_{t+1}(j), \quad t = T-1, \ldots, 1 \tag{4.45}$$

Initialization is a bit different to $\alpha_t(j)$. It is defined as follows:

$$\beta_T(i) = 1 \tag{4.46}$$

The initialization has been defined arbitrarily by [30, p.263]. As no observation, and no other knowledge is available this value seems reasonable.

Expected States and Components

The forward and backward variables $\alpha_t(i)$ and $\beta_{t+1}(j)$ can now be used to calculate the probability $\xi_t(i,j)$ of being in state S_i at time t and state S_j at time $t+1$, given the model λ and the complete observation sequence $\mathcal{Z}$. If we turn to the definitions and figure 4.12, intuitively this probability can be calculated from $\alpha_t(i)$ and $\beta_{t+1}(j)$, the transition probability from state S_i to S_j (a_{ij}) and the emission likelihood for $\mathbf{z}_{t+1}$. Suppose we want to calculate $\xi_3(2,N)$. This is done by multiplying $\alpha_3(2)$, $\beta_4(N)$, the transition probability a_{2N} and the likelihood that state S_N generated observation $\mathbf{z}_4$. This is divided by the total likelihood of the model λ, which can be calculated from $\alpha_t(i)$ and $\beta_{t+1}(j)$. Formulating it in a general way we get:

$$\begin{aligned} \xi_t(i,j) &= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{z}_{t+1})\beta_{t+1}(j)}{p(\mathcal{Z}|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(\mathbf{z}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{z}_{t+1})\beta_{t+1}(j)} \end{aligned} \tag{4.47}$$

If we sum $\xi_t(i, j)$ over t, we get a quantity which can be interpreted as the expected number of times state S_j was visited from state S_i, i.e the number of transitions from S_i. This is calculated as:

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{expected number of transitions from } S_i \text{ to } S_j \tag{4.48}$$

This is used when updating the transition probabilities a_{ij}. To be able to do this update, the number of total transitions from state S_i is also required. For this purpose $\gamma_t(i)$ is defined as:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \tag{4.49}$$

Inserting (4.47) into (4.49) yields:

$$\begin{aligned} \gamma_t(i) &= \frac{\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{z}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{z}_{t+1}) \beta_{t+1}(j)} \\ &= \frac{\alpha_t(i) \sum_{j=1}^{N} a_{ij} b_j(\mathbf{z}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_t(i) \sum_{j=1}^{N} a_{ij} b_j(\mathbf{z}_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{4.50}$$

Recalling the definition of $\beta_{t+1}(j)$ we can rewrite as:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{4.51}$$

Which can be seen as the probability of being in state S_i at time t. Now if we sum over t as in equation 4.52, a quantity yielding the number of expected transitions from S_i is obtained.

$$\sum_{t=1}^{T} \gamma_t(i) = \text{expected number of transitions from } S_i \tag{4.52}$$

Last part of the E-step is now to calculate the probability of being in state S_j at time t with the m^{th} mixture component generating $\mathbf{z}_t$. To calculate this, equation 4.50 and equation 4.39 can be combined into[1]:

$$\gamma_t(j, m) = \gamma_t(j) \frac{c_{jm} g(\mathbf{z}_t | \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})}{\sum_{m=1}^{M} c_{jm} g(\mathbf{z}_t | \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm})} = \gamma_t(j) \frac{b_{jm}(\mathbf{z_t} | \boldsymbol{\Theta})}{b_j(\mathbf{z_t} | \boldsymbol{\Theta})} \tag{4.53}$$

[1]Here it should be noted a change of notation. j is not the state considered to be going to. It has the same meaning as i on the left side of equation 4.49. In the case when γ_t is a "function" of only one variable as in (4.49) any arbitrary variable denoting state can therefore be used.

The last term on the right hand side is simply the probability of the t^{th} observation coming from the m^{th} mixture component in state S_j. If only a single mixture component were present, this term would reduce to 1 and equation 4.53 would simply reduce to $\gamma_t(j)$. This completes the E-step. The parameters can now be maximized using the values obtained in the E-step.

Transition Update

In the maximization step two jobs are at hand, one is to update the transition probabilities a_{ij} and the initial state distribution π_i, another is to update the parameters Θ of the GMMs for each state. First a_{ij} are updated. Having the expected number of transitions from S_i to S_j, and the expected number of transitions from S_i from equation 4.48 and equation 4.49 respectively, the elements of the transition probability matrix can now be calculated as:

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.54}$$

Because the sum goes over t, we get the expected number of transitions from S_i to S_j. If this is divided by the expected number of transitions from S_i, we get the transition probabilities of going from state S_i to S_j.
Remember that $\gamma_t(i)$ is the probability of being in state S_i at time t. We can simply update $\boldsymbol{\pi}$ by letting:

$$\pi_i = \gamma_1(i) \tag{4.55}$$

Emission Update

Update of the parameters defining the emission likelihoods is almost similar to those in section 4.2 for the GMM, except that states now have to be included. First the mixture weights are updated as:

$$c_{jm} = \frac{\sum_{t=1}^{T-1} \gamma_t(j,m)}{\sum_{t=1}^{T-1} \sum_{m=1}^{M} \gamma_t(j,m)} \tag{4.56}$$

Which is the ratio between the expected number of times the system is in state S_j utilizing the m^{th} mixture component and the expected total number of times the system is in state S_j. Updating the mean values $\boldsymbol{\mu}_{jm}$ is done by weighting each observation by the probability of the m^{th} component in state S_j generated the observation $\mathbf{z}_t$

$$\boldsymbol{\mu}_{jm} = \frac{\sum_{t=1}^{T-1} \gamma_t(j,m)\mathbf{z}_t}{\sum_{t=1}^{T-1} \gamma_t(j,m)} \tag{4.57}$$

The same explanation can be given for updating $\boldsymbol{\Sigma}_{jm}$:

$$\boldsymbol{\Sigma}_{jm} = \frac{\sum_{t=1}^{T-1} \gamma_t(j,m)(\mathbf{z}_t - \boldsymbol{\mu}_{jm})(\mathbf{z}_t - \boldsymbol{\mu}_{jm})^{\mathrm{T}}}{\sum_{t=1}^{T-1} \gamma_t(j,m)} \tag{4.58}$$

Here it should be noted that the $\boldsymbol{\mu}_{jm}$ used is from the previous iteration[2].

Termination

Termination can e.g. be done by calculating the ratio of the current likelihood and the previous likelihood and check if it is below a threshold ϵ. As we will see in section 4.3.5 scaling of likelihood is necessary and the log-likelihood is introduced instead. The ratio between likelihoods for two iterations is therefore calculated as a difference in log-likelihood. This difference is used as a distance measure between different models [30]. We believe that this distance therefore is suitable as the convergence criteria:

$$\frac{1}{T}(\log[L(\lambda^l|\mathcal{Z})] - \log[L(\lambda^{l-1}|\mathcal{Z})]) < \epsilon \tag{4.59}$$

This term can be seen as the absolute increase in log-likelihood from iteration $l-1$ to l. A scaling of T is included such that the term is the average increase in log-likelihood per observation. The scaling only have the affect that ϵ is different. The threshold ϵ has been set to 0.01, which was found to be a suitable value.

4.3.3 Initialization

It was assumed that the set of parameters λ was initialized in the Baum-Welch algorithm. The initialization is important for the Baum-Welch to end up in a "good" local or global maximum of the log-likelihood function. A procedure for obtaining good initial estimates has previous been presented by Rabiner[30], and will be adopted in this thesis.

The method is described by figure 4.13. The first step makes an initial guess of the transition probabilities, unlike the initial parameters for the Baum-Welch algorithm these can be chosen randomly for the initialization. In our case as a uniform distribution. The observations are clustered into N clusters and labeled such that each observation belong to a cluster. This division of observations can be thought of as dividing observations into states.

In the block Emission Update the parameters for the m^{th} mixture component in the j^{th} state is updated by clustering the observations labeled j into M clusters. The observations from the m^{th} component and j^{th} state are then utilized in the update of the GMM parameters for this particular mixture component and state. This update is similar to the one conducted for the GMM initialization in section 4.2.2 - equation 4.19, 4.20 and 4.21.

All the parameters in Θ, now have an initial value, which makes it possible to calculate the emission likelihoods $b_j(\mathcal{Z}|\Theta)$. From the emission likelihoods, the log-likelihood is obtained from the forward procedure and the convergence criteria is checked.

[2]Because of the variables which need to be stored in the algorithm, it requires a considerable amount of memory for large T. Implementation has been done such that iteration goes over t and the variables are summed continuously. Therefore we are restricted to use the $\boldsymbol{\mu}_{jm}$. If available, $\boldsymbol{\mu}_{jm}$ could advantageously be used to update $\boldsymbol{\Sigma}_{jm}$. This would most probably result in a faster convergence.

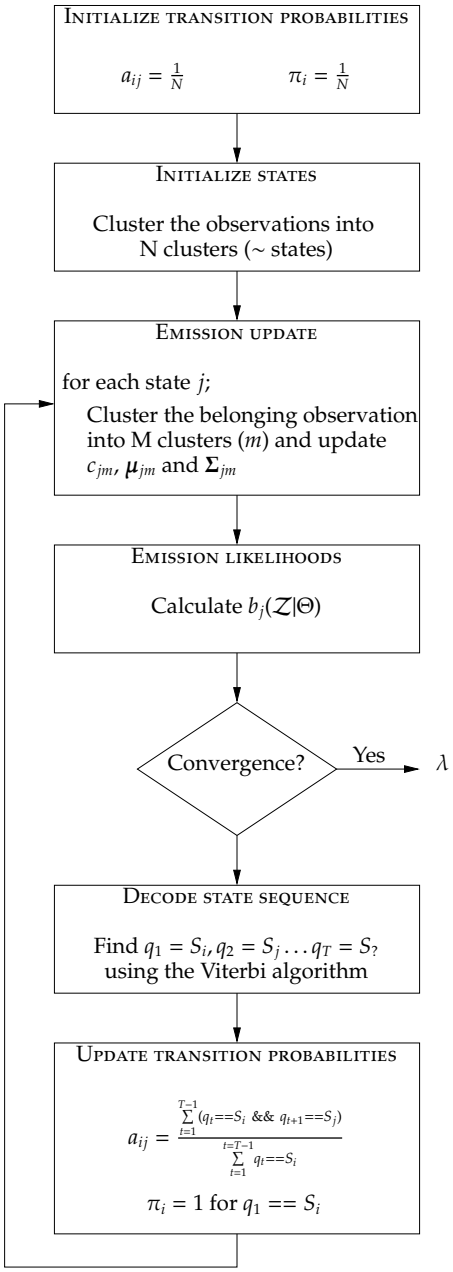


Figure 4.13: Flowchart for the initialization of the parameters for HMM. For the updating of a_{ij}, $(q_t == S_i)$ && $(q_t == S_j)$ yields 1 if a state transition from S_i to S_j occurred, otherwise 0.

The most likely state sequence is calculated to update the transition probabilities. The state sequence can be calculated efficiently using the Viterbi algorithm. The Viterbi algorithm finds the most optimal state sequence given the observations and model λ. The Viterbi algorithm is similar to the forward procedure except that it eliminates path-sequences along its way from the first observation to the last. When it reach the last observation at time T backtracking is performed, which result in the optimal sequence: $Q = [q_1 = S_i, q_2 = S_j, \ldots q_T = S_?]$. The obtained state sequence is used to label the observations with a state label (instead of the clustering procedure). The Viterbi algorithm is addressed shortly here and the reader is referred to [10, 30] for more information.

The update of the transition probabilities (the last block at the figure) is based on the sequence of states. That is, a_{ij} is updated as the number of transitions from state S_i to state S_j divided by the number of times the state sequence is decoded as S_i. The initial distribution $\boldsymbol{\pi}$ is set to one for the entry, which corresponds to the first decoded state. The initializing method iterates until the increase in log-likelihood is below a certain threshold. The same criteria as in equation 4.59 is used with an ϵ equal to 0.1.

The method mentioned above is the first part of the initialization. The complete initialization is performed in two steps:

Step 1: The initialization procedure in figure 4.13 is performed with 20% of the available data.

Step 2: The Baum-Welch algorithm runs for two iterations with the same data as in step 1, then the next 20% and so on, i.e. a similar approach as for the GMM initialization, described in section 4.2.2.

After ending step 2 the set of parameters λ is obtained and utilized in the Baum-Welch algorithm with all the available data.

4.3.4 Estimator

The available information for estimation is an HMM model, the current $\mathbf{x}_t$ and previous observations $\mathbf{x}_1^{t-1}$. The HMM model is described by its parameters: $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\Theta}\}$. From this information, an estimate of the extensionband feature $\mathbf{y}_t$ at time t is obtained. Estimation is carried out for each frame, i.e. for each t.

The estimator is chosen such that it minimizes the mean squared error between the estimated extension-band feature $\hat{\mathbf{y}}_t$ and the real extensionband feature $\mathbf{y}_t$. The Minimum Mean Squared Error (MMSE) estimator can then be written as:

$$\hat{\mathbf{y}}_t^* = \underset{\hat{\mathbf{y}}_t}{\operatorname{argmin}}\, E[(\hat{\mathbf{y}}_t - \mathbf{y}_t)^{\mathrm{T}}(\hat{\mathbf{y}}_t - \mathbf{y}_t)|\mathbf{x}_1^t, \lambda] \tag{4.60}$$

The equation is similar to the estimator derived in section 4.2.3 except for the time-index and the different model parameters. The solution for equation 4.60 is (see section 4.2.3):

$$\hat{\mathbf{y}}_t^* = E[\mathbf{y}_t|\mathbf{x}_1^t, \lambda] \tag{4.61}$$

This equation can be split into a weighted sum of estimates for each state weighted with the probability of being in this state at time t, $P(q_t = S_j|\mathbf{x}_1^t, \lambda)$:

$$\hat{\mathbf{y}}_t^* = \sum_{j=1}^{N} P(q_t = S_j|\mathbf{x}_1^t, \lambda) E[\mathbf{y}_t|\mathbf{x}_t, q_t = S_j, \lambda] \tag{4.62}$$

It should be noted that the given data in the expectation is reduced to only the current observation $\mathbf{x}_t$ instead of all the previous observations $\mathbf{x}_1^t$. This is due to the fact that the observation probability, for an HMM, only is dependent on the current state.

The Gaussian mixture components within states are independent of each other. Therefore we can write the expectation term in equation 4.62 as a linear combination of estimates from each component, weighted by the probability of this mixture component for a given state, i.e:

$$\hat{\mathbf{y}}_t^* = \sum_{j=1}^{N} P(q_t = S_j|\mathbf{x}_1^t, \lambda) \sum_{m=1}^{M} P(m|\mathbf{x}_t, \Theta_{jm}, q_t = S_j) \cdot E[\mathbf{y}_t|\mathbf{x}_t, q_t = S_j, \Theta_{jm}] \tag{4.63}$$

where $P(m|\mathbf{x}_t, \Theta_{jm}, q_t = S_j)$ is the probability for the m^{th} mixture component. Figure 4.14 is an graphical illustration of the estimator.

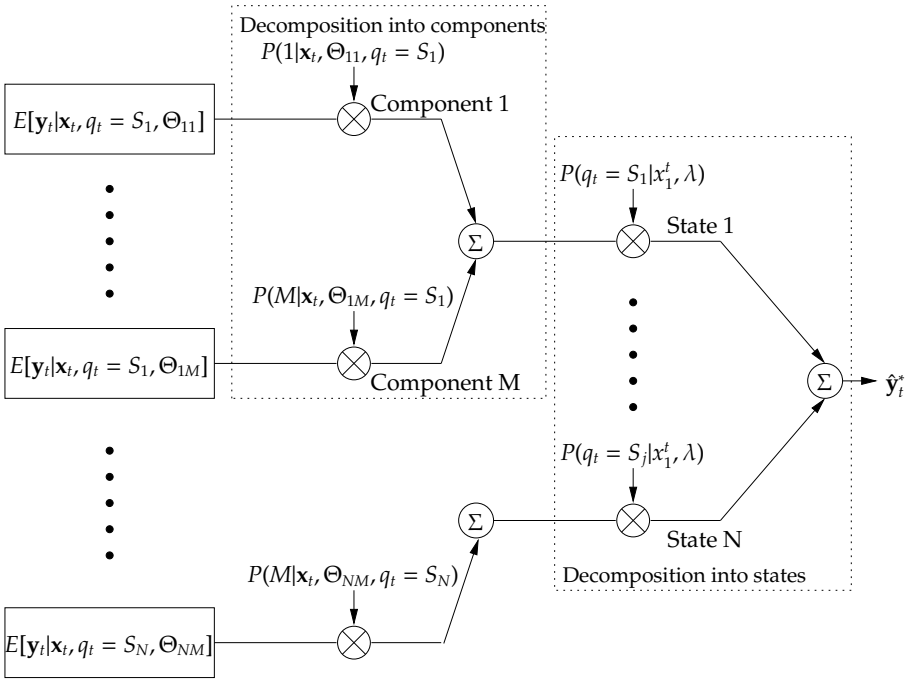


Figure 4.14: Graphical illustration of the MMSE estimator in equation 4.63. The expectation is calculated for each component in all states and weighted with the probability of being in this particular state and using this particular mixture component.

The three terms in (4.63) are now further explained one at the time. The estimator is a soft-decision estimator, i.e. it weights each expected value of $\mathbf{y}_t$ for each state and mixture component.

$P(q_t = S_j|\mathbf{x}_1^t, \lambda)$ is the probability of being in state S_j at time t. This probability can be found by solving the decoding problem for an HMM. The decoding problem can be solved by using e.g. the forward-backward variables or the Viterbi algorithm. The former method

will result in a probability for each state at each time t. The Viterbi algorithm makes a hard-decision and yields the most likely state sequence found from backtracking. For the backtracking to work efficiently it requires observations beyond $\mathbf{x}_t$ in order to estimate $\mathbf{y}_t$. Since the forward procedure yields a probability for each state, which can be used as soft-decision, it is chosen.

Equation 4.51 is applied to calculate the probability for each state. The equation is repeated here for convenience:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

The observations available at time t is $\mathbf{x}_1^t$. Therefore the backward variable is $\beta_t(i) = 1$ for all states, because nothing is known about the next observation $\mathbf{x}_{t+1}$. This future observation could have been used, but it results in additional delay, before the bandwidth extended speech signal can be played. Therefore $\mathbf{x}_{t+1}$ is not utilized and the calculation of $\gamma_t(i)$ reduces to:

$$P(q_t = S_j|\mathbf{x}_1^t, \lambda) = \gamma_t(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} \tag{4.64}$$

where $\alpha_t(j)$ is found from the forward procedure with the observations $\mathbf{x}_1^t$ as:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij}\right] b_j(\mathbf{x}_t)$$

where the initial $\alpha_1(j)$ is:

$$\alpha_1(j) = \pi_j b_j(\mathbf{x}_1)$$

and $b_j(\mathbf{x}_t)$ is the likelihood of observing $\mathbf{x}_t$ in state S_j:

$$b_j(\mathbf{x}_t) = \sum_{m=1}^{M} c_{jm} \cdot g(\mathbf{x}_t|\boldsymbol{\mu}_{jm}^x, \boldsymbol{\Sigma}_{jm}^{xx}) \tag{4.65}$$

The parameters $\boldsymbol{\mu}_{jm}^x$ and $\boldsymbol{\Sigma}_{jm}^{xx}$ are defined similar as in (4.26) with addition of the j^{th} state.

The second term in equation 4.63, $P(m|\mathbf{x}_t, \Theta_{jm}, q_t = S_j)$, is the probability of the mth GMM mixture component. The weighting is found in equation 4.25.

The last term in (4.63), $E[\mathbf{y}_t|\mathbf{x}_t, q_t = S_j, \Theta_{jm}]$, is the expectation of $\mathbf{y}_t$ given the present observation, the state, and the parameters for the m^{th} component of the GMM in the j^{th} state. This term differs depending on the form of the covariance matrix ($\boldsymbol{\Sigma}_{jm}^{zz}$). This is also the case for the GMM approach. From (4.28) and 4.63 we obtain the solution for a full covariance matrix:

$$\hat{\mathbf{y}}_t^F = \sum_{j=1}^{N} P(q_t = S_j|\mathbf{x}_1^t, \lambda) \sum_{m=1}^{M} P(m|\mathbf{x}_t, \Theta_{jm}, q_t = S_j) \left[\boldsymbol{\mu}_{jm}^y + \boldsymbol{\Sigma}_{jm}^{yx}(\boldsymbol{\Sigma}_{jm}^{xx})^{-1}(\mathbf{x} - \boldsymbol{\mu}_{jm}^x)\right] \tag{4.66}$$

The two probabilities which are included are found in (4.64) and (4.25). The index F denotes full covariance matrix. If the covariance matrix is diagonal (D) it reduces to:

$$\hat{\mathbf{y}}_t^D = \sum_{j=1}^{N} P(q_t = S_j | \mathbf{x}_1^t, \lambda) \sum_{m=1}^{M} P(m | \mathbf{x}_t, \Theta_{jm}, q_t = S_j) \boldsymbol{\mu}_{jm}^y \tag{4.67}$$

The estimators in (4.66) and (4.67) are used in the block ESTIMATE EB ENVELOPE in figure 3.4 on page 27.

4.3.5 Parameter flooring and scaling

In order to implement the HMM with success, two practical problems have to be solved. The first problem is parameter flooring of $\boldsymbol{\Sigma}_{jm}$, c_{jm}, $\mathbf{A}$ and $g(\mathbf{z}_t|\Theta_{jm})$. The other problem concerns the ability to represent the forward and backward variables on a computer. This problem is addressed first.

Scaling

Equation 4.43 for calculating the forward variable, causes some representational problems. The values of the transition probabilities a_{ij} are between zero and one. The emission likelihoods b_j takes on values from zero to infinity. The forward variable will therefore be able to take on values between zero and infinity. The variable is updated using induction which results in $\alpha_t(i)$ either decreasing or increasing exponentially towards zero or infinity as t increases. The same problem exist for the backward variable $\beta_t(i)$, when t decreases from T. This decreasing or increasing of α and β makes it difficult to represent them within the dynamic range of the computer.
To deal with this problem a scaling of α and β is introduced. This idea for scaling is found in [30, p. 272]. The scaling factor for the forward variable is defined as:

$$Sc_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)} \tag{4.68}$$

and is used to scale α_t to obtain the scaled version α'_t, which sums to 1 for each time instant t:

$$\alpha'_t(i) = Sc_t \cdot \alpha_t(i) \tag{4.69}$$

The scaling factor Sc_t is also applied to $\beta_t(i)$. Any scaling could have been chosen, but applying the same scaling as for alpha makes the scaling cancel out. The scaling is:

$$\beta'_t(i) = Sc_t \cdot \beta_t(i) \tag{4.70}$$

Inserting the scaled values of α and β into the calculation of the probability $\xi_t(i,j)$ from equation 4.47 yields:

$$\xi'_t(i,j) = \frac{\alpha'_t(i) a_{ij} b_j(\mathbf{z}_{t+1}) \beta'_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha'_t(i) a_{ij} b_j(\mathbf{z}_{t+1}) \beta'_{t+1}(j)} \tag{4.71}$$

Replacing $\alpha'_t(i)$ and $\beta'_{t+1}(j)$ with the right side of (4.69) and (4.70) we see that the scaling factors cancel each other out:

$$\xi'_t(i,j) = \frac{\alpha_t(i)(\prod_{k=1}^{t} Sc_k)a_{ij}b_j(\mathbf{z}_{t+1})\beta_{t+1}(j)(\prod_{l=t+1}^{T} Sc_l)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)(\prod_{k=1}^{t} Sc_k)a_{ij}b_j(\mathbf{z}_{t+1})\beta_{t+1}(j)(\prod_{l=t+1}^{T} Sc_l)} = \xi_t(i,j) \tag{4.72}$$

where $(\prod_{k=1}^{t} Sc_k)$ is the total scaling factor applied to $\alpha_t(i)$ and $(\prod_{l=t+1}^{T} Sc_l)$ is the total scaling factor for β_{t+1}. From 4.71 it is seen that the scaling do not influence the values of $\xi_t(i,j)$, which means that the rest of the Baum-Welch algorithm can be left unchanged, except for the likelihood calculation. The calculation of the likelihood is affected from the scaling. If equation 4.42 is used for calculating the likelihood with the scaled α' it yields:

$$p(\mathcal{Z}|\lambda) = \sum_{i=1}^{N} \alpha'_T(i) = 1 \tag{4.73}$$

It is seen that the scaling of alpha has to be removed in order to calculate the likelihood. Instead of calculating the likelihood, it is replaced by the log-likelihood:

$$\log(p(\mathcal{Z}|\lambda)) = \log\left(\frac{1}{\prod_{t=1}^{T} Sc_t}\sum_{i=1}^{N}\alpha'_T(i)\right) = \log\left(\frac{1}{\prod_{t=1}^{T} Sc_t}\right) = -\sum_{t=1}^{T}\log(Sc_t) \tag{4.74}$$

The log-operator is applied, because the product of the scaling factor increase towards infinity or decrease towards zero depending on the value of α. To represent this product the log-operator is applied, such that a sum of the log of the scaling factors are obtained instead of the product. The scaling factor will always be larger than zero, because lower limits on the emission likelihoods are applied. The lower limits also assures that the scaling factor can not increase to more than the inverse of the lower limit of $b_j(\mathbf{z})$. This is seen by inserting 4.43 into 4.68 and assuming $b_j(\mathbf{z})$ to have the lower limit b_{LL} for all states:

$$Sc_{max} = \frac{1}{\sum_{j=1}^{N}\alpha_t(j)} = \frac{1}{\sum_{j=1}^{N}\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_{LL}} = \frac{1}{\sum_{i=1}^{N}\alpha_t(i)\left[\sum_{j=1}^{N}a_{ij}\right]b_{LL}} = \frac{1}{\sum_{i=1}^{N}\alpha_t(i)b_{LL}} = \frac{1}{b_{LL}} \tag{4.75}$$

Since b_{LL} is chosen to be strictly larger than zero, it is possible to represent $\log(Sc_t)$.

Parameter flooring

The parameter flooring is applied similar for the covariance matrix and the mixture gains as in the GMM implementation (section 4.2.4).

Additional flooring of $\mathbf{A}$ makes the transition from all states to all states possible (ergodic model). It also helps to give a reliable estimate of the parameters [10]. The flooring is performed by adding a lower limit (10^{-4}) to all the entries of $\mathbf{A}$.

A lower limit of $b_j(\mathbf{z})$ has also shown to improve performance [30]. The lower limit on $b_j(\mathbf{z})$ is implemented by putting a lower limit of 10^{-20} on $g(\mathbf{z})$.

4.3.6 Perceptual Training

A lot of applications which have to do with speech, audio etc, has shown to benefit from taking the human auditory system into account. This would most probable also be the case for an application like BWE. Unfortunately no procedure exist which update the parameters of an HMM using a perceptual measure. An obvious candidate for incorporating a perceptual measure could therefore be to include it into the feature vector $\mathbf{z}$.
Recalling the training vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{y} \end{bmatrix}$$

where $\mathbf{x}_{acf}$ denotes the auto correlation coefficients representing the narrowband. $\mathbf{x}_{scl}$ denotes scalar features of the narrowband and $\mathbf{y}$ denotes the cepstral coefficients representing the extensionband. A lot of representations can be utilized to represent the extensionband envelope. One feature which has been used very successfully together with speech recognizers is the MFCC [4]. As described in section 2.2.2 on page 14, the MFCC is a perceptual representation which takes the human auditory system into account. The motivation for choosing the MFCC as feature, is that we would like to group the envelopes/features we perceive as humans to be "close" to each other. Put in a different way, the more perceptually close two envelopes are, the more likely we would it to be, that they come from the same state and same mixture component. It will therefore be expected that the perceived quality of the estimated speech signal is higher using this representation during training.
There is however no direct mapping from MFCC's to AR coefficients. To account for this missing mapping an alternative training procedure is proposed. The idea is to stack the training vector in a different way and then use the MFCC as feature during training. When the training procedure has converged some other features are used to make the final update of the means $\boldsymbol{\mu}_{jm}$ and the covariance matrices $\boldsymbol{\Sigma}_{jm}$. The values of the transition matrix $\mathbf{A}$ and the weighting coefficients c_{jm} remain the same as those found during training of the HMM.

Method 1

In this way, the complete feature vector $\mathbf{z}_{comp}$ is stacked as follows:

$$\mathbf{z}_{comp} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{y}_{mfcc} \\ \mathbf{y} \end{bmatrix}$$

Where $\mathbf{y}_{mfcc}$ denotes the mel frequency cepstral coefficients of the extensionband and $\mathbf{y}$ extensionband features from which it is possible to obtain AR coefficients of this band. The dimension of $\mathbf{y}_{mfcc}$ is eight and obtained from a filterbank with nine filters as described in section 2.2.2. During the first part of the training only

$$\mathbf{z}_{perc} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{y}_{mfcc} \end{bmatrix}$$

is utilized. When the training has converged

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{y} \end{bmatrix}$$

is used to calculate the means and covariance matrices, while keeping the other parameters of the HMM. This update is achieved by doing one last iteration in the Baum-Welch algorithm, using $\mathbf{z}$ as observation vectors.

Method 2

The complete feature vector used in this method is

$$\mathbf{z}_{comp} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{w}_{mfcc} \\ \mathbf{y} \end{bmatrix}$$

where $\mathbf{w}_{mfcc}$ is the MFCC of the whole wideband signal. The dimension of $\mathbf{w}_{mfcc}$ is 13 as studies has shown that 13 MFCC is enough to represent a wideband speech signal. During the first part of the training

$$\mathbf{z}_{perc} = \begin{bmatrix} \mathbf{x}_{scl} \\ \mathbf{w}_{mfcc} \end{bmatrix}$$

is used. After convergence

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{acf} \\ \mathbf{x}_{scl} \\ \mathbf{y} \end{bmatrix}$$

is used in a final iteration to update the means $\boldsymbol{\mu}_{jm}$ and covariance matrices $\boldsymbol{\Sigma}_{jm}$.
Figure 4.15 shows how an HMM would be trained by rearranging the complete vector $\mathbf{z}_{comp}$. First features are extracted to obtain $\mathbf{z}_{comp}$. Then only the features constituting $\mathbf{z}_{perc}$ is chosen when doing the training. After convergence one last iteration is executed where $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ are calculated using $\mathbf{z}$.

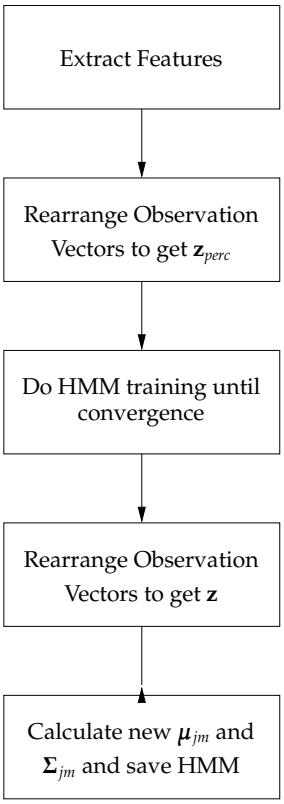


Figure 4.15: Update $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ by rearranging the training vector $\mathbf{z}$.

4.4 Discussion

Three different approaches for estimating the extensionband envelope have been presented. The simplest one taking a codebook approach has been discussed in section 4.1.4. The issues discussed in the codebook discussion could advantageously have been used in initialization of both GMM and HMM.

By using a GMM, any distribution can practically be represented provided a high enough order. Furthermore the GMM estimator is not limited to only estimate mean values. Estimation using a soft decision, can therefore be exploited by GMMs and significantly improves estimation. By introducing the HMM, time is included into estimation too. Informal listening test showed that it was almost impossible to distinguish the estimation method using GMM from the method using an HMM. A minor study of this carried out and it confirmed that the envelopes being estimated, is almost identical. The reason for this could be that a Markov model of order one is used, and the fact that a frame overlap of 50% is applied. The change in envelope from one frame to another might be to small to fully utilize the capabilities of the HMM. If a Markov model of order two, together with frame processing using no overlap, a better result might have been obtained. This hypothesis is supported by studying the transition matrices of the HMMs obtained from training. Figure 4.16 shows a plot of a transition matrix from an HMM with 32 states and 8 mixture components.

It is observed that the values are very high on the anti-diagonal of the matrix. This means that the probability of staying in the same state is very high. This is most probable because there is not much change from one frame to another. It would therefore be very interesting to see how a Markov model of order two would do. Furthermore it should be investigated

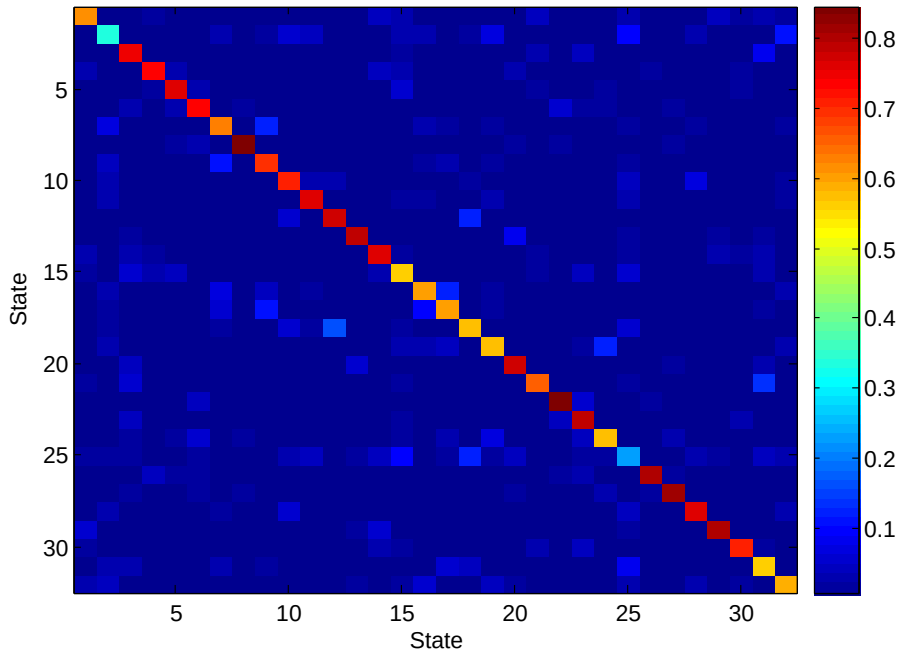


Figure 4.16: Transition matrix for an HMM with 32 states and 8 mixture components. It can be seen that the probabilities of staying in the same state is quite large compared to entering another state.

what impact processing using no overlap will have on the estimation. Because of the lack of ability of an HMM to model state durations [30], an evident improvement of the implementation would be to include state durations. As it is now the probability of going to the same state is $P(q_t = S_i, q_{t+1} = S_i, ...) = P(q_t = S_i) \times P(q_{t+1} = S_i) \times ...$ i.e exponential decreasing. A better model would perhaps be obtained using another distribution defining state durations. Because of time constraints, the ideas discussed in this section have unfortunately not been investigated. We still believe that the HMM does some smoothing in time. Therefore the HMM approach is chosen to extend the speech evaluated in the formal listening test (section 7.2).

Chapter 5

Extension of Excitation

This chapter will concern the extension of the excitation signal. This corresponds to Extension of excitation *in the block diagram for the BWE in figure 3.1 on page 24. Two methods are presented; spectral translation and a new observation based approach. Besides the estimation methods, a bypassing method for obtaining the original wideband excitation is presented. A perceptual comparison of the two methods are conducted.*

The input signal to the Extension of excitation block is the narrowband excitation signal, $u_n(k;m)$, and the output is an estimate of the wideband excitation, $\hat{u}_w(k;m)$. An important property of the block, is the block to be transparent, i.e. no modifications are made in the narrowband frequency region. Besides this property it should be able to give a good estimate of the missing excitation signal bands. A good and sufficient estimate would be an estimate, which results in no perceptual differences between the estimate and the original wideband. Two different methods for estimation will be presented. The first method is in literature known as spectral translation and the second method is developed from observations of the excitation signal.

To evaluate the estimated envelope, which was discussed in the previous chapter, without influence from possible estimation error in the excitation signal, a bypass method is included. This method is not a part of the bandwidth extension algorithm, but is useful for perceptually comparing different methods of the envelope estimation. We will denote this signal as the original wideband excitation signal.

5.1 Original Wideband Excitation

The original wideband excitation signal is used to evaluate the estimated envelopes. The original wideband excitation signal is assumed to be the most optimal "estimate" of the excitation signal. An analysis filter is found from the wideband speech signal and the excitation signal is obtained from this analysis filter. Figure 5.1 depicts the setup for obtaining the original wideband excitation signal and utilizing it in the synthesis filter.

The order of the synthesis filter to model the speech production system is discussed in section 3.2.3 on page 34. The rule of thumb is to set the model order as the sample frequency i kHz plus some additional poles for modeling the glottal and lip radiation characteristics. Based on these considerations the model order is set to 18.

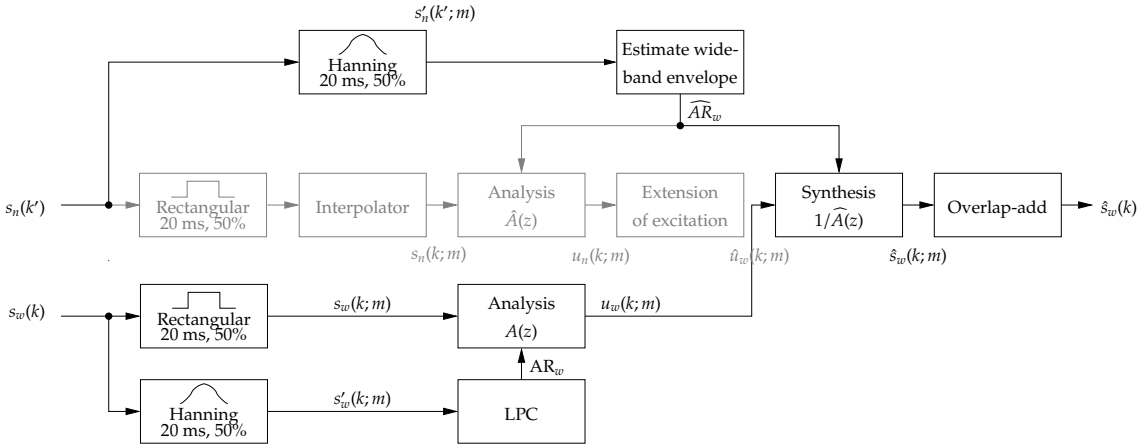


Figure 5.1: Block diagram of the bypassed system for obtaining the original wideband excitation signal. The gray blocks are bypassed. Compare e.g. with the non-bypassed block diagram in figure 3.1 on page 24.

5.2 Spectral Translation

Spectral Translation is one method of extending the excitation signal. The technic has previous been used in bandwidth extension algorithms and is described in literature [22, p.190]. The idea is to "shift" the spectrum of the narrowband excitation into the upper part of the spectrum, and then combine the two. This will result in the spectrum of the excitation being repeated. In the time domain this can be achieved by modulating $u_n(k;m)$ by a modulation frequency Ω_M, and then subsequently do a highpass filtering. Figure 5.2 depicts the principle of spectral translation.

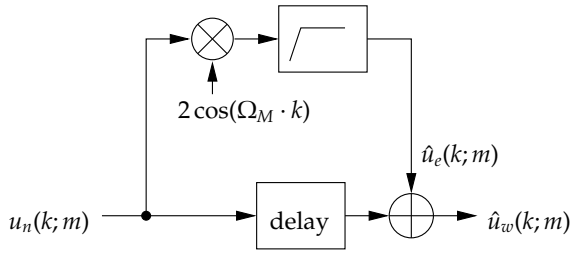


Figure 5.2: Block diagram for spectral translation. The delay is introduced to compensate for the delay in the highpass filter.

The modulation frequency and the cut-off frequency must be chosen to get the desired spectral characteristics. An illustration of spectral translation in the frequency domain can be seen in figure 5.3 and figure 5.4.

This shows how the modulation frequency is found by introducing $u_{n,l}$ and $u_{n,u}$ which are, respectively, the lower and upper cutoff frequency of the bandpass filter making it out for the telephone channel. In this illustration it is assumed that the narrowband signal is limited in frequency from $u_{n,l}$ = 300 Hz to $u_{n,u}$ = 3400 Hz. The estimate for the upper part of the spectrum of the excitation, is then just a highpass filtered version of the resulting signal. The spectrum of the extended signal is shown in figure 5.3. (note that only the upper part is used, that is, the dotted band is filtered out using a highpass filter.)

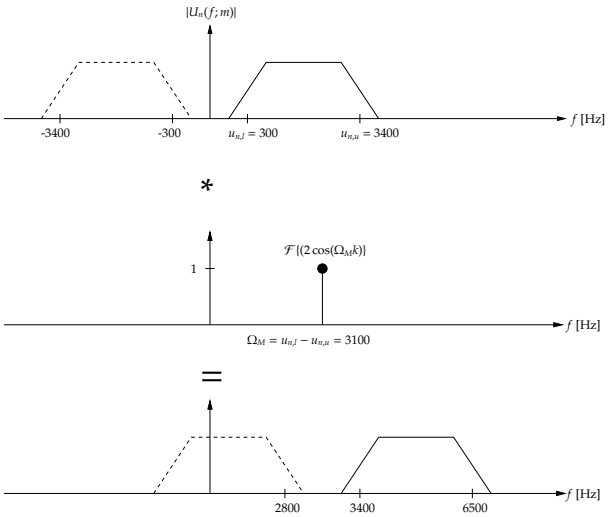


Figure 5.3: A narrowband excitation signal $u_n(k;m)$ being modulated by the frequency $\Omega_M = u_{n,u} - u_{n,l} = 3400 - 300 = 3100$ Hz. In the frequency domain this corresponds to folding marked with an $*$. For illustrative reasons only the positive modulation frequency is plotted.

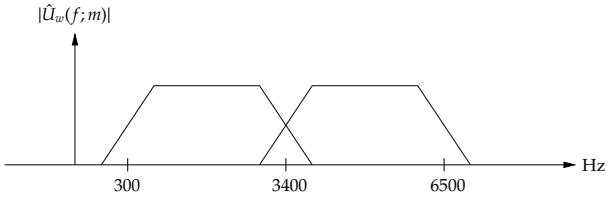


Figure 5.4: The resulting spectrum from adding the excitation signal from the narrowband and the extended signal

The estimate for the wideband excitation is then the sum of the narrowband excitation signal and the extended excitation signal. The resulting spectrum is shown in figure 5.4. This method is a good compromise between complexity and performance [19].

One special case of spectral translation is denoted spectral folding. If the modulation frequency Ω_M is chosen to be the Nyquist frequency $= \pi$ (8 kHz in our case), then a mirrored version of $u_n(k)$ would be obtained in the frequency domain mirrored around the Nyquist frequency. Because of the band limitations on the signals, no filtering would be required [22]. A consequence of using Spectral Folding would be a gap in the middle of the spectrum from 3700-4300 Hz. This is due to the band limitations on the signal, which would create a gap from 3700-4000 Hz and another one from the mirrored version from 4000-4300 Hz.

By informal listening test we have chosen to use Spectral Translation. Furthermore it can be argued that speech contains less energy, the higher the frequency. Hence the most upper part of the spectrum is omitted, instead of omitting frequencies near 3400 Hz as in Spectral Folding.

For the example in this section it is assumed that the narrowband signal is limited in frequency from $u_{n,l} = 300$ Hz to $u_{n,u} = 3400$ Hz. In the final implementation only the upper band is missing, i.e. the signal is limited from $u_{n,l} = 0$ Hz to $u_{n,u} = 3400$ Hz. Therefore the modulation frequency is chosen as:

$$\Omega_M = u_{n,u} - u_{n,l} = 3400 - 0 = 3400 \text{ Hz} \tag{5.1}$$

The cutoff frequency for the highpass filter is also chosen as Ω_M. The estimated wideband excitation signal, $\hat{u}_w(k;m)$, is then limited from 0 Hz to 6800 Hz.

5.3 Observation based approach

The section will describe a new approach for extending the excitation signal. First the motivation for a new solution is given, afterwards a method is developed. Finally results from the new method are presented.

5.3.1 Motivation

The motivation for this approach was found by inspecting the power spectra and spectrograms of the wideband excitation signal. The wideband excitation signal is derived from the output of the analysis filter, with the original wideband signal as input. The coefficients for the analysis filter are found from a LPC.

Figure 5.5 depicts the spectrogram of the wideband excitation signal. From the spectrogram the pitch structure is noticed. Looking at the pitch structure from approximately frame 17 to 26, where the phone "ae" in basketball is pronounced, we observe it to be very dominating and easily identified in the lower part of the spectrum. Above 3000 Hz it seems like the structure is less dominating or not even there. This trend can be found throughout the whole spectrogram. No dominating structure at all is present above approximately 5.5 kHz.

Frame 140 and the neighboring frames have a dominating pitch structure up to 5.5 kHz. Figure 5.6 shows the power spectrum of frame 140. This figure gives a closer look at this particular frame. The spectrum above 5 kHz does not contain as dominating dips as the lower part of the spectrum, i.e. the spectrum is more flat above 5 kHz and looks more like the spectrum of white noise.

Another interesting point is that the dips in the spectrum from 0 Hz to about 1500 Hz is in general lower than the dips in the middle of the spectrum from 1.5 kHz - 5.5 kHz. Frame 48 to 58 in figure 5.5 show this tendency. Above 1.5 kHz for these frames no green colors are present like it is the case below 1.5 kHz. It seems like there is a power floor.

Using spectral translation for extending the excitation signal will cause dominating dips in the power spectrum above 3.4 kHz. Furthermore spectral translation will yield the pitch structure to be repeated above 5.5 kHz. Another disadvantage with conventional spectral translation is the lack of spectral components above the double bandwidth of the signal.

5.3.2 Method

The objective is to produce an excitation signal with the properties that is observed for the wideband excitation signal, i.e no pitch structure above 5.5 kHz and no replication of the spectrum below 1500 Hz. Furthermore the new method should cover the spectrum all the way to 8 kHz.

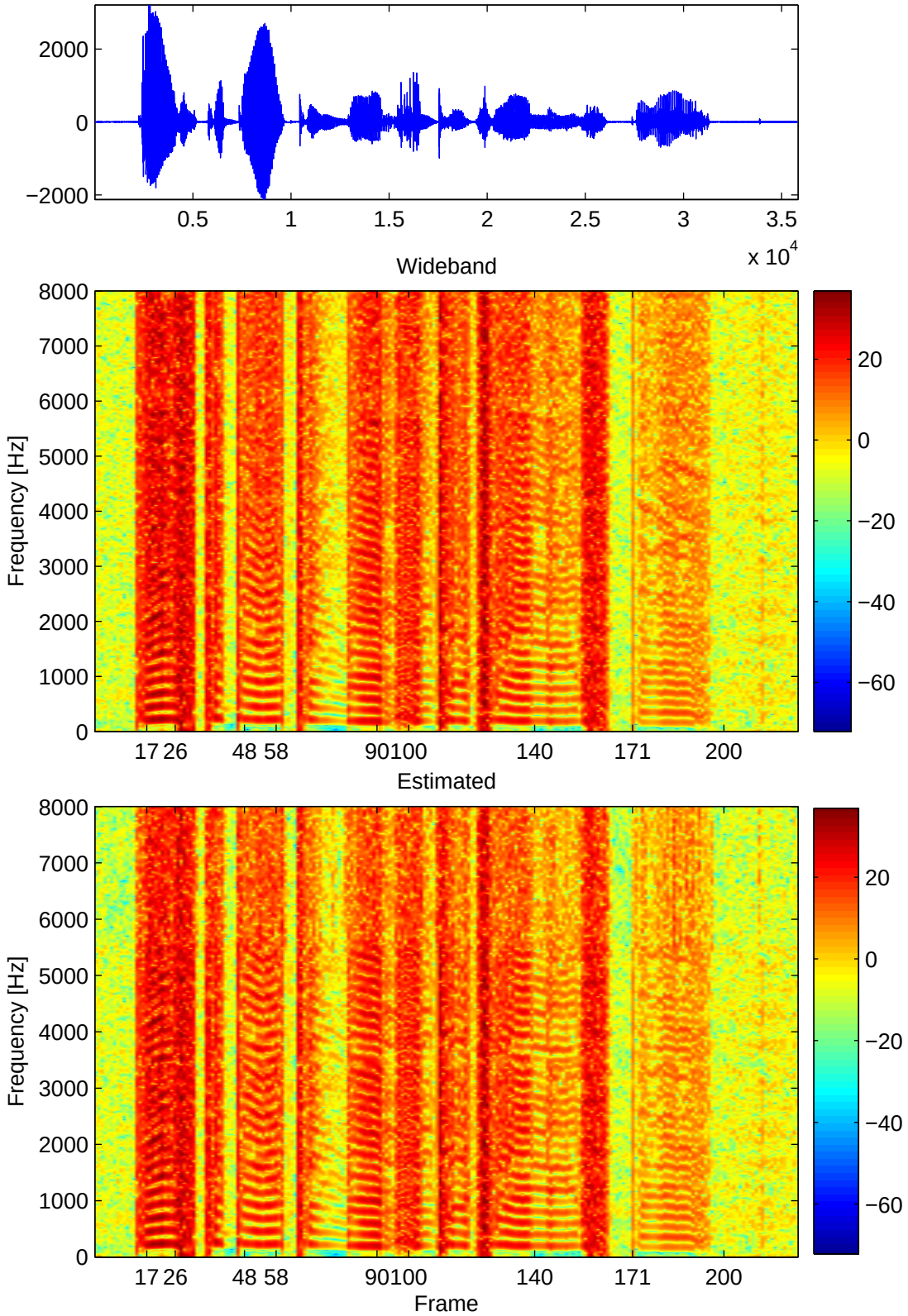


Figure 5.5: The speech signal is from the recording: DR2/FPAS0/SX44.WAV. A woman pronouncing:: "Basketball can be an entertaining sport". Time plot of speech signal plotted together with spectrograms of the original wideband and the estimated excitation respectively. The LPC order used is 18. A Hanning window is applied before the 2048 points DFT. Some specified frames are labeled, as they are referred to in the text.

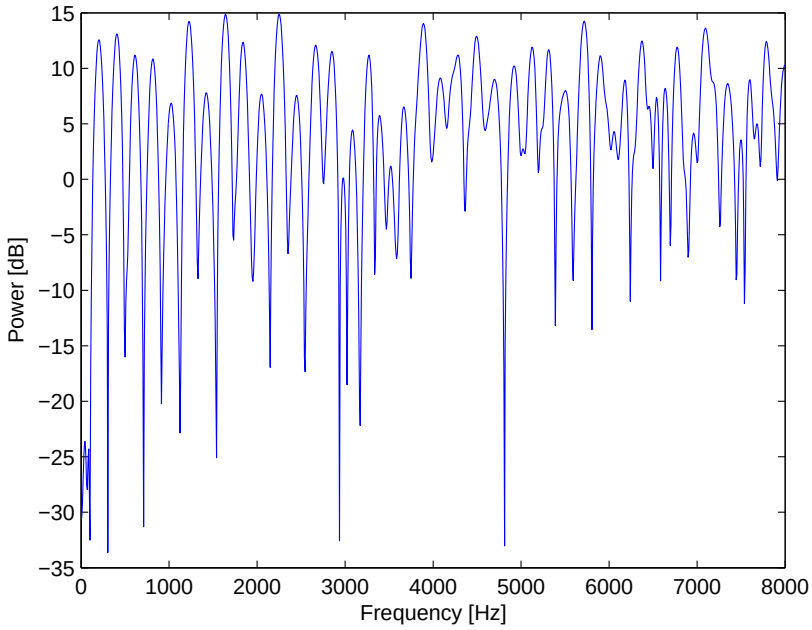


Figure 5.6: The power spectrum of frame 140 for the wideband excitation signal.

To avoid replicating the pitch structure above 5.5 kHz, as would be the case for conventional spectral translation, a lowpass filter can be applied to the signal. This will on the other hand still yield big dips in the power spectrum above 3.4 kHz, which is not observed.

Instead of the lowpass filter an adjustment of the modulation frequency (Ω_M) is possible. The adjustment should result in energy in the translated spectrum up to only 5.5 kHz. We denote this frequency as $u_{e,max}$ - the maximum frequency in the extensionband. The new modulation frequency can be found as:

$$\Omega_M = u_{e,max} - (u_{n,u} - u_{n,l}) \tag{5.2}$$

Inserting $u_{n,u} = 3.4$ kHz, $u_{n,l} = 0$ kHz and $u_{e,max} = 5.5$ kHz yield a modulation frequency of: $\Omega_M = 2.1$ kHz. The spectral translation is still followed by a highpass filter with a cutoff frequency of 3.4 kHz, such that the lower part of the spectrum from the translation is not repeated. That is the spectrum below $(u_{n,u} - u_{n,l}) - \Omega_M = 1.3$ kHz is not used for producing the final spectrum.

The extended excitation signal contains energy up to 5.5 kHz. In section 5.3.1 it was noticed that the wideband excitation signal had the same characteristics as white noise above this frequency. Therefore a noise generator is applied, followed by a highpass filter with a cutoff frequency of 5.5 kHz.

If too much noise is added it will make the extended signal sound noisy. On the other hand if too little noise is added it will not give the wanted wideband effect. It is therefore necessary to be able to control the variance of the noise.

The variance can be calculated from the narrowband excitation signal as:

$$\sigma^2_{u_n} = \frac{1}{N_k}\sum_{k=1}^{N_k}(u_n(k) - \mu)^2 \tag{5.3}$$

where N_k is number samples per frame and μ is the mean value.

The noise generator is implemented by drawing samples from a Gaussian distribution with zero mean and unit variance. The output of the noise generator is scaled by σ_{u_n}. The result is that the extensionband on average has the same spectral height as the narrowband. A block diagram of the final method can be seen in figure 5.7.

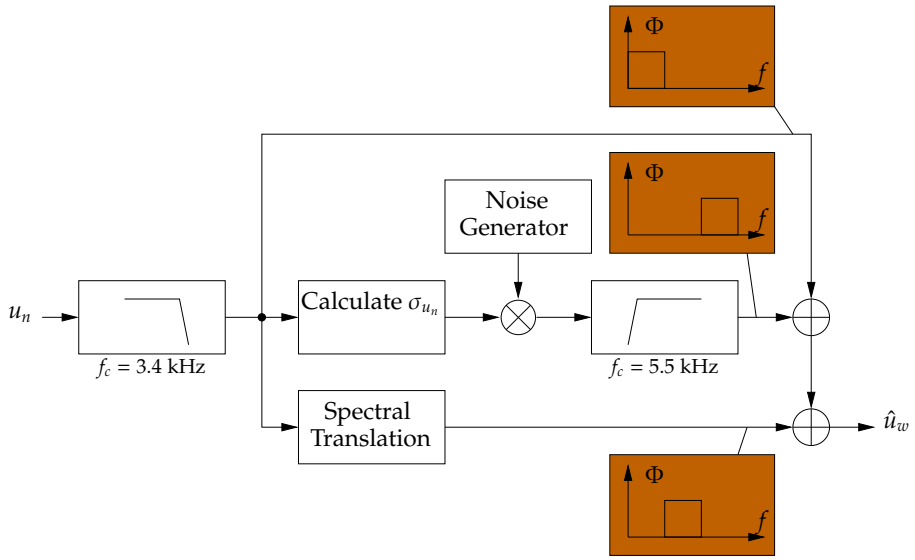


Figure 5.7: The block diagram of the new approach. The brown boxes illustrate which part of the spectrum the signal contribute to. The narrowband excitation signal has been upsampled. During this upsampling some aliasing has been introduced. When filtering the slightly aliased signal through the analysis filter this aliasing is amplified. In order to avoid overlapping in the extensionband an additional lowpass filter is included after the analysis filter.

5.3.3 Results

The result is presented as a spectrogram in figure 5.5. There are no gaps in the spectrogram as was the case for spectral translation. Differences between the original wideband excitation and the estimated can be seen in figure 5.8.

From this plot it is possible to see that spectra from spectral translation contains wrong pitch information. This is especially noticed around frame 90. The blue and yellow horizontal lines indicate that the translated spectrum was not modulated as a multiple of the pitch. It is also noticed from the constant color in narrowband, that the spectrum in narrowband is not modified. Another difference is noticed at frame 171. Here the phone "p" in "sport" is pronounced. An estimation error in the variance results in a vertical line being visible.

5.4 Discussion

In this section two methods for estimating the excitation signal have been presented. The former method is known in literature as spectral translation. The second method was developed from observations of the wideband excitation signal.

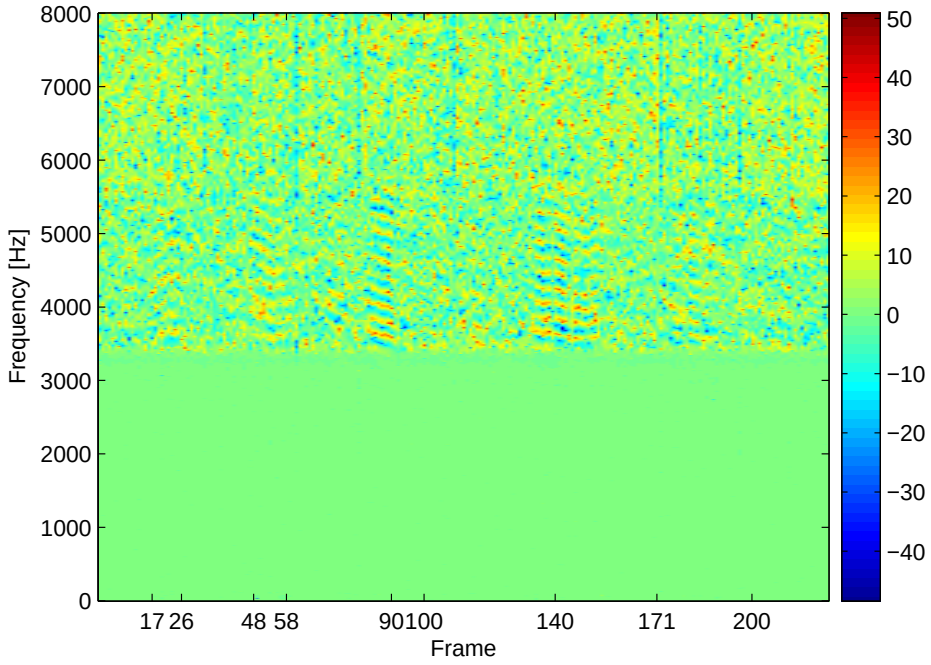


Figure 5.8: Spectrogram of the original excitation subtracted the estimated excitation signal.

The new approach has been compared with spectral translation in an informal listening test. To avoid errors in the shape of the envelope, the analysis filter was found from an LP analysis of the original wideband signal. Therefore only errors in the estimation of the excitation signal contributes to differences between the original wideband and the estimated signal. The informal listening test consisted of 10 randomly chosen speech files from the TIMIT test core. The excitation signal was extended using the two methods. The two versions was compared by the members of the group. In 3 out of 10 files spectral translation was the best, in 6 out of 10 the observation based method was preferred. For one file it could not be distinguished which version was the best.

The primary reason that spectral translation was disliked was due to "ringing" effects being introduced. The effect arises from misaligned harmonics and occurs only during voiced sounds. Even though the observation based method also contain misaligned pitch structure these artifacts was not as noticeable as for spectral translation. For two of the files, the new approach gave a more wideband feeling. This is likely the result of adding noise in the upper part of the spectrum. This addition on the other hand introduced some noisy characteristics in the speech signal, which were a bit unpleasant in some utterances. This noise sound occurred in bursts and sounded a bit like cicadas. For those version where severe ringing occurred the observation based method was preferred.

It is very individual how these artifacts are perceived by different listeners. Therefore it is difficult to state that the observation based approach is better than spectral translation.

In spite of these artifacts, the two methods gave a perceptual much better result than the narrowband speech signal.

Chapter 6

Verification of Algorithms

The algorithms presented, are both numerical methods which maximizes the log-likelihood function and the implementation is therefore difficult to verify, because they do not guarantee ending up in a global maximum of this cost function. Synthetic data is used to verify that the parameters estimated from the algorithms, comply with the distribution of the synthetic data.

Before the actual training using extracted speech features, the implemented EM and Baum-Welch algorithms are tested with synthetic data. This is done for a small scale experiment, such that visual verification is possible. The experiment is conducted to give a verification of the training procedures for GMMs and HMMs. The verification of the codebook implementation is not included in this section, because it has already been verified in section 4.1. The verification is described in two sections starting with GMM.

6.1 Expectation-Maximization

The generation of synthetic data is addressed first. The idea is to use synthetic data in the EM algorithm to find the parameters for the distribution of these data. The log-likelihood, the distribution and the parameters for the estimated model and the generating model are then compared to verify correct implementation.

The synthetic data is generated by drawing samples from a GMM. To be able to inspect the result visually, the observation vectors are restricted to a dimension of two. The two-dimensional vector $\mathbf{o}$ is drawn as:

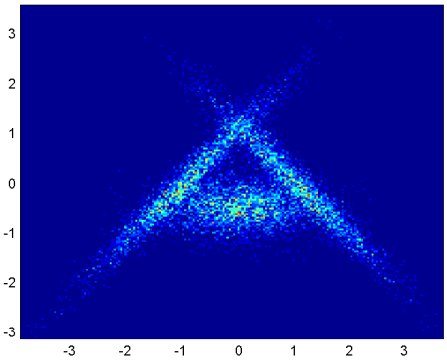
$$\mathbf{o} \sim \sum_{m=1}^{M} c_m g(\mathbf{o}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{6.1}$$

It is chosen to use three mixture components, i.e. $M = 3$, to ease comparison. The parameters for the three components are set as specified in table 6.1. The parameters are denoted Θ^g, to indicate that they were used to generate the observations.

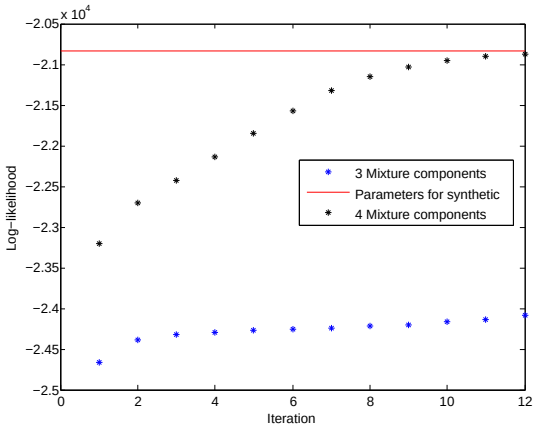
This distribution incorporates overlapping densities and cross-covariance within component two and three. It is therefore verified that the implemented EM-algorithm (including initialization) is able to model covariance and position of the individual mixture components in a suitable way.

The number of observations are set to T=10,000, i.e. $O = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{10,000}\}$. Drawing (6.1) of data is implemented by drawing $c_m \cdot T$ samples from each component. The function *randn* in Matlab is used to draw independent samples from a multivariate Gaussian dis-

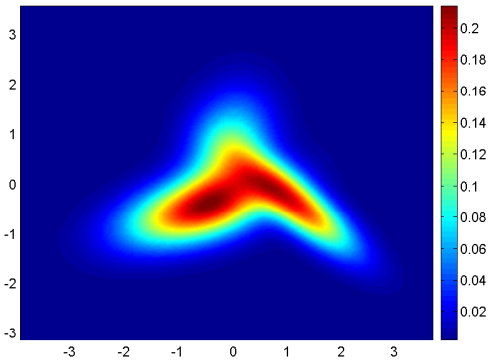
tribution with zero mean and unit variance. The samples are shaped with the covariance matrix (by using a Cholesky factorization) and added the mean value for this specific mixture component. When the samples are drawn for each component, the actual mean value and covariance matrix are calculated for this mixture component. These actual values are used to calculate the log-likelihood for the model: $\log(L(\Theta|O))$. This log-likelihood should be considered as the upper bound for the log-likelihood. A histogram of the generated data is shown in figure 6.1(a).



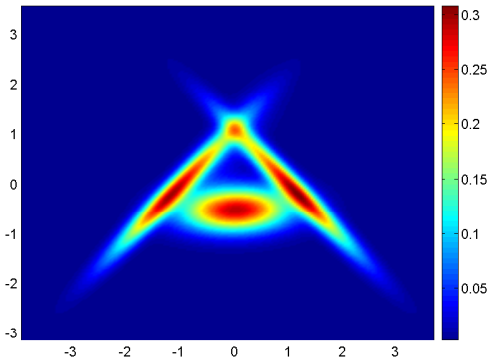
(a) Histogram of synthetic data



(b) Log-likelihood as function of iteration. The red line is the log-likelihood $\log(L(\Theta|O))$. It should be noted that the first ten iterations are a part of the initialization.



(c) Pdf of trained GMM with 3 mixture components



(d) Pdf of trained GMM with 4 mixture components

Figure 6.1: The result for estimating the parameters for the synthetic data.

The estimation of parameters are carried out for models with three and four mixture components. Figure 6.1(c) depicts the estimated pdf for a GMM with 3 components. It is seen that the densities do not follow the shape of the histogram completely. Estimation of model parameters must have ended up in a local maximum of the log-likelihood function. Increasing the number of components results in the pdf plotted in figure 6.1(d). This model follows the histogram better than with three mixture components. The log-likelihood for the generating model, $\log(L(\Theta|O))$, is depicted in figure 6.1(b). It is compared to the log-likelihood for the two estimated models. It is seen from figure 6.1(b) that the log-likelihood is increasing fr each iteration. The log-likelihood for the model with 3 mixture components do not reach the log-likelihood of the generating model,

which is also indicated by the plot of the pdf. The model with four components gets very close to the optimal log-likelihood. It is concluded that it ends up very close to the global maximum of the log-likelihood function. The estimated parameters are given in table 6.2 for three components and table 6.3 for four components. The parameters for the model with three components are difficult to compare to the parameters which generated the distribution (table 6.1), because the mean value and covariance matrix are different. But comparing the parameters for the model with 4 components and the generating parameter shows that these parameters are much alike. The mean and covariance for the first 3 components are very similar. The fourth mixture component is given a little mixture weight c_4 and the influence it has on the complete distribution is therefore quite limited.

m	1	2	3
c_m	0.4	0.3	0.3
$\boldsymbol{\mu}_m$	$\begin{bmatrix} 0 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 0 \end{bmatrix}$
$\boldsymbol{\Sigma}_m$	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.7 & -0.75 \\ -0.75 & 0.85 \end{bmatrix}$	$\begin{bmatrix} 0.7 & 0.75 \\ 0.75 & 0.85 \end{bmatrix}$

Table 6.1: Parameters for the synthetic data for the GMM verification.

m	1	2	3
c_m	0.47	0.24	0.29
$\boldsymbol{\mu}_m$	$\begin{bmatrix} -0.67 \\ -0.50 \end{bmatrix}$	$\begin{bmatrix} -0.08 \\ 0.65 \end{bmatrix}$	$\begin{bmatrix} 1.2 \\ -0.44 \end{bmatrix}$
$\boldsymbol{\Sigma}_m$	$\begin{bmatrix} 0.82 & 0.21 \\ 0.21 & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.27 & 0.07 \\ 0.07 & 0.59 \end{bmatrix}$	$\begin{bmatrix} 0.48 & -0.35 \\ -0.35 & 0.40 \end{bmatrix}$

Table 6.2: Estimated parameters for a GMM with 3 components.

m	1	2	3	4
c_m	0.39	0.29	0.27	0.06
$\boldsymbol{\mu}_m$	$\begin{bmatrix} 0.02 \\ -0.52 \end{bmatrix}$	$\begin{bmatrix} 1.05 \\ -0.06 \end{bmatrix}$	$\begin{bmatrix} -1.14 \\ -0.15 \end{bmatrix}$	$\begin{bmatrix} -0.07 \\ 0.83 \end{bmatrix}$
$\boldsymbol{\Sigma}_m$	$\begin{bmatrix} 0.48 & 0.01 \\ 0.01 & 0.09 \end{bmatrix}$	$\begin{bmatrix} 0.66 & -0.71 \\ -0.71 & 0.81 \end{bmatrix}$	$\begin{bmatrix} 0.60 & 0.65 \\ 0.65 & 0.74 \end{bmatrix}$	$\begin{bmatrix} 0.30 & 0.32 \\ 0.32 & 0.51 \end{bmatrix}$

Table 6.3: Estimated parameters for a GMM with 4 components.

It is therefore concluded that the implemented EM-algorithm increases the log-likelihood

for each iteration and it seems to fit the distribution well with the histogram of the data.

6.2 Baum-Welch

Verification of the Baum-Welch implementation takes same approach as for the EM algorithm. The idea is to generate synthetic data and then estimate HMM parameters which generated these data. These estimated parameters are then used to generate a new set of data. The distribution of the two sets of data are compared. The distance between the two models is also calculated and should be as low as possible.

The generation of synthetic data is different compared to the GMM, because the data should be observations from a Markov chain. The data is generated by the following procedure:

1. Choose the initial state S_i from the distribution of $\boldsymbol{\pi}$.
2. Draw a sample ($\mathbf{o}_1$) from the emission distribution in state S_i
3. Chose the next state for time $t+1$ by drawing a number from the discrete distribution of going from state S_i to S_j.
4. Draw a sample ($\mathbf{o}_{t+1}$) from the emission distribution in state S_j
5. If $t == T$; stop else goto 3.

The samples are drawn from a GMM, i.e. similar to the drawing in the verification of the EM algorithm. The difference compared to the GMM is the drawing of states. 10,000 samples are drawn from an HMM. The number of states is set to three. The implementation of the EM algorithm is verified in section 6.1, therefore it is sufficient to use one component per state. The transition matrix and the initial distribution is set to:

$$\mathbf{A}_g = \begin{bmatrix} 0.10 & 0.10 & 0.80 \\ 0.45 & 0.50 & 0.05 \\ 0.05 & 0.15 & 0.80 \end{bmatrix} \quad \boldsymbol{\pi}_g = \begin{bmatrix} 0.33 \\ 0.33 \\ 0.33 \end{bmatrix} \tag{6.2}$$

The parameters for the GMMs are listed in table 6.4. The parameter for generating the synthetic data is denoted λ^g. A histogram of the generated distribution is shown in figure 6.2(a).

The results of the estimation of parameters are now described. The Baum-Welch (and initialization) results in the following parameters for the HMM [1]:

$$\mathbf{A}_{est} = \begin{bmatrix} 0.11 & 0.12 & 0.77 \\ 0.45 & 0.51 & 0.04 \\ 0.05 & 0.15 & 0.80 \end{bmatrix} \quad \boldsymbol{\pi}_{est} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{6.3}$$

If the estimated parameters are compared to the generating parameters (6.2) it is seen that the transition probabilities matrix is quite similar.

[1] The indexing of the states was different after running the Baum-Welch algorithm, but it has been changed to ease comparison of the parameters which generated the data.

The estimated parameters for the GMMs for each state are found in table 6.5. As for the transition matrix these parameters correspond well to those of the generating model (table 6.4). This result is supported by the histogram of a set of data generated by an HMM with the estimated parameters. The histogram is shown in figure 6.2(b). The histogram is very similar to the histogram of the data generated using the parameter λ^g (figure 6.2(a)). Figure 6.2(c) depicts the increase in log-likelihood during training. It is seen that it almost reaches the log-likelihood of the generating model ($\log(L(\lambda^g|O))$).

State	1	2	3
c_{jm}	1	1	1
μ_{jm}	$\begin{bmatrix} 0 \\ -1.5 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 0 \end{bmatrix}$
Σ_{jm}	$\begin{bmatrix} 0.5 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.7 & -0.75 \\ -0.75 & 0.85 \end{bmatrix}$	$\begin{bmatrix} 0.7 & 0.75 \\ 0.75 & 0.85 \end{bmatrix}$

Table 6.4: GMM parameters for the synthetic data for the HMM verification. Note that there are only one Gaussian per state (k=1).

State	1	2	3
c_{jm}	1	1	1
μ_{jm}	$\begin{bmatrix} 0.00 \\ -1.49 \end{bmatrix}$	$\begin{bmatrix} 1.00 \\ 0.00 \end{bmatrix}$	$\begin{bmatrix} -1.99 \\ 0.01 \end{bmatrix}$
Σ_{jm}	$\begin{bmatrix} 0.52 & 0.00 \\ 0.00 & 0.10 \end{bmatrix}$	$\begin{bmatrix} 0.66 & -0.71 \\ -0.71 & 0.81 \end{bmatrix}$	$\begin{bmatrix} 0.69 & 0.74 \\ 0.74 & 0.84 \end{bmatrix}$

Table 6.5: Estimated GMM parameters for the HMM verification.

The distance between the two models (or actual dissimilarity) can be calculated by this equation [30]:

$$D_s(\lambda^1, \lambda^2) = \frac{D(\lambda^1, \lambda^2) + D(\lambda^2, \lambda^1)}{2} \tag{6.4}$$

where $D(\lambda^1, \lambda^2)$ is defined as:

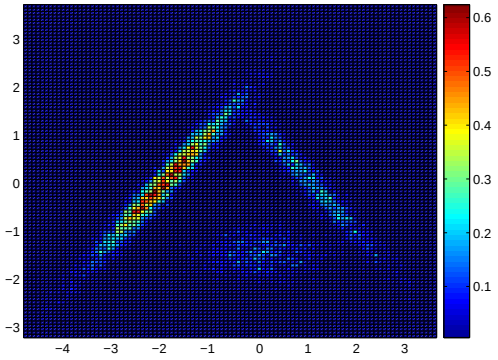
$$D(\lambda^1, \lambda^2) = \frac{1}{T}[\log(L(\lambda^1|O^2)) - \log(L(\lambda^2|O^2))] \tag{6.5}$$

O^2 is a sequence of observations generated by the model λ^2 and O^1 is generated by λ^1. The distance measure between λ^g and λ^{est} yields:

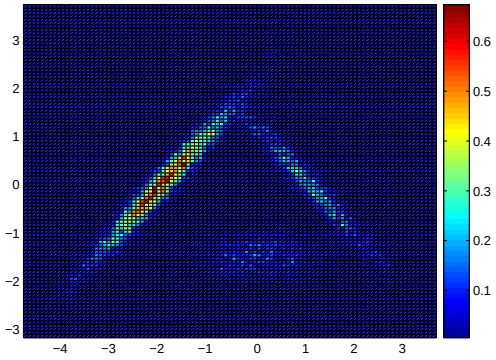
$$D_s(\lambda^g, \lambda^{est}) = \frac{D(\lambda^g, \lambda^{est}) + D(\lambda^{est}, \lambda^g)}{2} = \frac{-3.94 \cdot 10^{-4} + (-6.14 \cdot 10^{-4})}{2} = -5.05 \cdot 10^{-4}$$

The more negative the distance, the higher dissimilarity. The closer to zero the measure is, the more alike they are. $D_s(\lambda^g, \lambda^{est})$ of the size 10^{-4} tell us that the models are very much

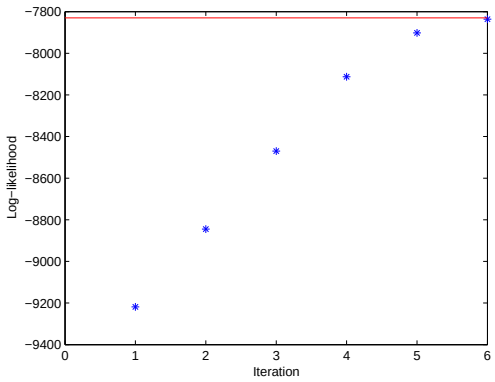
alike. Therefore it is concluded that the implementation of the Baum-Welch algorithm assures an increase in log-likelihood and that the generating model and the estimated model is very similar.



(a) Histogram of synthetic data generated with the parameter λ^g



(b) Histogram of the data generated from the estimated parameter λ^{est}



(c) Log-likelihood as a function of iteration. The red line is the log-likelihood $\log(L(\lambda^g|O))$

Figure 6.2: The result for estimating the parameters for the synthetic data.

Chapter 7 Evaluation

Before reading this section the reader is encouraged to read section 2.3, as these measures are used extensively throughout section 7.1. Furthermore the reader is also referred to appendix A for a description of the files used in this objective test. The reader is also strongly encouraged to study the figures in appendix B to get a visual impression of how the algorithm perform. These figures have been positioned in the appendix because of size and not because of minor relevance.

In the previous chapters we have seen how the spectral envelope and excitation is estimated. Synthesizing the speech combining these estimates, an estimate of the wideband speech is obtained. This chapter focuses on evaluation of the estimated speech signal. The reader is referred to figure B.2 in appendix B which verifies that the framework do not introduce any errors in the narrowband region.

7.1 Objective measures on extended speech

As mentioned in section 2.3 an objective measure is convenient to get an idea of how well the algorithm performs. The measures presented are concerning envelopes. For both HMM and GMM, the RMS LSD, RMS Itakura and RMS Itakura Saito distance are calculated. Results from the codebook based approach are deliberately omitted, because it only achieved an RMS LSD of 5.14 dB and because informal listening test demonstrated very low performance by this method. Glitches were recognized and the extended speech sounded a bit "sharp". As described in section 2.3 these measures indicate how "close" an envelope are to another. In this evaluation the envelopes, which are compared are the one estimated and the original wideband envelope. All files in TEST_COMPLETE from the TIMIT database have been used in this test setup. Each file has been extended using different model-orders for both GMM and HMM. For every frame in each file, all three measures have been calculated. For every measure a mean value has been calculated, by adding the values for each frame and then dividing it by the total number of frames, as e.g. in equation 2.25 on page 19.

7.1.1 GMM

Objective measures from estimating using the GMM based method can be seen in figure 7.1 and 7.2. All distributions in the following are represented using 3 different kinds of training/estimation.

Full: During training a full covariance matrix is estimated for each mixture (4.17) . The estimation is performed with the estimator $\hat{\mathbf{y}}_{MMSE_F}$ (4.29).

Full with diagonal estimation: The same training as the above, but estimation is performed with the diagonal estimator $\hat{\mathbf{y}}_{MMSE_D}$ (4.30). These results are included to

give a understanding of the differences between using diagonal and full covariance matrices during training.

Diagonal: During training a diagonal covariance matrix is estimated (4.18) and therefore a diagonal estimator $\hat{\mathbf{y}}_{MMSE_D}$ is applied.

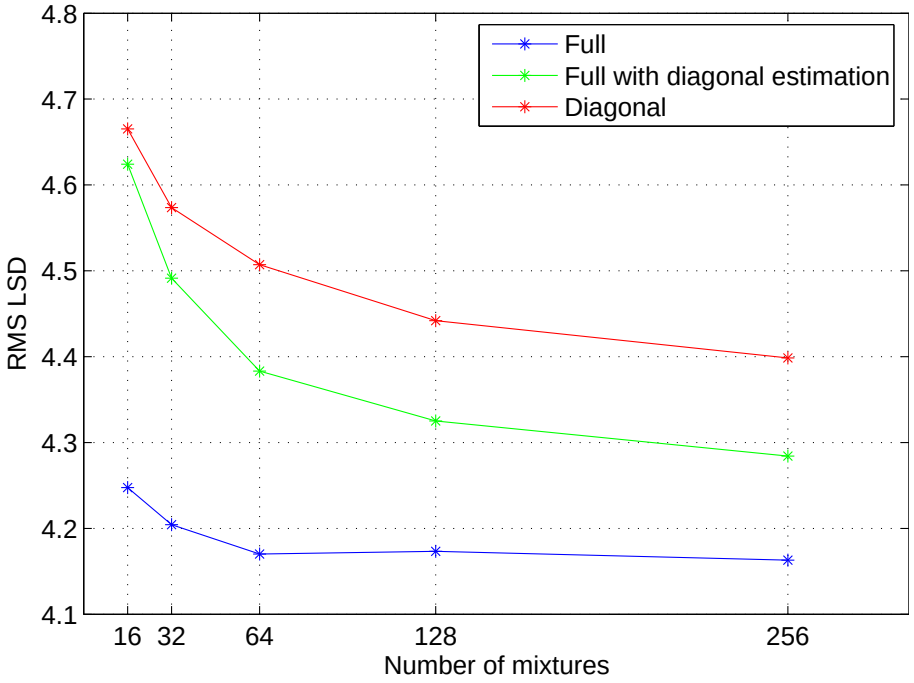
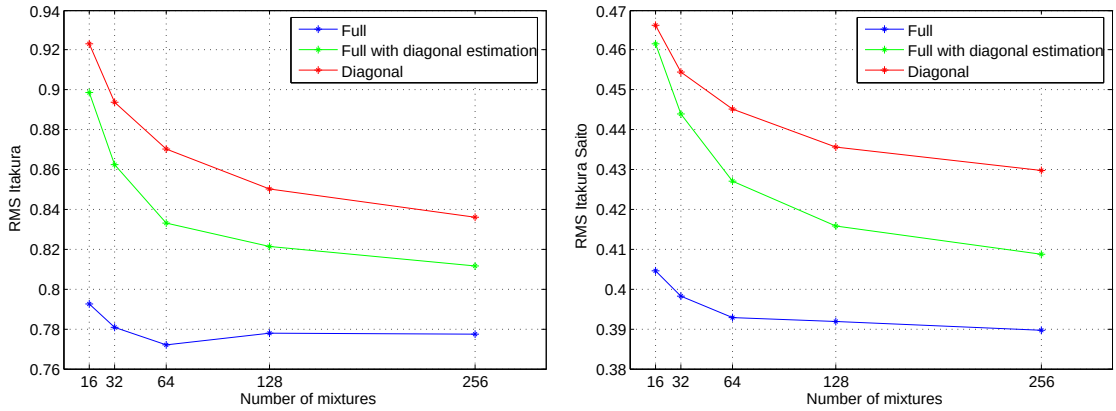


Figure 7.1: Average RMS Log Spectral Distortion plotted as a function of mixture components in a GMM. The difference between the best performing and worst performing, using full covariance matrices, can be seen to be very low. Only a difference of 0.1 dB from a GMM having 16 components to a GMM having 256 components. For GMMs using diagonal covariance matrices it is approx. 0.26 dB.

From figure 7.1 we observe an improvement in LSD by using a GMM with more components. For full covariance matrices the 16 component GMM achieves a distortion of approx. 4.25 dB whereas the GMM having 256 components achieves a distortion of 4.15 dB. The difference between the two however is only 0.1 dB and we would have expected it to be a bit higher, at least when going from 16 to 256 mixture components. In the case when using diagonal covariance matrices the difference is approx. 0.26 dB. For GMMs with 16 and 32 components there is a large difference between the two estimators with full covariance matrixes. This difference decreases when increasing the number of components increases. For 16 components the diagonal estimators perform almost similar even though the training is conducted differently.

Figure 7.2 shows that there is a reasonable relationship between the LSD and the Itakura measures. But again the difference from the GMM performing the best and the GMM performing the worst is very low. The performance in all three measures when using diagonal matrices is in general low compared to using full diagonal matrices. In all cases the GMM having 16 components and full covariance matrix actually gives a better result than a GMM with 256 components having diagonal covariance matrix. For example in the case of RMS LSD the difference is 0.15 dB. An informal listening test showed that the



(a) (b)

Figure 7.2: Itakura(a) and Itakura-Saito(b) distance calculated for GMMs with different number of components.

perceived quality of the GMM with 16 components and full covariance matrices, is higher than the GMM with 256 components and diagonal covariance matrices.

7.1.2 HMM

Because a higher performance was observed using full covariance matrices for the GMM method, all distributions used in each state of the HMMs are represented using full covariance matrices. Figures 7.3 and 7.4 depict the RMS LSD, RMS Itakura and RMS Itakura Saito distance. Figure 7.3 shows the average RMS Log Spectral Distortion as a function of number of states. Each color denotes the number of components used in each state. It can be seen that the best result achieved in terms of LSD has been achieved by using a 4 state HMM with 64 components in each state. The average RMS LSD when estimating is approximately 4.16 dB. If we turn to the highest value, it is obtained utilizing a 2 state HMM with 2 mixture components in each state. For this a value of approx. 4.35 dB is achieved. Compared to the best it is only a difference of approx. 0.2 dB. That is the difference between the best and worst performing model order is very low, and should not be noticeable at all when listening to it. Informal listening has also shown that it is almost impossible to distinguish speech files extended by using any of these plotted model orders. The Itakura and Itakura-Saito measures show the same trend as LSD. The plots show that increasing the number of states and mixture components results in lower values of LSD, Itakura-Saito and Itakura up to a certain order. For example for LSD an increase is seen from 32 states to 64 states with 2 components.

In table 7.1 we compare the ordinary way of stacking $\mathbf{z}$ during training with the perceptual stacking procedure. The table shows that perceptual method 2 results in higher values in all three measures, which indicate a greater difference between original and estimated envelope. Whereas the ordinary way and perceptual method 1 are almost identical. Informal listening tests also showed that both the ordinary way and perceptual method 1 were preferred over perceptual method 2.

It is difficult to distinguish these two methods, both are therefore included in the formal listening test. Due to time constraints the study carried out in figure 7.3 and figure 7.4, has not been conducted using both perceptual methods.

To get a more visual feeling of how "close" the original and estimated signal are, we urge

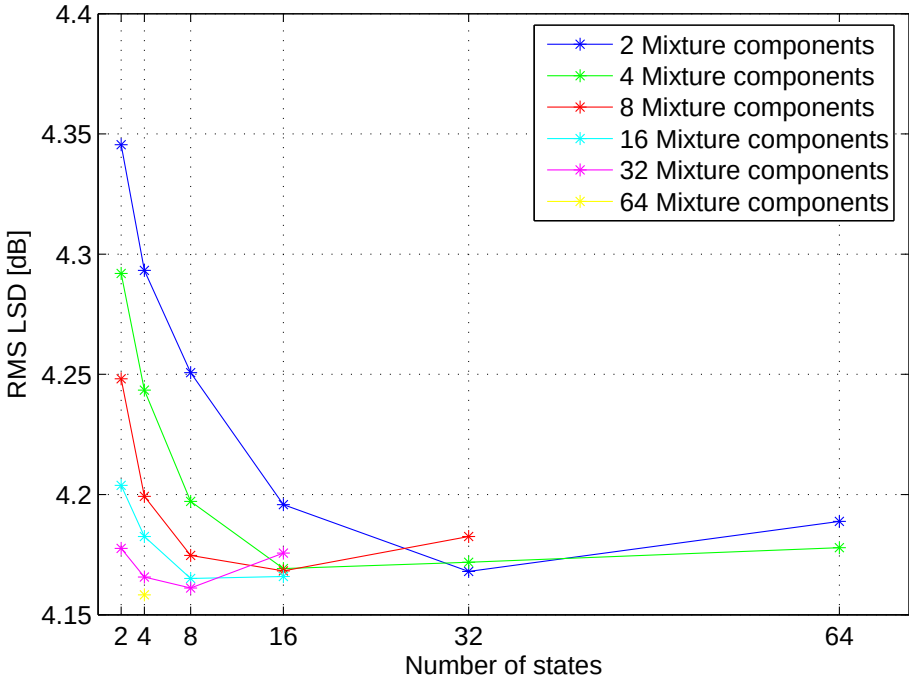
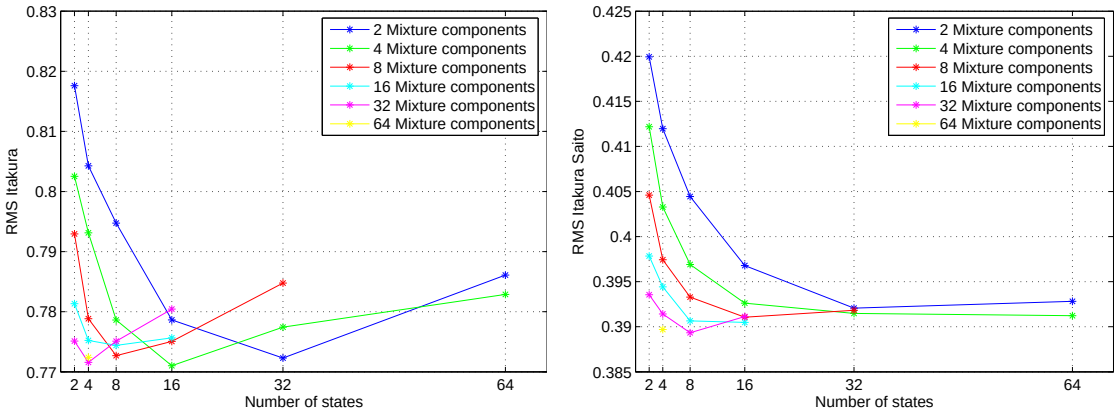


Figure 7.3: RMS Log Spectral Distortion plotted as a function of states in the HMM. Different colors denotes different number of components in each state. The difference between the best performing and worst performing can be seen to be very low. The HMMs have been trained the ordinary way.



(a) (b)

Figure 7.4: Itakura(a) and Itakura-Saito(b) distance calculated for different sizes of HMMs. In both figures it might look like the training is saturated for the highest number of states and components.

Measure/HMM	Ordinary	Perceptual method 1	Perceptual method 2
LSD	4.16 dB	4.16 dB	4.64 dB
Itakura distance	0.78	0.76	0.96
Itakura-Saito distance	0.39	0.39	0.44

Table 7.1: Comparison of ordinary way of stacking and perceptual way of stacking $\mathbf{z}$. All HMMs used in this comparison have 16 states and 8 mixtures components having full covariance matrices.

the reader to take a look at figure B.1 in appendix B.

7.1.3 Discussion

In this section obtained objective results from doing bandwidth extension have been presented. Results from estimation utilizing both HMM and GMM have been presented. Increasing the model order in both cases, does not have the desired effect of minimizing these measures. An increase in model order lead to a decrease in all measures for the HMMs up to a certain model order only. It seems like training of the model gets saturated for the highest orders. One thing could be that there is insufficient training data to train models of that size using full covariance matrices. Even though the measure decreases, only a difference of 0.2 dB, when comparing the best and the worst performing HMM can be presented. For a GMM the number is even smaller, 0.1 dB. This small difference is also supported by informal listening test, in which it was concluded that it was impossible to distinguish between speech files being extended using any of the presented models. Results for both the diagonal and full estimator are presented for the GMM. For 16 mixture components there is a difference of approx. 0.6 dB in LSD between a model with diagonal and full covariance matrix. In an informal listening test it was found that fewer artifacts were introduced and less lisp was introduced for the model with the full covariance matrix compared to the diagonal. For the GMM two estimators are used to estimate $\hat{\mathbf{y}}$. Both estimators were tried and for 16 and 32 components the result indicated that the cross covariance matrix $\mathbf{\Sigma}_l^{yx}$ has great impact on the results. It seems that this impact reduces when more mixture components are included in the model. The reason for this could be that for the low number of components, one group of closely spaced observations are modeled as one mixture component with cross covariance between $\mathbf{x}$ and $\mathbf{y}$. When the number of mixture components are increased several components may model this group of closely spaced observations and the cross covariance for each component is decreased. If the cross covariance tends toward zero the two estimators will give the same estimate. Even though no significant improvement was noticed when increasing the model order, informal listening tests showed that combining the estimated envelopes with spectral translation gave a much better sound experience compared to narrowband.

7.2 Listening test

To evaluate the subjective quality of the extended speech files a listening test is conducted. The test has primarily been designed such that the results it produces, should answer these three questions:

- How close are the estimated signal to the wideband in Mean Opinion Score (MOS) ratings. (Test 1a)
- How important is excitation and envelope respectively when doing estimation. (Test 1b,1c)
- Is the estimated speech signal better than the narrowband speech signal. (Test 2)

7.2.1 Setup

Eighteen people are going to be conducting the listening test. All of them having normal hearing capabilities. The evaluation is done using a modified version of the "The Audio Codec Benchmarking Program" by O. Niamut. See e.g. appendix C for screenshots and guide handed out to the test persons. Two HMM are used to estimate the wideband envelope. An HMM having 8 states and 32 components used to estimate the normal way (standard Baum-Welch training). To estimate using perceptual method 1, one with 16 states and 8 components are used. These two HMMs were the two yielding the lowest LSD, which were available at the time the test was conducted.

To achieve the before mentioned objective, fifteen different files total are used to create the samples used in the test. It is assumed that fifteen files is enough to be representative of the TIMIT database. Test 1a consist of five samples, which are going to be rated. Test 1b consist of ten samples where five of them are the same as those in test 1a. The same apply to test 1c. In test 2 the same files as in test 1a are compared to the corresponding narrowband version. The specific files used in each of the tests can be found in appendix A.

To assure no artifacts or anomalies are present, the speech files are listened to before they are bandwidth extended. If some files contain artifacts or an unacceptable level of noise, a new file is drawn randomly until fifteen files are obtained. The files which are used are chosen randomly from the TIMIT database.

Test 1

The test is divided into two tests. Test 1 is further divided into three subtests, 1a, 1b and 1c, to ease rating for the subjects. These tests have a certain amount of samples which are to be rated. A sample is a reference speech file together with different processed versions of this reference. In our case a wideband, and versions utilizing different ways of estimating the upper band are available. The number of versions which are going to be rated, varies from two to four per test. The wideband version is the reference and is given the highest rating. The test subject is therefore not required to rate the reference. That is, we say that the original wideband is the best in terms of quality. By quality is meant, what the subject prefers to listen to. The test subject is required to rate each of the versions according to a degradation MOS score listed by [20]. The range of possible ratings are shown in table 7.2. It should be noted that the versions of the speech file can be rated using steps of 0.25.

Test 1a: Joint evaluation

This test combines estimation of both the envelope and the excitation signal. The first two versions are speech files which are synthesized using estimated wideband envelope

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory	Very annoying and objectionable

Table 7.2: The different possible ratings in the listening test. From [20, p.578]

(normal way). This is combined with estimating excitation by spectral translation and the observation based approach. The two other files use perceptual training for estimation of the wideband envelope and same scenario for excitation as the first two files. Table 7.3 shows the versions which are going to be evaluated in test 1a.

Version	Envelope	Excitation	Comments
Reference	–	–	Original wb
Est-ST	Standard training	Spectral Translation	
Est-Obs	Standard training	Observation based method	
Perc-ST	Perceptual training	Spectral translation	
Perc-Obs	Perceptual training	Observation based method	

Table 7.3: Versions used for evaluation in test 1a - Joint Evaluation. Envelope and Excitation refers to the method used when synthesizing the different versions.

For each of the versions a MOS score is calculated. Because some of the same files are used in test 1b and 1c, the score obtained here would be comparable with those of test 1b and test 1c.

Test 1b: Evaluation of excitation estimation This test include a wideband speech signal and two other signals. One which has been extended using original wb envelope and spectral translation. The other one also uses original envelope but the observation method to extend the excitation. Table 7.4 shows how this test is going to be conducted.

Version	Envelope	Excitation	Comments
Reference	–	–	Original wb
Org-ST	Original	Spectral Translation	
Org-Obs	Original	Observation based method	

Table 7.4: Versions used for evaluation of excitation.

This test should show if the observation based method of estimating excitation performs better than regular spectral translation. Furthermore it should reveal if any unpleasant artifacts are introduced by extending the excitation using the two methods. This test also

gives a bound for the possible best which can be achieved, if the envelope was estimated perfectly. This test is comparable with test 1a and 1c.

Test 1c: Evaluation of envelope estimation
Besides the wideband, this test includes two other files. Both files have been synthesized using original excitation. The difference is that one of the files has been extended using "normal" estimation, the other has been extended using Perceptual training. Table 7.5 shows the versions which are going to be evaluated.

Version	Envelope	Excitation	Comments
Reference	–	–	Original wb
Est-Org	Standard training	Original	
Perc-Org	Perceptual training	Original	

Table 7.5: Versions used for evaluation of envelope.

By using original excitation and only estimating the envelope in two different ways, this test will show which of the two estimation methods produces the perceptually "best" speech signal. Furthermore it shows the possible best which could be achieved if the excitation was estimated perfectly.

Test 2

In test 2 the subjects are to decide whether or not he/she prefer the reference over each of the extended versions in the test. The results from the test subjects are both highly dependent on both rating-criterion and questions asked. They could e.g. be asked to rate the versions in terms of the broadband feeling they experience. This question would definitely bias the test person. The result we would get, is which one sounds the most broad banded and not which one they like to listen to. This however is not the desired outcome. Though a more detailed signal is created by the BWE algorithm it may introduce artifacts, such as ringing, metallic sounds etc. Hopefully these artifacts are kept at a minimum, such that the bandwidth extended versions are preferable compared to the narrowband speech. Therefore this test should reveal, if this possible degrading has reduced the overall perceptual quality compared to narrowband speech.

Test 2: Preference test
In this test the subject is to choose whether he/she prefers to listen to either the narrowband (reference) or the versions. The result from this test should reflect if estimation has been carried out such that the extended speech file is preferred to narrowband speech in terms of perceptual quality.
This test shows if the estimated speech signal has a perceived higher quality than the narrowband speech signal.

Version	Envelope	Excitation	Comments
Reference	–	–	nb
Est-ST	Standard training	Spectral Translation	
Est-Obs	Standard training	Observation based method	
Perc-ST	Perceptual training	Spectral translation	
Perc-Obs	Perceptual training	Observation based method	

Table 7.6: Versions used in preference test.

7.2.2 Discussion of setup

The optimal setting would of course be as many samples in each setup as possible and as many test subjects as possible. This is not always possible, therefore a compromise has to be made. Only five samples have been chosen in test 1a. This has been done to reduce the total time which it takes to do the test. By using the same five speech files in test 1b and 1c makes it possible to compare them with test 1a.
Development of the test has been carried out in an iterative way. Different types of tests/setups have been tried out, by letting people do the test. The test setup has then been discussed and adjusted properly. In one of the first test setups, narrowband speech was included in test 1a, 1b and 1c and test 2 was omitted. It was decided that narrowband speech was removed from the first test, because preliminary test showed that it is difficult to rate narrowband speech compared to wideband speech. By removing the narrowband speech version, the subject only has to rate a degraded[1] version of same bandwidth, and not decide how he/she should rate wideband compared to narrowband speech. But by removing the narrowband, no comparison of the estimated wideband signal and the narrowband is conducted. In literature a MOS score of approx. 4 corresponds to what we think of as "telephone-speech". One should therefore think that ratings obtained from test 1 could be compared to the score of "telephone-speech". That is not the case, because in a wideband test the same scale is used, hence a wideband codec may have a MOS score of 3.9 even though it sounds much better than a narrowband codec with a MOS of 4.1 [27]. Therefore test 2 was included in the setup. By using MOS score in test 1a-c it makes it possible to compare results from the test with MOS scores for different wideband speech codecs. In this test it was decided that the subject should not rate the versions compared to the narrowband, but instead make a preference. This was done in order to make it easier for the subject, and because the rating of the estimated signal is already covered by test 1.

Conduction of tests

The tests are conducted in a quiet room. During the test only the test person is in the room. The speech files are played back on a standard pc using a set of Bayer Dynamics DT990 headphones. Before the test, a short introduction was read aloud from a piece of paper. The test person was then told to read the listening test guide in appendix C. The

[1] It is actually not degraded from wideband, but (hopefully) enhanced compared to narrowband. The test subject however will perceive it as degraded compared to wideband. This is also what they are told in the guide (appendix C)

test person performed a "dummy" test to get familiar with the software. Until the real test was commenced, the test person was allowed to ask any question he/she might have. The duration of the test was on average 30 minutes for each person.

7.2.3 Results

Eighteen people have done the test. The results from test 1a, 1b, 1c can be seen in figure 7.5, 7.6, 7.7 and 7.8. Figure 7.5 shows how people have rated each of the versions from the tests in Test 1a. The x-axis denotes how the version rated, is synthesized in the BWE algorithm.

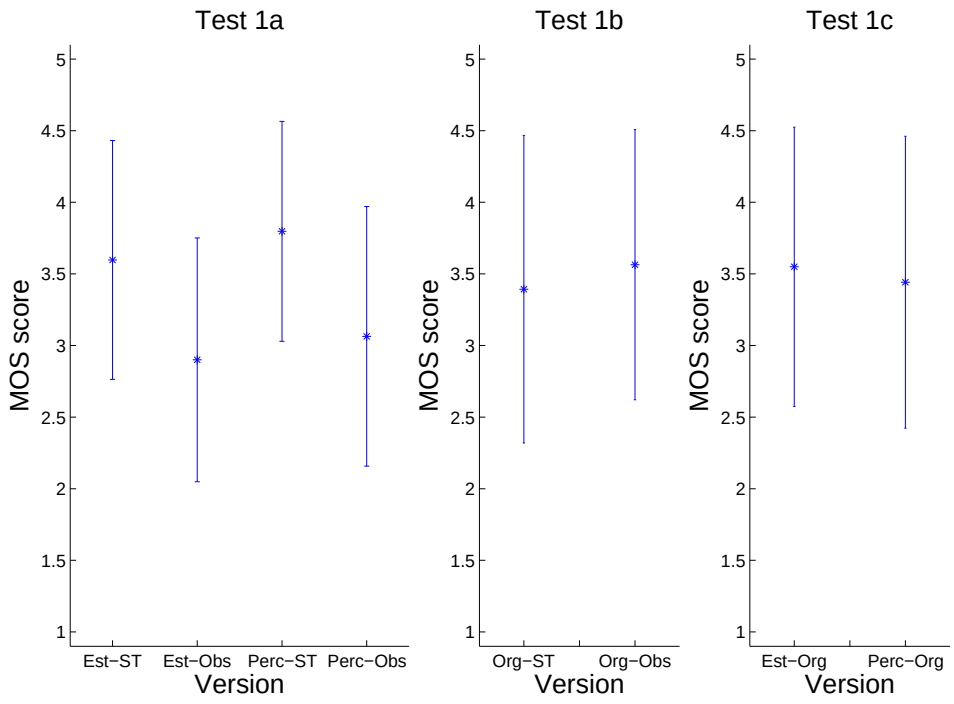


Figure 7.5: Mean Opinion Scores for Test 1a,1b and 1c. The y-axis shows the MOS score obtained in the test. The x-axis denotes the versions.

Mean and standard deviation is plotted for each of the versions. The asterisk denotes mean value and the line is the standard deviation. For all versions, the standard deviation is quite high, meaning that people have rated using both ends of the scale. From 1b and 1c it is observed, that the mean-values and standard deviation are almost similar. Therefore no method can be said to be preferred over another. The standard deviation in test 1a is also observed to be more or less the same for all versions. The mean values, however, are higher for the versions extended using spectral translation. In the case of both estimation of envelope and estimation of excitation, spectral translation must be claimed to perform better than the observation based method.

Figure 7.6 is included, as supplement to figure 7.5. This shows the frequency as a function of the ratings for each version. This gives a more detailed picture of the distribution compared to the previous figure. The method utilizing Perceptual envelope and spectral translation has higher frequencies for higher ratings, which also is indicated on figure 7.5 as a higher mean value. The distribution seems to be bell-shaped.

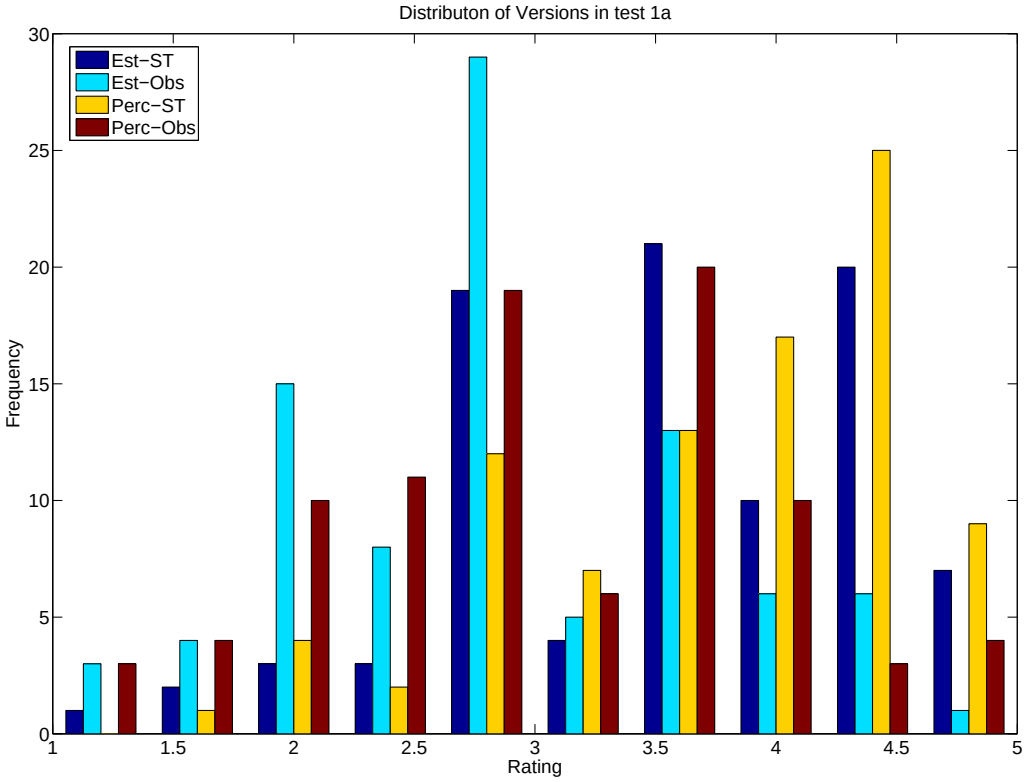


Figure 7.6: Distributions for each of the versions in test 1a.

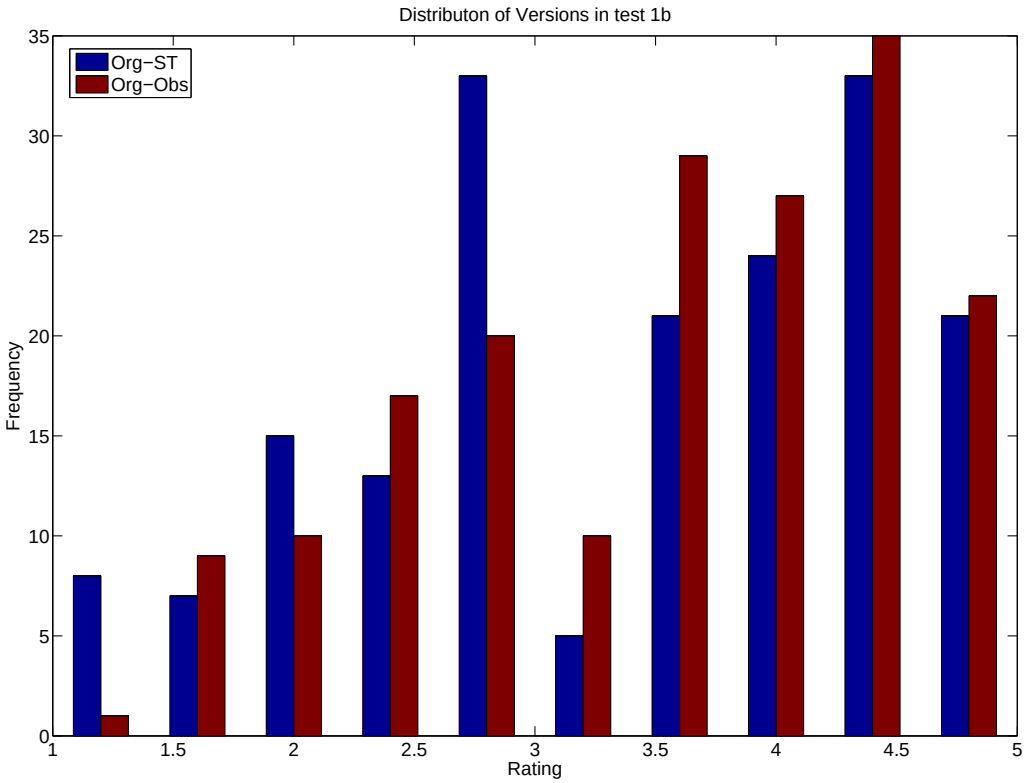


Figure 7.7: Distributions for each of the versions in test 1b.

Figure 7.7 and figure 7.8 is plotted the same way as figure 7.6. In both test 1b and 1c the two versions, i.e the red and blue color bars have almost equal frequency for each of the ratings. One exception is in test 1b. Only one person rated one version in one sample, to be very annoying and objectionable, for the method using original envelope and observation based method.

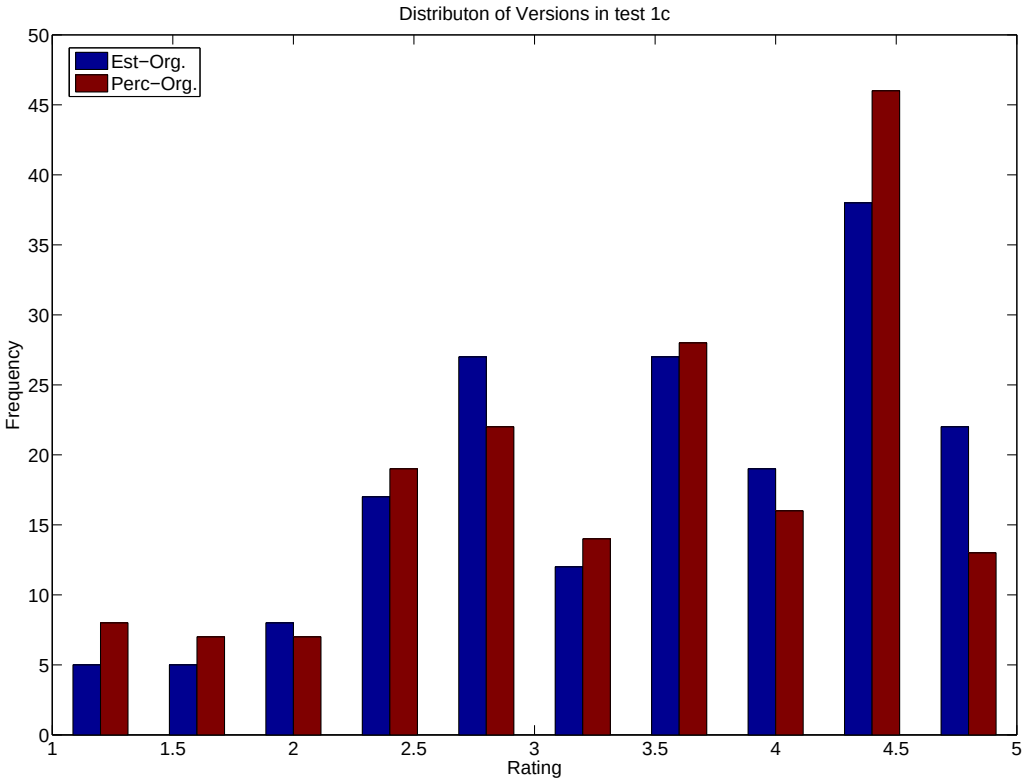


Figure 7.8: Distributions for each of the versions in test 1c.

The results from test 2 is presented in figure 7.9. More people prefer to listen to the versions Est-ST and Perc-ST compared to narrowband speech. Especially the version Perc-ST getting 74 votes against narrowband only getting 16. The two methods using the observation based method get less votes than narrowband. That is, most people prefer to listen to narrowband compared to using these methods.

The significance of the results for test 2 is evaluated using a simple sign test. The objective is find if the methods Est-ST and Perc-ST are better compared to narrowband. Since we have a set of matched-paired data the non-parametric sign test is applied. Assuming that the probability that the extended speech is better than narrowband is p and is equal for all test pairs (n), the following null and alternative hypotheses can be defined as:

- H_0: $p \leq 0.5$ "The extended speech is no better than narrowband"
- H_1: $p > 0.5$

The listeners could only choose whether or not they preferred the extended speech over narrowband. Therefore we can model the number of pairs (M), where extended speech is preferred as binomial distribution B(n,p). The binomial pdf is depicted in figure 7.10.

The null hypothesis can be rejected if M is greater than the critical point found from the cumulative binomial distribution function and a level of significance. The critical point

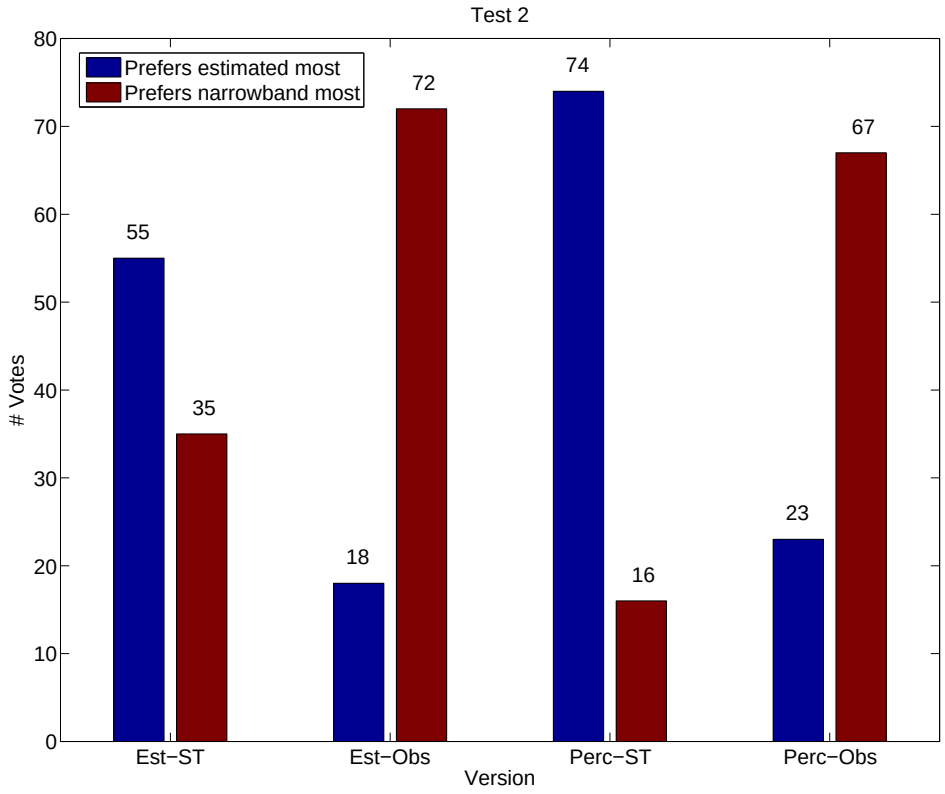


Figure 7.9: Number of votes given to each version in the preference test.

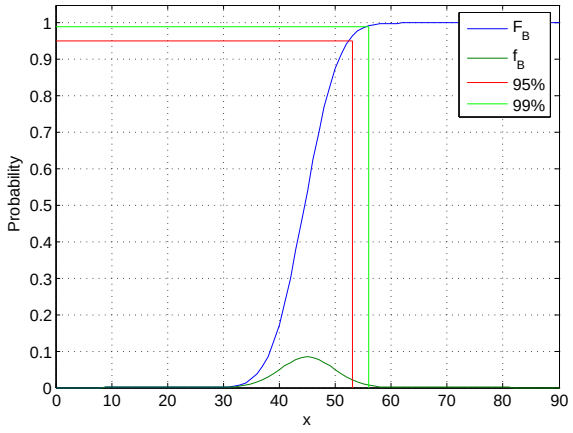


Figure 7.10: The cumulative binomial distribution, $F_B(x)$. With the probability for method 1 of 50% and 90 match paired data samples. The 95% and 99% level of confidence intervals are included.

in our case with 90 matched pairs is 53 and 56 for a 95% and 99% level of confidence respectively. From figure 7.9 it is seen that the methods Est-ST and Perc-ST score above 53, hence H_0 is rejected with 95% level of confidence. The p-values (p-value = 1-$F_B(M)$) for the sign test are:

Method	p-value
Perc-ST	$2.19 \cdot 10^{-10}$
Est-ST	0.022

The null hypothesis is definitively rejected for Perc-ST and is rejected for Est-ST with a 0.02 level of significance. Therefore the test shows the extended speech to be statistical significantly better than narrowband speech with a level of confidence of 98% for estimated (Standard Baum-Welch) and almost 100% for Perc-ST.

7.2.4 PESQ measure

To further evaluate the Bandwidth Extension Algorithm, the same speech files used in the formal listening test are evaluated using the PESQ measure. PESQ stands for Perceptual Evaluation of Speech Quality, and is an enhanced perceptual quality measurement for voice quality in telecommunications. It is a program which take a reference file and compare it to a degraded version. PESQ was specifically developed to be applicable to end-to-end voice quality testing under real network conditions, like VoIP, ISDN, GSM [13].
PESQ as defined by ITU-T P.862 is an originally an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs [13]. A wideband extension to ITU-T P.862 has been introduced with ITU-T P.862.2 which is used to evaluate wideband telephone networks and speech codecs. This is the one utilized in this test. The comparison between narrowband and the extended version can not directly be compared as the PESQ-program requires a sampling rate of 16 kHz. The narrowband speech signal is therefore upsampled before comparison. The narrowband version which is upsampled is the same version we have available when doing estimation. That is the lower band from 0-300 Hz is present. Figure 7.11 shows PESQ scores for the same files and versions as in the listening test.
It should be noted that the reason nb gets different PESQ scores is because different narrowband files are used in the tests. The first thing which is observed is that the variance using the PESQ measure is lower than in the formal listening test. In test 1a it is observed that narrowband gets a slightly better score than any of the estimated methods. The difference however is not significant. It is a bit surprising that the extended versions do not get a higher score than narrowband. It could be that the PESQ measure weighs a serious error in the upper band harder than no spectral components being there at all. This is only guessing and in order to fully understand the reason, a closer look at the standard is required. In test 1b, where original envelope is used to synthesize the speech, a high score is obtained for both methods with a turn of the scale for the observation based method of creating the wideband excitation. Test 1c shows that no better scores are achieved when using the PESQ evaluation software. Conclusion is that the extended speech is as good as narrowband speech and that the spectral envelope is more significant than excitation, when applying the PESQ measure.

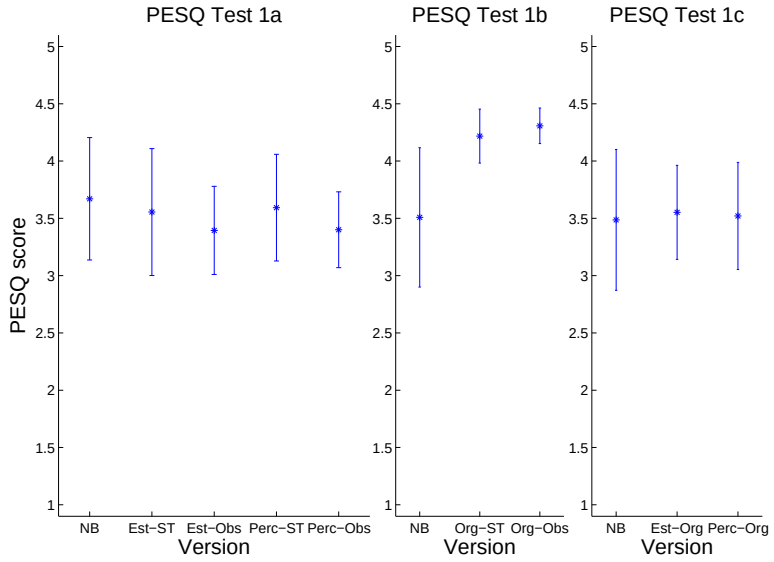


Figure 7.11: PESQ scores on same files used in listening test. In this plot however the upsampled nb is also included.

7.2.5 Discussion of results

The combination of estimating the excitation based on the Observation method together with original envelope, shows at least as good results as using spectral translation. When estimating the envelope, the performance of this method decreases drastically. We suspect the decrease in performance is caused by the noise we add in the upper band of the excitation signal. Because the envelope is not estimated perfectly, the noise is not shaped properly. This is audible and quite annoying to listen to. If however the extended speech signal was played back using a set of standard pc-speakers these artifacts reduced to being inaudible. Using this setup it was almost impossible to distinguish the extended from the original wideband speech, whereas a clear difference was audible when compared to narrowband speech. The test has shown that the method using perceptual envelope and spectral translation is promising in test 1a. Furthermore it shows that people prefer versions utilizing this method compared to narrowband with a level of confidence very close to 100%. Therefore bandwidth extension is definitely feasible, and the methods Org-ST or Perc-ST should be used.

Chapter 8
Summary and Conclusion

In this thesis we addressed the problem of artificial BandWidth Extension (BWE) of speech. Bandwidth extension is motivated by the limited frequency range in ordinary telephone networks. This limitation in bandwidth is due to standards from the old analogue telephone networks. By reducing bandwidth, we simultaneously reduce the quality of the speech signal. If the receiver is capable of playing wideband signals, quality of the speech signal could be increased by processing the speech before playback. This is attractive as the potential quality improvement comes without increasing the bit rate, i.e no extra information is required to be transmitted over the channel. All along this project, focus has been to process narrowband speech signals (perform artificial BWE) without increasing the amount of information sent over the channel and without modifying the transmitting side. From our point of view, if side information are to be sent, the solution might as well be to use a true wideband codec. The desire was to develop a feasible solution without modifying the existing network, and only introducing modifications at the receiver.
It is chosen to use a parametric approach by using the source filter model, and thereby exploiting characteristics in the speech production system. In this way it is possible to utilize statistical tools for estimation.
To realize the BWE algorithm, four frameworks were presented, which all were transparent in the narrowband region. An analysis of these four resulted in a single framework being chosen to realize the algorithm. The framework has the effect that the narrowband region of the analysis filter is derived by linear prediction. Furthermore estimation can be done in subbands, thus fewer features are needed during both estimation and training, compared to the other frameworks.

Estimation and objective results

Three estimation rules based on a conventional codebook (vector quantization), a Gaussian Mixture Model (GMM) and a Hidden Markov Model (HMM) were developed and fully implemented in Matlab®. Estimation using a simple and conventional codebook indicated that correlation between narrowband and extensionband were limited, when using the applied feature representation. An informal listening test also shows that estimation using this method was not sufficient. Glitches were recognized and the extended speech sounded a bit "sharp".
Applying a GMM and using full covariance matrices improved the speech quality considerably. This was supported both by objective measures and informal listening test. The codebook method achieves a maximum RMS Log Spectral Distortion (LSD) of 5.14 dB, whereas the best GMM (256 mixture components) achieves an RMS LSD of 4.16 dB.

GMMs with model orders of 16 to 256 components have been trained. The increase in performance however was not observed to be as high as expected when increasing the model order. A GMM of 16 components, achieves an RMS LSD value of 4.24 dB compared to a 256 component achieving 4.16 dB.

When training and estimation are done using diagonal matrices, the best result is achieved using a GMM of 256 components which achieves an RMS LSD of 4.39 dB. Meaning that the worst performing GMM with 16 components using full covariance matrices is actually 0.15 dB better than the best performing GMM of 256 components using diagonal covariance matrices. This shows that using full covariance matrices contributes more to a better estimate than increasing the order, at least for a model order of 16 or higher.

Implementing the HMM framework, and thereby being able to model frame dependencies, did not increase the quality of estimation as much as expected. The best achieving HMM was an HMM of 4 states and 64 mixture components. Estimation using this model, achieves an RMS LSD of 4.15 dB. Compared to the GMM it is an increase of 0.01 dB. An informal listening test also supported the fact that no audible enhancement was achieved compared to the GMM. This is most likely a combination of both frame length and overlap used, when processing the signal into frames.

The BWE algorithm was applied on narrowband speech using an HMM with 16 states and 8 mixture components as estimator. This showed that for the extensionband, the variations in the spectral envelope are very small and the formants are not estimated correctly. On the other hand, no dominating peaks outside the formant frequencies, are introduced in the upper part of the spectrum either. This would probably introduce unwanted artifacts.

Listening tests

Despite the fact that the formants were not estimated correctly, informal listening tests shows that the estimated speech signal provided good audible results. It is observed that small changes in the objective measures result in large differences in the perceived quality. In the case of the GMM with 256 components and diagonal covariance matrices versus the GMM with 16 components and full covariance matrices only a difference of 0.15 dB RMS LSD is observed. Despite the low difference in distortion, an informal listening test surprisingly revealed a higher perceived quality for the GMM with full covariance matrices. Informal listenings of the extended signal actually gave an experience of listening to real wideband. A small weakness is however audible when extending files with a lot of unvoiced sounds. The algorithm introduces a minor lisp to the speaker, which can be heard using a pair of high-end headphones. Playback of the extended signal having small artifacts on a set of pc-speakers reduced these effects to being practically inaudible. Because the BWE algorithm works on speech signals, playback capabilities equivalent to ordinary pc-speakers is more likely to be used. When artifacts are non-audible the extended speech signal sounds much more pleasant than the corresponding narrowband speech.

These results are furthermore supported in the formal listening test in which the extended signals are fully estimated using HMM. These signals were ranked alongside extended signals using both original envelope or original excitation. Some people even rated the fully estimated speech signal to be as good as the original wideband in some cases. In the preference test a total of 55 out of 90 votes were given to the estimated speech (standard

Baum-Welch algorithm) and 35 votes were given to the narrowband signal. It is concluded that the estimated speech is statistically significantly preferred over narrowband speech at a confidence level of 98%. A total of 74 out of 90 votes goes to the estimated speech using the proposed perceptual method and only 16 going to narrowband. For this new method we conclude that the estimated speech is more preferable compared to narrowband with a level of confidence of almost 100%

Conclusion

The promising results for the proposed method shows that including properties of the auditory system into the features used in training, improves the perceived quality of the extended speech. Artificial Bandwidth Extension, as presented here, is therefore considered to be an attractive solution to provide wideband speech, in the transition phase of going from the old telephone networks to real wideband telephony. Even though the suggested BWE algorithm most likely could be implemented and utilized as it is now, further research should be carried out to mature the technology completely.

Perspective

To further improve the artificial BWE algorithm and having implementation in mind, further research could be carried in the following areas:

- Other models could be utilized to model the speech, e.g. sinusoidal modeling.
- Pitch estimation when extending excitation signal
- Incorporate state durations if an HMM is used for estimation
- Increasing the framelength to 30 ms and having no overlap, the advantages of the HMM could probably be exploited better.
- A more intensive study of which features to utilize in both training and estimation
- Examine the impact of weighting or scaling of the elements of the feature vector.
- Make the algorithm robust towards noise

Bandwidth extension as presented here, has primarily focused on improving narrowband telephone speech on a wideband capable terminal. Even after a transition phase where true wideband exists in communication systems, there will be a need for BWE. It can be incorporated into wideband speech codecs, in order to maintain low bit rates. The concept of BWE can easily be extended to estimating "super-wideband", i.e. extending speech signals from e.g. 8 kHz to 16 kHz. Finally historical recordings could be bandwidth extended in order to obtain a wideband version of the recordings.

Abbreviations

AR	Auto Regressive
A-law	Compander used in telephone networks in whole world except North America and Japan[10, p. 343]
A/D	Analog to Digital conversion
BWE	Bandwidth Extension
CC	Cepstral Coefficients
CELP	Code Excited Linear Prediction
DFT	Discrete Fourier Transform
D/A	Digital to Analog conversion
eb	extensionband
EM	Expectation-Maximization
GMM	Gaussian Mixture Model
GSM	Global System for Mobile Communications
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
ISDN	Integrated Services Digital Network
LP	Linear Prediction
LPC	Linear Prediction Coding
LSD	Log Spectral Distortion

MFCC	Mel Frequency Cepstral Coefficients
MMSE	Minimum Mean Square Error
MOS	Mean Opinion Score
μ-law	Compander used in telephone networks in North America and Japan[10, p. 343]
nb	narrowband
pdf	Probability density function
PESQ	Perceptual Evaluation of Speech Quality
PSTN	Public Switched Telephone Network
RMS	Root Mean Square
SLP	Selective Linear Prediction
TIMIT	Speech data base database (See appendix A)
VoIP	Voice over IP (Internet Protocol)
VQ	Vector Quantizer
wb	wideband

Symbols

The notation $(k;m)$ denotes the k^{th} sample in the m^{th} frame for a given signal, e.g. $s_n(k;m)$. If k is replaced by ω, i.e. $(\omega;m)$ it denotes the angular frequency ω for the m^{th} frame, e.g. $\Phi_n(\omega;m)$. Estimates are mark with the symbol ($\hat{}$), e.g. $\widehat{\Phi}_e$ which denotes the estimated extensionband power spectrum.

Symbol	Meaning
s_w	Wideband speech
s_n	Narrowband speech
u_n	nb excitation signal
u_w	wb excitation signal
$\mathbf{x}_{acf}$	Vector containing the first ten coefficients of the autocorrelation function.
x_{zcr}	Zero crossing rate.
x_{gi}	Gradient index.
x_{nrp}	Normalized relative frame energy
x_k	Local kurtosis
x_{sc}	Spectral centroid
$\mathbf{x}_{scl}$	Feature vector containing scalar features: $\mathbf{x}_{scl} = [x_{zcr}, x_{gi}, x_{nrp}, x_k, x_{sc}]^{\mathrm{T}}$
$\mathbf{x}$	Vector containing features extracted from narrowband speech
$\mathbf{y}$	Vector representing the extensionband envelope
$\mathbf{z}$	Vector obtained from stacking $\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$
$\mathbf{y}_{mfcc}$	Vector containing the MFCC for the extensionband
$\mathbf{w}_{mfcc}$	Vector containing the MFCC for the wideband
$\mathbf{w}_{perc}$	Stacked vector consisting of $\mathbf{x}_{scl}$ and $\mathbf{w}_{mfcc}$ or $\mathbf{x}_{acf}$, $\mathbf{x}_{scl}$ and $\mathbf{y}_{mfcc}$
$\mathbf{z}_{comp}$	The complete feature vector in perceptual training.
$A(z)$	Analysis filter
$H(z)$	Synthesis filter
$G(z)$	Glottal pulse model

Continued on next page

Symbol	Meaning
$V(z)$	Vocal tract model
$R(z)$	Radiation model
$\Phi_n(\omega; m)$	Power spectrum of a narrowband signal
$\Phi_w(\omega; m)$	Power spectrum of a wideband signal
$\Phi_e(\omega; m)$	Power spectrum of a extensionband signal
AR_w	Set of AR coefficients from which a wb envelope can be obtained
AR_n	Set of AR coefficients from which a nb envelope can be obtained
AR_e	Set of AR coefficients from which a eb envelope can be obtained
$A_w(\omega; m)$	Frequency response of the wb analysis filter.
$A_e(\omega; m)$	Frequency response of the eb analysis filter found using SLP.
$A_{e_t}(\omega; m)$	Frequency response of the translated eb analysis filter found using SLP.
N_k	Number of samples per frame.
d^2_{LSD}	Squared Log Spectral Distortion
$\bar{d}_{LSD}$	RMS LSD
$\bar{d}_{LSD}$	RMS LSD
d_I	Itakura distance
d_{I-S}	Itakura-Saito distance
$c(n)$	The nth cepstral coefficient.
a_n	The nth AR coefficient.
f_s	Sampling frequency.
Ω_m	Modulation frequency for spectral translation.
$\omega_{e,l}/\omega_{e,u}$	Lower/upper frequency in extensionband.
$\omega_{n,l}/\omega_{n,u}$	Lower/upper frequency in narrowband.
$\sigma^2_{u_n}$	Variance of the nb excitation signal
σ^2_e	Gain factor for the extensionband.
$\bar{\Phi}^2_n$	Mean value of the power spectrum in the nb interval (nb gain factor).
σ^2_{rel}	The ratio between eb and nb gain factor, i.e $\sigma^2_{u_e}/\bar{\Phi}^2_n$
$r(\eta; m)$	The auto correlation function.
$\widehat{AR}_w$	AR coefficients which are found after assembly.
k	Sample index for wideband speech (16 kHz)

Continued on next page

Symbol	Meaning
k'	Sample index for narrowband speech (8 kHz)
T	The number of observations extracted from the TIMIT database
$\mathbb{C}_x$	Codebook for $\mathbf{x}$
$\mathbb{C}_y$	Codebook for $\mathbf{y}$
$\mathbb{C}_z$	Codebook for $\mathbf{z}$
M	Number of mixture components in a HMM
N	Number of states in a HMM
$\boldsymbol{\pi}$	Initial state distribution
$\mathbf{A}$	State transition matrix
a_{ij}	State transition probability
Θ	Set of parameters defining a GMM
λ	Set of parameters defining a HMM consist of; $\boldsymbol{\pi}$, $\mathbf{A}$, Θ
c_{jm}	Gain for the m^{th} component in the j^{th} state
$\boldsymbol{\mu}_{jm}$	Mean vector for the m^{th} component in the j^{th} state
$\boldsymbol{\Sigma}_{jm}$	Covariance matrix the m^{th} component in the j^{th} state
$p(\cdot)$	Likelihood
$P(\cdot)$	Probability
$L(\cdot)$	Likelihood function
$\mathbf{z}_t$	The t^{th} observation
S_j	State j
$\alpha_t(j)$	Forward-variable
$\beta_t(j)$	Backward-variable
$\xi_t(i, j)$	Probability of being in state S_i at time t and state S_j at time $t + 1$
$\gamma_t(j)$	Probability of being in state S_j at time t
$\gamma_t(j, m)$	Probability of being in state S_j at time t with the m^{th} component generating the observation
Sc_t	Scaling factor at time t
ϵ	Convergence threshold
$D_s(\lambda^1, \lambda^2)$	Distance measure between HMMs
η	Sample lag in the auto correlation function

Bibliography

[1] V. Berisha and A. Spanias.
A scalable bandwidth extension algorithm.
ICASSP, 2007.

[2] J. A. Bilmes.
A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models.
April 1998.

[3] L. D. Consortium.
Timit acoustic-phonetic continuous speech corpus.
CD-ROM.
ISBN 1-58563-019-5.
Version NIST Speech Disc 1-1.1.

[4] S. B. Davis and P. Mermelstein.
Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.
IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, ASSp-28(4):357–366, August 1980.

[5] N. Enbom and N. B. Kleijn.
Bandwidth expansion of speech based on vector quantization of themel frequency cepstral coefficients.
1999 IEEE Workshop on Speech Coding Proceedings, pages 171–173, 1999.

[6] J. Epps and W. Holmes.
A new technique for wideband enhancement of coded narrowband speech.
1999 IEEE Workshop on Speech Coding Proceedings, pages 174–176, 1999.

[7] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren.
DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus.
U.S. Department of commerce, nist speech disc 1-1.1 edition, February 1993.
Documentation for the TIMIT database.

[8] B. Geiser and P. Vary.
Backwards compatible wideband telephony in mobile networks: Celp watermarking and bandwidth extension.
ICASSP 2007, April 2007.

[9] A. Gersho and R. M. Gray.
Vector quantization and signal compression.
Kluver 1992, 1992.

[10] X. Huang, A. Acero, and H. Hon.

Spoken Language Processing.
Prentice Hall, 2001.
ISBN 0-13-022616-5.

[11] B. Iser and G. Schmidt.
Bandwidth extension of telephone speech.
Technical Report 2, Termic SDS Research, June 2005.

[12] ITU-T TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU.
ITU-T Recommendation G.712 Transmission performance characteristics of pulse code modulation channels, 11 2001.

[13] ITU-T TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU.
Perceptual evaluation of speech quality PESQ: An objective method for end-to-end speech quality assesment of narrowband telephone networks and speech codecs Amendment 2., 11 2005.

[14] V. Iyengar, R. Rabipour, P. Mermelstein, and B. R. Shelton.
Speech bandwidth extension method and apparatus, October 1995.
`http://www.freepatentsonline.com/5455888.html`.

[15] L. H. Jamieson.
Introduction to the physiology of speech and hearing.
`http://cobweb.ecn.purdue.edu/~ee649/notes/physiology.html`.

[16] P. Jax.
Enhancement of Bandlimited Speech Signals: Algoriths and Theoretical Bounds.
PhD thesis, Institut für Nachrichtengeräte und Datenverarbeitung, 2002.

[17] P. Jax and P. Vary.
Wideband extension of telephone speech using a hidden markov model.
2000 IEEE Workshop on Speech Coding, pages 133–135, 2000.
ISBN 0-7803-6416-3.

[18] P. Jax and P. Vary.
Artificial bandwidth extension of speech signal.
ICASSP, 2003.

[19] P. Jax and P. Vary.
On artificial bandwidth extension of telephone speech.
Signal Processing, 83:1707–1719, 2003.

[20] J. D. jr., J. H. Hansen, and J. Proakis.
Discrete-time Processing of Speech Signals.
Wiley-interscience, 1993.

[21] K. R. Krishnamachari, R. E. Yantorno, J. M. Lovekin, and et al.
Use of local kurtosis measure for spotting usable speech segments in co-channel speech.

[22] E. Larsen and R. M. Aarts.
Audio Bandwidth Extension.
John Wiley and Sons Ltd, 2004.
ISBN 0-470-85871-0.

[23] Y. Linde, A. Buzo, and R. Gray.
An algorithm for vector quantizer design.

IEEE Transactions on Communications, 28(1):84–95, Jan 1980.

[24] J. D. Markel and A. H. Gray.
Linear Prediction of Speech.
Springer-Verlag, 1976.
ISBN 3-540-07563-1.

[25] M. Nilsson, H. Gustafson, S. V. Andersen, and W. B. Kleijn.
Gaussian mixture model based mutual information estimation between frequency bands in speech.
ICASSP, 1:525–528, 2002.

[26] K.-Y. Park and H. S. Kim.
Narrowband to wideband conversion of speech using gmm based transformation.
IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings., 3:1843–1846, 2000.
ISBN 0-7803-6293-4.

[27] A. C. Ph.D.
Tech note: Voice quality measurement.
`http://www.tmcnet.com/tmcnet/articles/2005/voice-quality-measurement-vo%ip-alan-clark-telchemy.htm`, February 2005.

[28] S. R. Quackenbush, T. P. B. III, and M. A. Clements.
Objective Measures of Speech Quality.
Prentice Hall, 1988.
ISBN 0136290566.

[29] L. Rabiner and B.-H. Juang.
Fundamentals of speech recognition.
Prentice Hall, 1993.
ISBN 0-13-015157-2.

[30] L. R. Rabiner.
A tutorial on hidden markov models and selected applications in speech recognition.
Proceedings on the IEEE, 77(2), 1989.

[31] L. R. Rabiner and R. W. Schafer.
Digital processing of speech signals.
Prentice Hall, 1978.

[32] D. A. Reynolds and R. C. Rose.
Robust text-independent speaker identification using gaussian mixture speaker models.
IEEE transactions on speech and audio processing, 3(1):72–83, January 1995.

[33] K. S. Shanmugan and A. M. Breipohl.
Random Signal: Detection, Estimation and Data Analysis.
Wiley, 1988.
ISBN 0-471-81555-1.

[34] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler.
Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music.
Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR), 2006.

[35] S. Voran.
Listener ratings of speech passbands.
Speech Coding For Telecommunications Proceeding, pages 81–82, September 1997.

Appendix

A Speech Database and Telephone Channel

This appendix concerns the database which has been used for training and test. For a quick overview of the TIMIT database see table A.1. The degrading of speech is presented in section A.2. See figure A.2 for a plot of the magnitude response of the filter used for degrading the speech.

In order to train and test the bandwidth extension algorithm recorded wideband speech is necessary. This appendix clarifies the most important points about the wideband speech database, which is used. Furthermore it discusses how this wideband speech is degraded, such that it can be used in the bandwidth extension algorithm. The last section makes a listing of the files from the database used in the listening test.

A.1 The TIMIT corpus

To be able to model the speech it is important to have a large set of data. This will give a good representation of the distribution of data.

The TIMIT speech database is used. The TIMIT database is a corpus of speech utterances spoken by American's with 8 different accents (DR1-DR8). The database consists of wideband speech recordings, sampled at 16 kHz.

All the information presented about the database originate from [7] and [3].

The database consists of 6300 utterances spoken by 630 speakers (70% male and 30% female). Each speaker spoke two SA sentences, three SI sentences and five SX sentences. SA, SI and SX defines the type of the sentence [7, p. 19]. SA is a dialect sentence. SX are phonetically-compact sentences and were hand designed. They were designed to give a good coverage of pairs of phones. SI sentences originate from existing text sources.

There are 2 SA, 450 SX and 1890 SI sentences. Each speaker read the same two SA sentences. Each SX sentence was read by seven persons. Each SI sentence was only read by one person.

To be able to evaluate the bandwidth extension algorithm some files (utterances) should be reserved for evaluating, such that the evaluation is unbiased of the training. The TIMIT documentation suggests a division of the material into a test and a training part and this partition is adopted in our work. No speakers are present in both the training and test parts. The division between training/test is 73%/27%.

The complete test set consist of 1344 sentence distributed as 8 sentences from 168 speakers (112 males and 56 females). A subset of the complete test set has been suggested and consist of 192 sentences (8 sentences from 24 speakers). The sentences originates from two males and one female per dialect. The test subset is denoted "Core test set".

The training set consist of 3696 sentences (8 sentences from 462 speakers). No sentences with the same text appears in both the training and test material. It should be noted that the two SA sentences from each person are not used. According to [7, p. 23] it might skew the training models since the words contained in the SA sentences would be over-represented.

Table A.1 gives an overview over the division of the TIMIT database for training and testing. The models for extending the envelope is trained using approximately 3hrs of speech.

Database	Training	Test Core	Test Complete	Listening test
Speakers	462 (326/136)	24 (16/8)	168 (112/56)	15 (9/6)
Sentences	3696 (2608/1088)	192 (128/64)	1344 (896/448)	15 (9/6)
Frames (Vectors)	1,131,655	58,269	413,418	
Time	3hrs 8mins	9mins	1hrs 9mins	

Table A.1: Overview of the TIMIT database. The numbers in parentheses denotes the distribution between (Male/female). The number of frames is based on 20 ms frames and 50% overlap. The column listening test is included to indicate the distribution between male and female in the listening test.

A.2 Simulating the telephone channel

The database consists of data recorded with a sampling frequency at 16 kHz. To evaluate the algorithm the signal should be filtered such that it corresponds to transmitting the speech over a telephone channel.

The telephone system was previously an analog system but is nowadays converted into a digital system. The minimum bandwidth for the analog system was specified to be from 300 Hz to 3.4 kHz [22].

After the switch to a digital system the requirements from the analog system are still used. The speech in the digital system is sampled at 8 kHz and the samples are quantized using A-law or μ-law PCM (Pulse Code Modulation) [22]. Since the speech is sampled at 8 kHz the sampled speech does not contain information above 4 kHz. The ITU-T recommendation G.712 deals with transmission performance in PCM channels. It recommends an attenuation on the analog input signal of at least 12.5 dB at 4 kHz [12, Fig. 10]. This means that the speech at the receiver also is attenuated at these frequencies. In the telephone channel it is likely to introduce different attenuation and phase shift depending on frequency. These dependencies will change depending on the telephone network, the country of the network etc. Therefore the same narrowband speech signal transmitted in a different network will appear different at the receiver.

The bandwidth extension algorithm is implemented at the receiver and will therefore

be affected by these influences. In section 1.3 it is chosen to avoid these influences, and assuming a channel with no ripples, phase shift etc.

The narrowband speech signal is obtained from the TIMIT database by either lowpass or bandpass filtering it with a linear filter without ripple. If a bandpass filter is applied, both the band below 300 Hz and above 3.4 kHz are removed. Using a lowpass filter only attenuates the signal above 3.4 kHz.

The operation is seen in figure A.1. The filter is followed by a downsampling of a factor 2. The downsampling assures that the bandwidth extension algorithm can not take advantage of the attenuated signal in the upper band (>4 kHz). Furthermore a sampling frequency of 8 kHz is achieved as in the telephone network.

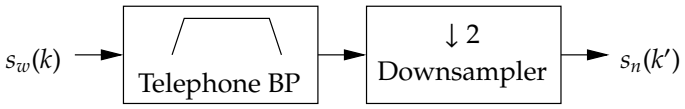


Figure A.1: Block diagram for reducing the bandwidth of the speech signals ($s_w(k)$) from the TIMIT database. The output data ($s_n(k')$) has sample frequency 8 kHz. The discrete indexes k' and k denote 8 kHz and 16 kHz sample frequency respectively.

In section 1.3 it was chosen to only estimate the upperband >3.4 kHz. Therefore the filter in figure A.1 is implemented as a lowpass filter. The filter is chosen as a 300 tap FIR (linear phase) filter with the following specifications:

Stop band	3.7 - 8 kHz
Transition band	3.4-3.7 kHz
Pass band	0 Hz - 3.4 kHz

The magnitude response of the filter is plotted in figure A.2.

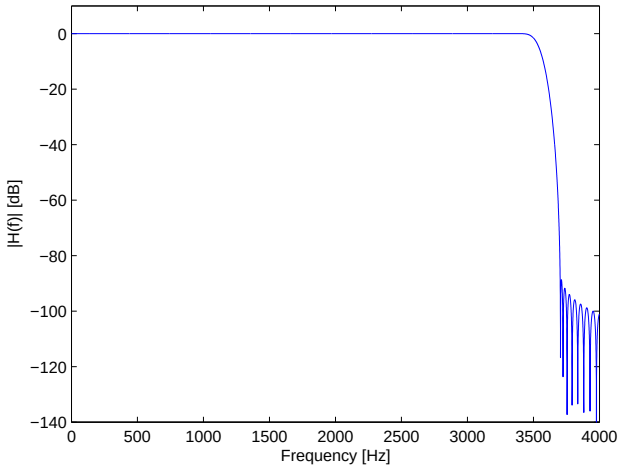


Figure A.2: Magnitude response of the filter for degrading the wideband speech.

A.3 Utterances used in listening test

The files used in the listening test is presented in table A.2, A.4, A.5, A.3 for test 1a, 1b, 1c and 2 respectively. It should be noted that all the files originates from the complete test set. The directories name DRX indicates it is the Xth dialect. The first letter in the subfolder to DRX indicates (M)ale or (F)emale. E.g.

`TEST_COMPLETE\DR3\MTHC0\SI1015.WAV`

means the utterance SI1015 is spoken by a male speaker with dialect number 3 (North Midland).

TEST_COMPLETE\DR3\MTHC0\SI1015.WAV
TEST_COMPLETE\DR4\FJMG0\SX191.WAV
TEST_COMPLETE\DR6\FMGD0\SI2194.WAV
TEST_COMPLETE\DR7\MRMS1\SX137.WAV
TEST_COMPLETE\DR2\MMDM2\SX102.WAV

Table A.2: Files used in test 1a

TIMIT\TEST_COMPLETE\DR3\MTHC0\SI1015.WAV
TIMIT\TEST_COMPLETE\DR4\FJMG0\SX191.WAV
TIMIT\TEST_COMPLETE\DR6\FMGD0\SI2194.WAV
TIMIT\TEST_COMPLETE\DR7\MRMS1\SX137.WAV
TIMIT\TEST_COMPLETE\DR2\MMDM2\SX102.WAV

Table A.3: Files used in test 2

TIMIT\TEST_COMPLETE\DR5\FGMD0\SX143.WAV
TIMIT\TEST_COMPLETE\DR4\FDMS0\SX318.WAV
TIMIT\TEST_COMPLETE\DR3\MRTK0\SI1093.WAV
TIMIT\TEST_COMPLETE\DR6\MESD0\SX372.WAV
TIMIT\TEST_COMPLETE\DR8\MRES0\SI1847.WAV
TIMIT\TEST_COMPLETE\DR3\MTHC0\SI1015.WAV
TIMIT\TEST_COMPLETE\DR4\FJMG0\SX191.WAV
TIMIT\TEST_COMPLETE\DR6\FMGD0\SI2194.WAV
TIMIT\TEST_COMPLETE\DR7\MRMS1\SX137.WAV
TIMIT\TEST_COMPLETE\DR2\MMDM2\SX102.WAV

Table A.4: Files used in test 1b

TIMIT\TEST_COMPLETE\DR3\FKMS0\SX320.WAV
TIMIT\TEST_COMPLETE\DR1\MWBT0\SI1553.WAV
TIMIT\TEST_COMPLETE\DR4\MBNS0\SI1220.WAV
TIMIT\TEST_COMPLETE\DR7\FCAU0\SX47.WAV
TIMIT\TEST_COMPLETE\DR6\MESD0\SI2262.WAV
TIMIT\TEST_COMPLETE\DR3\MTHC0\SI1015.WAV
TIMIT\TEST_COMPLETE\DR4\FJMG0\SX191.WAV
TIMIT\TEST_COMPLETE\DR6\FMGD0\SI2194.WAV
TIMIT\TEST_COMPLETE\DR7\MRMS1\SX137.WAV
TIMIT\TEST_COMPLETE\DR2\MMDM2\SX102.WAV

Table A.5: Files used in test 1c

Appendix

B Verification of Extended Speech

This chapter is included to convince the reader, that the bandwidth extension algorithm do not modify the narrowband region of the extended speech compared to narrowband. Furthermore it visualizes how the extended speech signal looks like compared to the original wideband. The estimation method used in this example, is the proposed perceptual method for estimating the envelope using a HMM with 8 states and 16 mixture components. and spectral translation for extending the excitation signal. Estimation was carried out using an HMM with 16 states and 8 mixture components. Figure B.1 shows a speech signal in the time domain. This signal is plotted together with the original (wideband) envelopes obtained from an LP analysis, the estimated envelopes from the BWE algorithm and the difference between the two. We observe that all the nuances and peaks does not get estimated correct, nevertheless the overall picture matches the plot of the original wideband envelopes fairly well. For the extensionband, the variations in the spectral envelope is very small and the formants are not estimated correctly. On the other hand, no dominating peaks outside the formant frequencies, are introduced in the upper part of the spectrum either. This will probably introduce unwanted artifacts.
The difference shows small variations in the narrowband region. These variations are however constant in each frame and it shows like streaks in the plot. This is the result of slight estimation errors of the relative gain. But as the analysis and synthesis are mutual inverse it only modifies the extensionband slightly.

Figure B.2 shows the same speech signal together with short term power spectra of the narrowband, wideband, estimated wideband and the difference between original and estimated respectively. The plot shows that wideband and estimated wideband are almost identical, which also is supported by the difference plot in the narrowband region. If looking closely, very small variations can be observed in the difference plot in the narrowband region. This is due to aliasing from spectral translation. A small study, in which the estimated speech signal was low pass filtered and listened to, confirmed that it was not possible to hear any audible differences between narrowband and the lowpass filtered version.

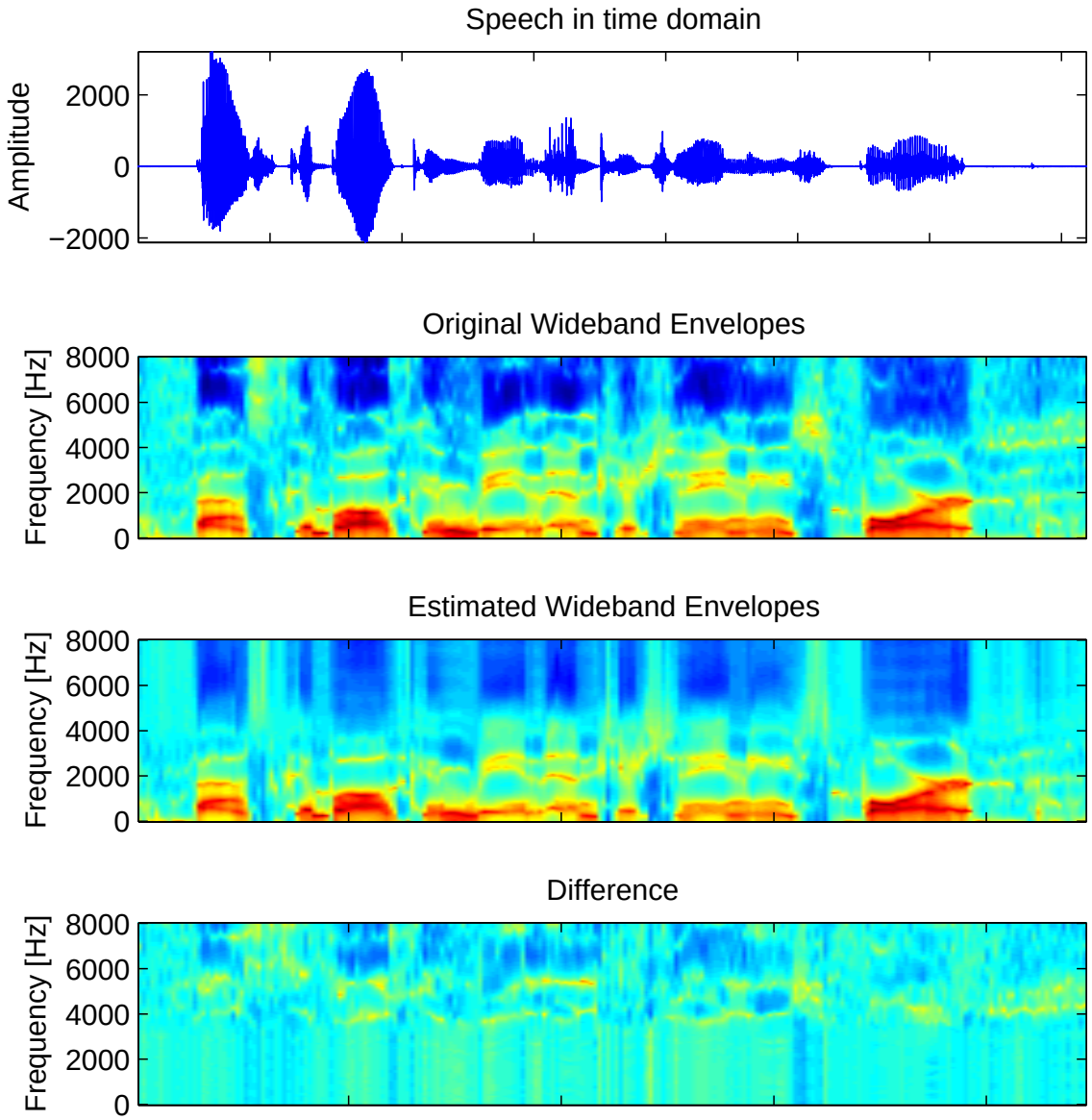


Figure B.1: Speech signal plotted together with original envelopes of an LP analysis, estimated envelopes and the difference between the two. The small varying colors in the narrowband region is a slightly wrong estimation of gain. The speech recording is /DR2/FPAS0/SX44.WAV.

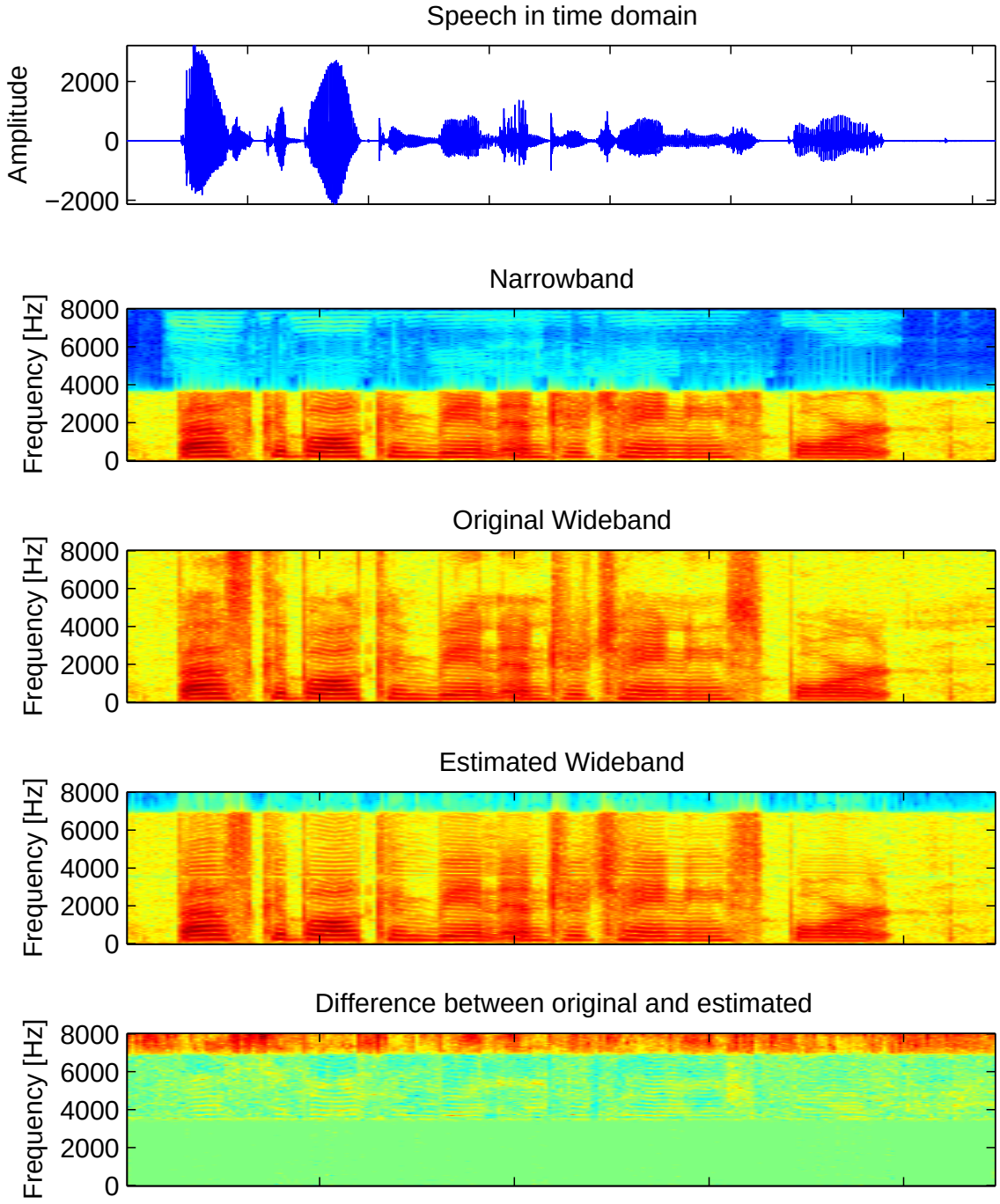


Figure B.2: Speech signal, the corresponding short power spectrum for narrowband, wideband and estimated wideband respectively. The bottom plot shows the difference between estimated and original wideband. The speech recording is /DR2/FPAS0/SX44.WAV.

Appendix

C Listening Test Guide

Test 1

In the first test you have a reference version of speech, and then you have to rate the other versions which have been more or less degraded. The ratings you can make use of are listed in table C.1. The files can be rated from 1 to 5. The rating can be done in steps of 0.25.

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory	Very annoying and objectionable

Table C.1: The different possible ratings in the listening test.

Figure C.1 shows the program which you will use to rank the different versions in test 1.

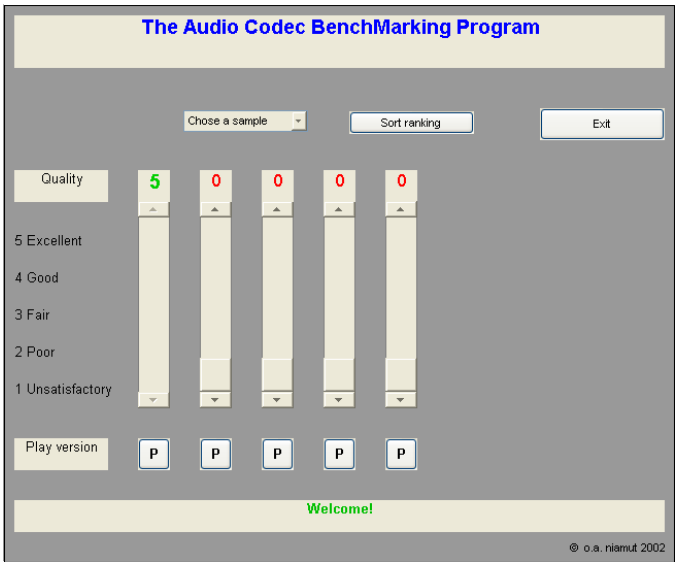


Figure C.1: Screenshot of test 1.

Procedure for Test 1:

1. Choose a sample to rate in the dropdown box.
2. Start the ranking: Begin by listening to the reference (press the "P" button below the left bar). Listen to the remaining versions. Rate the versions based on your preferences according to table 7.2 . If you can not distinguish which is the best of the two versions, rate them equally. You can use the "Sort ranking" button to ease comparison.
3. Continue until you have rated all versions in this sample. Chose the next sample in the dropdown box.
4. Continue with this procedure until all samples have been evaluated.
5. When finished, push the "Exit" button. Please do not close the window, as the results will not be stored.

Test 1 is divided into three subtests; 1a, 1b and 1c. Information in the prompt will guide you through the three tests.

Test 2

In the second test you have a reference and four versions. For each of version you have to specify whether you prefer to listen to the version or the reference. This is indicated by either choosing "Better than reference" or "Worse than reference".
Figure C.2 shows the program which you will be using in test 2.

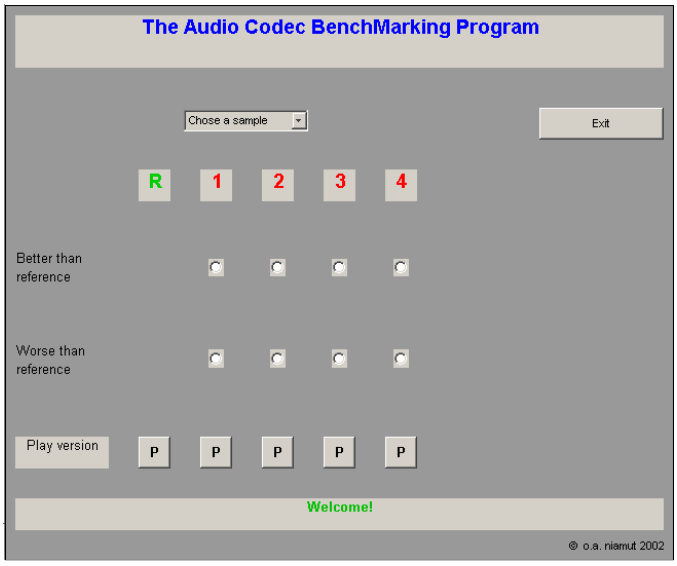


Figure C.2: Screenshot of test 2.

Procedure for Test 2:

1. Choose a sample to rate in the dropdown box.
2. Play the reference and then version 1. Choose if you prefer listening to version 1 or the reference. In doubt take the best guess.

3. Repeat 1. for all versions
4. Continue with this procedure until all samples have been evaluated.
5. When finished, push the "Exit" button. Please do not close the window, as the results will not be stored.

The complete test is estimated to last approximately 20 minutes. Please feel free to ask questions.

Thank you for your help!

Group 1092