HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Laboratory of Acoustics and Audio Signal Processing

Arttu Laaksonen

Bandwidth extension in high-quality audio coding

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Helsinki, May 30, 2005

Supervisor: Professor Vesa Välimäki
Instructor: Pasi Ojala, PhD.

HELSINKI UNIVERSITY OF TECHNOLOGY

ABSTRACT OF THE MASTER'S THESIS

Author:	Arttu Laaksonen
Name of the thesis:	Bandwidth extension in high-quality audio coding
Date:	May 30, 2005 **Number of pages:** 49+13
Department:	Electrical and Communications Engineering
Professorship:	S-89 Acoustics and Audio Signal Processing
Supervisor:	Prof. Vesa Välimäki
Instructor:	Pasi Ojala, PhD.

In mobile telecommunications the transmission speed is currently very limited. Advanced coding methods are used to pack transmitted information into a smaller memory space. The coding of audio signals has developed a lot in the past years, and one method enabling better coding efficiency has been bandwidth extension.

In this thesis, the current bandwidth extension methods are studied, and analysis is made to find out if the methods could be improved. Two new methods have been developed by the author. A method based on modified discrete cosine transform (MDCT) has been used to examine how different parameters affect the result of bandwidth extension. The second method uses linear prediction (LPC) in modeling the properties of audio signals.

The new methods were compared against each other and one previous bandwidth extension method in listening tests. The results of the tests were that the new methods can be used to improve coding efficiency in high-quality audio coding.

Keywords: audio coding, bandwidth extension, spectral band replication, acoustic signal analysis, acoustic signal processing

TEKNILLINEN KORKEAKOULU DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Arttu Laaksonen
Työn nimi:	Kaistanlaajennus korkealaatuisessa audiokoodauksessa
Päivämäärä:	30.5.2005 **Sivuja:** 49+13
Osasto:	Sähkö- ja tietoliikennetekniikka
Professuuri:	S-89 Akustiikka ja äänenkäsittelytekniikka
Työn valvoja:	Prof. Vesa Välimäki
Työn ohjaaja:	TkT Pasi Ojala

Langattoman viestinnän siirtonopeudet ovat tällä hetkellä varsin rajalliset. Kehittyneiden koodausmenetelmien avulla siirrettävä tieto saadaan pakattua pienempään muistiin. Äänisignaalien pakkaaminen on edistynyt paljon kuluneina vuosina, ja yksi tämän mahdollistanut menetelmä on ollut kaistanlaajennus.

Tässä diplomityössä tutkitaan nykyisiä kaistanlaajennusmenetelmiä. Menetelmiä analysoidaan tavoitteena löytää keinoja niiden parantamiseen. Diplomityön tekijä on kehittänyt kaksi uutta menetelmää. Modifioituun diskreettiin kosinimuunnokseen (MDCT) perustuvaa menetelmää on käytetty selvittämään kuinka eri parametrit vaikuttavat kaistanlaajennuksen lopputulokseen. Toinen menetelmä käyttää lineaarista ennustusta (LPC) äänisignaalin ominaisuuksien mallintamiseen.

Uusia menetelmiä vertailtiin toisiinsa ja yhteen olemassaolevaan menetelmään kuuntelutesteissä. Testien tuloksina todettiin, että uusia menetelmiä voidaan käyttää pakkaustehokkuuden parantamiseen korkealaatuisessa audiokoodauksessa.

Avainsanat: audiokoodaus, äänenpakkaus, kaistanlaajennus, äänenkäsittely, äänianalyysi

Acknowledgements

This Master's thesis has been done for Nokia Research Center during the years 2004 and 2005, in a project studying audio coding methods.

First I want to thank Jari Hagqvist and Pasi Ojala for taking me into this project to do my master's thesis. I really could not have thought of a much better place to do the thesis in, and the subject of the thesis was also about just what I wanted.

I wish to thank Mikko Tammi for working with me in the area of bandwidth extension. His comments and especially the handling of the listening tests with Henri Toukomaa has been very valuable for this thesis. My gratitude also goes to our project members for helping with various details of my work.

Of course, I would like to thank my supervisor Vesa Välimäki for the interest in the thesis, and commenting it when needed.

Special thanks to Julia Jakka, so I did not have to be the only new worker here.

Finally, I would like to thank my family for providing me with affordable accommodation during my studies.

Helsinki, May 30, 2005

Arttu Laaksonen

Contents

Abbreviations

AAC	Advanced Audio Coding
ABE	Artificial Bandwidth Extension
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DMOS	Degradation Mean Opinion Score
ERB	Equivalent Rectangular Bandwidth
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IFFT	Inverse Fast Fourier transform
IMDCT	Inverse Modified Discrete Cosine Transform
LSF	Line Spectral Frequencies
MDCT	Modified Discrete Cosine Transform
MPEG	Moving Picture Experts Group
MUSHRA	Multistimulus Test With Hidden Reference and Anchors
SBR	Spectral Band Replication
WMA	Windows Media Audio

Chapter 1

Introduction

Coding in the context of this thesis means packing information into a smaller memory space. How much the space needed by the information is reduced is one of the most important aspects in coding. Other points of concern include for example the processing power and memory required by the coding. As the technology advances coding can be done more efficiently.

With powerful computers and broadband Internet connections the importance of coding efficiency has somewhat diminished, as there is not as much need for reduction in bits required by information, and even computationally heavy coding algorithms can be performed. However, there are some situations where development is still useful. The quality of video distributed through networks is very dependent on the coding used. Usually larger file size provides better quality, but takes more time or bandwidth to be transmitted. Also the coding of unpacked video into smaller space takes still quite a lot of time at least with most home computers, so improvement in that area would also be welcome.

One other area where coding efficiency is currently important is mobile applications. The coding algorithms must be fairly simple because mobile processors are relatively not very powerful, and less processing leads to lesser use of battery. In addition the wireless transmission speed and storage capacity are currently limited. And as for mobile applications, where it usually is cheaper to transmit less information than more, packing the data into small space with efficient methods is beneficial.

Overview of the thesis

This thesis concentrates on developing new methods to improve coding in high-quality audio applications. In chapter 2 background of perceptual audio coding is covered, and one specific approach to the coding, bandwidth extension is presented. Chapter 3 introduces two new methods of bandwidth extension, and how they are implemented in a way that they can

be compared with each other and existing bandwidth extension methods. The comparison is done with listening tests, which are described and their results presented in chapter 4. The new methods and test results are discussed in chapter 5, and finally conclusions are made and future work is planned in chapter 6.

Chapter 2

Perceptual audio coding

In order to give the reader some background information on the work done in this thesis, this chapter first briefly describes some parts of the history of audio coding, and then one specific way to improve the coding, bandwidth extension, is introduced. Some technical terms needed in this thesis are also explained.

2.1 MPEG-1 Audio

In November 1992 Moving Picture Experts Group (MPEG) finalized the international standard ISO/IEC 11172 "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s" [5]. It consists of three parts: system, video and audio. The standard is also known as the MPEG-1 standard.

One part of the standard, MPEG-1 Audio, deals with audio bitrate reduction techniques. It includes three different levels of coding capabilities, Layers I, II and III, of which the Layer III is the most widely known audio compression technique, "mp3".

MPEG-1 Audio bases its audio compression mainly on the properties of the human hearing. The human hearing has a feature that a high-amplitude tone at some frequency may "mask" lower amplitude tones with frequencies close to it, so that the lower tones can't be heard. This is called the masking phenomenon [29]. The MPEG-1 Audio uses this phenomenon as one of the ways how to reduce bitrate.

In an MPEG-1 Audio encoder the input signal is converted to discrete frequency domain. This presentation of the signal is then analyzed, and frequencies where higher amplitudes are masking lower levels of the signal are noted. This information is then used in determining how precisely different parts of the frequency domain are to be quantized — masked regions do not need as good precision as others, as they are not heard as accurately. Therefore less bits can be used in quantizing the masked parts, as the noise caused by more coarse

quantization is not much of a problem. In addition to the masking, MPEG-1 Audio uses several other techniques to reduce the bitrate, such as Huffman coding, these are described more thoroughly for example in [5].

2.2 Improvements after MPEG-1

The MPEG standardization body has continued its work to develop new audio and video coding standards. In 1994 two new standards were defined, MPEG-2 BC (backward compatible) and MPEG-2 LSF (lower sampling frequencies) [4]. The MPEG-2 BC was made as an backward compatible multichannel extension to the MPEG-1 Audio. The MPEG-2 LSF was introduced to enable audio coding at sampling rates of 16, 22.5 and 24 kHz, as the MPEG-1 operated at 32, 44.1 and 48 kHz.

In 1994, development for a new, non-backward-compatible (NBC), audio standard was started. The standard was later named MPEG-2 AAC (advanced audio coding), and it was finalized in 1997. It was formally an extension to MPEG-2 Audio, but in reality a completely new coder that offered a 2:1 improvement in bitrate efficiency when compared to MPEG-1/2 Layer II [10, 14]. However, when compared to the most used coder, MPEG-1 Layer III, the improvement of AAC is not as high [30].

After MPEG-2 AAC, already in 1998, the MPEG-4 General Audio coder (Version 1) was standardized [10]. AAC was used as the base coder, and some new functionalities, concerning for example coding at very low bitrates, were introduced to the standard. Other tools included for example coding natural and synthetic audio objects and composing them into an “audio scene” [28].

Work on Version 2 of MPEG-4 was already ongoing when Version 1 was completed, and the amendment of Version 2 was finalized in 1999 [27]. MPEG-4 Version 2 added again new features to the existing coder, features which were not mature enough at the time of Version 1’s standardization. These features include error resilience, low-delay audio coding, small step scalability, parametric audio coding, and environmental spatialization. More information on these can be found in [28].

Several other audio coders (for example Microsoft’s WMA, RealNetworks’ RealAudio, Ogg Vorbis) have also been introduced in the past years, but they have not been adapted as wide international standards as the MPEG-coders.

Current state-of-the-art coders will be presented briefly later in this chapter.

2.3 Bandwidth extension

2.3.1 Spectral band replication

As MPEG-2 AAC was already coding audio into fairly small space, finding new ways to achieve even lower bitrates while maintaining quality was increasingly difficult. However, in 2001 software containing a new approach to audio coding was released. The new approach was bandwidth extension, and this use of it in the context of audio coding was named Spectral Band Replication (SBR) by its inventors [7]. The basic idea behind SBR is shown in figure 2.1.

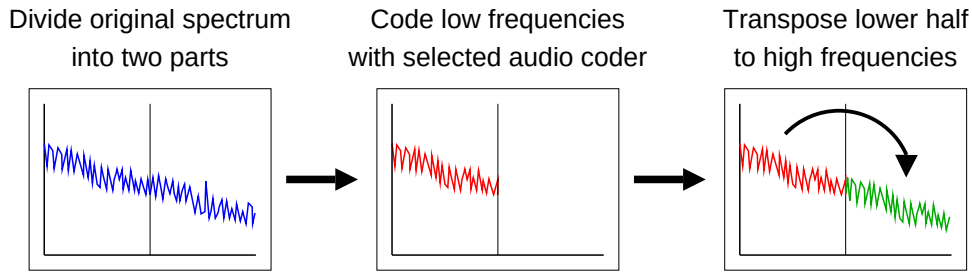

Figure 2.1: How SBR works.

The approach was again based on the properties of the human hearing. The hearing threshold for high frequencies is higher than for lower frequencies (except very low frequencies), so high frequency tones are not heard as loud as the same amplitude tones at lower frequencies [29]. Also the frequency resolution of hearing is better on lower frequencies. On higher frequencies two tones must be relatively far from each other (in frequency) in order to be considered different by the listener.

Another useful feature of many types of audio samples is that the level of the higher frequencies is usually lower than the level of lower frequencies. And finally, the sound of many musical instruments is harmonic, which means that some properties of the frequency spectrum are very similar in lower and higher frequencies. An example frequency spectrum of a very harmonic sound is shown in figure 2.2. The example is from a pitch pipe. The spectrum is mostly harmonic, but there is some noise between the spectral peaks.

In figure 2.3 there is an example frequency spectrum of a non-harmonic sound. This sound is from a sample with castanets.

In both figures 2.2 and 2.3 the spectrum has been calculated from 2048 samples at 32 kHz sampling frequency, so the length of the samples is 64 ms.

In an encoder using SBR only the lower frequencies of the spectrum are coded using conventional techniques (like MPEG-1 Layer III). Only some specific data is extracted from

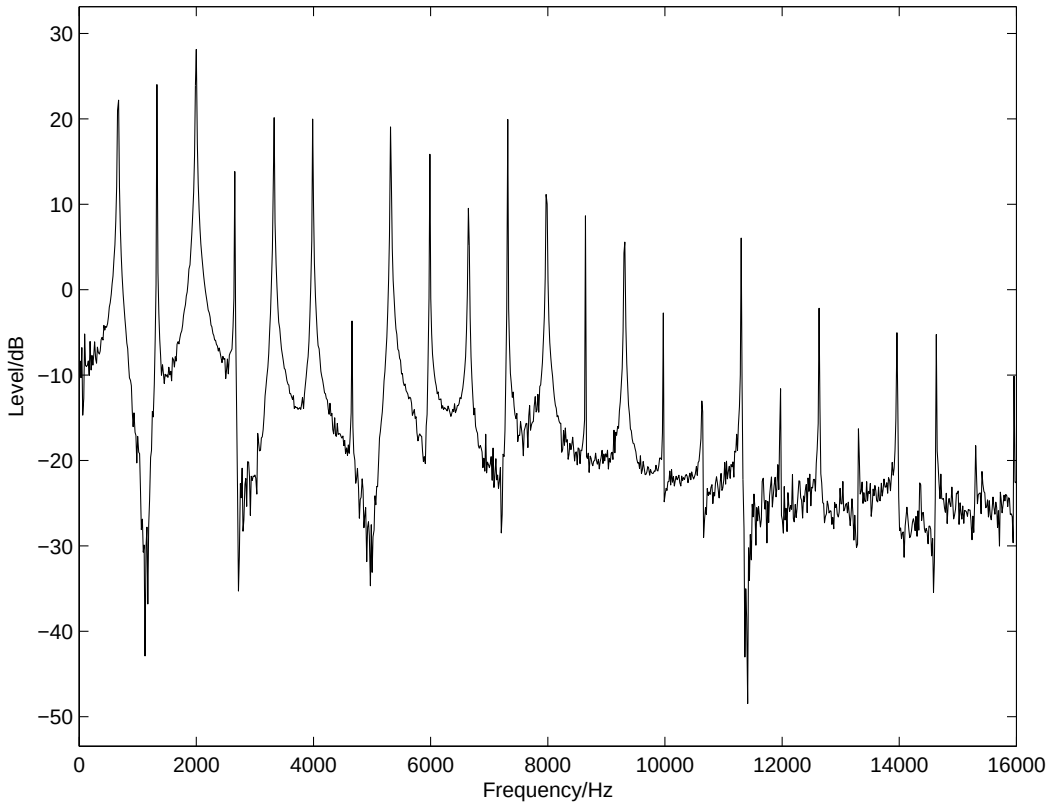

Figure 2.2: Example of a harmonic spectrum.

the higher frequencies. The limiting frequency between the lower and higher frequencies is often chosen to be at the middle of the audio bandwidth (for example at 8 kHz for 32 kHz sampling frequency), but other limit frequencies can be also used. In a simple case the data extracted from the higher frequencies is just the shape of the frequency spectrum. Some other information can be extracted as well, as will be described later. A basic block diagram of the SBR encoder is shown in figure 2.4. In the figure “Core codec” refers to the basic codec upon which the SBR is applied. “Bitstream multiplex” is a component where the data from the core codec and SBR part are joined together, so that they can be transmitted appropriately.

An SBR decoder gets the lower frequencies into its input, along with the additional data. In the decoder the lower frequencies are transposed to the upper frequencies, and using the additional data their level is shaped so that the level of the new upper band is similar to the original upper band. In a case where the signal is very harmonic as in figure 2.2 the method will produce a signal very much like the original, as the spectral peaks from the lower band will replicate the peaks in the original upper band. Also in cases where the original signal

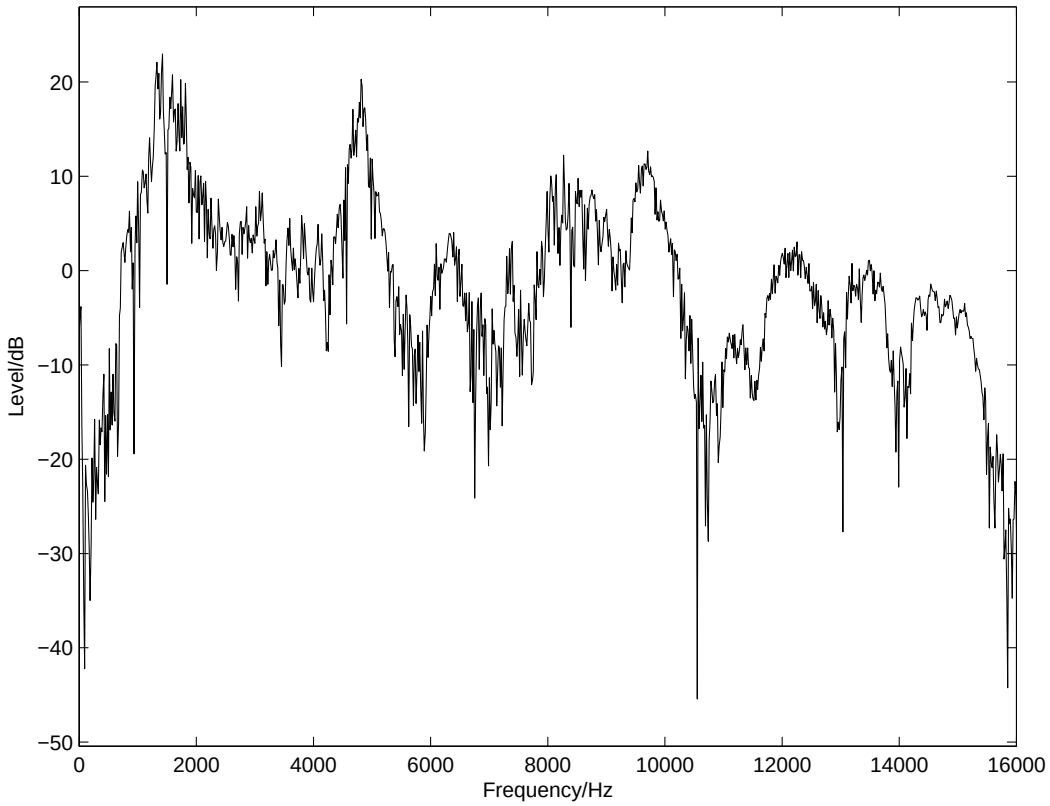

Figure 2.3: Example of a non-harmonic spectrum.

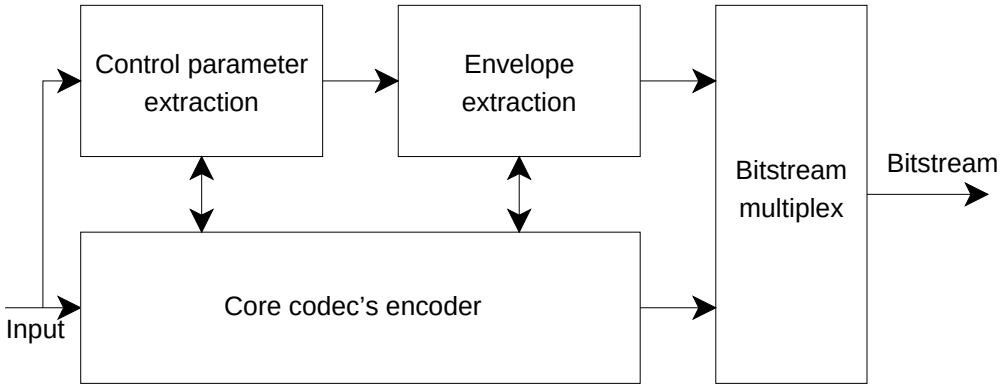

Figure 2.4: SBR encoder [7].

is not harmonic the method can work well, as long as the data from the higher frequencies is detailed enough.

However, in some signals there can be spectral peaks in the upper band, which are not present at the corresponding location at the lower band that gets transposed to the upper

band, so the original peak will not be reproduced. Also it can be so that the original lower band has spectral peaks not present in the original upper band, and transposing those to the upper band would create incorrect characteristics for the signal, even if the level around the peak is adjusted to be correct. The former case can be solved by adding sinusoidal peaks to the upper band near the location of the missing peak. The latter case can be handled by flattening the area in the generated upper band near the unnecessary peak. In SBR the addition of sinusoids is called "sine synthesis" and removing them is "inverse filtering" [8]. Both of these methods require some additional data to be transmitted, at least the location and amplitude of the sinusoidal peaks in the sine synthesis and the area where the flattening is needed in the inverse filtering. A basic block diagram of the SBR decoder is presented in figure 2.5. In the figure "Bitstream demux" is a "de-multiplex" which works oppositely to a multiplex, and separates the data for SBR and the core codec from the transmitted bitstream.

Applying SBR over a normal coder can make it possible to achieve almost equal quality with a bitrate almost half as low as earlier. This assumption is based on a case where coding the lower band takes half the amount of bits compared to normal situation, and the SBR part only needs a few bits. However, in many cases the quality will not be equal, and the normal coders usually have more bits allocated to the lower band of the sound signal, where most of the information of the sound often is, and dropping the upper band from normal coding does not really reduce the bits as much as one might imagine. In any case SBR can be used to reduce the bitrate somewhat while achieving equal quality, or getting better quality at the same bitrates. The latter is made possible by having more precision to code the lower band normally when the upper band needs less bits. Test results for applying SBR over a normal coder can be found for example from [7], [11] and [31].

SBR applied over "mp3" is called mp3PRO [34]. mp3PRO is backwards compatible with existing mp3-decoders, which means that the files encoded using mp3PRO can be also played on an older decoder. However, the older decoder can only play the traditionally encoded lower band, and the SBR-coded upper band is completely left out, so the sound quality will be far from optimal.

MPEG-4 AAC with SBR has been named MPEG-4 High Efficiency AAC (HE AAC) [33]. It is nowadays also called aacPlus [9]. The latest development in the MPEG audio coder branch is Enhanced aacPlus, which adds parametric stereo functionality to aacPlus [1]. Enhanced aacPlus is a very efficient audio coder and can be used for example at multimedia messaging and mobile streaming of audio.

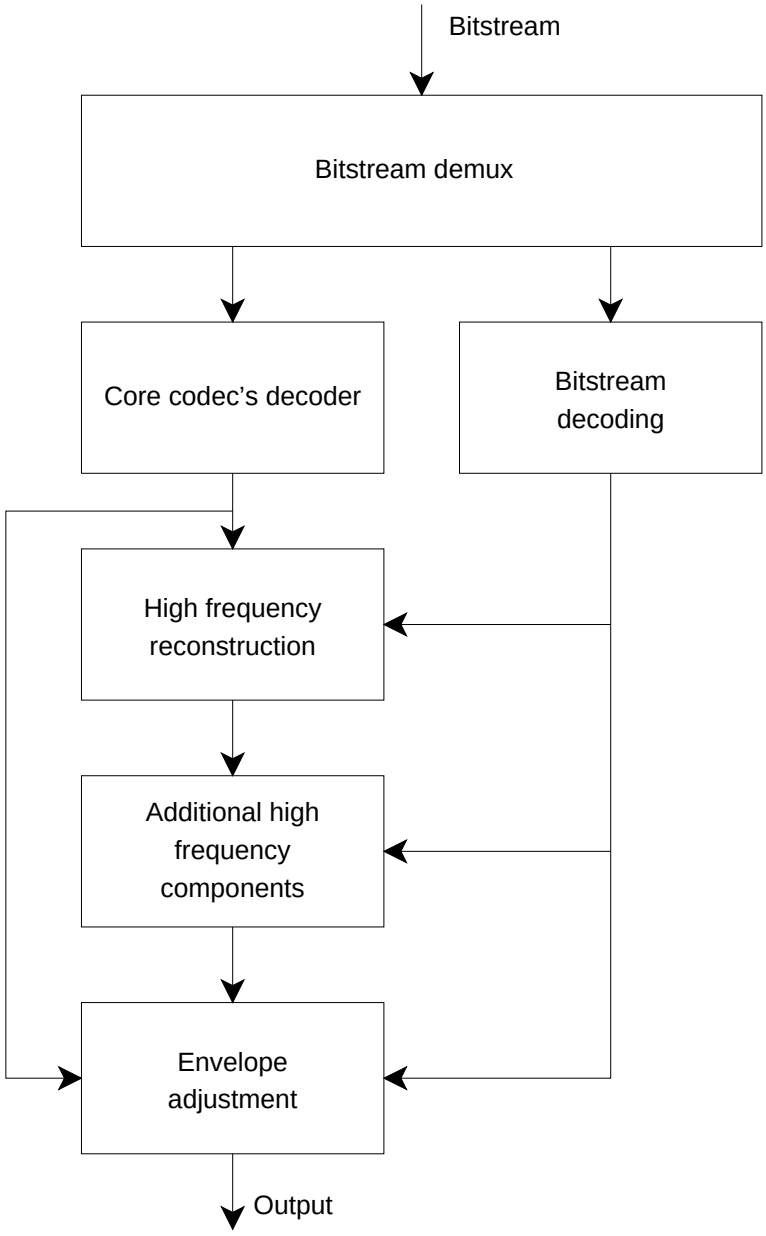

Figure 2.5: SBR decoder [7].

2.3.2 Other uses of bandwidth extension in audio or speech coding

"Blind" bandwidth extension

SBR is not the only way to perform bandwidth extension for an audio signal. A different method has been introduced for example in [22]. The method is fundamentally different from SBR in the sense that the decoder does not require any information known from the

upper band of the original signal. This means that the method does not even need anything specific done in the encoder, and can be applied to any signal. A block diagram for the method is shown in figure 2.6.

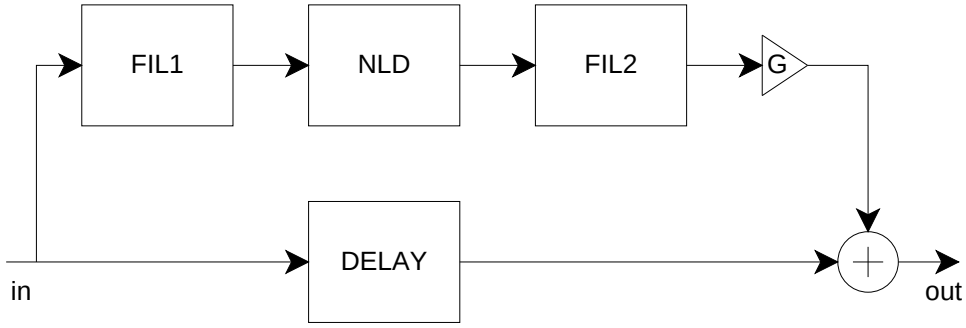

Figure 2.6: Another method for high-frequency bandwidth extension [22].

In the diagram `FIL1` is a bandpass filter picking the lower frequencies from the input audio signal. As the method assumes that the higher frequencies should be missing from the signal, the filter removes any possible unwanted material from the upper band. `NLD` is a non-linear device, which performs most of the bandwidth extension in this method. The non-linear device can be for example a full-wave rectifier, which is a component that outputs the absolute value of the input signal, usually causing non-linearities for signals that have both positive and negative values. The non-linearities generally mean unwanted frequency components around the original signal. Some of these components should become created to the missing upper band, and as they originate from the original signal, they can be used as additional high-frequency components in the method. Another bandpass filter `FIL2` is used to pick only the high-frequency components from the signal, removing generated lower frequency components. Scaling component `G` is used to adjust the level of the new upper band to be suitable for using with the orignal lower band. Finally the upper band is summed with the original signal, which has been delayed for some small time that the high-frequency processing components use during processing.

As the method does not know anything of the original upper band, the generated upper band will be just some random high-frequency components in most cases, so it is not very suitable for high-quality audio coding. However, the sound may be perceptually pleasant, and many listeners may find it to be better that just the original lower frequencies. The method also has low computational complexity, so it does not need much processing power from the decoder.

The authors of this method have also published a method which can extend bandwidth for low frequencies as well [2], but in this thesis the aim is to study high-frequency bandwidth extension, so the method will not be described here. More low-frequency bandwidth

extension methods can be also found from [21].

An artificial bandwidth extension (ABE) algorithm

An algorithm called ABE has been first presented in [23], further developed in [20] and [18].

ABE is meant to be used for speech signals, and it can extend the higher frequencies without knowledge of the original upper band. However, the method takes advantage from analyzing the lower band and trying to know what type of phoneme is present in the current part of speech being processed. The method uses this information to shape the upper band, which has been generated from the transmitted lower band.

The method is computationally quite simple, and can be applied to any existing speech decoder, as it does not need any side information from the original signal. However, in a listening test the test subjects rated non-expanded samples slightly higher than samples processed with ABE [18], so some more work is needed to make this method more useful.

2.4 Some technical terms explained

The reader is assumed to know the basics of signal processing and related things, but some more audio coding specific terms will be needed to understand the thesis. The most important ones are briefly explained next.

Discrete frequency domain

As bandwidth extension mostly works with the frequency information of a signal, working in time domain is not suitable for the methods. Therefore the time domain signal is often transformed into another domain, one of which is the discrete frequency domain. The transform to this domain is usually made with Fourier transform, especially with its discrete version (DFT). The transform is defined as follows [12]:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \text{ for } 0 \leq k < N-1. \tag{2.1}$$

The transform is defined for finite length sequences (N). The inverse transform goes as follows [12]:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \text{ for } 0 \leq k < N-1. \tag{2.2}$$

For transforming the time domain signals into the discrete frequency domain there are very efficient computational algorithms called the fast Fourier transform (FFT), and its inverse, IFFT. Such algorithms are described for example in [26].

MDCT

Modified discrete cosine transform (MDCT) is in this work used to transform sound signals from time domain to a domain better suitable for bandwidth extension, here named as the MDCT-domain. The MDCT-domain is a frequency presentation of a signal, as is the discrete frequency domain, and is ofter used in audio coding. The transform is defined as follows [13]:

$$X_{i,k} = 2 \cdot \sum_{n=0}^{N-1} z_{i,n} \cos\left(\frac{2\pi}{N}\left(n + n_0\right)\left(k + \frac{1}{2}\right)\right) \text{ for } 0 \leq k < N/2, \tag{2.3}$$

where:

$X_{i,k}$ = MDCT spectral coefficient
$z_{i,n}$ = windowed time domain input sequence
n = sample index
k = spectral coefficient index
i = block index
N = window length of the one transform window
n_0 = $\left(N/2 + 1\right)/2$.

To transform the signal from MDCT-domain back to the time domain, an inverse transform, IMDCT, is needed. It is defined as follows [13]:

$$x_{i,n} = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} X_{i,k} \cos\left(\frac{2\pi}{N}\left(n + n_0\right)\left(k + \frac{1}{2}\right)\right) \text{ for } 0 \leq n < N, \tag{2.4}$$

where:

$X_{i,k}$ = MDCT spectral coefficient
n = sample index
i = window index
k = spectral coefficient index
N = window length of the one transform window
n_0 = $\left(N/2 + 1\right)/2$
$x_{i,n}$ = time domain signal values.

"Normal" discrete cosine transform (DCT) is often used in signal compression, and not only for audio signals [26, 27]. Advantages of MDCT in audio coding are presented for example in [32].

Linear prediction

Linear prediction is a way to model spectral characteristics of a signal. Linear prediction itself means predicting the next value of a signal from the immediately preceding values, by multiplying them with a specific prediction coefficients. These coefficients can be calculated by various methods, so that the error in the prediction is as small as possible [24].

If the prediction coefficients are calculated for a signal or a signal frame, they can be also used to generate a filter whose impulse response models the spectral envelope of the signal. This can be useful in bandwidth extension, where such information of a signal can be highly valued. An example LPC envelope calculated for a signal can be found in figure 2.7. The signal frame in the example is from a speech signal. In the next chapter a BWE method using linear prediction will be described, and in that method the level of the spectral envelope is adjusted closer to the mean level of the signal, to give needed information from the mean level of the signal.

Linear prediction is widely used in speech processing, as described for example in [6].

Line spectral frequencies

Line spectral frequencies (LSFs) are linear prediction coefficients converted into another form. Usually the coefficients are statistically concentrated around zero, being both positive and negative, and not having a lower or upper limit. This is not optimal situation when considering the quantization of the coefficients. Using calculations presented for example in [3], the coefficients can be transposed to range between 0 and π, which is much more efficient to quantize.

Transient detection

In sound signal analysis there is often need for two things. The frequency resolution in the analysis is wanted to be as precise as possible, which is achieved by analysing long segments of the signal. Also the time resolution is important in the analysis, which in contrast demands short parts of signal, so both of these requirements may not be possible at the same time.

In the BWE methods described in the next chapter the analysis is mainly done on frames with 2048 samples. However, this does not give enough time resolution for all situations, so transient detection is used to find such cases.

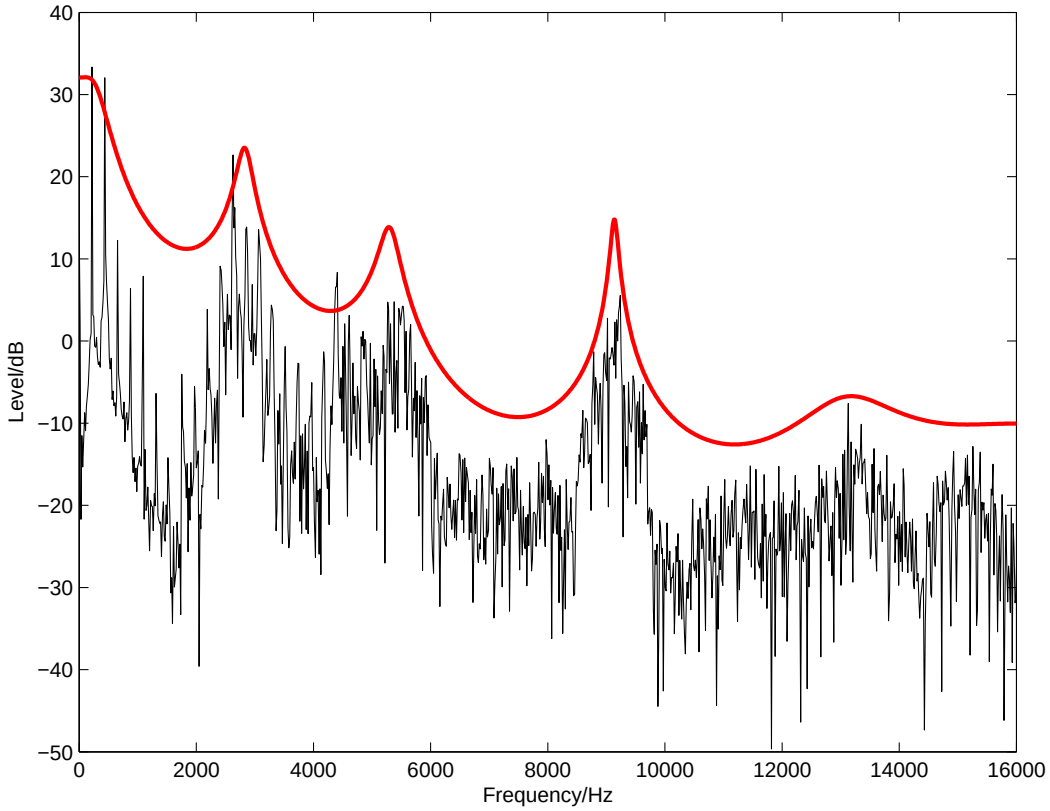

Figure 2.7: Example LPC envelope.

The transients are short sharp variations of sound, and they require slightly different processing than other parts of the sound signal. For detecting the transients, the input signal is high-pass filtered. If the energy of the high-pass signal changes significantly from one window to another, it is concluded that a transient has occurred. The information is used to decide what length and type of windows and frames should be used for the signal.

In figure 2.8, the top figure is an example of a transient occuring during a signal (a castanet hit), and at the bottom is a fairly stationary signal (vowel /i/ from a speech signal), which can be fairly well analysed using long frames.

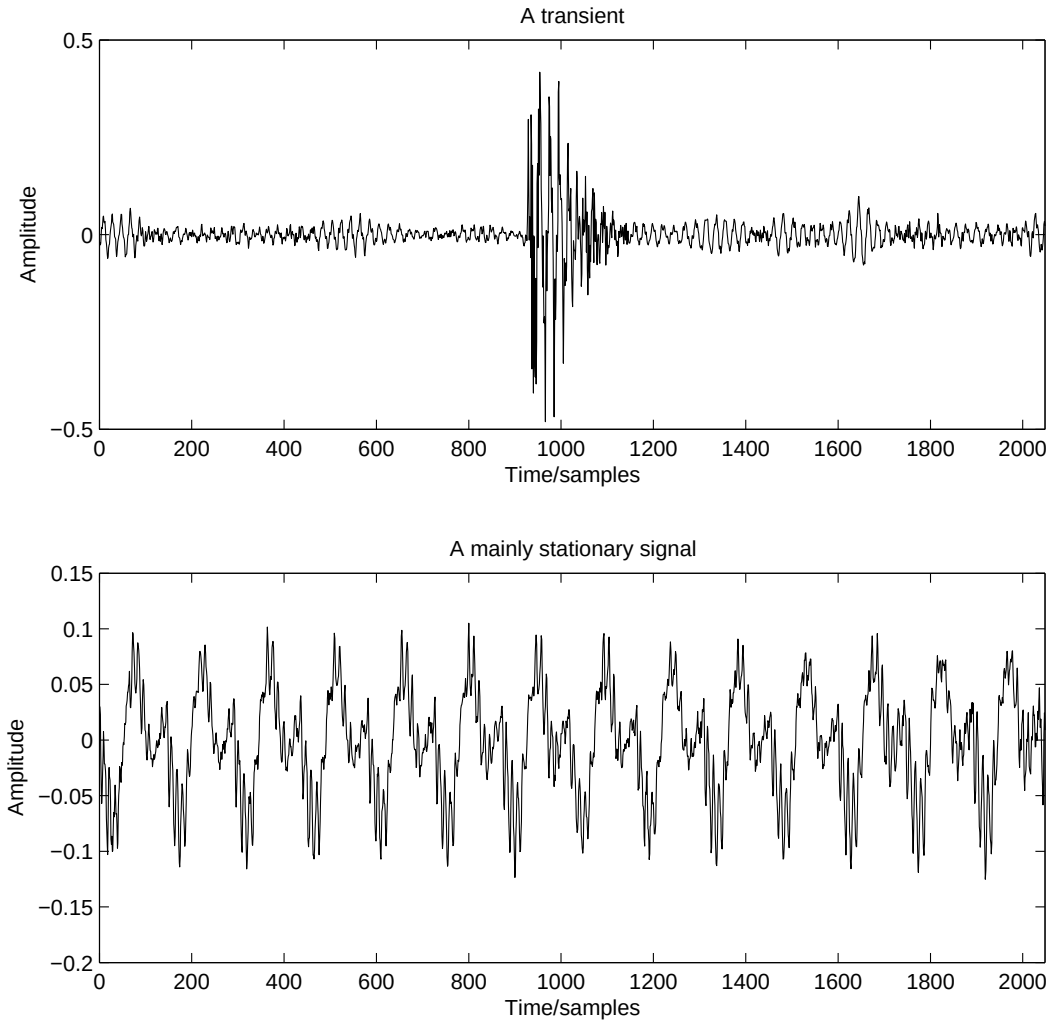

Figure 2.8: Example of a transient and a mainly stationary signal, in time domain.

Chapter 3

New bandwidth extension methods

The objective in the work upon which this thesis is based on was to study current bandwidth extension methods, try to find out if they could be done better, and also create new methods if ideas for those would come up. As several different methods were expected to be tested, a general framework for processing the methods was created first. All methods were processed similarly, only the part where bandwidth was extended was different for the methods. In this chapter the general processing will be described first, then the two different extension methods developed by the author, the modified discrete cosine transform –based (MDCT) method and the linear predictive coding –based (LPC) method. In the MDCT-based method, the input signal is transformed to the MDCT domain, the signal's upper frequency band is divided into subbands, and their energies are calculated and afterwards used to generate the synthesized upper band. In the LPC-based method the shape of the upper band is calculated and transferred as LPC-coefficients, which are then used to generate the new upper band.

3.1 General processing

This section describes the general processing part of the developed bandwidth extension framework.

3.1.1 Encoder

A block diagram for the general processing is shown in figure 3.1.

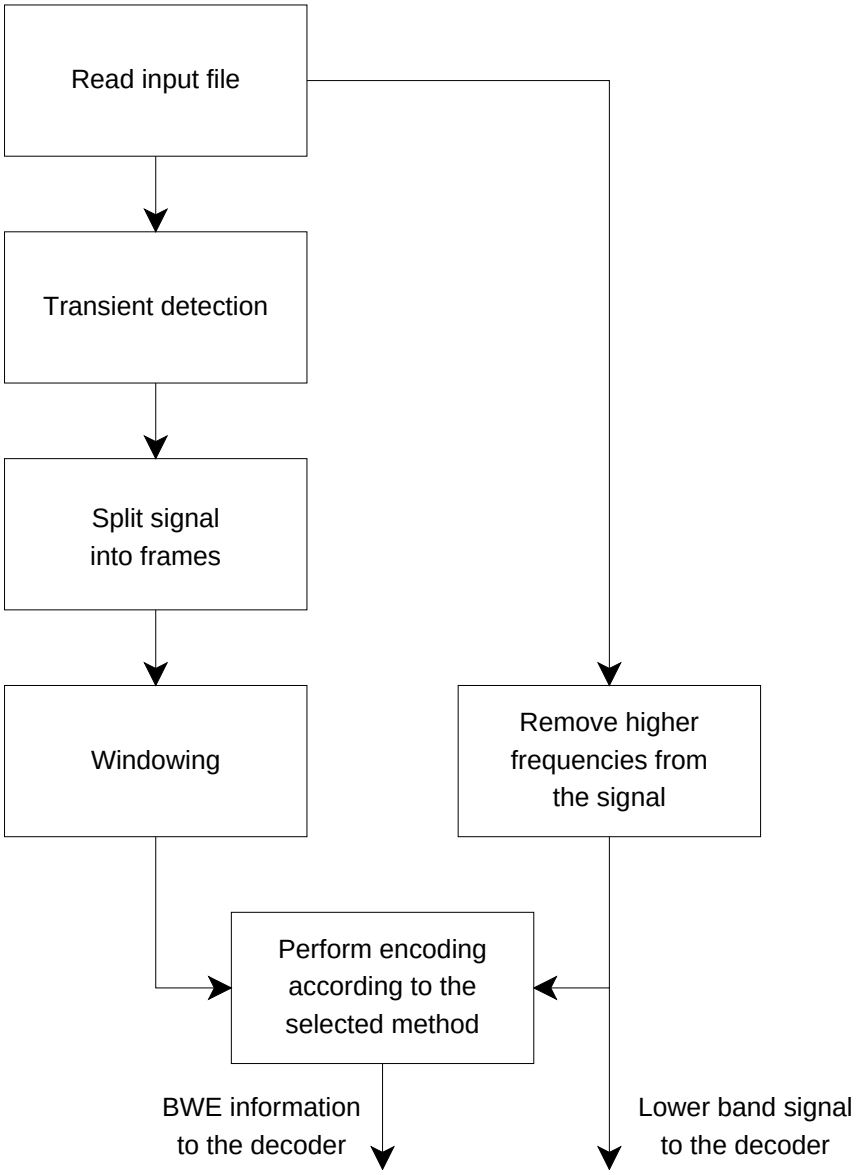

Figure 3.1: General processing in the encoder.

BWE limit frequencies other than middle of the audio bandwidth

The methods presented in this chapter always work so that the limit frequency where the bandwidth extension begins is in the middle of the audio bandwidth. Sometimes it might be desired that the limit frequency is somewhere else. The methods can be made to work with higher limit frequencies by upsampling the input signal to a sampling frequency for which the limit frequency is now in the middle of the audio bandwidth. Doing this the methods work internally at a higher sampling rate than what the original rate was. The procedure is

reversed after decoding by downsampling back into the original sampling rate.

Transient detection and frame division

In audio coding the input signal is usually first divided into frames before it is studied and processed. The frame division and windowing used here are the same as in MPEG-4 Audio [13], but some parts have been simplified a bit.

However, before frame division transient detection is done for the input signal. The detection gives information about the location of the transients in the signal, and is briefly described in section 2.4. After the detection, the input signal is divided into 50% overlapping frames with length of 2048 samples.

Lower band signal to the decoder and comparison signal

Before continuing with the frames, a new signal is generated by removing the upper band frequency information from the input signal. This is done by first downsampling the input signal by a factor of 2, and then upsampling it by the same factor. Aliasing cancellation filters are used in both resamplings. The result from this processing is a signal with the original sampling frequency, but with just the lower frequencies.

The new signal also divided into 50% overlapping frames. It is also windowed and transformed into the MDCT-domain, using equation 2.3. Reason for generating the signal is that the decoder needs a signal which has only the lower band frequencies. The modified signal resembles a signal which in real situation would be input to the decoder.

The signal can also be used in the encoder, if some specific methods are used in the processing. These can include performing the bandwidth extension already in the encoder, and comparing the acquired signal with the original, and calculating some parameters based on the result.

Windowing

The frames are next windowed. The windows used are sinusoidal windows [13]. The window length N can be 2048 or 256. Longer windows are used for normal frames, and shorter windows for frames which contain transients. There are also specific windows for transitions from normal windows to short windows and back. The windows are defined as follows:

$$W_{SIN\_LEFT,N}(n) = \sin\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right) \quad \text{for} \quad 0 \leq n < \frac{N}{2} \tag{3.1}$$

$$W_{SIN\_RIGHT,N}(n) = \sin\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)\right) \quad \text{for} \quad \frac{N}{2} \leq n < N. \tag{3.2}$$

A 2048 samples long window for a normal frame which is preceded and followed by a frame of same type, is defined as follows:

$$W(n) = \begin{cases} W_{SIN\_LEFT,2048}(n), & \text{for} \quad 0 \leq n < 1024 \\ W_{SIN\_RIGHT,2048}(n), & \text{for} \quad 1024 \leq n < 2048 \end{cases}. \tag{3.3}$$

Time domain values after windowing ($z_{i,n}$) for all 2048 samples long window types can be expressed as:

$$z_{i,n} = W(n) \cdot x'_{i,n}. \tag{3.4}$$

A 2048 samples long window which is preceded by a normal frame and followed by a transient frame, is defined as follows:

$$W(n) = \begin{cases} W_{SIN\_LEFT,2048}(n), & \text{for} \quad 0 \leq n < 1024 \\ 1.0, & \text{for} \quad 1024 \leq n < 1472 \\ W_{SIN\_RIGHT,256}(n), & \text{for} \quad 1472 \leq n < 1600 \\ 0.0, & \text{for} \quad 1600 \leq n < 2048 \end{cases}. \tag{3.5}$$

A 2048 samples long window which is preceded by a transient frame and followed by a normal frame, is defined as follows:

$$W(n) = \begin{cases} 0.0, & \text{for} \quad 0 \leq n < 448 \\ W_{SIN\_LEFT,256}(n), & \text{for} \quad 448 \leq n < 576 \\ 1.0, & \text{for} \quad 576 \leq n < 1024 \\ W_{SIN\_RIGHT,2048}(n), & \text{for} \quad 1024 \leq n < 2048 \end{cases}. \tag{3.6}$$

Finally, the frames containing transients are windowed in the following fashion:

$$W(n) = \begin{cases} W_{SIN\_LEFT,256}(n), & \text{for} \quad 0 \leq n < 128 \\ W_{SIN\_RIGHT,256}(n), & \text{for} \quad 128 \leq n < 256 \end{cases}. \tag{3.7}$$

A transient frame contains 2048 samples as the other frames do. However, the transient frame is processed quite differently. The middle part of the 2048 samples are divided into eight overlapping segments, and windowed as follows:

$$z_{i,n} = \begin{cases} x'_{i,n+448} \cdot W(n), & \text{for} \quad 0 \leq n < 256 \\ x'_{i,n+576} \cdot W(n-256), & \text{for} \quad 256 \leq n < 512 \\ x'_{i,n+704} \cdot W(n-512), & \text{for} \quad 512 \leq n < 768 \\ x'_{i,n+832} \cdot W(n-768), & \text{for} \quad 768 \leq n < 1024 \\ x'_{i,n+960} \cdot W(n-1024), & \text{for} \quad 1024 \leq n < 1280 \\ x'_{i,n+1088} \cdot W(n-1280), & \text{for} \quad 1280 \leq n < 1536 \\ x'_{i,n+1216} \cdot W(n-1536), & \text{for} \quad 1536 \leq n < 1792 \\ x'_{i,n+1344} \cdot W(n-1792), & \text{for} \quad 1792 \leq n < 2048 \end{cases} . \tag{3.8}$$

After windowing, the eight short segments are distributed evenly on the 2048 samples long frame, so that they no longer overlap. This enables the BWE encoding function to easily process each of the short frames.

An example sequence of frames and corresponding windows is presented in figure 3.2. From left to right first is a normal frame, then a transition frame from normal frames to short frames, eight short frames, transition frame back to normal frames, and two more normal frames. The 50% overlap of the frames is also seen in the figure.

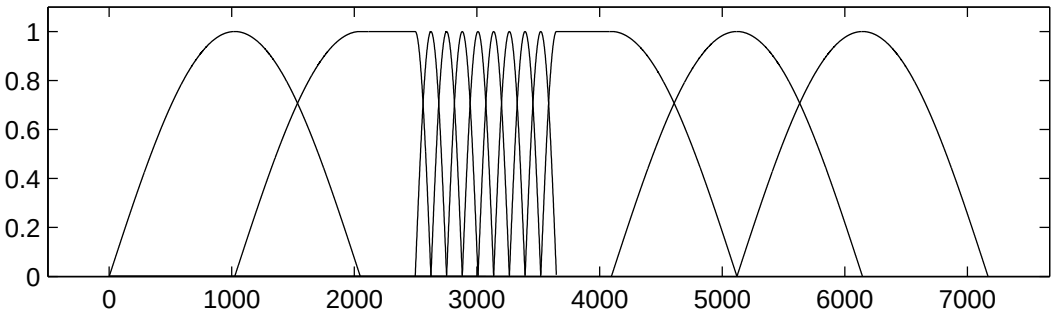

Figure 3.2: Example window sequence.

After windowing, the frames are passed to the specific BWE encoder part.

3.1.2 Decoder

After being processed in the BWE decoder, the frames still have to be treated so that they form a sound signal as the original input signal was. A block diagram for the general processing of the decoder is shown in figure 3.3.

Windowing

The windowing is again similar as in [13]. For long frames of the 3 different situations the same windows are used as in previous section (3.1.1). Also the time domain values are

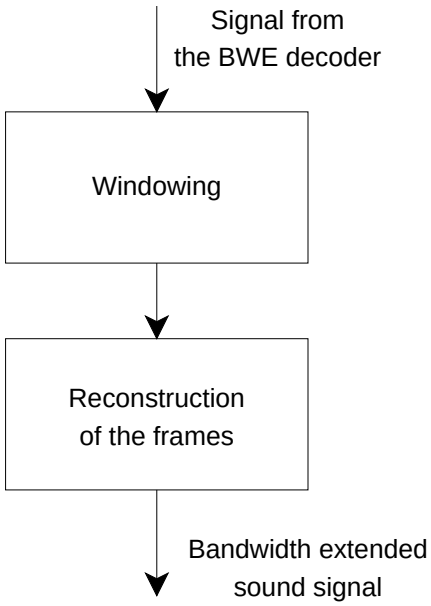

Figure 3.3: General processing in the decoder.

calculated with equation 3.4.

For transient frames the windowing in the decoder is not as straightforward. The BWE part of the decoder outputs the short frames as they were in the input, non-overlapping. The general processing in the decoder requires the short frames to be again overlapping in the middle part of the 2048 samples long frame. The short frames are windowed individually with the same windows as in the previous section, and overlap-and-added as follows:

$$z_{i,n} = \begin{cases} 0, & \text{for} \quad 0 \leq n < 448 \\ x_{i,n-448}, & \text{for} \quad 448 \leq n < 576 \\ x_{i,n-448} + x_{i,n-320}, & \text{for} \quad 576 \leq n < 704 \\ x_{i,n-320} + x_{i,n-192}, & \text{for} \quad 704 \leq n < 832 \\ x_{i,n-192} + x_{i,n-64}, & \text{for} \quad 832 \leq n < 960 \\ x_{i,n-64} + x_{i,n+64}, & \text{for} \quad 960 \leq n < 1088 \\ x_{i,n+64} + x_{i,n+192}, & \text{for} \quad 1088 \leq n < 1216 \\ x_{i,n+192} + x_{i,n+320}, & \text{for} \quad 1216 \leq n < 1344 \\ x_{i,n+320} + x_{i,n+448}, & \text{for} \quad 1344 \leq n < 1472 \\ x_{i,n+448}, & \text{for} \quad 1472 \leq n < 1600 \\ 0, & \text{for} \quad 1600 \leq n < 2048 \end{cases} . \tag{3.9}$$

So in the overlap-and-add the second half of a short frame is added with the first half of

the next short frame (as they overlap each other), for all short frames except the first half of the first short frame and the second half of the last short frame. The modified frames are placed around the middle of the 2048 samples long frame.

Reconstruction of the frames

Now the decoded signal is in overlapping frames, quite like in figure 3.2. To get a usable continuous sound signal, the frames need to be overlap-and-added. As described in [13], the operation goes as follows:

$$out_{i,n} = z_{i,n} + z_{i-1,n+\frac{N}{2}} \text{ for } 0 \leq n < \frac{N}{2}, N = 2048. \tag{3.10}$$

As the overlap-and-add used for the eigth short frames, equation 3.10 sums the first half of a frame with the second half of the next frame. The result from the output is a completely processed bandwidth extended audio signal.

3.2 MDCT-based extension method

The modified discrete cosine transform –based extension method was built to be much like the model of the SBR, but different types of copying the frequency spectrum and adjusting the frequency envelope were developed.

3.2.1 Encoder

A block diagram for the MDCT-based extension method's encoder is shown in figure 3.4.

The encoder receives windowed time domain frames one at a time from the general processing part, and some control parameters (the number of subbands to do energy calculation and the frequency scale being used). In the method's encoder the frames are first converted to MDCT-domain with equation 2.3. In the MDCT-domain the frequency information of the signal is practicably available. The MDCT-domain is also good presentation of the signal, as it only requires the same amount of discrete spectral values as the original signal has time domain values, whereas the Fourier transformed signal would have double the amount of values, because discrete frequency domain also includes the phase values of the signal. The upper band of the MDCT information is then divided into subbands.

The subbands can have either uniform width, so that all of them have equal number of MDCT values, or they can be divided according to the equivalent rectangular bandwidth (ERB) scale [25]. The ERB scale models the analysis bandwidth of human hearing, which affects for example the masking phenomenon described in section 2.1. In the MDCT-based method the scale was chosen to see if dividing the subbands according to the scale would

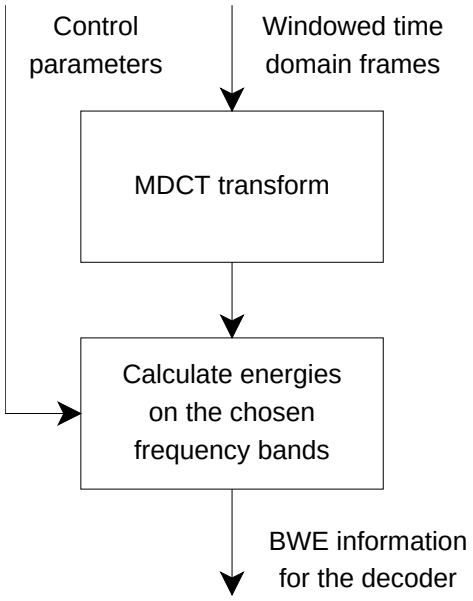

Figure 3.4: MDCT-based method encoder.

have effect on results of the bandwidth extension. The bandwidth of an ERB in Hz on a center frequency F in kHz can be calculated with the following equation [25]:

$$ERB = 24.7(4.37F + 1). \tag{3.11}$$

Applying equation 3.11 the borders of the ERBs can be calculated, and the subbands in the MDCT-domain will be placed accordingly.

If the ERB scale was chosen, the number of subbands is selected according to how many ERBs the calculation found for the upper band. For example 32 kHz and 44.1 kHz sampling rates have seven subbands, and 48 kHz has eight. When using uniform width subbands, the number can be chosen. For example 8, 16 or 32 subbands can be used.

After deciding the subbands, the energies e of the MDCT-values $m(n)$ are calculated separately on each of the subbands as follows:

$$e = \sqrt{\sum_{n=0}^{N-1} m(n)^2}. \tag{3.12}$$

In equation 3.12 N is the length of the subband. The subbands start from the middle of the MDCT-transformed frame, so the border frequency for the BWE is at the center of the audio band of the original signal.

The output from the MDCT-based method's encoder is just the energies on the MDCT subbands.

3.2.2 Decoder

A block diagram for the MDCT-based extension method's decoder is shown in figure 3.5.

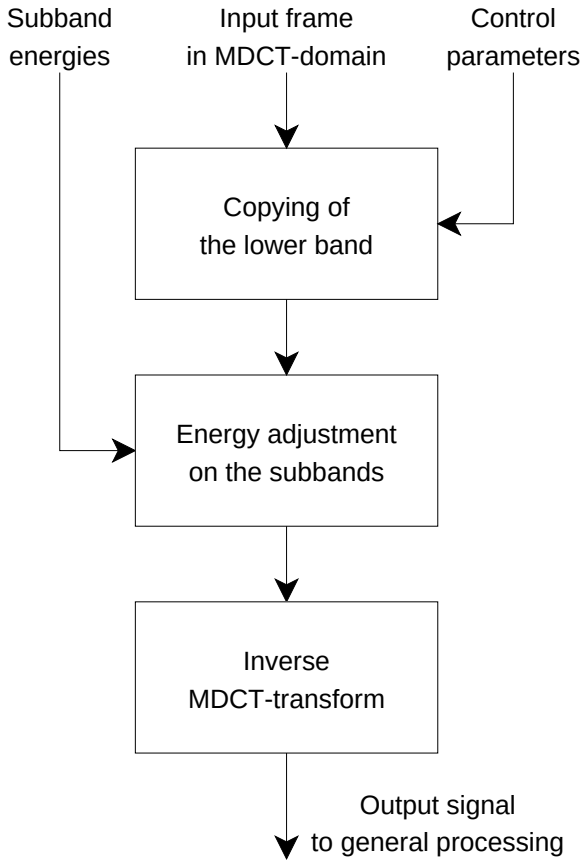

Figure 3.5: MDCT-based method decoder.

The decoder receives transmitted signal in MDCT-form without any upper band information, subband energies the encoder has calculated, and some control parameters (the copying type of the extension and the frequency scale being used). First the decoder copies the lower band to the upper band. This can be done in four different ways, which are shown in table 3.1.

After copying, the upper band is again divided into subbands, same as were used in encoding the corresponding frame. The energies in the copied upper band are calculated as in the encoder, using equation 3.12. The copied upper band must still be modified so that the levels on the subbands are same as in the original upper band. This is done by multiplying every MDCT value with the ratio of the original and copied energies, separately on each of the subbands:

Table 3.1: Copying types for the MDCT-based method.

Copying type	Description
`full`	The lower band is copied completely
`half`	The upper half of the lower band is copied twice one after another
`mirror_full`	The lower band is copied completely, but mirrored horizontally
`mirror_half`	The upper half of the lower band is copied twice mirrored

$$m_{adjusted}(n) = m_{copied}(n) \cdot \frac{e_{original}}{e_{copied}} \text{ for } 0 \leq n < N-1, \tag{3.13}$$

where N is the number of MDCT values on the corresponding subband. Note that when using uniform width subbands N is same for every subband, but with the ERBs the width of the subbands (and N) changes according to the center frequency of the subband.

After the adjustment, an inverse MDCT is performed for the frames using equation 2.4. Resulting time domain frames are passed to the general processing part of the decoder.

Example figures of signals extended with the MDCT-based method can be found in appendix A. The figures include the frequency spectrum of the same frame coded with different number of subbands and different type of copying. There are also time domain figures of a transient and a mainly stationary signal, and the frequency spectrum of a situation where the MDCT-based method has some problems.

3.3 LPC-based extension method

In the linear predictive coding –based method the focus of study was on how well a set of coefficients could be used to model the frequency envelope of a signal, and then used to generate a synthesized upper band.

3.3.1 Encoder

A block diagram of the LPC-based method encoder is shown in figure 3.6.

The LPC-method's encoder receives windowed time domain frames one at a time from the general processing part, and one control parameter, the order of the LPC calculation to be used. The frames are first high-pass filtered. The filter used is a FIR-filter with the cut-off frequency at the middle of the frequency scale, and the order of the filter is high enough to keep the level of the lower frequencies from interfering with the upper band processing. The filtered signal is next downsampled by a factor of two, to isolate just the upper band of the signal. The downsampling has caused the frequency spectrum of the upper band to be mirrored horizontally, so the mirroring is negated by mirroring again, by multiplying every

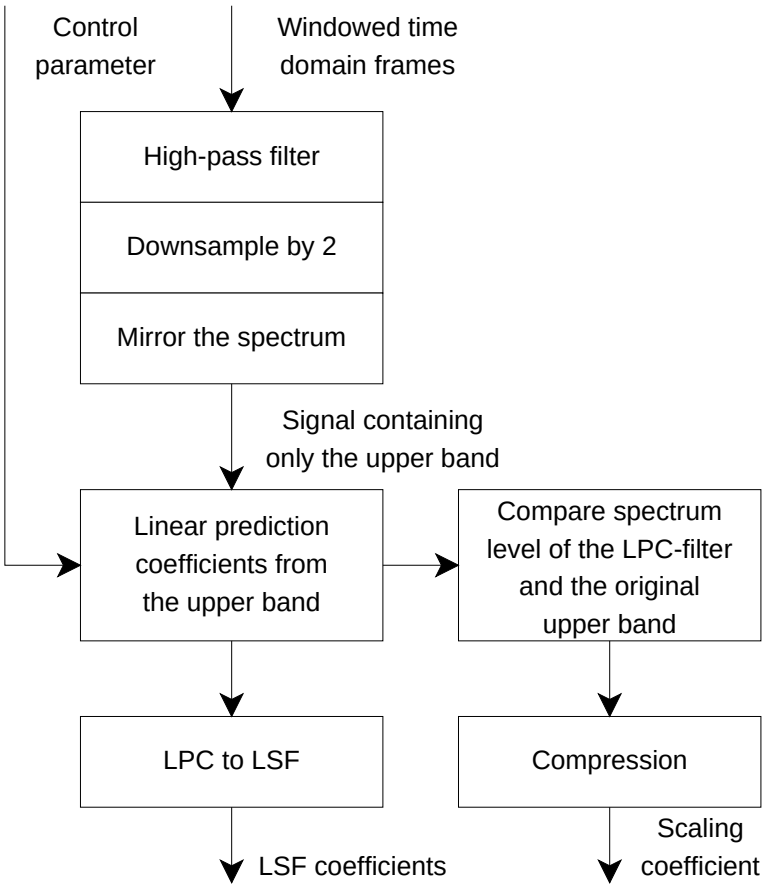

Figure 3.6: LPC-based method encoder.

other value of the time domain signal by -1.

Now linear predictor coefficients are calculated from the upper band time domain values. The coefficients are later transformed into line spectral frequencies (LSF), which is a efficient form for the coefficients in quantization.

Just the LPC coefficients are not enough to model the desired characteristic of the upper band. The LPC model models the spectral envelope of the signal, but the average level of the spectrum is also required in this extension method. Therefore a scaling factor is used to measure the distance of the envelope from the average level, their shape is mostly similar but they are at different levels initially. The scaling factor is calculated by synthesizing the upper band from the predictor coefficients and comparing its average level to the average level of the original upper band.

The synthesizing of the upper band is done by first calculating the impulse response of a filter defined by the LPC coefficients. Then the LPC upper band and the original signal are

transformed to the discrete frequency domain by using FFT. Sum of the discrete frequency domain values of the original upper band is divided by the sum from the LPC upper band, and the result is the scaling coefficient.

During the design of the method, it was found out that the scaling coefficient was not accurate enough to set the level of the LPC spectral envelope. It was noticed that with signals having relatively low levels the scaling produced too low level envelopes. On somewhat high levels the behaviour was correct. This is fixed by compressing the scaling coefficients, so that most of the coefficients are amplified, smaller ones more than the larger coefficients. The compression is done using the μ-law compression:

$$y = \frac{V \ln(1 + \mu|x|/V)}{\ln(1 + \mu)} \mathrm{sgn}(x), \tag{3.14}$$

where x is the input signal value, V is the "maximum" value of compression, taken from the largest scaling coefficient of the signal being processed, μ is the μ-law parameter, informal testing has proved that 1.6 is a good value here, $\mathrm{sgn}(x)$ is the signum function of x, and y is the compressed output value.

The output from the LPC-based method's encoder is the LPC coefficients in LSF-form, and the scaling coefficient.

3.3.2 Decoder

A block diagram of the LPC-based method's decoder is presented in figure 3.7.

The decoder receives input frames in MDCT-form, which are first inverse transformed to time domain using equation 2.4. Other inputs to the decoder are the LPC coefficients in LSF form, and the scaling coefficient.

In the decoding process the lower band again has to be transposed to the upper band. This can be done for example the same way as in the MDCT-based method, section 3.2.2. However, only the `mirror_full` –type of copying is used in the LPC-based method. After getting the information also to the higher frequencies, FFT is done for the upper band.

The LSF values are then converted back to LPC-form. Impulse response of the LPC-filter is then calculated, as in the method's encoder. With FFT this is transformed to the discrete frequency domain, and the level of the resulting synthesized frequency envelope is adjusted to the correct level by multiplying the spectral values with the scaling coefficient.

To shape the current upper band to be similar as the original, a moving average is first calculated of the copied upper band. The moving average is calculated from $N/64$ spectral values at a time, where N is the length of the frame being processed. Result from this calculation is a smoothed version of the upper band, which is used in the upper band level

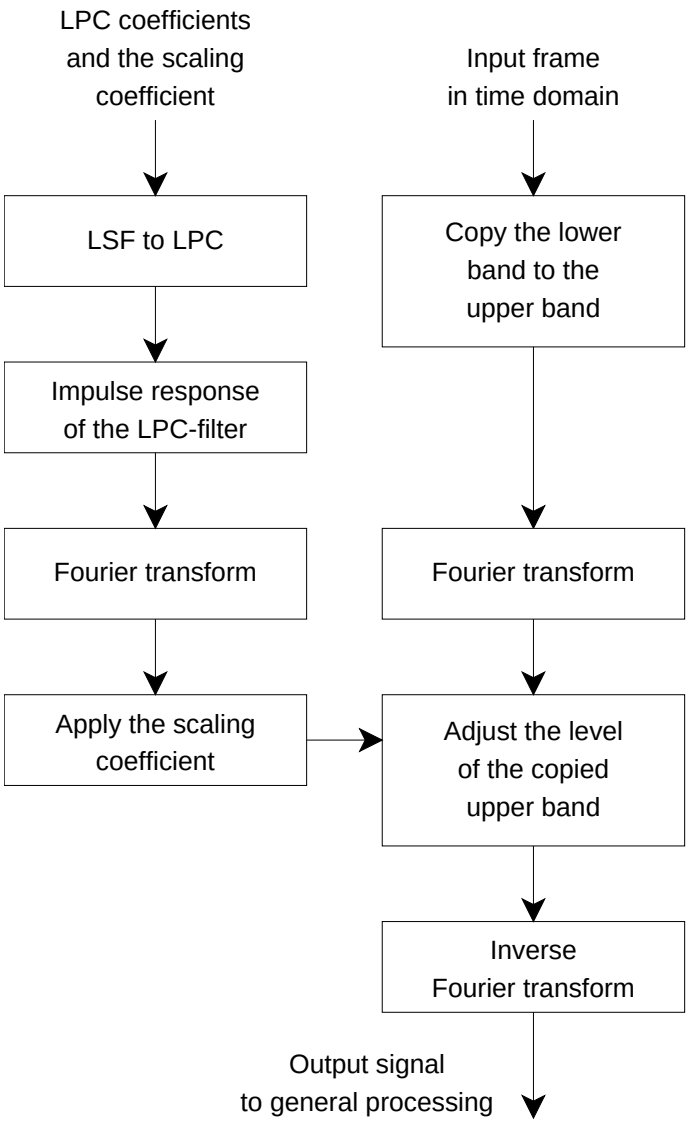

Figure 3.7: LPC-based method decoder.

adjustment as s_{copied}. The level adjustment goes similarly to as in the MDCT-based method:

$$f_{adjusted}(n) = f_{copied}(n) \cdot \frac{s_{LPC}(n)}{s_{copied}(n)} \text{ for } 0 \leq n < N-1, \tag{3.15}$$

where $s_{LPC}(n)$ are the synthesized frequency envelope values, $f_{copied}(n)$ the copied upper band values and $f_{adjusted}(n)$ the final extended upper band spectral values. N is the total number of spectral values on the upper band.

As the extension process is done in discrete frequency domain, an inverse fourier transform (IFFT) is needed to get the signal back to the time domain. Resulting frame is then

passed to the general processing part of the decoder.

Example figures of signals where the LPC-based method is used in the extension can be found in appendix A. In the figures there is the frequency spectrum of the same frame when 7 and 20 coefficients are used for the LPC. Also included is a problematic situation for the LPC-based method.

Chapter 4

Listening tests

In order to be able to reliably evaluate the sound quality produced by the extension methods, two listening tests were held by Nokia Research Center at Tampere. The first test was performed with non-quantized bandwidth extension parameters, and the purpose of the test was to find out how the copying parameters would affect the result of extension. For the second test the best parameters were chosen, quantization schemes were optimized for these, and the test was done with quantized parameter values. The author did not conduct the tests or process the results, but participated in choosing the test samples, test conditions, and test methods.

4.1 Listening test with non-quantized parameters

4.1.1 Test samples

Samples used in the test were 12 “usual” samples used by MPEG in listening tests. These items are known to be very critical to audio coding [7, 33]. The samples are listed and briefly described in table 4.1.

The non-vocal –samples were shortened so that their length was between 6–8 seconds. This was done to make sure that the test would not be too long for the listeners. However, shortening was done carefully such that the samples remained long enough to give the listener the possibility to hear the sample’s characteristics. The vocal (“es”) samples were from 7.5 to 10.5 seconds long. All of the samples were single channel (mono) samples.

4.1.2 Test conditions

The test was done separately with 16 kHz and 32 kHz sampling rates. The higher sampling rate represents a situation where BWE is used in high-quality audio coding. The lower

Table 4.1: Samples used in listening tests.

Sample name	Description
es01	Vocal (Suzanne Vega)
es02	German male speech
es03	English female speech
sc01	Trumpet solo and orchestra
sc02	Classical orchestral music
sc03	Contemporary pop music
si02	Castanets
si03	Pitch pipe
sm01	Bagpipes
sm02	Glockenspiel
sm03	Plucked strings

was chosen because informal test listening had proved that the differencies between the methods are more audible at lower rates. In addition using two different sampling rates provides information how BWE methods work with different rates.

10 conditions were selected to both tests. However, some of the methods used in the test were not developed by the author, so they are left out from the lists and results. Methods and their parameters used in 16 kHz and 32 kHz test are listed in tables 4.2 and 4.3.

Table 4.2: Conditions in test, 16 kHz.

Condition name	Method	Parameters
direct	No processing	
lowpass4	Lowpass filter	Cutoff at 4 kHz
lowpass6	Lowpass filter	Cutoff at 6 kHz
MDCT8_full	MDCT	8 bands, uniform scale, full copy
LPC7	LPC	LPC order 7
LPC20	LPC	LPC order 20
MDCT8_mirrorfull	MDCT	8 bands, uniform scale, mirrored full copy
MDCT32	MDCT	32 bands, uniform scale, full copy

A couple of reference conditions were included in the test with the BWE methods. The references were direct, where no processing was done for the test sample, and two lowpass filtered references with different cutoff frequencies.

The 32 kHz sampling rate test also included a special reference, AAC+. AAC+ is an existing codec, which uses SBR for bandwidth extension, and was included to give some comparison between it and the new methods described in chapter 3. However, as AAC+ is a real codec, all its coefficients are quantized, and the coefficients of the other methods

Table 4.3: Conditions in test, 32 kHz.

Condition name	Method	Parameters
direct	No processing	
lowpass8	Lowpass filter	Cutoff at 8 kHz
lowpass12	Lowpass filter	Cutoff at 12 kHz
AAC+	AAC+	AAC+ at 64 kbit/s
LPC7	LPC	LPC order 7
LPC20	LPC	LPC order 20
MDCT8	MDCT	8 bands, uniform scale, mirrored full copy
MDCT32	MDCT	32 bands, uniform scale, mirrored full copy

were not, so complete comparison can not be done in this test. The AAC+ samples were done at 64 kbit/s bitrate. 64 kbit/s is sufficiently high to keep the non-BWE –related coding artifacts low enough, so they do not interfere with the characteristics tested here.

For the 32 kHz BWE methods, the limit frequency of the BWE was set to 9 kHz, which is not in the middle of the 16 kHz audio bandwidth of that sampling rate. The limit was chosen because AAC+ uses the same limit at 48 kHz sampling rate, from which the AAC+ samples were converted to 32 kHz. To make the comparison between AAC+ and the other methods fair, the lower band of the AAC+ –coded samples were used in the generation of the all 32 kHz output samples, which means that all the upper bands were generated using the AAC+ –coded lower band.

For the 16 kHz methods, the limit frequency was 4 kHz, which is in the middle of the 8 kHz audio band, and the output samples were generated completely with the original uncoded samples.

Different number of bands and coefficients were chosen to find out if and how much there is difference in the methods when different parametrization is used for the upper band. In the MDCT-method there also was the option to choose which part of the lower band was copied to the upper band. Informal listening had proved that the copying type did not have much effect, and as mirrored full was at least as good method as the others and also used in the LPC-based methods, it was chosen to be used in the test. The full copying type was also included in the 16 kHz test, as there was no AAC+ for that sampling rate. The MDCT methods using ERBs were not tested, as the number of ERBs and their width was very close to eight uniform width bands.

No quantization was done for any of the coefficients in the new BWE methods in this test. The LPC7-condition used order 4 LPC for every short frame, other methods had half the number of long frame coefficients during short frames. The limiting of the coefficients during transients was chosen to be done to keep the bitrates of the methods lower, as one

transient frame contains 8 short frames, and therefore requires 8 times more coefficients than a normal frame if the number of coefficients per processed frame would be the same. High bitrates would be acceptable in a synthetic test, but for real coders the bitrates are required to be low. The lower number of coefficients during transients is possible because those situations do not need very exact frequency resolution as the time resolution is more important.

4.1.3 Test procedure

The listening was done according to the paired comparison method [15, 16, 27], and grading was according to the degradation mean opinion score (DMOS) [19]. The subjects heard two samples, the first was the original and the second was one of the 10 conditions. There was no possibility to repeat the samples. As presented in section 4.1.1, 12 different audio samples were used. The subjects listened to either the 16 kHz or the 32 kHz samples, not both. The test was designed such that its duration was less than 35 minutes. Three different sets were generated for both sampling rates. 80 sample pairs were included in every set. Every condition appeared 8 times in one set, and at the end of the set each audio sample had been played 5–8 times. Because of the limited amount of time, every listener did not listen to every sample-condition –pair. But the sets were generated so that after all three sets had been listened, every pair had appeared twice.

The subjects graded the sample pairs with a scale presented in table 4.4.

Table 4.4: DMOS grading scale.

Grade	Difference between the second sample and the first sample
5	Inaudible
4	Audible, but not annoying
3	Slighly annoying
2	Annoying
1	Very annoying

4.1.4 16 kHz test results

In the 16 kHz test the differences between the original and the coded samples were quite clear, and so it was not necessary to use expert listeners. In total 22 persons, of which 2 were expert listeners, participated in the test. The main test results and their 95% confidence intervals are presented in figure 4.1.

The sound samples are divided into four groups, each having distinguishable characteristics. The “es” samples are all vocal samples. The “sc” samples are comprised of complex

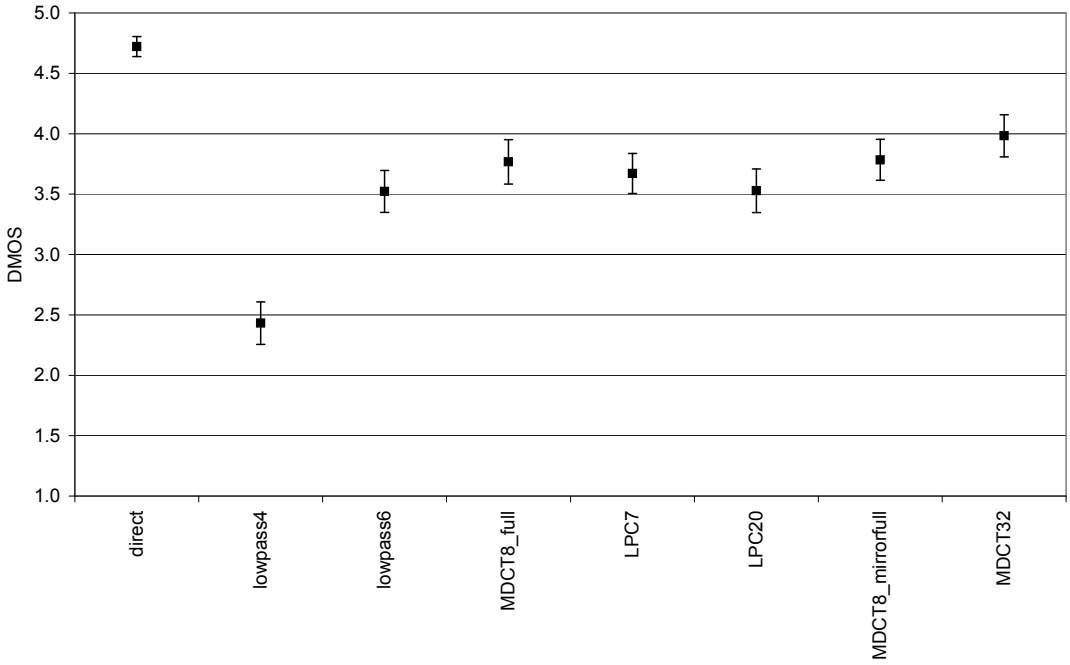

Figure 4.1: 16 kHz listening test results.

sounds. The “si” samples are fairly simple sounds from single instrumets. In the “sm” sounds there also is only one instrument, but the melodies played are more complicated. Grouped test results can give more information from the performance of the methods, and for the 16 kHz test, they are shown in table 4.5.

Table 4.5: Average results at 16 kHz grouped by sample types. Two best scores are printed in bold in each column.

Condition	es	sc	si	sm	Grand total
direct	**4.82**	**4.77**	**4.71**	**4.58**	**4.72**
lowpass4	1.98	2.55	2.59	2.64	2.43
lowpass6	3.23	3.79	3.35	**3.72**	3.52
MDCT8_full	3.32	4.41	3.85	3.45	3.77
LPC7	3.63	3.96	3.69	3.40	3.67
LPC20	3.96	3.63	3.44	3.09	3.53
MDCT8_mirrorfull	3.73	3.98	**3.96**	3.44	3.78
MDCT32	**4.36**	**4.43**	3.61	3.52	**3.98**

4.1.5 32 kHz test results

It was known in advance that in 32 kHz test it is difficult to hear differences between samples, and therefore only experienced listeners were used. In total 15 expert listeners partic-

ipated in the test. The main test results are shown in figure 4.2.

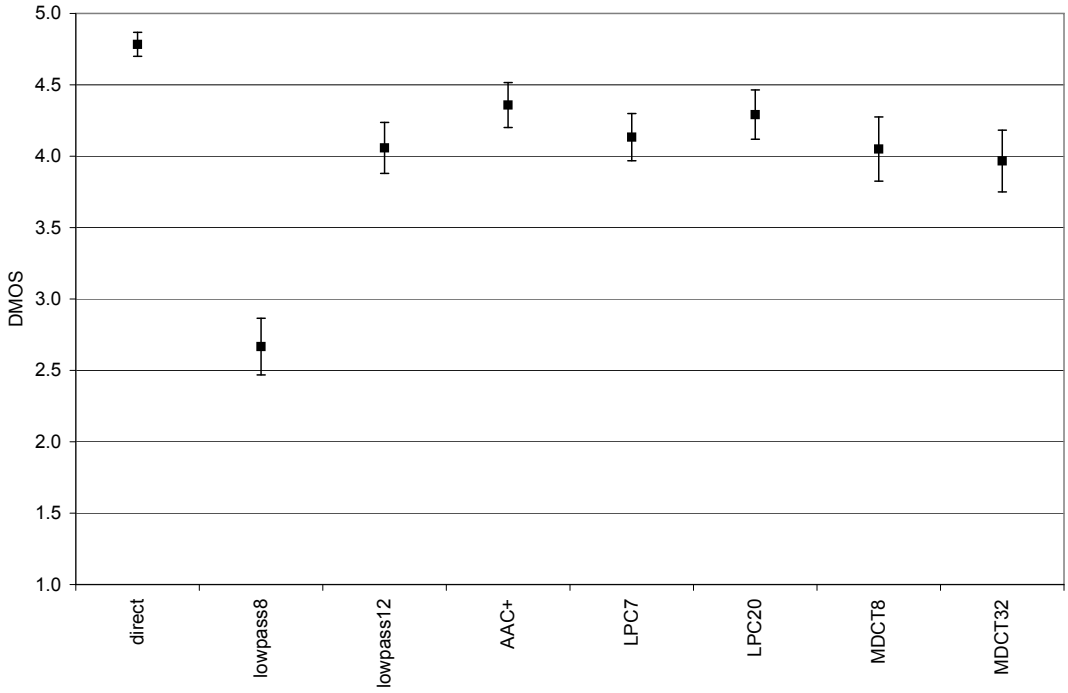

Figure 4.2: 32 kHz listening test results.

As with the 16 kHz samples, the grouped test results for the 32 kHz conditions are shown in table 4.6.

Table 4.6: Average results at 32 kHz grouped by sample types. Two best scores are printed in bold in each column.

Condition	es	sc	si	sm	Grand total
direct	**4.82**	**4.91**	**4.81**	**4.62**	**4.78**
lowpass8	2.43	3.46	2.41	2.50	2.67
lowpass12	4.04	4.19	3.88	4.14	4.06
AAC+	4.07	4.68	**4.22**	**4.47**	**4.36**
LPC7	4.35	4.04	4.13	3.96	4.13
LPC20	4.50	4.50	4.11	4.03	4.29
MDCT8	**4.69**	**4.69**	3.50	3.19	4.05
MDCT32	4.50	4.50	3.46	3.33	3.97

4.2 Listening test with quantized parameters

From the results of the previous listening test, some BWE methods were chosen to be further developed, and quantization for their parameters was implemented. The quantization methods were not the work of the author, so their operation will not be described here. The resulting bit consumption will be presented. The test also included some methods not done by the author, so they are also left out from the results.

The methods selected for quantization are briefly described in table 4.7.

Table 4.7: BWE methods for quantization.

Acronym	Method	Quantized parameters
mdct8	MDCT-based method with 8 subbands, mirrored full copy, uniform width subbands	One scaling factor for every subband
lpc7	LPC-based method with 7 LPC parameters	LPC parameters and scaling coefficients

4.2.1 Quantized parameters

The overall bit consumption of the methods used in this test is presented in table 4.8. The estimated average bit rates are given at 16 and 32 kHz sampling rates. The average bit rates are estimated based on the assumption that 78% of 2048 sample windows consist of one single window and 22% consist of eight short windows. These estimations are based on the data used in training the quantizers. The data used in training was the 12 test samples at 16, 32 and 48 kHz sampling rates, and eight short samples of pop music, at 32 and 48 kHz sampling rates.

Table 4.8: Bit consumptions and average bit rates for the methods.

	Bits per 2048 sample windows		Average bit rate (kbit/s)	
Method	Transients	Normal	32 kHz	16 kHz
mdct8	112	28	1.45	0.73
lpc7	144	23	1.55	0.78

A more detailed description of the employed quantization methods is given in the following subsections.

Quantization of parameters in the MDCT-based extension method

The energy values of the MDCT-method are not bounded in any way, they are directly proportional to the level of the original signal's upper band. However, the energies have a

tendency to decrease as the frequency of the subband rises, so that the second subband has lower energy that the first and so on. This may be helpful in quantizing.

In the method there are energy values on 8 subbands to quantize in the long frames, and on 4 subbands in frames that are considered transients. The energy values are quantized as vectors of 8 samples in normal frames, and 4 samples in short frames. Bits used per sample is shown in table 4.9.

Table 4.9: Quantization of energy values in the MDCT-based method.

Method name	Frame type	Vector length	Bits/sample
MDCT8	Transient	4	3.5
	Normal	8	3.5

Both the 16 and 32 kHz sampling rates use the same quantizers.

Quantization of parameters in the LPC-based extension method

The linear predictor coefficients in the LSF-form are bounded to the range from 0 to π, and they are always in ascending order in the coefficient vector. This makes is possible to quantize them with vector quantization very efficiently. However, the order of the coefficients must be retained after quantization, otherwise the result from bandwidth extension may be very erroneous. If the used vector quantization method is not foolproof, correction methods must be used to ensure correct operation.

During normal frames, the LPC-based method produces 7 linear predictor coefficients. In transient frames there are 4 coefficients per short frame. Bit usage of the quantizers is shown in table 4.10.

Table 4.10: Quantization of predictor coefficients in the LPC-based method.

Method name	Frame type	Vector length	Bits/sample
LPC7	Transient	4	3
	Normal	7	2.3

The LPC-based method also needs one scaling coefficient per frame. These are not bounded, their value depends on the level of the signal being processed. The scaling coefficients are quantized with a scalar quantizer, table 4.11 shows the used bits.

Table 4.11: Quantization of scaling coefficients in the LPC-based method.

Method name	Frame type	Bits/sample
LPC7	Transient	6
	Normal	7

The same quantizers are employed for the both sampling rates in the test, 16 and 32 kHz.

4.2.2 Listening test description

The test for the methods using quantized parameters was made as a multistimulus test with hidden reference and anchors (MUSHRA, [17, 27]). The MUSHRA test can potentially provide more reliable results than a DMOS test used in the previous test, but it also requires more time. In the test the listeners compared multiple conditions of a sample at the same time, and could repeat the samples. The listeners could also repeat the reference when they so wanted. The test produces results for the conditions between 0 and 100, with 100 being same quality as the reference.

Test samples were the same as in the previous test, as presented in section 4.1.1. The test was again done separately for 16 and 32 kHz samples. AAC+ was again used as a reference method in the 32 kHz test. In both tests the BWE limit frequency was in the middle of the audio band, at 4 kHz in the 16 kHz test and at 8 kHz in the 32 kHz test. This time that AAC+ was operated at 32 kHz sampling rate, and its limit frequency was at 8 kHz instead of the 9 kHz in the previous test. The lower band that the BWE methods used in the 32 kHz test was the same as in the AAC+ samples, so the synthetic upper bands were also generated using frequency data comparable with AAC+.

Conditions in the 16 and 32 kHz tests are shown in tables 4.12 and 4.13. The total number of conditions was eight in the 16 kHz test and nine in the 32 kHz test, so three conditions have been left out from the descriptions and results.

Table 4.12: Conditions in second test, 16 kHz.

Condition name	Description
direct	No processing
lowpass4	Lowpass filter, cutoff at 4 kHz
lowpass6	Lowpass filter, cutoff at 6 kHz
mdct8	MDCT-based method with 8 subbands
lpc7	LPC-based method with order 7 coefficients

4.2.3 Test results

In this test there were 12 listeners in the 16 kHz test, and 11 in the 32 kHz test. All of the listeners were experienced listeners. Main test results are shown in figures 4.3 and 4.4. Test results grouped as in the previous test can be found from tables 4.14 and 4.15.

Table 4.13: Conditions in second test, 32 kHz.

Condition name	Description
direct	No processing
lowpass8	Lowpass filter, cutoff at 8 kHz
lowpass12	Lowpass filter, cutoff at 12 kHz
mdct8	MDCT-based method with 8 subbands
lpc7	LPC-based method with order 7 coefficients
aacplus	AAC+ at 64 kbit/s

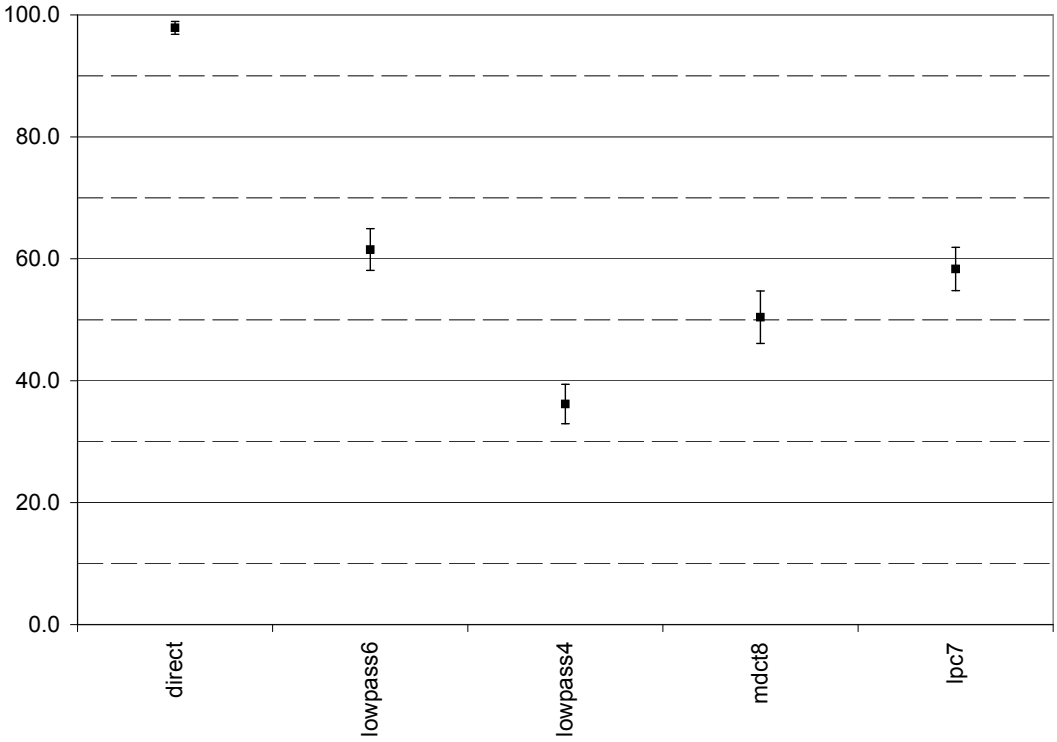

Figure 4.3: 16 kHz second listening test results.

Table 4.14: Average results from the second test, at 16 kHz grouped by sample types. Two best scores are printed in bold in each column.

Condition	es	sc	si	sm	Grand total
direct	**98.81**	**96.50**	**99.11**	**97.14**	**97.89**
lowpass6	**61.17**	63.11	55.39	**66.42**	**61.52**
lowpass4	38.08	40.83	32.28	33.56	36.19
mdct8	40.61	59.83	50.44	50.78	50.42
lpc7	48.11	**67.42**	**63.06**	54.75	58.33

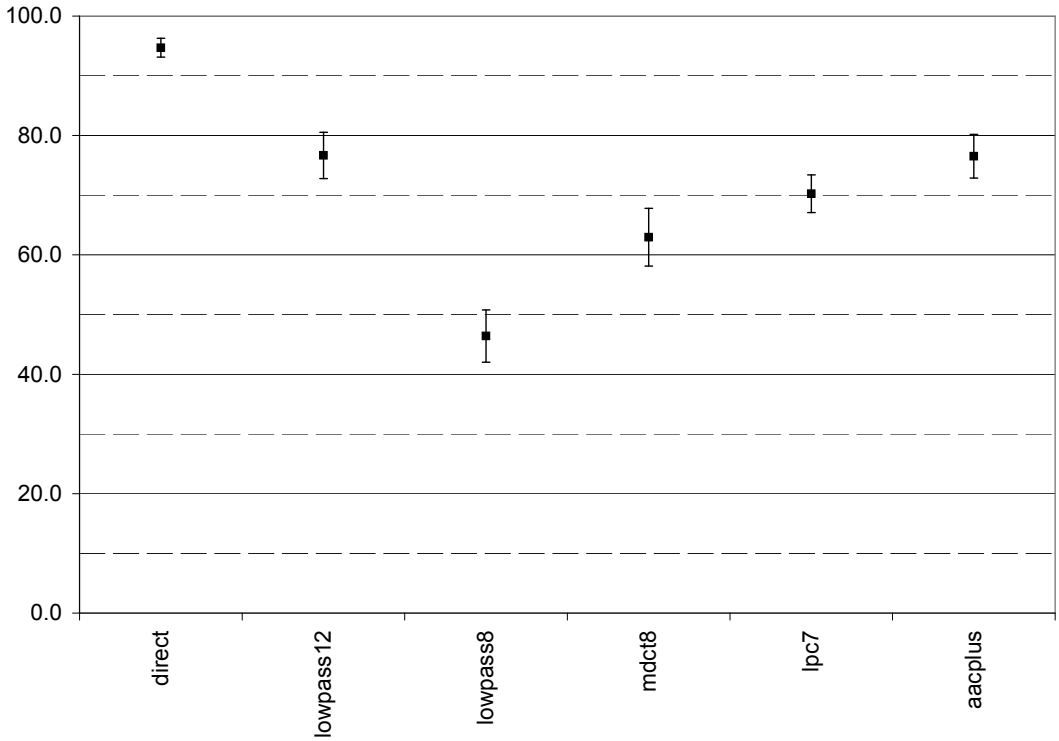

Figure 4.4: 32 kHz second listening test results.

Table 4.15: Average results from the second test, at 32 kHz grouped by sample types. Two best scores are printed in bold in each column.

Condition	es	sc	si	sm	Grand total
direct	**95.45**	**92.67**	**96.55**	**94.15**	**94.70**
lowpass12	**77.55**	83.21	74.15	71.73	**76.66**
lowpass8	44.27	60.42	37.91	43.03	46.41
mdct8	63.12	68.82	59.33	60.55	62.95
lpc7	67.30	69.76	74.03	69.88	70.24
aacplus	62.48	**86.82**	**75.91**	**80.94**	76.54

Chapter 5

Discussion

In the previous chapters the algorithms behind the new bandwidth extension methods were presented, as well as the results of the listening tests. This chapter looks into the strengths and weaknesses of the methods and discusses the test results.

5.1 New bandwidth extension methods

MDCT-based extension method

The MDCT-based extension method is fairly simple both computationally and on theoretical basis. The idea of copying the lower band of the frequency spectrum and adjusting its level on several subbands is the same as in spectral band replication [7]. The parameters of the copying were varied to find out if any of them could be better than others, but informal test listening did not provide such results, so the mirrored full type of copying and energy adjustment on eight subbands were chosen for the further development.

LPC-based extension method

The LPC-based extension method is somewhat more complicated than the MDCT-based method, as the calculations require processing power in the filtering and moving average computations. The method however can give theoretically more exact spectral information on the upper band, as the resulting envelope is continuous, compared to the discrete subbands of the MDCT-based method and SBR.

At first it was thought that the LPC-based method would provide much lower bitrate than the MDCT-based method. This assumption was based on the thought that the quantization of the LSFs is very efficient, but the comparation was made against scalar quantized MDCT-based method's energy values. Later is was found out that vector quantization could also

be used on those, and the amount of bits the MDCT-based method needed dropped considerably. The LPC-based method also needs the scaling coefficients during processing, and quantizing those with scalar quantization is not very efficient, at least not yet.

Common for both methods

Both methods do only spectral level adjustment after the synthetic signal has appeared on the upper band. In this sense they are inferior to SBR, which can at least add missing sinusoids or noise to the upper band. Such extra processing could be beneficial for the methods in some cases, but it would also mean that more data is needed to be transmitted and the bit consumption of the methods would rise somewhat. This could perhaps move the new methods away from their probably intended best purpose, bandwidth extension at low bitrates. The effect of doing the additional processing on quality and bit consumption should still be studied.

Another issue which may be a weak link for the new methods is the frame selection during transients. Currently a very short transient requires two long frames to be converted into 16 short frames. The transient always appears in two frames, as they have the 50% overlap. The transients require much more bits than normal frames as is shown on table 4.8. The position of the transients could be taken into account more accurately, and the windowing selection done with the lowest possible amount of short windows. This could also mean having different length transition windows, but the effect should anyway be looked into.

Third issue which might still need attention is the quantization. The quantization scheme presented in the previous chapter is only the first version, and could be developed more. Apparently prediction during quantization could be used, as the coefficients in concurrent frames are often very similar, if the sound signal is mainly stationary. Also in the quantization, the training data used for the quantizers should contain much more different samples as was used here, to give more general result for the training. The quantization results used for the second listening test are also a bit questionable, as the training data included all the test samples, and not really that much more different sound samples.

5.2 Listening test results

Listening test with non-quantized parameters

For the 16 kHz samples, as seen in figure 4.1, the confidence intervals of all the tested methods overlap (except MDCT32 versus LPC20), so statistically the methods should be considered equal. However, as the average scores for the LPC-based methods are slightly lower than the scores of MDCT-based methods, the LPC-based methods may be though to

provide lowest quality at this sampling rate. All the BWE methods rate better than the 4 kHz low-pass filtered samples and are at least equal with the 6 kHz low-pass filtered versions, so the bandwidth extension can be considered to work even at this low BWE limit bandwidth.

As for the differences between different versions of the methods, the MDCT-based method with 32 subbands obtains the highest score in the test, with 8 subbands the result is slightly lower. For the different type of copying in the MDCT8-methods, the scores are very even, so it should not matter much how the spectrum is copied to the upper band. There are some differencies between the sample groups however, full copy getting better results with the "sc" –type of samples, while mirrored full copy does a higher score with the "es" samples. For the description of the sample groups, see table 4.1 and section 4.1.4. With the different LPC-based versions, the less accurately modeled LPC7 gets a slightly higher average score than the 20-coefficient version, but with the confidence intervals overlapping, no clear winner can be distinguished.

In the 32 kHz test the average results in figure 4.2 are again very close and the confidence intervals overlap for all tested methods. This time, without taking into account the AAC+ score, the highest average is obtained by the LPC20, which has the lowest score in the 16 kHz test. LPC7 comes next, and then the MDCT-based methods. According to the results, the new methods can produce quality comparable to the existing codec AAC+, but it must be noted that the parameters of the other methods were not quantized. The best average scores of the tested methods are also above 4, which is by the test definition considered "audible, but not annoying (difference)". However, it should be noted that the scores of the MDCT-based methods for the "si" and "sm" –type of samples are quite low. And again, the bandwidth extension is working, as the 8 kHz low-pass filtered samples get clearly the lowest score.

Listening test with quantized parameters

The results for the quantized 16 kHz test can be found in figure 4.3, and they are not especially good if the quality of the BWE methods is considered. These results are not directly comparable with the non-quantized results, as the test was a MUSHRA test instead of paired comparison, but it seems that quantized bandwidth extension for this sampling rate is not very feasible. From the two BWE methods in test, the LPC-based method obtains a higher score, and as the confidence intervals of the methods do not overlap, can be considered better. However, the 6 kHz lowpass filtered samples get a better average score than the extended methods. Most of the problems for the methods seem to come from the "es" –type of samples.

32 kHz test results in figure 4.4 look better, and at least for the LPC-based method, it can be said that the quantization has not significantly reduced the quality. These results are also

directly comparable with AAC+. The lower band of the signals is the same, it is used to generate the upper band, and all coefficients are quantized. AAC+ get the highest average score, but its confidence intervals overlap with the intervals of the LPC-based method, so statistically they are equal in quality. The MDCT-based method gets the lowest average score of the methods, and as both of the new methods require almost the same amount of bits, the LPC-based method can be considered better of the two. The grouped results do not vary much for the methods, but AAC+ seems to have a clear advantage in the more complex and melodical groups “sc” and “sm”.

Common for both tests

If more tests were to be done for the bandwidth extension methods, the selection of the test samples should probably be done again. The test samples may be critical for audio coding, but not necessarily for bandwidth extension. Especially in the “sc01” and “sc02” –samples there is not very much high-frequency information, so the samples could possibly be removed from the test or at least changed. Also the length of the samples could be considered again. Other MUSHRA tests have included even 31 seconds long samples [30], using longer samples could give the listener more time to concentrate on the different characteristics of the samples.

Chapter 6

Conclusions and future work

The aim in this thesis work was to study bandwidth extension in high-quality audio coding, and develop new methods to perform the extension. The objective was achieved as two working methods were implemented and tested.

From the methods the MDCT-based one basically just realizes the idea of SBR, but different ways to do the copying of frequency information and spectral envelope adjusting were tested. Based on the listening test results, conclusions from these are that the larger number of subbands where the spectral envelope is adjusted does not necessarily lead to better quality, as the average scores of the different versions of the method are very close. Also the use of ERBs as subbands is not clearly beneficial, as the number and width of the subbands would be almost the same as with eight uniform width subbands. The effect of which part of the spectrum is copied to the upper band was also tested. As informal listening did not give much difference between the four different copying types, and the difference between the two types in the 16 kHz sampling rate listening test was also very small, it can be concluded that the full and mirrored full –types of copying provide results of equal quality.

The LPC-based method provides a slightly different approach to the extension. The processing is done in the discrete frequency domain, and the spectral information from the upper band is saved as a continuous LPC envelope. The method was tested with 7 and 20 LPC coefficients, but according to the first listening test, either of those did not appear to provide better quality and the other. As the 7 coefficient version requires less bits in transmission, it can however be considered superior, and was selected to the test with quantized parameters. In the quantized test the LPC-based method scored better results than the MDCT-based one, but the confidence intervals have overlap. The conclusion is that the methods provide at least equal quality at about the same bitrate. The quality of the LPC-based method is also comparable with AAC+.

Future work

As noted in chapter 5, the new methods do not use any additional processing to add missing sinusoids or noise to the upper band. Such feature could be useful in some situations and should be tested. However, the functionality might not really be needed in low bitrate bandwidth extension.

On the other hand the quantization is still only the first version, and could be made better. Currently especially the scaling coefficients during transients in the LPC-based method may use too many bits. Also the quantization does not use any prediction between the frames. More advanced quantization methods could also save some bits. The training data used in quantization should also be more diverse than it currently has been. Another object of development could be the transient processing. Less short frames for the transients would also mean less bits.

The bandwidth extension is a very good way to reduce bitrate for the current audio coders. For example the mobile applications with limited transmission and storage capacity clearly benefit from smaller data rate. However, the network speeds have been increasing steadily, and in the future there might be a time when the limits are not a problem. At that time the coding will be made differently, but until then bandwidth extension is very useful in high-quality audio coding.

Bibliography

[1] 3GPP. *General audio codec audio processing functions; Enhanced aacPlus general audio codec; General description*. 3rd Generation Partnership Project technical specification TS 26.401 V6.2.0, March 2005.

[2] R. M. Aarts, E. Larsen, and O. Ouweltjes. A unified approach to low- and high-frequency bandwidth extension. In *AES 115th Convention*, New York, USA, October 2003.

[3] T. Bäckström. *Linear Predictive Modelling of Speech - Constraints and Line Spectrum Pair Decomposition*. PhD thesis, Helsinki University of Technology, February 2004.

[4] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa. ISO/IEC MPEG-2 Advanced audio coding. *J. of the Audio Engineering Society*, 45(10), October 1997.

[5] K. Brandenburg. ISO-MPEG-1 Audio: A generic standard for coding of high-quality digital audio. *J. of the Audio Engineering Society*, 42(10), October 1994.

[6] J. R. Deller, J. H. L. Hansen, and J. G. Proakis. *Discrete-time processing of speech signals*. Institute of Electrical and Electronic Engineers, Inc., 2nd edition, 2000.

[7] M. Dietz, L. Liljeryd, K. Kjörling, and O. Kunz. Spectral Band Replication, a novel approach in audio coding. In *AES 112th Convention*, Munich, Germany, May 2002.

[8] A. Ehret, M. Dietz, and K. Kjörling. State-of-the-Art Audio Coding for Broadcasting and Mobile Applications. In *AES 114th Convention*, Amsterdam, The Netherlands, March 2003.

[9] A. Ehret, K. Kjörling, J. Rödén, H. Purnhagen, and H. Hörich. aacPlus, only a low-bitrate codec? In *AES 117th Convention*, San Francisco, USA, October 2004.

[10] B. Grill. The MPEG-4 general audio coder. In *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, September 1999.

[11] A. Gröschel, M. Schug, M. Beer, and F. Henn. Enhancing audio coding efficiency of MPEG Layer-2 with Spectral Band Replication for DigitalRadio (DAB) in a backwards compatible way. In *AES 114th Convention*, Amsterdam, The Netherlands, March 2003.

[12] M. H. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, Inc., 1996.

[13] ISO/IEC. *Coding of audio-visual objects - Part 3: Audio*. ISO/IEC Int. Std. 14496-3:2001, 2001.

[14] ISO/IEC JTC1/SC29/WG11 MPEG. *Report on the formal subjective listening tests of MPEG-2 NBC multichannel audio coding*, November 1996. Document N1419 of the Maceio MPEG Meeting.

[15] ITU-R. *Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*. Recommendation ITU-R BS.1116-1, October 1997.

[16] ITU-R. *General methods for the subjective assessment of sound quality*. Recommendation ITU-R BS.1284-1, December 2003.

[17] ITU-R. *Method for the subjective assessment of intermediate quality level of coding systems*. Recommendation ITU-R BS.1534-1, January 2003.

[18] L. Kallio. Artificial Bandwidth Expansion of Narrowband Speech in Mobile Communication Systems. Master's thesis, Helsinki University of Technology, 2003.

[19] M. Karjalainen. *Kommunikaatioakustiikka*. Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, 1999.

[20] K. Käyhkö. *A Robust Wideband Enhancement for Narrowband Speech Signal*. Research Report, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, 2001.

[21] E. Larsen and R. M. Aarts. *Audio Bandwidth Extension - Application to Psychoacoustics, Signal Processing and Loudspeaker Design*. John Wiley & Sons, Ltd, 2004.

[22] E. Larsen, R. M. Aarts, and M. Danessis. Efficient high-frequency bandwidth extension of music and speech. In *AES 112th Convention*, Munich, Germany, May 2002.

[23] J. Mahkonen. Äänen laadun parantaminen puheensiirrossa keinotekoisella taajuuskaistan laajennuksella. Master's thesis, Teknillinen korkeakoulu, 1999.

[24] J. Makhoul. Spectral Analysis of Speech by Linear Prediction. *IEEE Transactions on Audio and Electroacoustics*, AU-21(3), June 1973.

[25] B. C. J. Moore, R. W. Peters, and B. R. Glasberg. Auditory filter shapes at low center frequencies. *J. of the Acoustical Society of America*, 88(1), July 1990.

[26] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-time signal processing*. Prentice-Hall, Inc., 2nd edition, 1998.

[27] F. Pereira and T. Ebrahimi, editors. *The MPEG-4 book*. Prentice Hall, 2002.

[28] H. Purnhagen. An overview of MPEG-4 Audio Version 2. In *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, September 1999.

[29] T. D. Rossing, F. R. Moore, and P. A. Wheeler. *The science of sound*. Addison Wesley, 3rd edition, 2002.

[30] G. A. Soulodre, T. Grusec, M. Lavoie, and L. Thibault. Subjective Evaluation of State-of-the-Art 2-Channel Audio Codecs. In *AES 104th Convention*, Amsterdam, The Netherlands, April 1998.

[31] G. A. Soulodre and M. Lavoie. Subjective Evaluation of MPEG Layer II with Spectral Band Replication. In *AES 117th Convention*, San Francisco, USA, October 2004.

[32] Y. Wang and M. Vilermo. The Modified Discrete Cosine Transform: Its Implications for Audio Coding and Error Concealment. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, Espoo, Finland, June 2002.

[33] M. Wolters, K. Kjörling, D. Homm, and H. Purnhagen. A closer look into MPEG-4 High Efficiency AAC. In *AES 115th Convention*, New York, USA, October 2003.

[34] T. Ziegler, A. Ehret, P. Ekstrand, and M. Lutzky. Enhancing mp3 with SBR, Features and Capabilities of the new mp3PRO Algorithm. In *AES 112th Convention*, Munich, Germany, May 2002.

Appendix A

Example figures

In this appendix some example figures are presented from the new bandwidth extension methods. Spectrum figures A.1 – A.8 show a situation where BWE should be working well, as a strongly harmonic signal is being extended. From the figures slight differences can be seen between the different methods and their parameters. In the figures using the LPC-based method the LPC envelope is shown. In the MDCT-based methods using less than 32 subbands the borders of the subbands are separated by vertical lines. A red vertical line in every example denotes the limit frequency where the bandwidth extension starts. The average levels of the original and processed signals are shown in the figures, as the local differences of the signals can be quite large, but the average difference is mostly somewhat small. Here it should be noted that the BWE methods are not trying to replicate the exact frequency spectrum of the original signal. The average level is a moving average calculated from 129 points, length of the whole frame in the figures is 2048 samples.

Figures A.9 and A.10 present examples of BWE in time domain. In the transient example there is a quite large difference between the original and processed signal during the transient. In the tonal example the difference is fairly small for the whole duration of the frame, but there is some error all the time.

Finally figures A.11 and A.12 depict a situation where the BWE methods of this thesis have some difficulties. In the original signal there are spectral peaks in the upper band, which can not be transposed back in the decoder, as there are no peaks at suitable locations in the lower band. The LPC-based method can recreate one of the peaks as the LPC envelope has a peak there itself, but the LPC of this order is not able to model all the peaks. The MDCT-based method has similar problems, and in addition it has not diminished a transposed peak in the lowest subband.

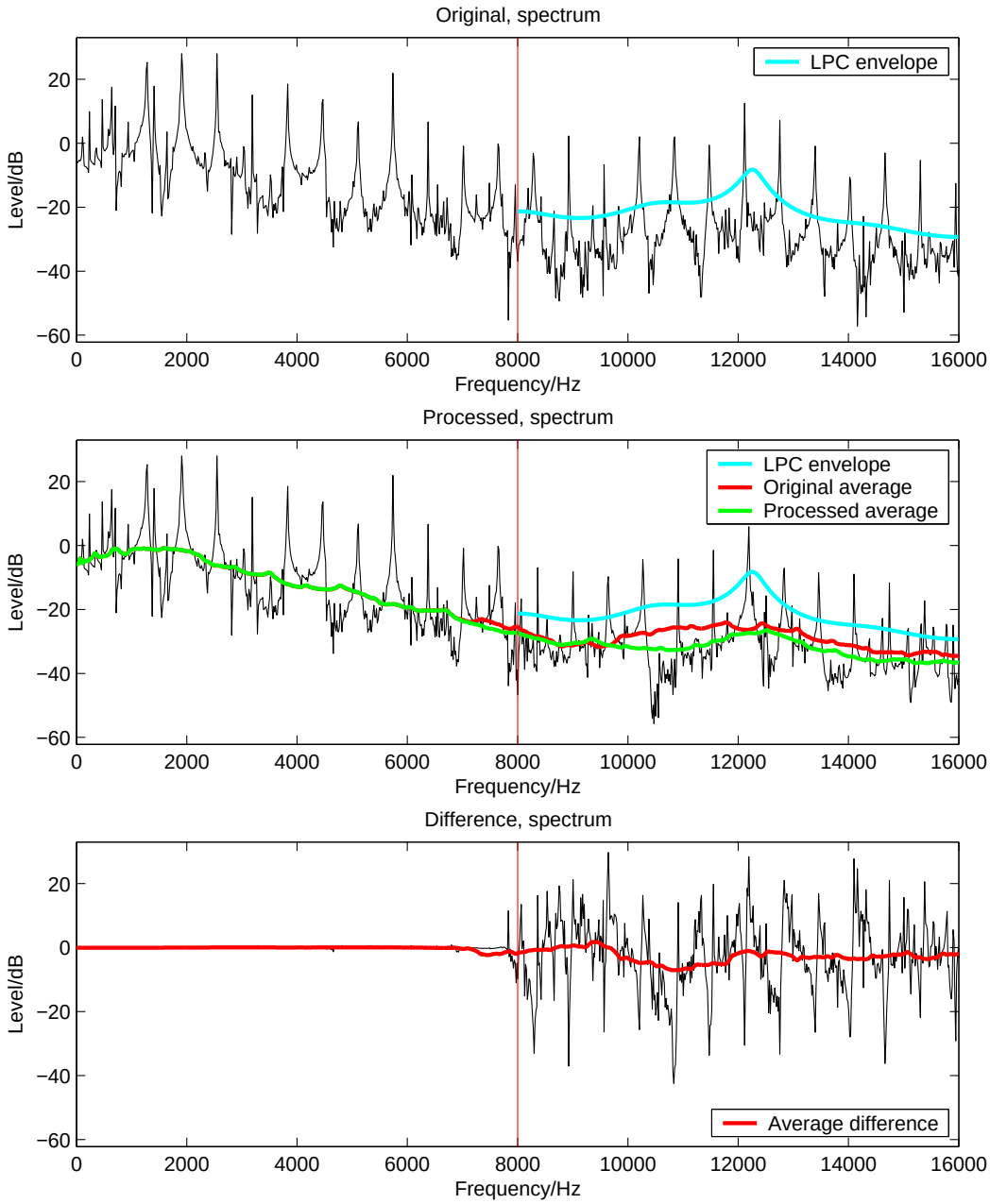

Figure A.1: Example frame, “sm01”, LPC-based method, 7 coefficients.

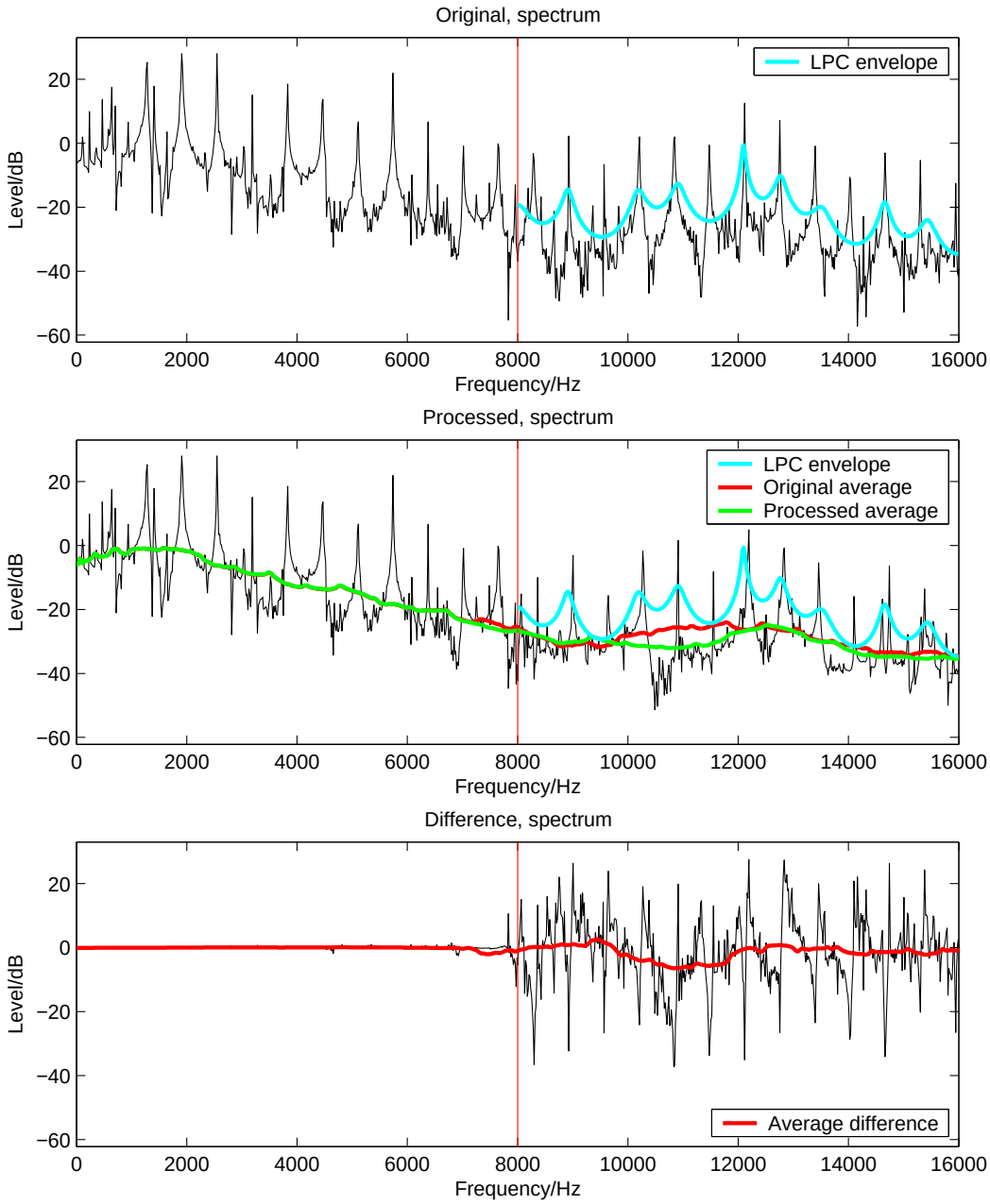

Figure A.2: Example frame, “sm01”, LPC-based method, 20 coefficients.

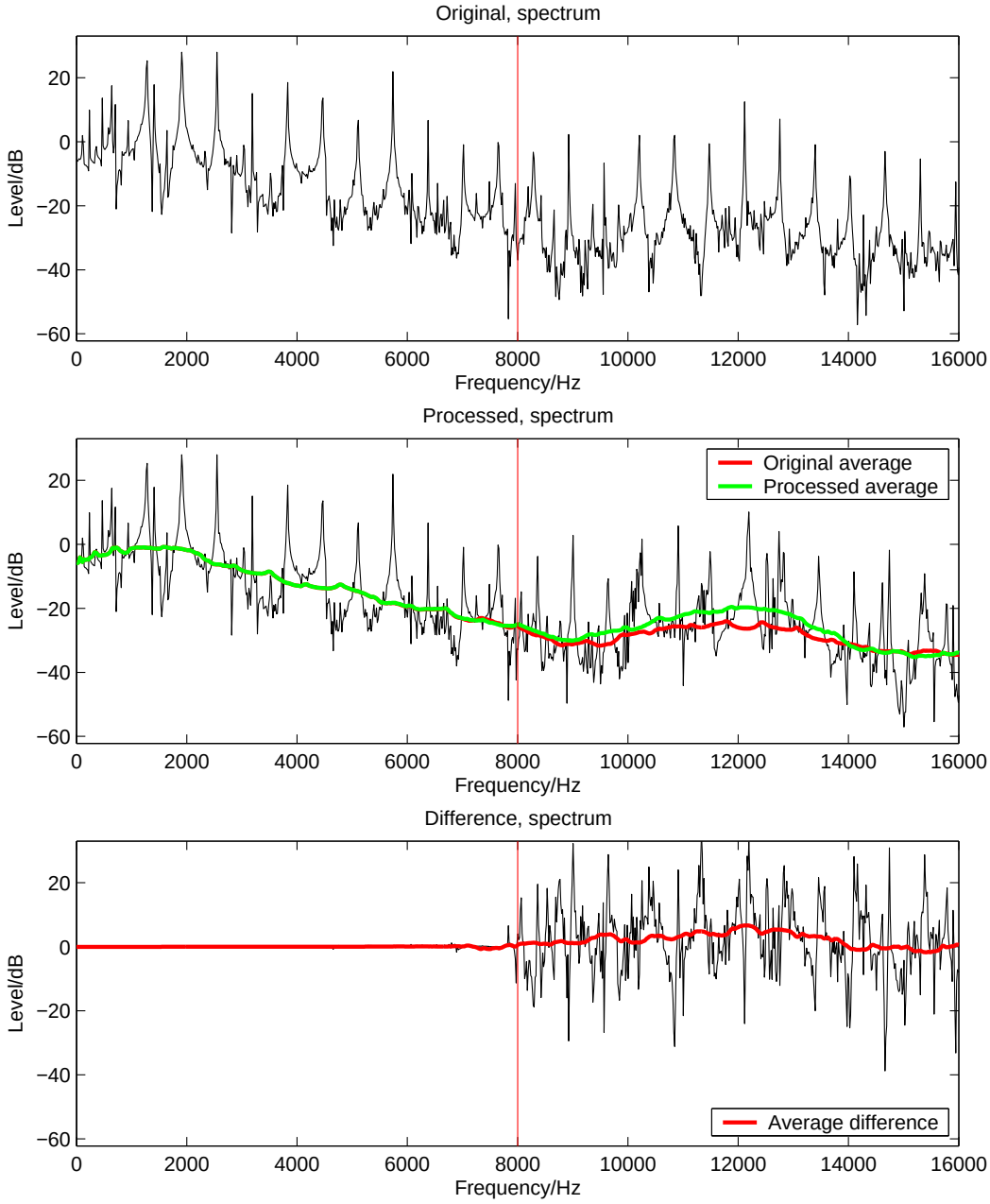

Figure A.3: Example frame, “sm01”, MDCT-based method, 32 uniform subbands, mirrored full copy.

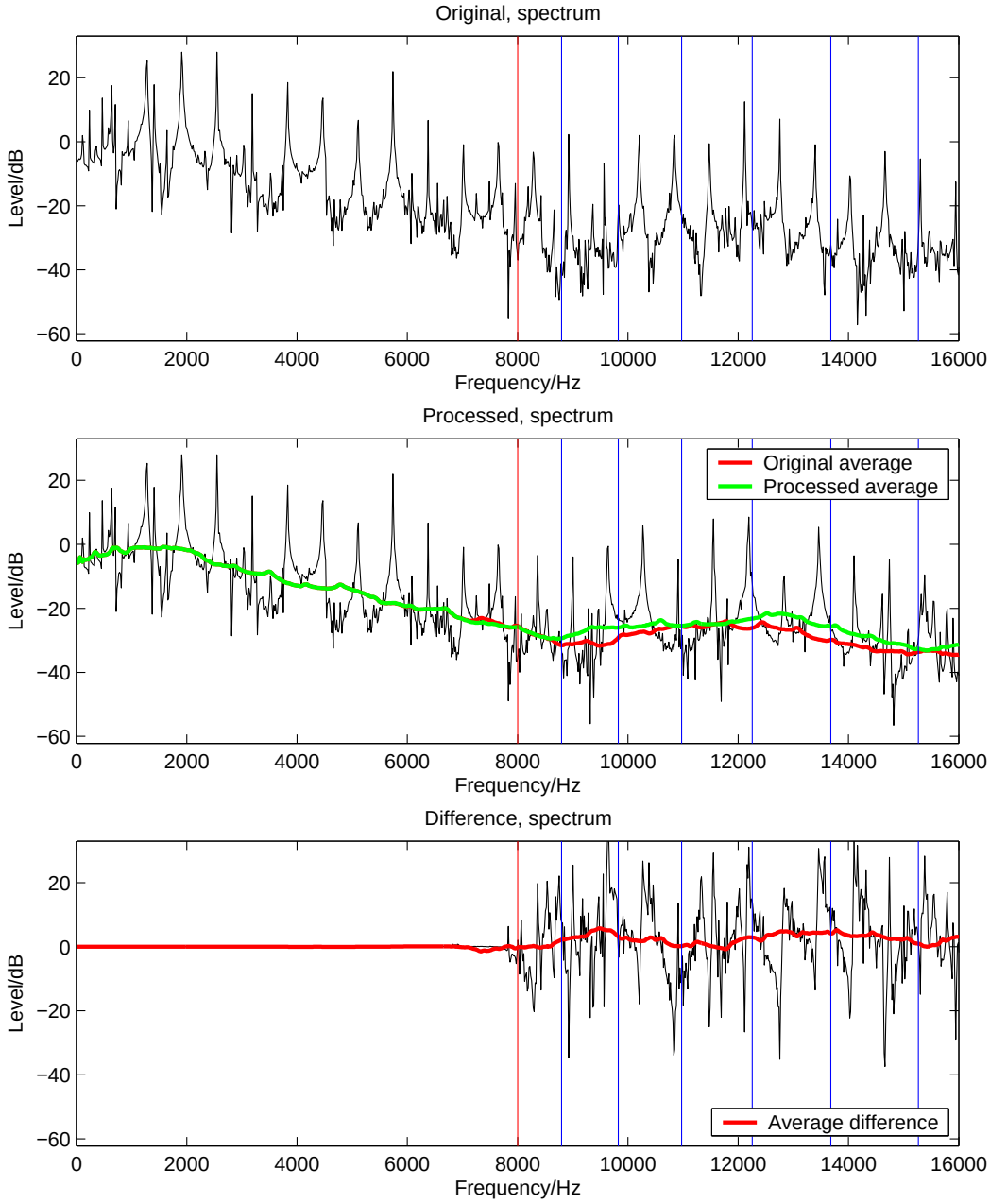

Figure A.4: Example frame, “sm01”, MDCT-based method, 7 ERB subbands, mirrored full copy.

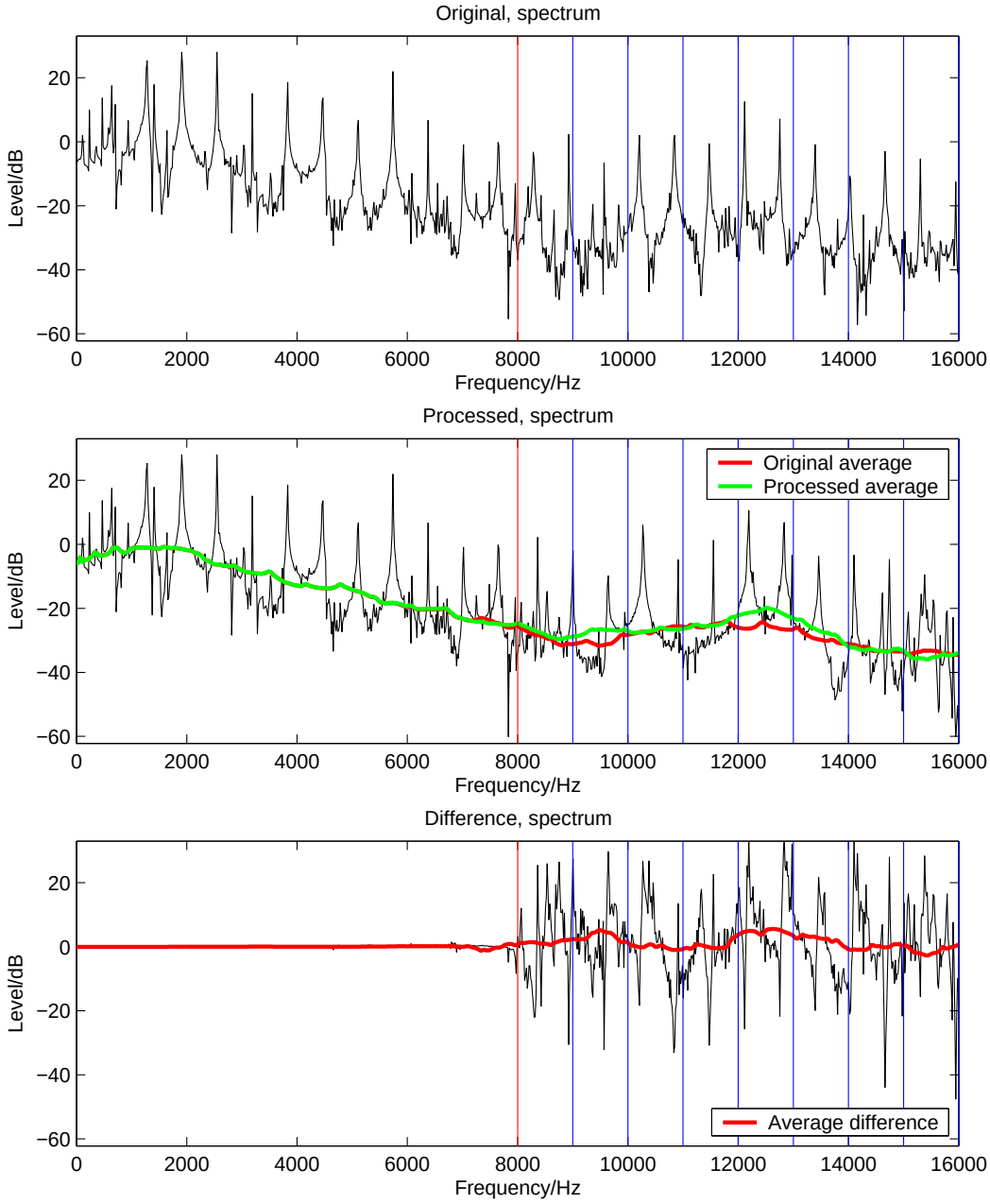

Figure A.5: Example frame, “sm01”, MDCT-based method, 8 uniform subbands, mirrored full copy.

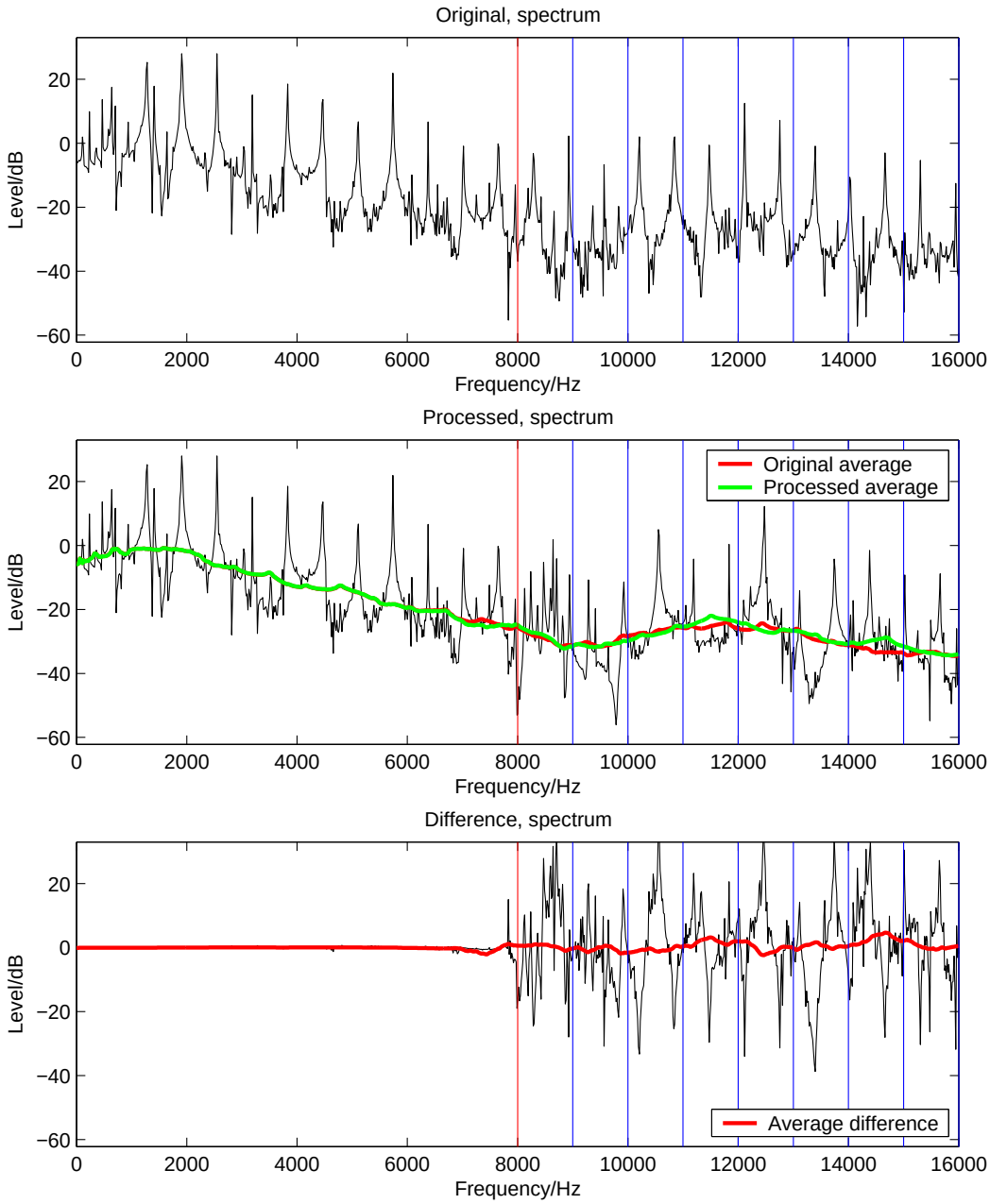

Figure A.6: Example frame, “sm01”, MDCT-based method, 8 uniform subbands, full copy.

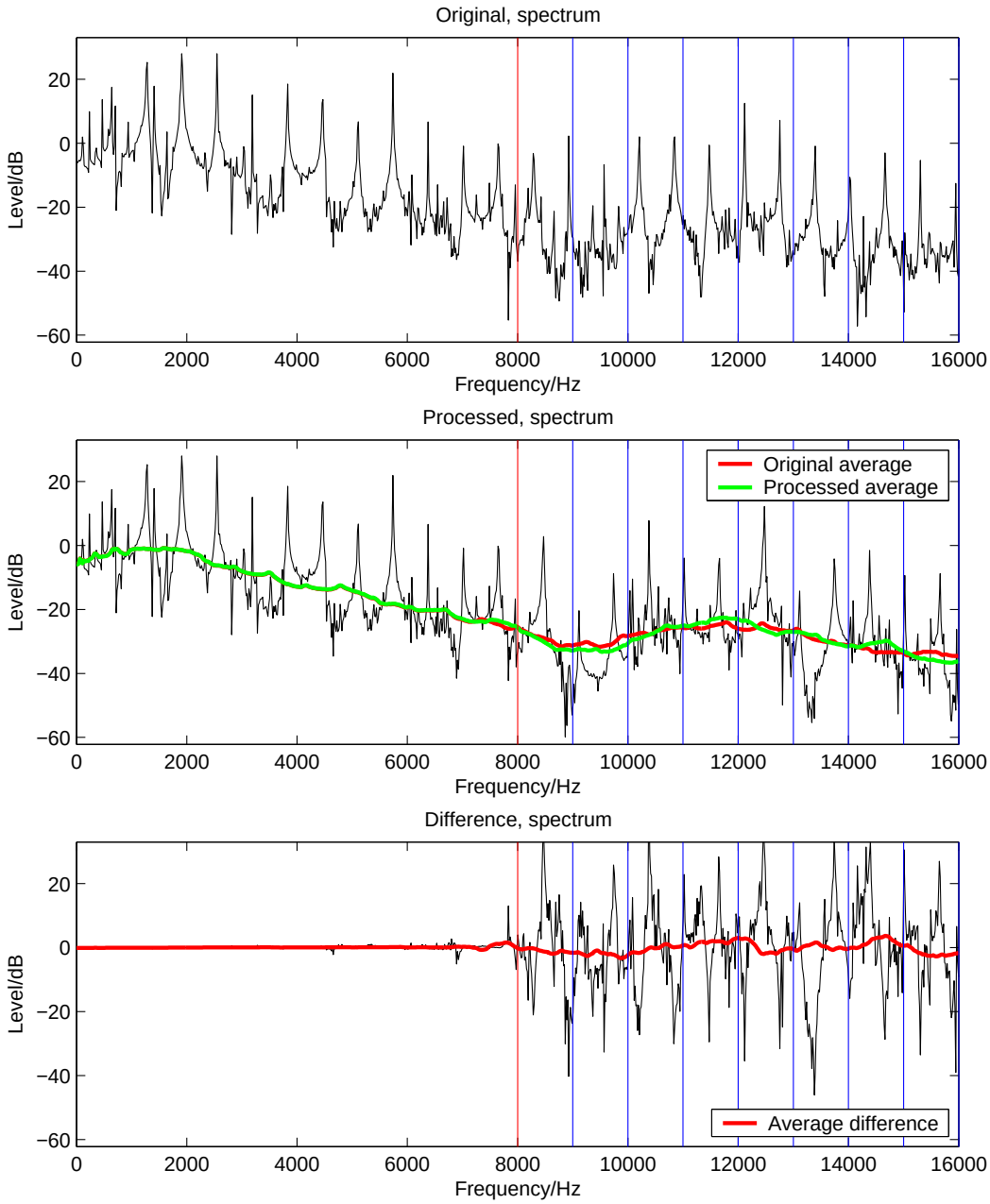

Figure A.7: Example frame, “sm01”, MDCT-based method, 8 uniform subbands, half copy.

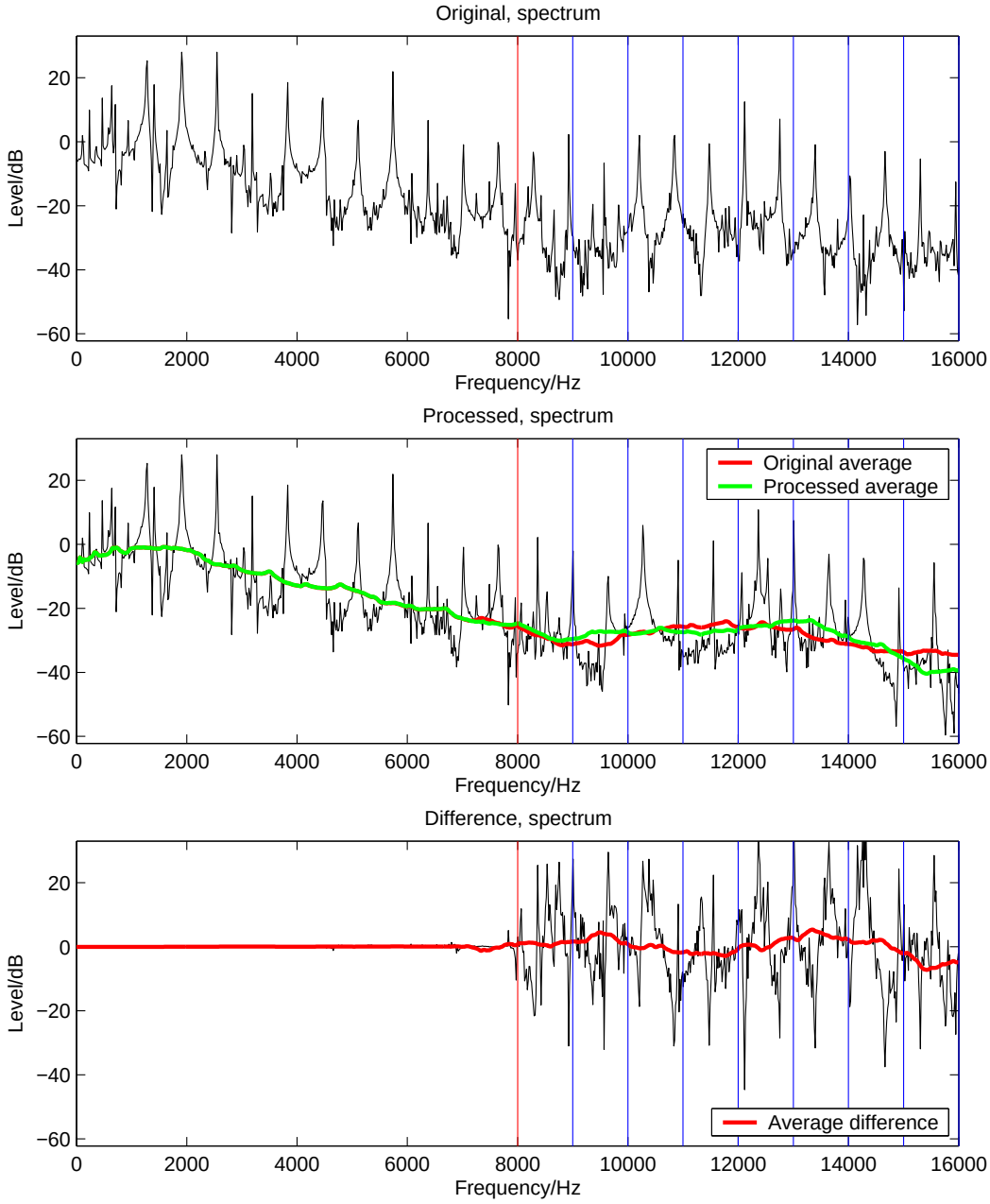

Figure A.8: Example frame, “sm01”, MDCT-based method, 8 uniform subbands, mirrored half copy.

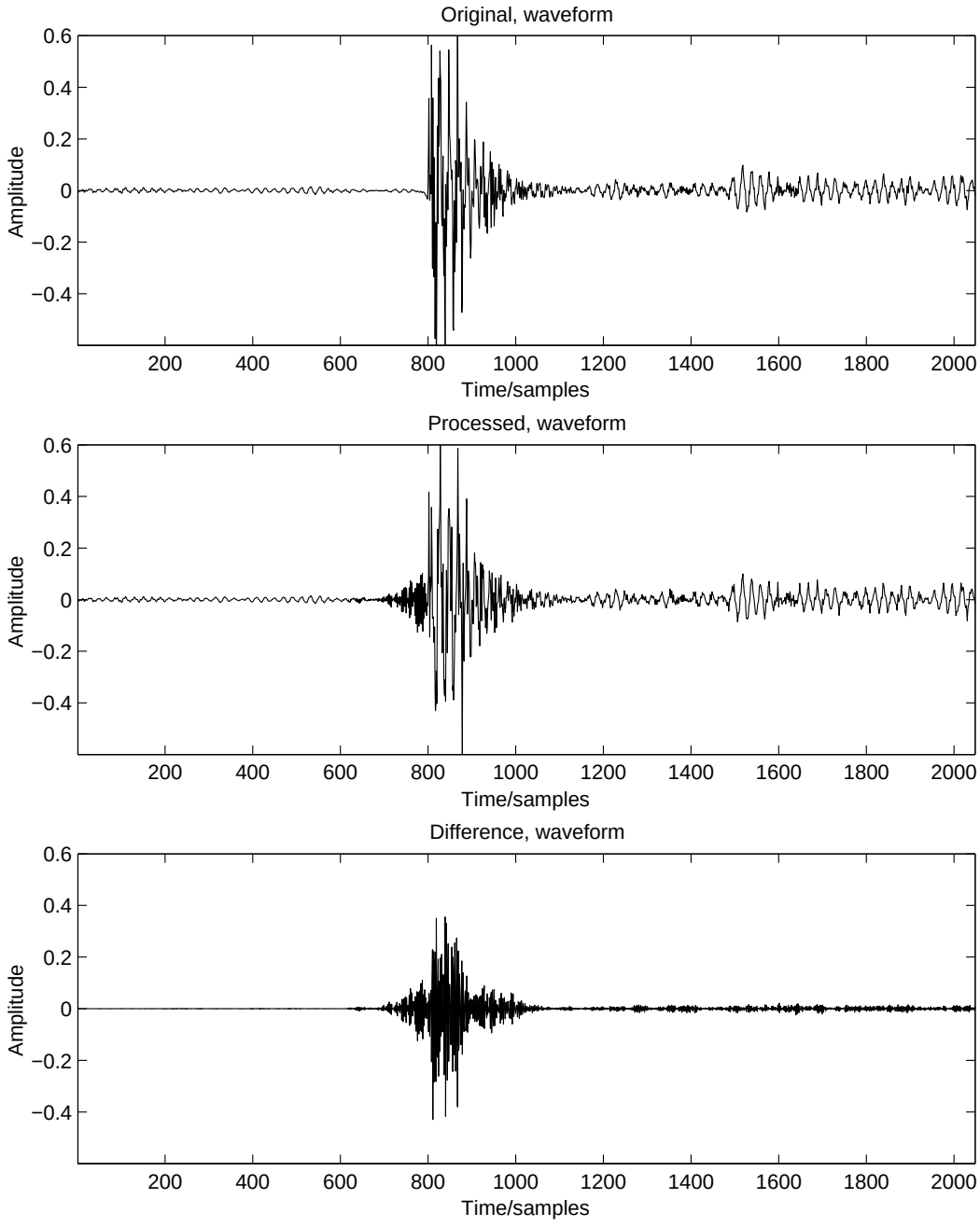

Figure A.9: Example frame in time domain, “si02”, during a transient.

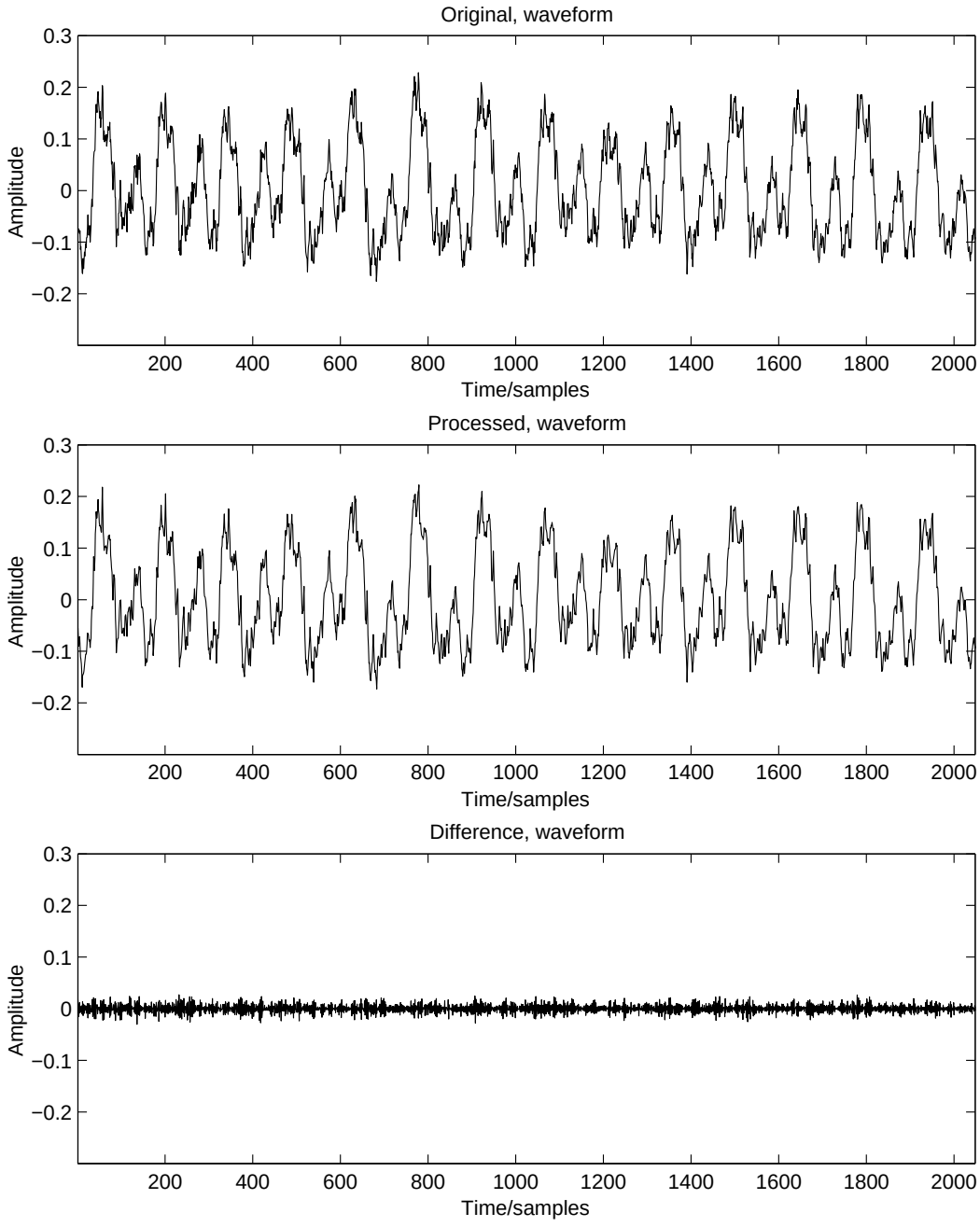

Figure A.10: Example frame in time domain, “si01”, tonal situation.

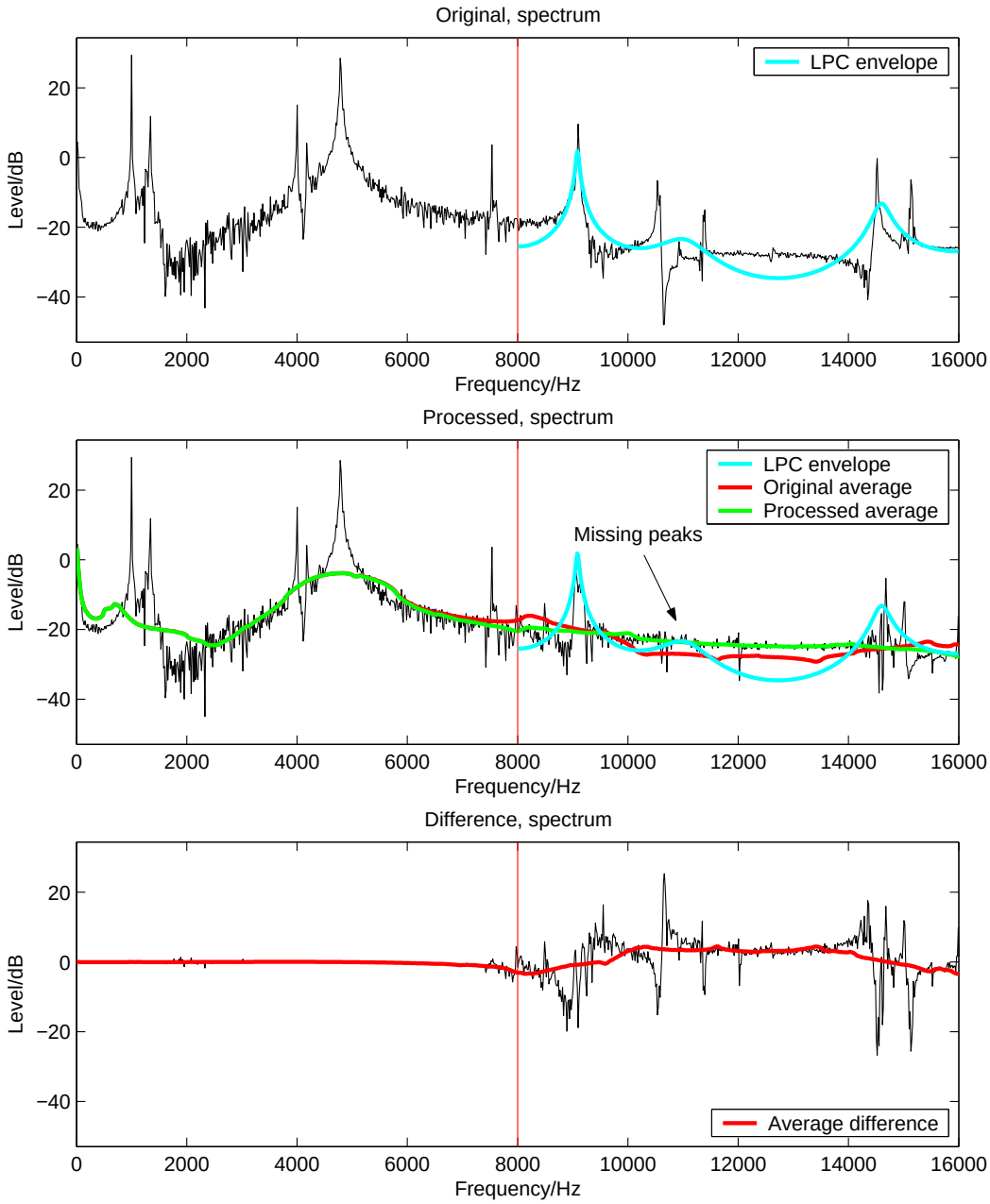

Figure A.11: Problematic situation, “sm02”, LPC-based method, 7 coefficients.

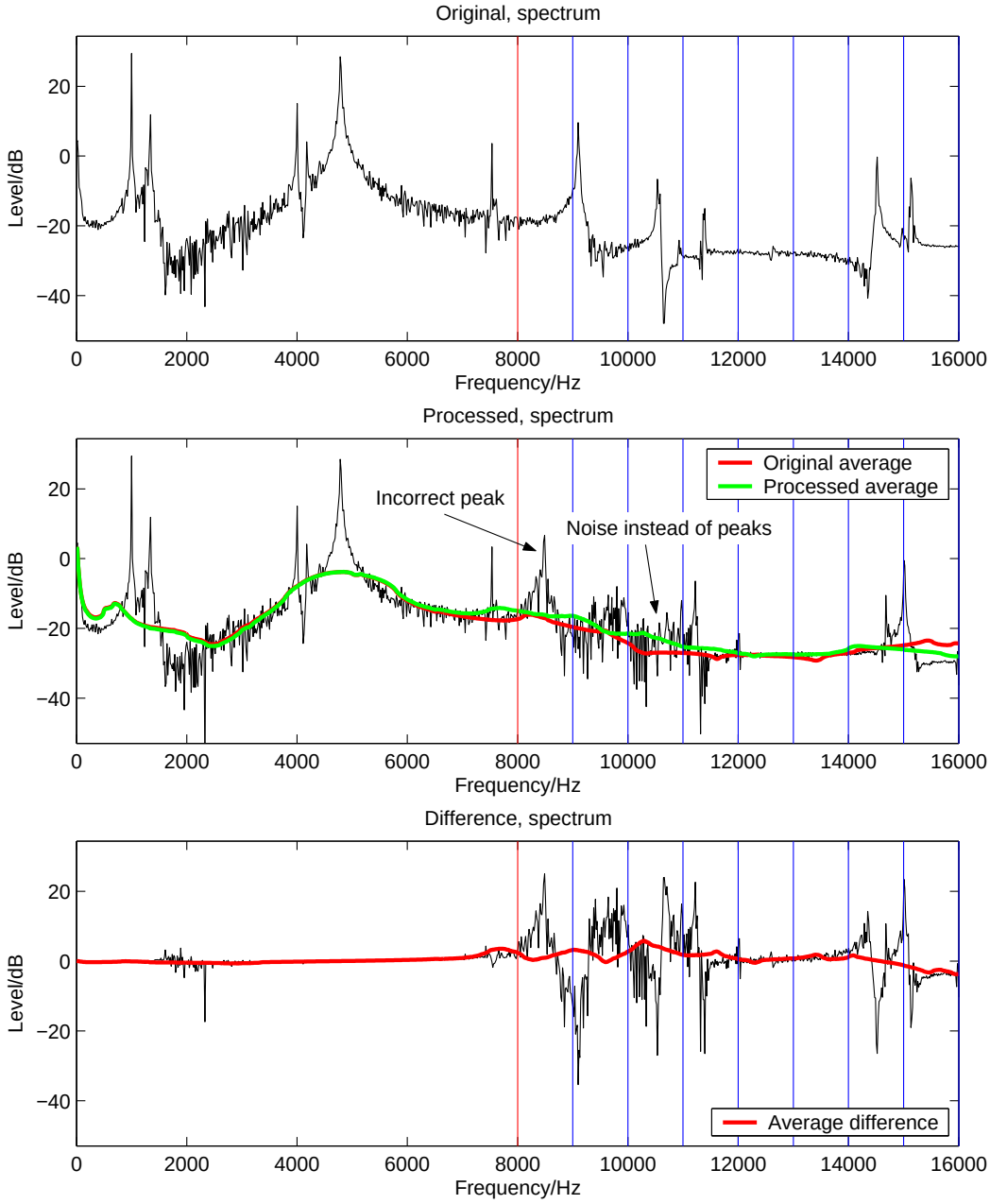

Figure A.12: Problematic situation, “sm02”, MDCT-based method, 8 uniform subbands, mirrored full copy.