Master's Thesis

Frequency-Domain Bandwidth Extension for Low-Delay Audio Coding Applications

by

Stanislaw Gorlow

A thesis submitted to
the Faculty of Electrical Engineering and Information Technology
at Ilmenau University of Technology
in partial fulfillment of the requirements
for the degree of
Master of Science in Electrical Engineering and Information Technology

Thesis Advisor: Univ.-Prof. Dr.-Ing. Gerald Schuller
Supervising Tutor: Dipl.-Ing. Michael Werner
Date of Issue: July 1, 2009

January 7, 2010
Ilmenau, Thuringia

URN: urn:nbn:de:gbv:ilm1-2014200076

Publisher: Universitätsbibliothek Ilmenau / ilmedia, 2014

ACKNOWLEDGEMENTS

First of all, I would like to thank Professor Gerald Schuller and Michael Werner for their guidance and support during the entire thesis process. Moreover, I would like to thank Michael in particular for being not only a tutor but also a reviewer providing me with constructive advice. I would also like to express my deep gratitude to my parents, who supported me financially over the last two years, allowing me to fully concentrate on my studies. Last but not least, my appreciation is due to everyone who participated in the listening test bringing a whole lot of patience. Their names in alphabetical order are Benjamin, Christian, Dominik, Florian, João Paulo, Kristina, Maija, Marie, Markus, Martin, Michael S., Michael W., Dr.-Ing. Mike Wolf, Tanja, Tolomej, and Vadim.

CONTENTS

1. INTRODUCTION

In 1935 Homer Dudley from Bell Telephone Laboratories had the United States Patent Office grant him a patent for an invention that should go down in the annals as the voice encoder, which is more commonly known as the *vocoder* [1] today. Instead of sending the complex signal wave itself, the vocoder makes use of the principle that speech may be resolved into invariable factors, such as the vibrations of the vocal cords, and into variable factors, such as the changes of pitch of the vocal cords and the various modulations effected by the lips, tongue, palate, etc. The fixed features appearing in a speech signal are understood to be oscillatory in nature and the variable features are modulatory.

By the arrangement disclosed in the vocoder, speech is instantaneously analyzed to determine the set of variable parameters, which define the unknown or variable elements of the speech signal. The fixed factors, such as the relatively high frequency vibration due to the vocal cords or to the hissing sounds of the air rushing through passages, are not transmitted. On the other hand, the variable parameters are transmitted in separate channels to the speech synthesizer. These parameters may correspond to the volume of energy in different frequency ranges of the voice and in the variations of pitch of the voice sounds. To reproduce the original sound, the information received in the several channels is combined by the synthesizer with waves from local sources corresponding to the invariable characteristics of speech.

In speech, two types of frequency spectrum are used alternatively: a continuous spectrum in the case of hissing or unvoiced sounds, and, in the case of voiced sounds, a discrete spectrum with a variable fundamental frequency with upper harmonics present to a relatively high frequency. Hence, the local source provided at the synthesizer preferably is such that the waves supplied can have either type of spectrum. The type is determined in response to the information transmitted from the sending end of the

system with regard to the presence or absence of a fundamental frequency in the speech wave and the magnitude of any such fundamental frequency.

Advantage is taken of the fact that much of the information ordinarily transmitted is of an invariable or predictable character, due to the general uniformity of the speech producing organs from person to person. By reproducing such predictable information artificially at the receiving end, it need not be transmitted from the sending end. Thus, effective use is made of the foreknowledge of the fixed characteristics of the signal source, with the result that the frequency bandwidth necessary for transmission can be reduced.

In a concrete realization, the vocoder is a system in which a speech signal is analyzed for, e. g., its fundamental frequency and for the average power in properly chosen frequency subbands. These derived signals should give as many independent variable quantities as the number of independent variables involved in the production of speech. This information is transmitted and then used at the receiving end by means of a synthesizer to fashion waves from a local multi-frequency oscillator into a very close copy of the speech at the sending end, so far as the ear can determine. Since the ear is the ultimate observer of a speech sound, the characteristic of the ear is very important in determining what frequency bands to use. It has long been known that the ear is essentially of such a nature as to observe the logarithm of frequency rather than frequency directly. The original speech band is therefore divided into frequency bands having a constant percentage of increment from the lowest frequency up to the top frequency, rather than with frequency bands of equal width [2].

Since the late 1970's, most vocoders avail themselves of linear prediction for the estimation of a speech signal's spectral envelope. In *linear prediction coding* (LPC) an all-pole filter replaces the bandpass filter bank of its predecessor. One advantage of the LPC model is its inherently low algorithmic delay coming from the fact that the future values are a linear superposition of previous samples. Modern vocoders provide a reasonably good simulation of voice with as little as 2.4 kbps[1] of data. The ITU G.729, which is encountered in many telephone networks, offers great voice quality at a data rate of 8 kbps. Other LPC-based vocoding schemes are LPC-10, *Code Excited*

[1] Kilobit per second

Linear Prediction (CELP), or *Mixed Excitation Linear Prediction* (MELP). More on speech coding is found, e. g., in [3].

During the last years, the Fraunhofer Institute for Digital Media Technology (IDMT) has developed a perceptual coder suitable for speech and music likewise, based on a concept that pretty much resembles the LPC approach. The new coding scheme, which was given the name *Ultra Low Delay* (ULD), separates irrelevancy reduction, i. e. temporal and spectral noise shaping, from redundancy reduction. Inaudible and hence irrelevant information is removed from the signal by an all-zero pre-filter, which is controlled by a psychoacoustic model in such a way that the frequency response of the pre-filter coincides with the inverse of the masked threshold. Predictive coding is used for this purpose, whereas now the masked threshold rather than the short-term power spectrum is modeled. The perceptual model itself is based on a subband decomposition of the observed signal. Redundancy in the pre-filtered and uniformly quantized signal is reduced via lossless coding using an extended version of the well-known *Least Mean Squares* (LMS) algorithm. The prediction error is entropy coded and transmitted to the decoder. A reciprocal all-pole post-filter reshapes the decoded residual keeping the quantization noise below the masked threshold, and thus inaudible. Further details on ULD coding are given in [4], [5], [6], [7], [8], [9], [10], and [11].

Due to its very low delay (less than 10 ms) and a good compression performance, the ULD coder opens the floodgates for time-critical applications like video conferencing, digital microphones in live or studio settings, in-ear monitoring, wireless loudspeakers, or live music sessions over the Internet, e. g. These applications are out of range for the traditional transform coders, as their encoding-decoding delay reaches up to several hundreds of milliseconds [12].

One of the current state-of-the-art codecs for natural audio is MPEG's *Advanced Audio Coding* (AAC) scheme [13] providing transparent or near transparent audio quality at 64 kbps per audio channel. AAC is a conventional transform coder that uses a filter bank to decompose the signal into its constituent frequency components. A sophisticated perceptual model together with enhanced noise shaping techniques removes the irrelevant part of the information. The allocated bits for the audible part are redundancy minimized by the use of noiseless or lossless coding, to provide

the maximum compression within the constraint of giving the highest possible audio quality. A high compression ratio, however, demands for many filter bank channels or subbands to implement an efficient masking model. The high number of subbands in turn leads to a high encoding-decoding delay. Further delay sources are a look-ahead buffer for block switching decisions and a buffer for bit-rate smoothing. An overview of the basics of high-quality low-bit-rate audio coding is given, e. g., in [14], [15].

The MPEG-4 low-delay codec AAC-LD [16] is derived from the architecture of AAC inheriting the capabilities and the shortcomings of the latter. Its main application area is full-duplex real-time communications. To reduce the algorithmic delay of AAC, the frame size is halved, and so is the length of the analysis-synthesis window. Block switching is removed, while instead, to lessen pre-echo artifacts in the case of transient signals, two window shapes are used alternatively. A sine window is employed for stationary signal sections, whereas a low-overlap window is used in transient passages. In order to reach the desired target delay, no bit reservoir is used at all. Despite of the described modifications, the 20 ms of delay at 48 kHz sampling rate are still considerably high. Moreover, the reduced number of subbands implicates a reduced coding efficiency, resulting in a higher data rate or a lower audio quality. For a comprehensive overview of audio coding principles, the interested reader is referred to [17].

For very low bit rates, i. e. bit rates under 64 kbps, the MPEG-4 standard provides for additional tools, which yet allow the content author to achieve a high coding quality, with *Spectral Band Replication* (SBR) [18] being one of them. SBR is a bandwidth extension tool that recreates the high-frequency portion of a downsampled audio signal from its low-frequency portion in a perceptually accurate manner. The combination of AAC and SBR, which is better known as the *High-Efficiency Advanced Audio Coding* (HE-AAC) profile, significantly enhances the coding performance of a stand-alone AAC system, by either lowering the bit rate for a given quality level or by improving the audio quality at a given bit rate [19]. Because of its excellent compression performance, HE-AAC is widely spread in digital broadcasting and in streaming networks with limited resources [20], [21], [22].

Taking a closer look inside SBR, one may be astonished by its similarity with the vocoder principle. First, both schemes represent an analysis-synthesis system built of

bandpass filters. Second, the SBR tool as well as the vocoder measures the short-term energy spectrum, thus following the signal's envelope over time. In either case a data reduction is achieved by transmitting a quite compact description of the analyzed signal, rather than the signal itself, be it in the raw or in a perceptually coded form. And finally, both schemes require a local excitation signal to be envelope adjusted according to the signaled data, in order to obtain a replica of the original wave.

But for all the parallels, to acquire a high-quality copy it is not sufficient to model the excitation process by a pulse train or random noise, or a mixture of both, as the number of independent variables involved in the creation of music and other complex sounds is much higher, whereas the predictable factors are strongly reduced. Since a wideband excitation spectrum can stem from one or from multiple sources, like vocal cords, strings, reeds, etc., it constitutes a multifaceted mixture of different frequency components. The excitation sound is then filtered by a resonator, such as the vocal tract or the body of a musical instrument, giving the voice or the musical instrument its characteristic tone color or timbre. A bandwidth limitation of a quasi-stationary excitation signal, e. g., is equivalent to a truncation of the harmonic series, which alters the timbre and the sound is perceived as "muffled" or "dull". The SBR tool recreates these seemingly lost high-frequency components without creating annoying artifacts. More bandwidth extending techniques are presented in [23]. However, as they are outside the scope of this thesis, they are not discussed here.

The thesis is organized in seven chapters. After stating the goal and the problem in Chpt. 2, the former is pursued through the following stages: First, Chpt. 3 gives a detailed description of the standard SBR method. Chpt. 4 then presents the new features of the delay-optimized derivative, namely *Low-Delay Spectral Band Replication* (LD-SBR), which is incorporated into the MPEG-4 *Enhanced Low-Delay Advanced Audio Coding* (AAC-ELD) scheme [24]. The proposed frequency-domain low-delay bandwidth extension is explained in Chpt. 5, followed by the results of a subjective listening test in Chpt. 6. Chpt. 7 finally summarizes the thesis by drawing conclusions and giving suggestions for potential future work.

2. PROBLEM STATEMENT

2.1 Goal

MPEG's low-delay audio codecs AAC-LD and AAC-ELD are capable of providing natural audio in perceptually high quality with an algorithmic system delay[1] in the range from 20 to 46.9 ms. The AAC-ELD scheme, in particular, which is basically formed out of AAC-LD and SBR, produces high-quality wideband audio at bit rates between 24 kbps and 48 kbps maintaining a relatively low algorithmic delay of 31.3 ms, where 1.3 ms are due to SBR, at 48 kHz sampling rate. These are three crucial factors for AAC-ELD to conquer the telecommunications sector. To draw a comparison, at the same sampling rate Fraunhofer's latest ULD coding scheme delivers comparable high-quality audio at a constant bit rate of 64 kbps per channel [6], [8], [11]. Its algorithmic delay of merely 5.3 ms corresponds to a frame size of 256 samples.

Even though low-delay solutions for speech and audio do exist, particular time-critical applications require a coding system that offers both: an algorithmic delay of less than 10 ms *and* a very low bit rate, while delivering high-quality digital audio at the same time. One might conclude that an adaptation of the SBR technique to the stringent delay constraint could –in combination with the ULD coder– turn out to be the remedy sought.

2.2 Problem

The SBR system is a compound of distinct algorithms that are strongly interwoven with the frame size, since the estimator functions in use require a minimum number of subsamples for consistency. The common block size for SBR in HE-AAC, e. g., is 2048,

[1] Includes encoding and decoding

bearing 32 subsamples for each of the 64 filter bank channels. LD-SBR in AAC-ELD operates on 960-length blocks, reducing the number of subsamples to 15. For a ULD plus SBR scheme, however, as few as 4 subsamples per filter bank channel are available. The 640-length filters of the filter bank with a correspondingly long delay are another issue.

Although a delay-optimized modulated filter bank would provide the solution to the latter problem, an effortless replacement of the filter bank yet does not guarantee the correct operation of SBR under the new conditions. The task is therefore to elaborate a low-delay version of SBR, which can be used together with ULD or any other coder with similar characteristics. The new scheme should preferably preserve the overall performance of SBR with regard to audio quality without considerably increasing the bit rate.

3. MPEG-4 SPECTRAL BAND REPLICATION

3.1 Introduction

Spectral Band Replication is a hybrid[1] high-quality bandwidth extension technique for speech and natural audio. It is an add-on to a conventional waveform coder, referred to as the *core* coder, rather than an audio coder itself. SBR can be seen as a preprocessor to the core encoder and as a postprocessor to the core decoder as depicted in Fig. 3.1. In the preprocessing step signal characteristics are analyzed and a moderate amount

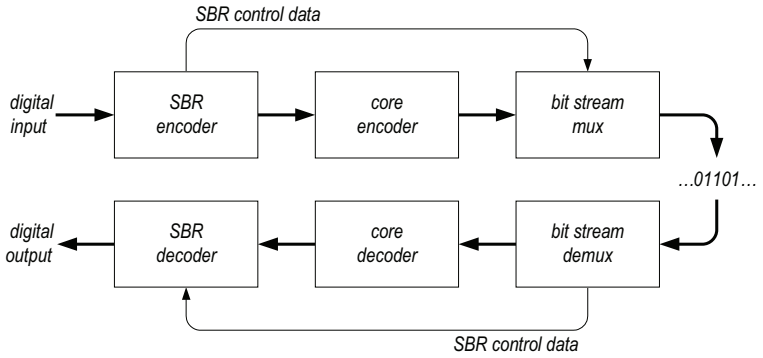

Fig. 3.1: *SBR as an add-on to a core codec*

of SBR specific data –usually a small fraction of the overall data rate– is stored, which is then used in the postprocessing step to reconstruct the broadband signal. The core encoder codes the low-frequency portion, alias the *low band*, of the original audio signal up to a chosen cutoff frequency. This frequency is labeled the *crossover frequency* between the low-frequency band and the high-frequency band, or short the *high band*. The SBR postprocessor reconstructs the high band from the decoded low band in a perceptually accurate manner, forming a wideband output signal. In the general case the core encoder operates at half the sampling frequency of SBR, resulting in a

[1] In the used context, the term "hybrid" denotes a parametric coding system that describes the correlation of two waveform signals, and therefore needs a reference signal for the excitation.

better frequency resolution of the core encoder's filter bank, which is beneficial for the exploitation of simultaneous-masking effects. This and the fact that only the low band needs to be redundancy coded boosts the coding gain of the entire system.

The driving idea behind SBR is the assumption that the low- and the highband characteristics of an audio signal are strongly correlated with each other. A signal with a distinct harmonic character in the low-frequency region is assumed to basically maintain its harmonic structure in the high-frequency region. Similarly, a noisy signal is assumed to carry its noise-like character from the low frequencies over to the high frequencies.

SBR also incorporates additional tools like *inverse filtering*, *adaptive noise addition*, and *sinusoidal regeneration* for signals that do not fit this simple model. The necessity to track partials, which is essential to sinusoidal modeling [25], spots SBR closely related to parametric methods. Moreover, due to the method of highband regeneration, the short-term synchronization of the high band with the low band, i. e. the temporal alignment, is close to optimal. A transient in the low band translates almost perfectly to the high band. A partial in the high band persists in time as long as the corresponding fundamental frequency is found in the low band. A coarse overview of SBR principles is also given in [26].

3.2 Complex quadrature mirror filter bank

Cosine modulated filter banks are often used in perceptual waveform coders and there are various efficient implementations [27], [28]. A modification of real-valued subband signals, however, creates audible distortion in the processed signal which makes this class of filter banks unsuitable for spectral shaping operations. These impairments do not occur, if a cosine modulated filter bank is extended to form a complex exponential modulated filter bank with complex-valued subband signals. The following pages give a brief overview of the basic concept behind a complex exponential modulated filter bank which in the forthcoming chapters will be termed as *complex quadrature mirror filter* (CQMF) bank [29].

3.2.1 Cosine modulated filter bank

The analysis filters $h_k(n)$ of a cosine modulated filter bank are obtained by cosine modulating an evenly symmetric lowpass prototype filter $p_0(n)$ according to

$$h_k(n) = p_0(n) \cos\left\{\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}-\frac{M}{2}\right)\right\}, \tag{3.1}$$

where $k = 0, 1, \ldots, M-1$, M is the number of filter bank channels, and $n = 0, 1, \ldots, N$, where N is the prototype filter order. Each of the analysis filters $h_k(n)$ bears real-valued subband samples in the case of real-valued input signals. The subband samples are decimated by a factor of M, so that the filter bank is *critically sampled*. The choice of the prototype filter $p_0(n)$ decides whether the filter bank fulfills the *perfect reconstruction* (PR) criterion.

As the FOURIER transform of the cosine function is given by two real-valued axially symmetric DIRAC delta pulses, any cosine modulated filter bank possesses the inherent property of having two passbands for each of its subband filters: one in the positive frequency range and one in the negative. Staying in the frequency domain it can be shown that the principal alias terms arise from overlapping images of either the filter's negative passband with the positive passband, or vice versa. In the general case, a modification of the subband samples or the spectral coefficients leads to severe aliasing artifacts in the output signal. The issue of overlapping images is overcome by extending the cosine modulation to complex exponential modulation as described in the following section. For the sake of completeness, the synthesis filters are [30]

$$g_k(n) = p_0(n) \cos\left\{\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}+\frac{M}{2}\right)\right\}. \tag{3.2}$$

3.2.2 Complex exponential modulated filter bank

Extending the cosine modulation to complex exponential modulation yields the analysis filters

$$h_k(n) = p_0(n) \exp\left\{\jmath\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}-\frac{M}{2}\right)\right\} \tag{3.3}$$

and the synthesis filters

$$g_k(n) = p_0(n) \exp\left\{\jmath\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}+\frac{M}{2}\right)\right\} \tag{3.4}$$

using previous notation, where the imaginary unit is denoted by $\jmath$. Obviously the extension is formed by additionally sine modulating the same prototype filter $p_0(n)$ on the imaginary axis. Deriving from a real-valued input signal, the output of the complex exponential filter bank can thus be interpreted as the analytic version of the output from the cosine modulated filter bank, where the imaginary part is obtained by HILBERT transforming the real part. Hence, Eqs. (3.3) and (3.4) imply that the output of the synthesis is complex-valued. It can also be shown that the cosine modulated filter bank and its sine modulated counterpart share the same behavior with respect to the PR property.

Because of its analytic nature, the complex exponential modulated filter bank has one single passband in the positive frequency range for each channel. Eliminating the source for principal alias terms makes the aliasing cancellation constraint from the cosine (sine) modulated filter bank obsolete, and both the analysis and synthesis filters can be written as [30]

$$h_k(n) = g_k(n) = p_0(n) \exp\left\{\jmath\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n-\frac{N}{2}\right)\right\}, \tag{3.5}$$

with k, n, M, and N as defined in Eq. (3.1).

Due to the absence of principal alias terms, the resulting aliasing depends only on the spectral leakage characteristics of the prototype filter $p_0(n)$. The minimization of the overall amplitude distortion is subject to the filter bank design procedure. M-channel filter banks that do not achieve perfect reconstruction are collectively known as *pseudo* QMF (PQMF).

3.3 Encoding process

3.3.1 Overview

A schematic block diagram of the SBR encoder is shown in Fig. 3.2. Prior to passing the *pulse code modulated* (PCM) input signal on to the waveform encoder, it is band limited and decimated by a factor of two. In parallel the input audio is fed to the analysis CPQMF bank, which inherently provides an instantaneous energy measure

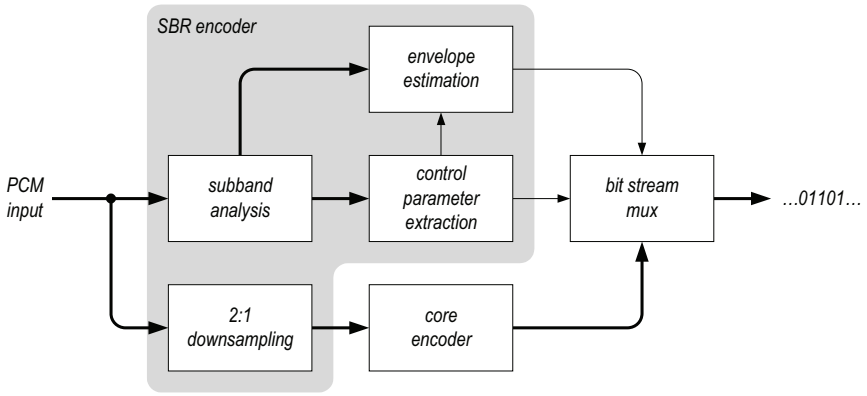

Fig. 3.2: *Generic scheme of the SBR encoder*

for each subband signal. Therefore, following PARSEVAL's theorem,

$$E = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X\left(e^{\jmath\Omega}\right)|^2 d\Omega, \tag{3.6}$$

the output of the filter bank is used to estimate the spectral envelope for every incoming data vector. As the core encoder operates at a different sampling rate from the SBR encoder, the whole encoding system is termed to operate in *dual rate* mode.

Apart from the envelope estimation, the subband samples undergo further analysis stages: Control parameter extraction –not necessarily in that order– includes signal-adaptive temporal subsample grouping, collecting guidance information for the *high-frequency reconstruction* (HFR) in the decoder, determining the tonal-to-noise ratio in the high band, and the detection of missing harmonics which cannot be reconstructed by merely shifting the low band towards higher frequencies. The spectral envelope coefficients together with the control parameters form the SBR data stream, which is entropy coded. If a constant overall bit rate is desired, the number of bits spent on SBR encoding is to be signaled to the underlying core encoder. One can even think of subsystems which exchange information in such a way that the optimum crossover frequency between the waveform coded low band and the SBR coded high band can be found adaptively for every processed time segment. Finally, the output of the SBR encoder is serially multiplexed with the output from the core encoder into one concatenated bit stream.

In the following, an in-depth description of SBR encoding tools and algorithms with reference to [31], [32], and [33] is provided. Since the cited 3GPP specifications do not

explain the basic principles, but only present the results in the form of pseudo-code algorithms, the insights that follow are to a great extent subject to my interpretation. In order to keep things clear and simple, the SBR tool is further postulated to operate on discrete-time mono signals.

3.3.2 Subband analysis

A uniform M-channel complex pseudo QMF bank (see Sec. 3.2.2) decomposes the real input signal into a time-indexed series of M complex subband signals, or spectral coefficients, each representing the signal amplitude localized within a frequency range of π/M rad. Using a 64-channel filter bank, at 48 kHz sampling rate this corresponds to a nominal bandwidth of 375 Hz per subband and a temporal resolution of 1.3 ms per time slot.

The input signal $x(n)$ is processed by a parallel bank of N-th order *finite-impulse-response* (FIR) bandpass filters $h_k(n)$. The analysis outputs

$$u_k(n) = h_k(n) * x(n) = \sum_{\nu=0}^{N} h_k(\nu)x(n-\nu), \tag{3.7}$$

$k = 0, 1, \ldots, M-1$, are decimated by a factor of M, yielding the maximally decimated subband sequences

$$x_k(n) = u_k(nM) = \sum_{\nu=0}^{N} h_k(\nu)x(nM-\nu). \tag{3.8}$$

Each of the bandpass filters $h_k(n)$ is a modulated version of a causal prototype filter $p_0(n)$ with the impulse response symmetric about $N/2$. It can be shown that if $p_0(n) = p_0(N-n)$, $0 \leqslant n \leqslant N$, with N an even integer, its FOURIER transform is of the form

$$P_0\left(e^{\jmath\Omega}\right) = \sum_{n=0}^{N} p_0(n)e^{-\jmath\Omega n} = W\left(e^{\jmath\Omega}\right)e^{-\jmath\Omega\frac{N}{2}}, \tag{3.9}$$

where $W\left(e^{\jmath\Omega}\right)$ is a real, even function of the normalized frequency Ω. Using Eq. (3.5), the frequency response of the respective analysis filters $h_k(n)$ is

$$H_k\left(e^{\jmath\Omega}\right) = e^{-\jmath\Omega_k\frac{N}{2}}P_0\left(e^{\jmath(\Omega-\Omega_k)}\right), \tag{3.10}$$

where $\Omega_k = \pi/M(k + 1/2)$ is the normalized center frequency of the k-th bandpass filter. Plugging Eq. (3.9) in into Eq. (3.10), we get

$$H_k\left(e^{\jmath\Omega}\right) = W\left(e^{\jmath(\Omega-\Omega_k)}\right) e^{-\jmath\Omega\frac{N}{2}}, \tag{3.11}$$

with $W\left(e^{\jmath(\Omega-\Omega_k)}\right)$ evenly symmetric about the modulation frequency Ω_k. From Eq. (3.11) it is easy to see that the filter bank is a generalized linear-phase system, the phase of which is the linear function $-\Omega N/2$. In this particular case, the constant group delay amounts to $N/2$, which is an integer.

The analysis filter bank that is found in [32] is a complex exponential modulated filter bank that employs a *discrete cosine transform* (DCT) for the calculation of the real part and a *discrete sine transform* (DST) for the calculation of the imaginary part. Just like the *discrete* FOURIER *transform* (DFT), both the DCT and the DST are finite-length transforms of the form

$$X(k) = \sum_{n=0}^{L-1} x(n)\phi_k^*(n), \tag{3.12}$$

where the basis sequences $\phi_k(n)$ in the orthonormal representation are either cosine or sine functions. The type-IV DCT, e. g., is defined as

$$X(k) = \sum_{n=0}^{L-1} x(n) \cos\left\{\frac{\pi}{M}\left(k + \frac{1}{2}\right)\left(n + \frac{1}{2}\right)\right\}. \tag{3.13}$$

At this point, let us redefine the analysis filters from Eq. (3.5) as

$$h_k(n) = p_0(n)e^{\jmath\Omega_k\left(n+\frac{1}{2}\right)}, \tag{3.14}$$

which effectively does not change anything on the modulation but only moves the constant phase offset from $-\Omega_k N/2$ to $+\Omega_k/2$. Evaluating the convolution sum from Eq. (3.8) yields

$$x_k(n) = \sum_{\nu=0}^{N} p_0(\nu)x(nM - \nu)e^{\jmath\Omega_k\left(\nu+\frac{1}{2}\right)}. \tag{3.15}$$

Let us further denote the product of L filter coefficients $\{p_0(\nu)\}_{\nu\in[0,L)}$, see Fig. 3.3, and L past input samples $\{x(nM - \nu)\}_{\nu\in[0,L)}$ counted backwards from the time instant nM

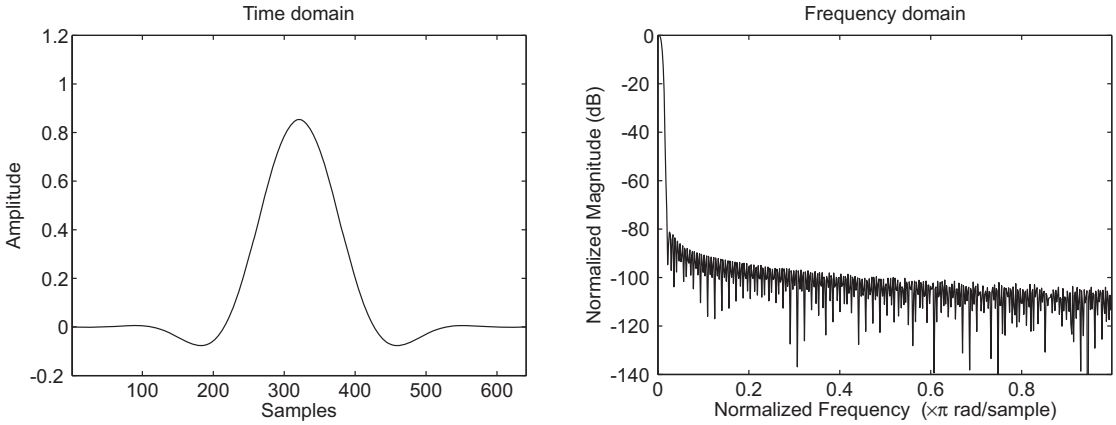

Fig. 3.3: *Time and frequency domain plot of the* 641*-length SBR window. The relative sidelobe attenuation is* -81.2 *dB. The mainlobe width* (-3 *dB) is* 0.015381.

as $\hat{x}^{-}_{nM}(\nu)$.[2] Eq. (3.15) can then be simplified to

$$x_k(n) = \sum_{\nu=0}^{L-1} \hat{x}^{-}_{nM}(\nu) e^{\jmath\Omega_k\left(\nu+\frac{1}{2}\right)}. \tag{3.16}$$

By assuming that the length of the filter's impulse response is an integer multiple of the number of subbands, i. e. $L = 2mM$, $m \in \mathbb{N}$, Eq. (3.16) can be reduced in complexity due to the $2M$-periodicity of the complex sequences $e^{\jmath\Omega_k\left(\nu+\frac{1}{2}\right)}$. In consequence we can rewrite Eq. (3.16) equivalently as

$$U(k) = \sum_{\nu=0}^{2M-1} u(\nu) e^{\jmath\Omega_k\left(\nu+\frac{1}{2}\right)}, \tag{3.17}$$

where

$$u(\nu) = \sum_{\mu=0}^{m-1} \hat{x}^{-}_{nM}(\nu + 2\mu M), \tag{3.18}$$

which is exactly the calculation rule in [13]. The connection between Eq. (3.17) and an M-length DCT (DST) can be established following, e. g., the *decimation-in-frequency* (DIF) approach [34]. For this purpose, the sequence $u(\nu)$ is split into two subsequences

$$v(\nu) = u(\nu), \tag{3.19}$$

$$w(\nu) = u(\nu + M), \tag{3.20}$$

[2] Equivalent to windowing in spectral analysis, where $p_0(\nu)$ is the L-point periodic window function and $\hat{x}^{-}_{nM}(\nu)$ is the windowed time sequence; here $L = N - 1$.

$\nu = 0, 1, \ldots, M-1$, so that Eq. (3.17) can be reformulated according to

$$\begin{aligned} U(k) &= \sum_{\nu=0}^{2M-1} u(\nu) e^{\jmath \Omega_k \left(\nu + \frac{1}{2}\right)} \\ &= \sum_{\nu=0}^{M-1} \left[v(\nu) e^{\jmath \Omega_k \left(\nu + \frac{1}{2}\right)} + w(\nu) e^{\jmath \Omega_k \left(\nu + M + \frac{1}{2}\right)} \right] \\ &= \sum_{\nu=0}^{M-1} \left[v(\nu) + \jmath e^{\jmath \pi k} w(\nu) \right] e^{\jmath \Omega_k \left(\nu + \frac{1}{2}\right)}. \end{aligned} \tag{3.21}$$

The factor $e^{\jmath \pi k}$ is $+1$ for even values of k and -1 for odd values of k. The even-numbered frequency samples and the odd-numbered frequency samples are therefore computed separately.

$$U(2k) = \sum_{\nu=0}^{M-1} \left[v(\nu) + \jmath w(\nu) \right] e^{\jmath \Omega_{2k} \left(\nu + \frac{1}{2}\right)}, \tag{3.22}$$

$$U(2k+1) = \sum_{\nu=0}^{M-1} \left\{ \left[v(\nu) - \jmath w(\nu) \right] e^{\jmath \frac{\pi}{M} \left(\nu + \frac{1}{2}\right)} \right\} e^{\jmath \Omega_{2k} \left(\nu + \frac{1}{2}\right)}, \tag{3.23}$$

$k = 0, 1, \ldots, M/2 - 1$. Using EULER's formula $e^{\jmath \varphi} = \cos \varphi + \jmath \sin \varphi$, with

$$u_e(n) = v(n) + \jmath w(n), \tag{3.24}$$

$$\begin{aligned} u_o(n) &= \left[v(n) - \jmath w(n) \right] e^{\jmath \frac{\pi}{M} \left(n + \frac{1}{2}\right)} \\ &= u_e^*(n) e^{\jmath \frac{\pi}{M} \left(n + \frac{1}{2}\right)}, \end{aligned} \tag{3.25}$$

Eqs. (3.22) and (3.23) are rearranged to

$$\begin{aligned} U(2k) &= \sum_{\nu=0}^{M-1} u_e(\nu) \cos \left\{ \frac{\pi}{M} \left(2k + \frac{1}{2}\right) \left(\nu + \frac{1}{2}\right) \right\} \\ &\quad + \jmath \sum_{\nu=0}^{M-1} u_e(\nu) \sin \left\{ \frac{\pi}{M} \left(2k + \frac{1}{2}\right) \left(\nu + \frac{1}{2}\right) \right\}, \end{aligned} \tag{3.26}$$

$$\begin{aligned} U(2k+1) &= \sum_{\nu=0}^{M-1} u_o(\nu) \cos \left\{ \frac{\pi}{M} \left(2k + \frac{1}{2}\right) \left(\nu + \frac{1}{2}\right) \right\} \\ &\quad + \jmath \sum_{\nu=0}^{M-1} u_o(\nu) \sin \left\{ \frac{\pi}{M} \left(2k + \frac{1}{2}\right) \left(\nu + \frac{1}{2}\right) \right\}. \end{aligned} \tag{3.27}$$

Comparing Eqs. (3.26) and (3.27) with Eq. (3.13), one can immediately see that the even-numbered frequency samples $U(2k)$ and the odd-numbered frequency samples $U(2k+1)$ are computed using the type-IV M-point DCT (DST) of the M-point complex

sequences $u_e(\nu)$ and $u_o(\nu)$, which are obtained from the two halves $v(\nu)$ and $w(\nu)$ of the real input sequence $u(\nu)$ on the basis of Eqs. (3.24) and (3.25). The procedure suggested by Eqs. (3.26) and (3.27) is once more summarized in Eq. (3.28).

$$U(k) = \begin{cases} \mathrm{DCT}_M^{\mathrm{IV}}\{u_e(n)\} + \jmath\mathrm{DST}_M^{\mathrm{IV}}\{u_e(n)\}, & k = 0, 2, \ldots, M-2 \\ \mathrm{DCT}_M^{\mathrm{IV}}\{u_o(n)\} + \jmath\mathrm{DST}_M^{\mathrm{IV}}\{u_o(n)\}, & k = 1, 3, \ldots, M-1 \end{cases}. \tag{3.28}$$

The magnitude response of a uniform 64-channel CPQMF bank designed using Eq. (3.14) is shown in Fig. 3.4. One can observe that all of the bandpass magnitude

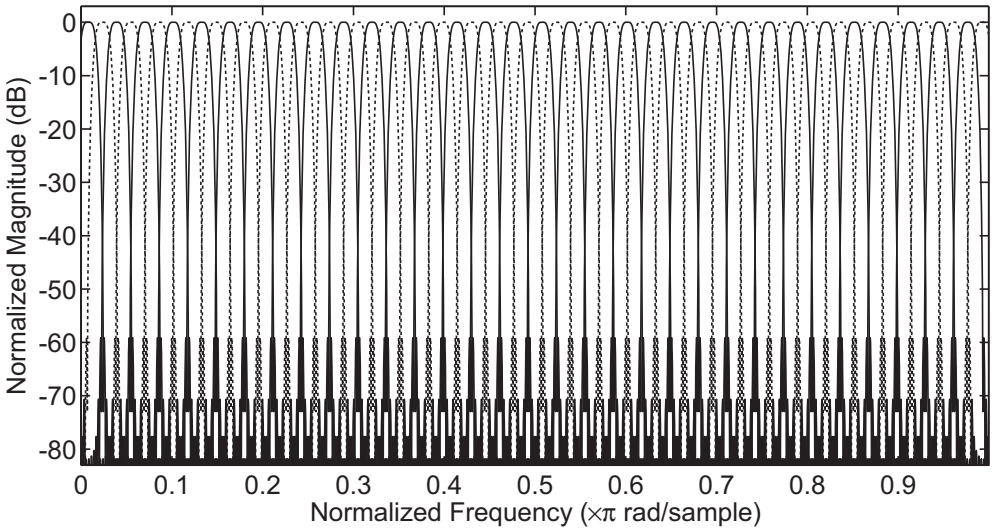

Fig. 3.4: *64-channel filter bank: magnitude response ($L = 640$). Even-numbered channels are marked with solid lines, odd-numbered with dotted lines.*

responses $|H_k\left(e^{\jmath\Omega}\right)|$, $k = 0, 1, \ldots, 63$, are frequency modulated versions of the lowpass prototype depicted in Fig. 3.3, and hence identical in their characteristics. All the subsequent processing steps are attached to the output of the filter bank, i. e., they are performed in the CPQMF domain.

Even though the phase of the filter bank is still linear with the constant group delay $-d/d\Omega \arg\left\{H_k\left(e^{\jmath\Omega}\right)\right\} = N/2$ unchanged, the bandpass filters from Eq. (3.14) introduce a constant phase term $\Omega_k(M+1/2)$, $M = N/2m$, m an odd integer, uniformly over all band $|\Omega| < \pi$, such that $\arg\left\{H_k\left(e^{\jmath\Omega}\right)\right\} = \Omega_k(M+1/2) - \Omega N/2$ is a function of the channel index k. Regardless of the discontinuities caused by the prototype filter characteristics, the phase within a band $|\Omega - \Omega_k| \lessapprox \pi/2M$ is continuous. The phase

of type-III DCT (DST) based analysis filters has a constant term of $\Omega_k M = (-1)^k e^{\jmath\frac{\pi}{2}}$, due to the fact that the phase shift, which is inherent to the modulated prototype filter, is not compensated by the particular type of transforms.

3.3.3 Transient detection

The transient detection algorithm, which is part of the *control parameter extraction* module (see Fig. 3.2), operates on subband samples, the number of which amounts to the length of one SBR frame. Aside from the presence of a transient, the algorithm estimates the position of the transient onset n_t. The time slot index n_t specifies the series of M subsamples $\{x_k(n)\}$, $k = 0, 1, \ldots, M-1$, in which the transient onset was detected.

The transient detection in frame i is accomplished through the following steps: Firstly, for each filter bank channel k a threshold value $t_i(k)$ is calculated by taking the square root of an unbiased estimator of the variance of the running short-term energies $|x_k(n)|^2$ according to

$$t_i(k) = \sqrt{\frac{1}{n_2 - n_1} \sum_{n=n_1}^{n_2} \left(|x_k(n)|^2 - \bar{x}_k \right)^2}, \tag{3.29}$$

where

$$\bar{x}_k = \frac{1}{n_2 - n_1 + 1} \sum_{n=n_1}^{n_2} x_k(n) \tag{3.30}$$

is the long-term energy average, and $[n_1, n_2]$ is the detection range. Then, the threshold from Eq. (3.29) is weighted by $1-\alpha$, $0 \leqslant \alpha \leqslant 1$, using the threshold value $t_{i-1}(k)$ from the previous frame and limited to a lower bound $t_{\min}$, yielding the final threshold value

$$\tilde{t}_i(k) = \max\left\{(1-\alpha)t_i(k) + \alpha t_{i-1}(k), t_{\min}\right\}. \tag{3.31}$$

In a second step, the mean energy of two adjacent subsamples in the negative distance $-d$ from the current position n is subtracted from the mean energy of two adjacent subsamples in the positive distance d, such that

$$\begin{aligned} \Delta e_k^{(d)}(n) = \frac{1}{2} \left\{ |x_k(n+d)|^2 + |x_k(n+d+1)|^2 \right\} \\ - \frac{1}{2} \left\{ |x_k(n-d)|^2 + |x_k(n-d-1)|^2 \right\}. \end{aligned} \tag{3.32}$$

Subsequently, the energy difference $\Delta e_k^{(d)}(n)$ is compared to the threshold value $\tilde{t}_i(k)$, and in case the inequality $\Delta e_k^{(d)}(n) > \tilde{t}_i(k)$ holds true, the time slot n is added to the round of potential onset candidates.

The "significance" of the energetic non-stationarity $s(n)$ at the time instant n is calculated as

$$s(n) = \sum_{k=0}^{M-1} \sum_{d=1}^{d_{\max}} \frac{\Delta e_k^{(d)}(n) - \tilde{t}_i(k)}{\tilde{t}_i(k)}, \tag{3.33}$$

with $\Delta e_k^{(d)}(n) > \tilde{t}_i(k) \ \forall \ d \in \{1, 2, \ldots, d_{\max}\}$. The first time slot n, $n_1 \leqslant n \leqslant n_2$, for which $s(n+1)$ falls below nine-tenths of $s(n)$, while $s(n)$ is above some heuristic threshold value $s_{\min}$, is declared the position of the transient onset n_t. The latter is passed on to the *time-frequency* (TF) grid selection algorithm, which is described in the following section.

3.3.4 Time-frequency grid selection

One major challenge of crucial importance for any encoding apparatus is to find the optimum time-frequency signal representation, which in most of the cases demands for a trade-off between coding gain and sound quality. As commonly known, transient signal passages require a high temporal resolution to preserve the dynamic structure of the observed segment and yet to have the quantization noise concealed by temporal masking. On the contrary, stationary regions are efficiently coded through the effects of simultaneous masking, for which a high-resolution signal spectrum is key.

Since the filter bank at hand was primarily designed for the purpose of high-frequency component regeneration, i. e. spectral signal shaping, the main objective of its application is to avoid pre- and post-echoes. The echoes are colored noise that is smeared over the analysis (synthesis) window and in the course becomes audible in the originally silent passages. The signal-adaptive time-frequency processing is achieved by grouping of subsamples in time and subbands in frequency, which is less restrictive compared to switching between a finite number of different modes of a non-uniform filter bank, as proposed in [35], to best match the short-term signal characteristics. Generically speaking, SBR divides quasi-stationary signal passages into longer time segments with higher frequency resolution, whereas transient passages are partitioned

into shorter time segments with lower frequency resolution, respectively.

The choice of the appropriate TF resolution for the current SBR frame is made upon the output of the transient detector (see Sec. 3.3.3). Based on this information, the TF grid selection algorithm determines the number of envelopes, the start and stop time borders of the envelopes, as well as their frequency resolution. There are two possible frequency resolutions for SBR envelopes: `high` and `low`. Each resolution is associated with a frequency band table. Frequency band tables contain subband indices that represent the start frequency borders of consecutive frequency bands, such that each frequency band is defined by a start frequency border and a stop frequency border. The frequency band table for low-resolution SBR envelopes is obtained by extracting a subset of the borders from the high-resolution table, while reducing the number of frequency bands by a factor of two. The high-resolution table is derived from a master frequency band table with linear or logarithmic scale, where the number of bands per octave may vary. Critical bands of the human auditory system are approximated by means of frequency warping [36]. The SBR range that is covered by the master frequency band table is implicitly derived from the opted bit rate and the sampling rate.

Before the grid generation, the current SBR frame is classified based on the knowledge of the preceding frame class and the position of the transient onset. The four available frame classes are: `FIXFIX`, `FIXVAR`, `VARFIX`, and `VARVAR`. `FIXFIX`-class frames are non-transient SBR frames, with the leading and the trailing time borders placed at the nominal frame boundaries. A supplementary function, the *frame splitter*, helps to decide whether the non-transient SBR frame is in addition to be divided into two envelopes of the same duration, e. g. when the energy levels of the two frame halves significantly differ. `FIXVAR`-/`VARFIX`-class sequences are indicative for sparsely transient signal passages, so they are separated by at least one non-transient `FIXFIX`-class frame. The leading time border of a `FIXVAR`-class frame equals the leading nominal frame boundary and is fixed, whereas the variable position of trailing time border is determined by the transient onset. `VARFIX`-class frames conclude transient signal segments having the trailing time border equal the trailing nominal frame boundary. If two (or more) successive SBR frames have high signal fluctuation, `VARVAR`-class

frames with variable time borders at both ends are used.

As the greater part of transient passages is locally limited, upon transient detection, with the predecessor frame of class `FIXFIX`, the grid selection algorithm expects a `FIXVAR-VARFIX` frame transition to occur. Rooted in this assumption, the present frame is regularly `FIXVAR` formed, while preparing borders compliant with the `VARFIX` scheme for the successor frame. Should the next SBR frame happen to be also transient, a new grid will be calculated, and the frame will be classified as `VARVAR`. Any sequences of SBR frame class transitions are thinkable, as long as the algorithm makes sure that the transitional phases between two transient SBR frames smoothly merge, and the boundaries (leading and trailing borders) of two consecutive frames coincide. To warrant this, the transient detection features a time look-ahead that reaches as far as a quarter-frame into the future frame.

Figure 3.5 illustrates an example situation when a transient sound is detected in the running SBR frame. One border is drawn at the position of the transient onset.

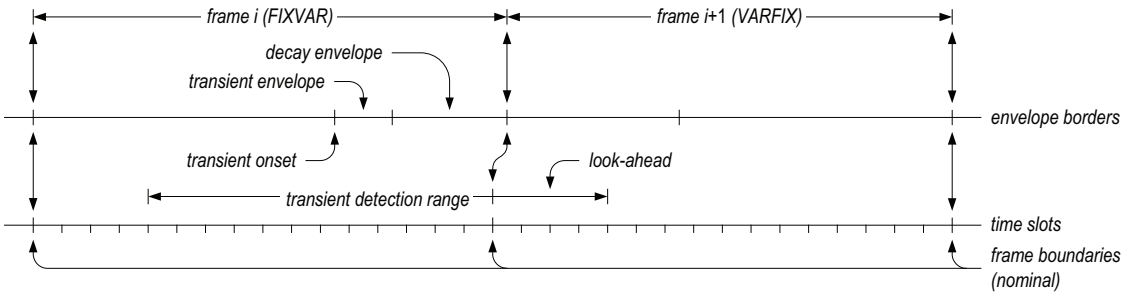

Fig. 3.5: *Example of a "sparse" transient: frame sequence, transient detection range inclusive of look-ahead, transient onset, and borders of spectral envelopes*

Two more borders are placed right behind to account for the decay period. Additional borders are inserted before and after the aforementioned borders in order to preserve the signal dynamics in the respective time segments. Both the transient and the decay envelope have a high temporal resolution, where the the decay envelope is twice as long, and therefore their frequency resolution is low. The rather stationary pre-transient and post-decay segments all have a high frequency resolution, as their time resolution is reduced. Restrictions relating to the number of SBR envelopes can be found in [13].

3.3.5 Envelope estimation

The envelope estimation falls back on the internal representation of the TF grid, which consists of the start (stop) time borders of all spectral envelopes within the range of the current SBR frame and the frequency resolution of each of these time segments. For each of the frequency bands from the signaled frequency band table a scaling factor is calculated by averaging the energy over the subbands that lie within the start and the stop borders of that band. The output of the envelope estimator is therefore given by

$$\bar{e}_\kappa(\nu) = \frac{1}{k_2(\kappa) - k_1(\kappa) + 1} \frac{1}{n_2(\nu) - n_1(\nu) + 1} \sum_{k=k_1(\kappa)}^{k_2(\kappa)} \sum_{n=n_1(\nu)}^{n_2(\nu)} |y_k(n)|^2, \tag{3.34}$$

where $\kappa = 1, 2, \ldots, \kappa_{\max}$ denotes the frequency band and $\nu = 1, 2, \ldots, \nu_{\max}$ refers to the envelope or time segment. Moreover, $k_1(\kappa)$ is the start border of frequency band κ and $n_1(\nu)$ is the start border of time segment ν, while $k_2(\kappa)$ and $n_2(\nu)$ are the corresponding stop borders. As is apparent from Fig. 3.6, $\{\bar{e}_\kappa(\nu)\}_{\kappa=1,2,\ldots,\kappa_{\max}}$ designates a staircase-shaped trajectory of the spectral envelope ν over a non-uniformly segmented frequency range.

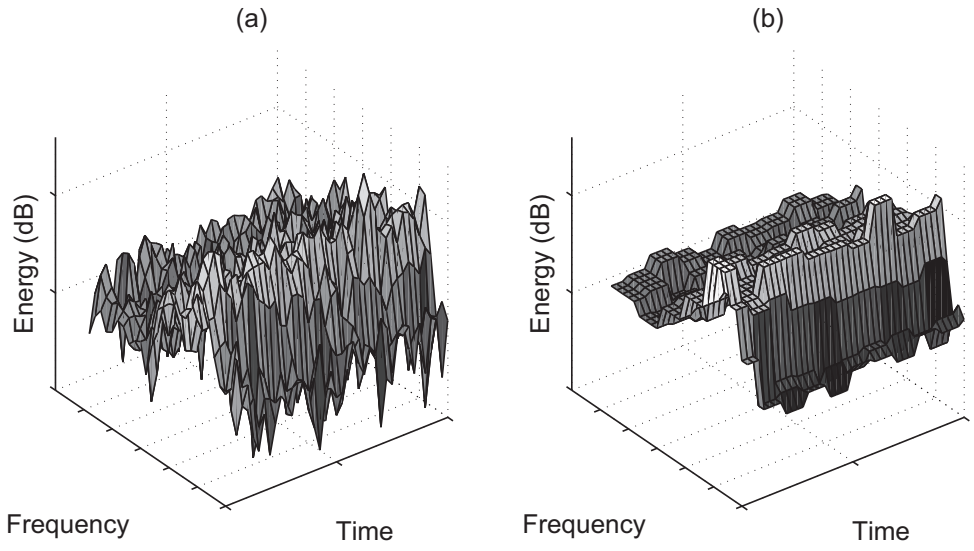

Fig. 3.6: *Highband signal: (a) original, (b) estimated envelope*

3.3.6 Additional control parameters

The high-quality performance of SBR for all kinds of sounds can be traced back to the fact that this technique employs auxiliary algorithms, which beneficially contribute to its perceptual ratings. They do so by tackling three key issues that occur with any HFR based method: improper noise level, spectral vacancies, and missing tonal components. The parameters that are necessary to compensate for these effects are estimated upon the original signal on the encoder side by means of a tonality measure that is explained below.

Tonality estimation

In this section it is shown how a tonality measure can be derived from the prediction gain of a linear predictor. Before doing so, however, some basic principles of linear prediction are presented to provide the reader with sufficient background knowledge for the subject of adaptive filtering. More details on adaptive filters can be found in [37].

A forward prediction error filter is a non-recursive (FIR) system that is defined by the polynomial

$$P(z) = 1 - \sum_{n=1}^{N} p_n^* z^{-n}, \tag{3.35}$$

where N is the prediction order, and p_n^* are the complex-valued filter coefficients. The prediction error $e(k)$ is given in the time domain by the difference equation

$$e(k) = x(k) - \hat{x}(k) = x(k) - \sum_{n=1}^{N} p_n^* x(k-n), \tag{3.36}$$

where $x(k)$ denotes a *wide-sense stationary* (WSS) random process with zero mean and $\hat{x}(k)$ is the estimated process. From Eq. (3.36) it is obvious that the estimate is determined by the past $x(k-n)$ of the WSS process. By introducing the observation vector of the input signal in the form of a finite time series

$$\mathbf{x}(k) = \begin{bmatrix} x(k-1) & x(k-2) & \cdots & x(k-N) \end{bmatrix}^{\mathrm{T}} \tag{3.37}$$

and the coefficient vector

$$\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \cdots & p_N \end{bmatrix}^{\mathrm{T}}, \tag{3.38}$$

Eq. (3.36) can be written more compactly by forming the inner product of the two vectors, as

$$e(k) = x(k) - \hat{x}(k) = x(k) - \mathbf{p}^{\mathrm{H}}\mathbf{x}(k), \tag{3.39}$$

with $\mathbf{p}^{\mathrm{H}}$ representing the HERMITIAN or conjugate transpose of $\mathbf{p}$. The task is to find the predictor coefficients $\mathbf{p}_{\min}$ that minimize the power of the prediction error $\mathrm{E}\left\{|e(k)|^2\right\}$, where $\mathrm{E}\{\cdot\}$ is the expected value operator, i. e. to determine $\mathbf{p}_{\min}$ such that

$$\mathrm{E}\left\{|e(k)|^2\right\} = \mathrm{E}\left\{e(k)e^*(k)\right\} \stackrel{!}{=} \min. \tag{3.40}$$

In a first step, we plug Eq. (3.39) in into Eq. (3.40) and we obtain, after some rearrangements, an expression for the cost function $J(\mathbf{p})$ according to

$$J(\mathbf{p}) \stackrel{\mathrm{def}}{=} \mathrm{E}\left\{|e(k)|^2\right\} = \sigma_x^2 + \mathbf{p}^{\mathrm{H}}\mathbf{R}_{xx}\mathbf{p} - \mathbf{p}^{\mathrm{H}}\mathbf{r}_{xx} - \mathbf{r}_{xx}^{\mathrm{H}}\mathbf{p}, \tag{3.41}$$

where $\mathbf{R}_{xx}$ denotes the autocorrelation matrix and $\mathbf{r}_{xx}$ denotes the autocorrelation vector, respectively. The elements of the autocorrelation matrix

$$\mathbf{R}_{xx} = \mathrm{E}\left\{\mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k)\right\} \tag{3.42}$$

conform to the the values of the autocorrelation sequence

$$r_{xx}(\kappa) = \mathrm{E}\left\{x^*(k)x(k+\kappa)\right\} = \mathrm{E}\left\{x(k)x^*(k-\kappa)\right\}, \tag{3.43}$$

for $\kappa = 0, 1, \ldots, N-1$. Hence, using the property that the complex conjugate of the autocorrelation sequence is equal to the original sequence with the argument changed in sign,

$$r_{xx}(-\kappa) = r_{xx}^*(\kappa), \tag{3.44}$$

the autocorrelation matrix $\mathbf{R}_{xx}$ can be expressed as

$$\mathbf{R}_{xx} = \begin{bmatrix} r_{xx}(0) & r_{xx}^*(1) & \cdots & r_{xx}^*(N-1) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}^*(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(N-1) & r_{xx}(N-2) & \cdots & r_{xx}(0) \end{bmatrix}. \tag{3.45}$$

The Eq. (3.45) further reveals that $\mathbf{R}_{xx}$ is a so-called HERMITIAN TOEPLITZ matrix, since $\mathbf{R}_{xx} = \mathbf{R}_{xx}^{\mathrm{H}}$. The autocorrelation vector $\mathbf{r}_{xx}$ is defined with the Eqs. (3.37) and

(3.43) according to

$$\mathbf{r}_{xx} = \mathrm{E}\left\{x(k)\mathbf{x}^*(k)\right\} = \left[\begin{array}{cccc} r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(N) \end{array}\right]^{\mathrm{T}}. \tag{3.46}$$

The mean signal power $\mathrm{E}\left\{|x(k)|^2\right\}$ is given by the value of the autocorrelation sequence at lag $\kappa = 0$,

$$\mathrm{E}\left\{|x(k)|^2\right\} = \mathrm{E}\left\{x(k)x^*(k)\right\} = r_{xx}(0) = \sigma_x^2, \tag{3.47}$$

which, as the process $x(k)$ is assumed to have zero mean, is the variance $\mathrm{var}\left\{x(k)\right\} = \sigma_x^2$ of the process.

In a second step, the problem stated in Eq. (3.40) is addressed. The coefficient vector $\mathbf{p}_{\min}$, for which the cost function from Eq. (3.41) attains its minimum value, is determined by setting the partial derivative $\partial J(\mathbf{p})/\partial \mathbf{p}^*$ of the cost function to zero, as stipulated by

$$\frac{\partial J(\mathbf{p})}{\partial \mathbf{p}^*} = \mathbf{R}_{xx}\mathbf{p} - \mathbf{r}_{xx} \stackrel{!}{=} \mathbf{0}. \tag{3.48}$$

This is motivated by the fact that the cost function $J(\mathbf{p})$ is a second-order function of the prediction coefficients. Consequently, the dependence of the cost function on $\mathbf{p}$ may be visualized as a bowl-shaped $(N+1)$-dimensional surface with N degrees of freedom. This surface possesses a unique minimum.

Finally, the set of homogeneous linear equations from Eq. (3.48) is solved by inversion of the autocorrelation matrix, provided that $\mathbf{R}_{xx}$ is nonsingular, yielding the well-known discrete Wiener-Hopf equation

$$\mathbf{p}_{\min} = \underset{\mathbf{p}\in\mathbb{C}^N}{\arg\min}\, J(\mathbf{p}) = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}. \tag{3.49}$$

In the following, one practical solution to Eq. (3.49), which bases upon a finite sample $\tilde{x}(k)$ of the stochastic process $x(k)$, is presented. In the literature, the described method is known as the *covariance method* or as the non-stationary approach [38].

The estimated linear prediction coefficients $\hat{\mathbf{p}}$, which are identical to the sign flipped parameters of an N-th order *autoregressive* (AR) model, are obtained from the estimate $\hat{\mathbf{R}}_{xx}$ of the autocorrelation matrix $\mathbf{R}_{xx}$ and from the estimate $\hat{\mathbf{r}}_{xx}$ of the autocorrelation vector $\mathbf{r}_{xx}$. The elements of both tensors are given by the values of the asymptotically unbiased estimator $\hat{r}_{xx}(\kappa)$ for the autocorrelation sequence $r_{xx}(\kappa)$. Assuming that $x(k)$

is ergodic, i. e. replacing the ensemble average by the time average, the estimate of the autocorrelation sequence is defined using the sample sequence $\tilde{x}(k)$, $0 \leqslant k < M$, $M \gg N$, according to

$$\hat{r}_{xx}(\mu, \lambda) = \hat{r}_{xx}(\mu - \lambda) = \frac{1}{M - N} \sum_{k=N-\mu}^{M-1-\mu} \tilde{x}^*(k)\tilde{x}(k + \mu - \lambda), \tag{3.50}$$

with $\mu = 1, 2, \ldots, N$, $\lambda = 0, 1, \ldots, N$. By substituting the true quantities in (3.49) by their estimates, we finally get the optimum coefficients of a second-order linear predictor. These are

$$\begin{aligned} \begin{bmatrix} \hat{p}_1 \\ \hat{p}_2 \end{bmatrix} &= \frac{1}{\det \hat{\mathbf{R}}_{xx}} \begin{bmatrix} \hat{r}_{xx}(2,2) & -\hat{r}_{xx}(1,2) \\ -\hat{r}_{xx}(2,1) & \hat{r}_{xx}(1,1) \end{bmatrix} \begin{bmatrix} \hat{r}_{xx}(1,0) \\ \hat{r}_{xx}(2,0) \end{bmatrix} \\ &= \frac{1}{\det \hat{\mathbf{R}}_{xx}} \begin{bmatrix} \hat{r}_{xx}(1,0)\hat{r}_{xx}(2,2) - \hat{r}_{xx}(1,2)\hat{r}_{xx}(2,0) \\ \hat{r}_{xx}(1,1)\hat{r}_{xx}(2,0) - \hat{r}_{xx}(1,0)\hat{r}_{xx}(2,1) \end{bmatrix}, \end{aligned} \tag{3.51}$$

where $\det \hat{\mathbf{R}}_{xx} = \hat{r}_{xx}(1,1)\hat{r}_{xx}(2,2) - |\hat{r}_{xx}(1,2)|^2$.

The prediction gain as a measure for the predictor's performance is given by

$$G_P = 10 \log_{10} \frac{\mathrm{E}\left\{|\hat{x}(k)|^2\right\}}{\mathrm{E}\left\{|e(k)|^2\right\}}, \tag{3.52}$$

i. e., it is the ratio between the mean signal power of the estimated signal and the mean error power in decibels (dB). The stronger the correlation is between the samples of the input sequence, the higher the gain [3]. This simple rule can likewise be used to determine the tonality of a signal. Therefore, let us suppose the tonality is defined accordingly to the prediction gain as

$$T_P = \frac{\mathrm{E}\left\{|\hat{x}(k)|^2\right\}}{\mathrm{E}\left\{|e(k)|^2\right\}}. \tag{3.53}$$

In the converged state, the prediction error power is minimized and thus given by the cost of the vertex $\mathbf{p}_{\min}$, with

$$\min \mathrm{E}\left\{|e(k)|^2\right\} = \min_{\mathbf{p} \in \mathbb{C}^N} J(\mathbf{p}) = J(\mathbf{p}_{\min}) = \sigma_x^2 - \mathbf{r}_{xx}^{\mathrm{H}}\mathbf{p} = \sigma_e^2. \tag{3.54}$$

The mean signal power of the estimate is formulated using the dot product notation for $\hat{x}(k)$ from (3.39), yielding

$$\mathrm{E}\left\{|\hat{x}(k)|^2\right\} = \mathrm{E}\left\{\mathbf{p}^{\mathrm{H}}\mathbf{x}(k)\mathbf{x}^{\mathrm{H}}(k)\mathbf{p}\right\} = \mathbf{p}^{\mathrm{H}}\mathbf{R}_{xx}\mathbf{p}. \tag{3.55}$$

Substituting $\mathbf{p}$ in (3.55) by $\mathbf{p}_{\min}$ from (3.49) and taking the complex conjugate of Eq. (3.55), we end up with a handy expression for the sought after signal power

$$\mathrm{E}\left\{|\hat{x}(k)|^2\right\} = \mathbf{r}_{xx}^{\mathrm{H}}\mathbf{p} = \sigma_{\hat{x}}^2. \tag{3.56}$$

In conclusion, we plug the results from (3.54) and (3.56) in into (3.53) and obtain a calculation rule for the tonality, alias the tonal-to-noise ratio [39], of the input signal, which is

$$T_P = \frac{\sigma_{\hat{x}}^2}{\sigma_e^2} = \frac{\sigma_x^2 - \sigma_e^2}{\sigma_e^2} = \frac{\mathbf{r}_{xx}^{\mathrm{H}}\mathbf{p}}{r_{xx}(0) - \mathbf{r}_{xx}^{\mathrm{H}}\mathbf{p}}. \tag{3.57}$$

In a practical realization, the second central moments are replaced by the respective estimates, so that, e. g., the second-order tonality measure is calculated in agreement with [32] as

$$\hat{T}_P = \frac{\hat{\mathbf{r}}_{xx}^{\mathrm{H}}\hat{\mathbf{p}}}{\hat{r}_{xx}(0,0) - \hat{\mathbf{r}}_{xx}^{\mathrm{H}}\hat{\mathbf{p}}} = \frac{\hat{p}_1\hat{r}_{xx}^*(1,0) + \hat{p}_2\hat{r}_{xx}^*(2,0)}{\hat{r}_{xx}(0,0) - \hat{p}_1\hat{r}_{xx}^*(1,0) - \hat{p}_2\hat{r}_{xx}^*(2,0)}, \tag{3.58}$$

with the prediction coefficients $\hat{p}_1$ and $\hat{p}_2$ given by Eq. (3.51).

Noise floor estimation

One of the features that enhance the perceptual performance of the SBR method is *adaptive noise floor addition* [40]. Adaptive noise floor addition addresses the issue of insufficiently audible noise content in the regenerated high band in such a way that the SBR processed output signal is perceived as naturally sounding. Thereby, in order to attain a noise level similar to the one of the original signal, stochastic components are added to the reconstructed high band. This approach is based upon the observation that the harmonic patterns of many musical instruments confirm a higher noise level in the high-frequency region put into relation to the low-frequency region. Moreover, there are harmonic sounds with little or no similarity between the noise levels of the high and the low band. A sole frequency transposition would therefore, in either case, suffer from lacking noise in the replicated high band.

Depending on the frame class, and hence the number of spectral envelopes within the frame, one or at most two noise floors are estimated. The noise floor level is defined as the ratio between the energy of the noise that is to be added to a particular frequency band and the energy of the high-frequency generated SBR signal in that

band. The estimation relies on a time-frequency grid for noise in the style of spectral envelopes. In particular, the start and stop borders of noise floors are by definition a subset of the start and stop time borders of spectral envelopes. The frequency resolution of noise floors is given by the noise floor band table, which is extracted from the low-resolution frequency band table via nonlinear decimation, granting an adequate temporal resolution.

The determination of noise content as such draws on the second-order tonality measure given by Eq. (3.58), where a large value of $\hat{T}_P$ suggests a tonal signal. For every noise floor frequency band κ in noise floor ν a tonality value for both the original and the regenerated high band (indicated by the $\prime$ symbol) is calculated through subband averaging according to

$$\bar{T}_\kappa(\nu) = \frac{1}{k_2(\kappa) - k_1(\kappa) + 1} \sum_{k=k_1(\kappa)}^{k_2(\kappa)} \hat{T}_P(k, \nu) \tag{3.59}$$

and

$$\bar{T}'_\kappa(\nu) = \frac{1}{k_2(\kappa) - k_1(\kappa) + 1} \sum_{k=k_1(\kappa)}^{k_2(\kappa)} \hat{T}'_P(k, \nu), \tag{3.60}$$

$\forall\ \kappa,\ \nu$, with $k_1(\kappa)$ and $k_2(\kappa)$ semantically identical to Eq. (3.34), where $\hat{T}_P(k, \nu)$ is the tonality estimate for the k-th subband of the ν-th noise floor envelope. The tonality of the HF reconstructed high band, $\hat{T}'_P$, is acquired in a manner resembling the analysis-by-synthesis approach, that is to say in compliance with the frequency transposition procedure employed in the decoder (see Sec. 3.5.4). Hence, given the tonality values, the actual noise floor level is calculated according to

$$\bar{q}_\kappa(\nu) = \min \left\{ \frac{\max \left\{ \frac{1}{4} \frac{\bar{T}'_\kappa(\nu)}{\bar{T}_\kappa(\nu)}, 1 \right\}}{\bar{T}_\kappa(\nu)}, q_{\max} \right\}, \tag{3.61}$$

where $q_{\max}$ is a limiting constant. The so obtained noise floor levels are additionally smoothed in time by third-order FIR lowpass filtering. If the inverse filtering level for noise band κ is below `high` (see below), the noise floor is obtained from

$$\bar{q}_\kappa(\nu) = \min \left\{ \frac{1}{\bar{T}_\kappa(\nu)}, q_{\max} \right\}. \tag{3.62}$$

Inverse filtering estimation

Adaptive spectral whitening [39] is another key feature, which makes SBR superior to other high-frequency reconstruction techniques that extrapolate a high band from a low band. It offers the means to control the tonal character of the reconstructed high band beyond the coarse envelope adjustment. The necessity comes from the fact that various audio signals, such as voice and mostly woodwind instruments like the saxophone, show a distinctive harmonic structure below approximately 5 kHz and a rather non-tonal or noise-like character like, e. g., the hi-hat towards higher frequencies. The adaptive noise floor addition is in the general case not sufficient to suppress the tonal character of the lowband regions, which serve the reconstruction of the high band. Following the analysis-by-synthesis approach, the detector on the encoder side therefore assesses the degree of spectral whitening to be applied to the decoder-generated high band, in order to have the tonal character within the SBR range as close to the original as possible. Adaptive spectral whitening varies over time and frequency, granting the maximum flexibility to impact on the harmonic content of the replicated high band.

Once more, two tonality values –one for the original signal and one for its replica– are computed for every noise floor frequency band from the noise floor band table. As the calculation rule is identical to the one given by the Eqs. (3.59) and (3.60), the tonality values $\bar{T}_\kappa(\nu)$ and $\bar{T}'_\kappa(\nu)$ from the noise floor estimation part are reused. After first-order FIR lowpass filtering in time, the smoothed tonality values are ranked by specific energy regions. A comparison of this regions yields one of the four possible inverse filtering levels: `off`, `low`, `mid`, and `high`. The level, which de facto regulates the degree of spectral whitening, is obtained by energy compensation, i. e. by also taking the energy level of the original highband signal into account.

Missing harmonics detection

The missing harmonics addition responds to the case when distinctive tonal components are lost in the HF generation process due to the lacking similarity of the frequency-shifted low band to the original highband signal. To overcome this problem, synthesized sinusoids are placed at the spectral position of the missing highband partials or simply

in the middle of the affected frequency band. As the magnitude of these sinusoidal components is implicitly given by the particular SBR envelope and the associated noise floor levels, only their frequency location needs be determined. This is accomplished by the missing harmonics detection algorithm in the encoder, see also [41], [42].

The detection of missing components is performed based on guide vectors using the highest frequency resolution table available. There are two detection stages with two different guide vectors. The length of both guide vectors accounts to the number of high-resolution frequency bands. Aside from its algorithm-specific role, each guide vector indicates whether the previous envelope already had a missing harmonic detected for a particular frequency band.

Similarly to the adaptive noise floor estimation, the first detection stage assays the ratio between the tonality estimate of the original highband signal and the tonality estimate of the decoder-like produced high band, given by

$$r_{\kappa,\max}(\nu) = \frac{\max\limits_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \hat{T}_P(k,\nu)}{\max\limits_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \hat{T}'_P(k,\nu)}, \tag{3.63}$$

where $k_1(\kappa)$ and $k_2(\kappa)$ are the frequency borders from the high-resolution table for band κ, and ν is the considered noise floor envelope. These ratios are then compared to adaptively selected threshold values, which are derived from the values stored in the guide vector. Should the tonality ratio in a frequency band exceed the respective threshold, a missing harmonic for that band will be assumed, and the guide vector will be assigned the current ratio. One restriction yet applies: the birth of a new sinusoidal partial is exclusively confined to transient signal passages. If not for the purpose of sinusoidal tracking, a detected harmonic in a stationary segment will therefore be discarded.

The death of a harmonic partial is covered by the second detection stage. There, in place of the tonal ratio $r_{\kappa,\max}(\nu)$, the decay of the original tone is tracked in the basically exact same manner as illustrated above. The decision, whether a harmonic needs to be inserted in a particular frequency band, is consequently made upon the tonality values

$$T_{\kappa,\max}(\nu) = \max_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \hat{T}_P(k,\nu), \tag{3.64}$$

which are equivalently used to set up the thresholds for the next run.

The detection algorithm also guards against situations, in which a strong tonal component in the original signal is replicated by multiple sinusoids in the SBR signal. Would the situation arise that the inserted tones are too close in pitch, coding artifacts in the form of roughness would become audible. Moreover, an energy compensation vector is determined that helps to properly adjust the spectral envelope, accounting for the position of a missing partial with regard to the borders of the respective frequency band as much as the spectral characteristics of the patched signal.

In case that the detection of missing partials was successful, several modifications relating to energy estimation become necessary. Firstly, the energy in the frequency band that contains the tonal component is to be calculated as the maximum energy instead of the average energy as suggested by Eq. (3.34), according to

$$e_{\kappa,\max}(\nu) = \max_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \frac{1}{n_2(\nu) - n_1(\nu) + 1} \sum_{n=n_1(\nu)}^{n_2(\nu)} |y_k(n)|^2. \tag{3.65}$$

Secondly, the tonality values in the Eqs. (3.59) and (3.60) for the affected noise floor frequency band are to be replaced by

$$T_{\kappa,\max}(\nu) = \max_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \hat{T}_P(k,\nu), \tag{3.66}$$

$$T'_{\kappa,\max}(\nu) = \max_{k_1(\kappa)\leqslant k\leqslant k_2(\kappa)} \hat{T}'_P(k,\nu), \tag{3.67}$$

and the noise floor level in Eq. (3.61) should instead be estimated as

$$q_{\kappa,\max}(\nu) = \min\left\{\frac{1}{T_{\kappa,\max}(\nu)}, q_{\max}\right\}. \tag{3.68}$$

3.3.7 Quantization & encoding

Dependent on the time-frequency signal characteristics of the considered SBR frame, the spectral envelope scale factors $\bar{e}_\kappa(\nu)$ are linearly quantized in either 1.5 dB or 3 dB steps. `FIXFIX`-class frames consisting of one single SBR envelope are categorically quantized using the 3 dB scale. Hence, the round-to-nearest quantization is performed according to

$$e^{\ulcorner}_\kappa(\nu) = \left\lfloor a \max\left\{\log_2 \frac{\bar{e}_\kappa(\nu)}{M}, 0\right\} + 0.5 \right\rfloor, \tag{3.69}$$

where the amplification factor a is defined as

$$a = \begin{cases} 2, & \text{for 1.5 dB-level quantization} \\ 1, & \text{for 3.0 dB-level quantization} \end{cases}, \tag{3.70}$$

and M is the number of filter bank channels. The noise floor scale factors $\bar{q}_\kappa(\nu)$ are asymmetrically quantized in 3 dB steps, yielding

$$q_\kappa^{\lceil}(\nu) = \min\left\{\max\left\{\lfloor q_0 - \log_2 \bar{q}_\kappa(\nu) + 0.5\rfloor, 0\right\}, q_{\max}^{\lceil}\right\}, \tag{3.71}$$

where q_0 denotes the noise floor offset, and $q_{\max}^{\lceil}$ is the maximum quantized noise floor scale factor. The term *envelope scale factor* is used synonymously with the averaged signal energy over a region described by a frequency band κ and a time segment ν, whereas a *noise floor scale factor* represents the ratio between the energy of the noise to be added to the envelope adjusted high-frequency generated signal and the energy of the latter.

The quantized envelope scale factors $e_\kappa^{\lceil}(\nu)$ together with the noise floor scale factors $q_\kappa^{\lceil}(\nu)$ are then delta or differentially coded in either the time or the frequency direction. Further data reduction is achieved by HUFFMAN [43] coding. More details on envelope and noise floor coding are given in [13], [32], and particularly in [44].

3.4 Bit stream

The SBR data stream is divided into two parts: side information and raw data. Side information denotes all the parameters that allow for a correct interpretation of the raw data plus decoder tunings. It specifically includes the TF grid information (Sec. 3.3.4), the information on how the raw data is encoded (Sec. 3.3.7), the inverse filtering data (Sec. 3.3.6), and the sinusoidal coding data (Sec. 3.3.6). Raw data includes entropy encoded envelope scale factors (Sec. 3.3.5) and noise floor levels (Sec. 3.3.6).

3.5 Decoding process

3.5.1 Overview

The SBR decoder consists of the following components which all operate in the complex subband domain: a high-frequency generation module, an envelope adjustment module, and a module that adds the required amount of noise and missing sinusoids to the highband signal –denoted as "additional high-frequency components" in the block diagram of Fig. 3.7. The necessary guidance information for the HFR process is read

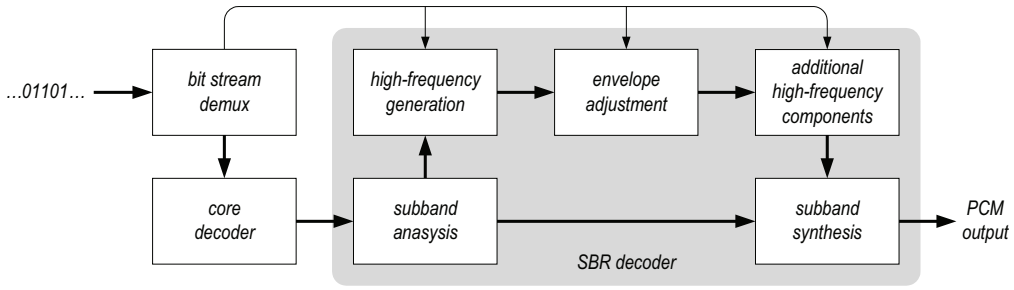

Fig. 3.7: *Generic scheme of the SBR decoder*

from the demultiplexed bit stream and decoded. The downsampled lowband signal is provided by the embedded core decoder. The high-frequency generation and envelope adjustment is performed upon the subband samples of the filter bank decomposed lowband data using the SBR control information. The control data further indicates the amount of adaptive filtering that is to be applied to the transposed lowband signal to preserve the spectral characteristics of the original high band. The patched and filtered low band is envelope adjusted according to the transmitted time-frequency grid that was selected on the encoder side. If required, ancillary signal components are added to compensate for a weak correlation between the low- and highband characteristics of the audio material with respect to noise content and dominant harmonics. Finally, the implicitly upsampled low band and the replicated high band are combined and synthesized to form the wideband PCM output signal.

3.5.2 Subband analysis

The time-domain audio signal from the output of the core decoder, which carries the low-frequency part of the original signal, is split by the analysis CPQMF bank into

$M/2$ complex-valued subband signals, with M being the number of channels of the synthesis CPQMF bank. As the core decoder operates at half the sampling frequency of the SBR superstructure, the number of analysis channels is reduced according to the downsampling factor, which in the considered case is two. The signal processing basics for the analysis CPQMF bank are covered in Sec. 3.3.2.

The calculation rule for the subband signals that is presented in [13] is obtained analogously to the derivation shown in the above-mentioned section. With the analysis filters $h_k(n)$ defined as

$$h_k(n) = p_0(2n)e^{\jmath\Omega_k\left(2n+\frac{1}{2}-1\right)}, \tag{3.72}$$

where $p_0(2n)$ are the L coefficients of the factor-2 decimated window function from Fig. 3.3, the subband sequences $x_k(n)$ are

$$x_k(n) = \sum_{\nu=0}^{L-1} h_k(\nu)x\left(n\frac{M}{2}-\nu\right) = \sum_{\nu=0}^{L-1} p_0(2\nu)x\left(n\frac{M}{2}-\nu\right)e^{\jmath\Omega_k\left(2\nu-\frac{1}{2}\right)}. \tag{3.73}$$

Given that the complex sequences $e^{\jmath\Omega_k\left(2\nu-\frac{1}{2}\right)}$, where $\Omega_k = 2\pi/M(k+1/2)$, are periodic with M, while $L = mM, m \in \mathbb{N}$, Eq. (3.73) can be equivalently formulated as

$$U(k) = \sum_{\nu=0}^{M-1} u(\nu)e^{\jmath\Omega_k\left(2\nu-\frac{1}{2}\right)}, \tag{3.74}$$

where

$$u(\nu) = \sum_{\mu=0}^{m-1} \hat{x}^{-}_{n\frac{M}{2}}\left(\nu+\mu M\right) \tag{3.75}$$

is a sequence derived from the windowed time-reversed input signal

$$\hat{x}^{-}_{n\frac{M}{2}}(\nu) = p_0(2\nu)x\left(n\frac{M}{2}-\nu\right). \tag{3.76}$$

3.5.3 Decoding & dequantization

Scale factors for spectral envelopes as well as for noise floors are differential coded in either the time or the frequency direction for each signaled envelope and noise floor. Whenever time coding is chosen over frequency coding and the nominal frame boundaries are not adhered to, the first envelope in the present frame is differentially

coded bearing upon the last envelope of the preceding frame. The decoding of the scale factors is sufficiently explained in [13].

In accordance with Sec. 3.3.7, the dequantized envelope scale factors are obtained from

$$\bar{e}_\kappa(\nu) = M \cdot 2^{e_\kappa^{\lceil}(\nu)/a}, \tag{3.77}$$

where M is the number of filter bank channels, and a is defined as

$$a = \begin{cases} 2, & \text{for 1.5 dB-level quantization} \\ 1, & \text{for 3.0 dB-level quantization} \end{cases}, \tag{3.78}$$

while the quantization level is supplied by the bit stream parser. Correspondingly, the dequantized noise floor levels are obtained from

$$\bar{q}_\kappa(\nu) = 2^{q_0 - q_\kappa^{\lceil}(\nu)}, \tag{3.79}$$

with q_0 representing the noise floor offset for the quantized noise floor levels $q_\kappa^{\lceil}(\nu)$.

3.5.4 High-frequency generation

The high-frequency generation unit patches, or copies, sets of consecutive subband signals the lower frequencies to higher frequencies. The number of sets, also termed as *patches*, and their frequency location is determined by the patch construction algorithm [13]. The so acquired highband signals are spectrally whitened to bear resemblance to the original high band with regard to the tonal-to-noise ratio. This is necessary since the tonal character of the signal is usually more pronounced in the low band compared to the high band [39]. The amount of spectral whitening to be applied is determined in the encoder and signaled to the decoder in the form of inverse filtering levels. The generated high band is subsequently processed by the envelope adjustment unit. Further details on the transposition process and inverse filtering are given below.

Frequency transposition

A subset of critically sampled lowband signals $\{x_k(n)\}$ from the $M/2$-channel analysis filter bank forms the band-limited source signal for the *spectral translation* [45]. The

lowermost not used filter bank channel is

$$k_c = \left\lfloor \frac{f_c}{f_s} 2M \right\rfloor, \tag{3.80}$$

$k_c \leqslant M/2$, where f_c is the crossover frequency between the low and the high band, f_s is the sampling frequency of the wideband signal, and M is the number of synthesis channels. The high band, which by definition covers the frequency range from f_c to circa 16 kHz, is constructed out of linearly frequency-shifted lowband sequences $x_k(n)$. The patches are built from consecutive subband signals $\{x_k(n)\}_{k\in[k_c-N_i,k_c]}$, where N_i is the size of the i-th patch, skipping the near-DC component at $k = 0$. Several patching steps are necessary, whenever the crossover frequency f_c lies below 8 kHz.

Inverse filtering

Based on the assumption that an audio signal can accurately enough be approximated by an autoregressive system model, the current envelope $H(z)$ is described by

$$H(z) = \frac{1}{A(z)}, \tag{3.81}$$

where

$$A(z) = 1 + \sum_{n=1}^{N} a_n z^{-n} \stackrel{!}{=} 1 - \sum_{n=1}^{N} p_n z^{-n} = P(z), \tag{3.82}$$

such that

$$H(z) = \frac{1}{P(z)}. \tag{3.83}$$

As can be seen from Eq. (3.83), the inverse of $H(z)$ is the linear prediction error filter $P(z)$, which was derived earlier in Sec. 3.3.6. The degree of spectral whitening is controlled by a bandwidth expansion factor, or *chirp factor* β, $0 \leqslant \beta \leqslant 1$, which is applied to the frequency parameter z of the N-th order polynomial $P(z)$, yielding the inverse or *pre-whitening* filter

$$P(z,\beta) = P\left(\frac{z}{\beta}\right) = 1 - \sum_{n=1}^{N} p_n \beta^n z^{-n}. \tag{3.84}$$

The complex-valued filter coefficients p_n are obtained using the covariance method, which is described in Sec. 3.3.6.

The linear prediction coefficients are calculated independently for each patched lowband signal $x_k(n)$. The prediction filter order is as low as $N = 2$, since the expected number of tonal components in each frequency band is relatively small for a system with $M = 64$ filter bank channels. The frequency-dependent chirp factors $\beta(k')$ that flatten out the formants in the replicated highband envelope are calculated based on the transmitted inverse filtering levels $\{\texttt{off}, \texttt{low}, \texttt{mid}, \texttt{high}\}$. The HF generated highband sequences $x_{k'}(n)$ are hence defined as

$$x_{k'}(n) = x_k(n) - p_1(k)\beta(k')y_k(n-1) - p_2(k)\beta^2(k')y_k(n-2), \tag{3.85}$$

where the subband mapping $k \mapsto k'$ is conducted following the patching algorithm.

3.5.5 Envelope adjustment

The envelope adjustment is performed upon the frequency range of the generated high band. To be able to do so, however, the envelope of the current SBR signal needs to be estimated first. This is accomplished by averaging the magnitude-squared subband samples $x'_k(n)$ over different time regions in compliance with the time-frequency grid from the SBR data stream, according to

$$\bar{e}'_k(\nu) = \frac{1}{n_2(\nu) - n_1(\nu) + 1} \sum_{n=n_1(\nu)}^{n_2(\nu)} |x'_k(n)|^2, \tag{3.86}$$

which provides a frequency-interpolated highband envelope representation with the resolution of the CPQMF bank. Beyond that, the level of noise and the level of missing sinusoids is scaled to the proper amplitude value. The magnitude of the noise signals $\bar{n}_k(\nu)$ is derived from the transmitted noise floor levels $\bar{q}_k(\nu)$ as

$$\bar{n}_k(\nu) = \sqrt{\bar{e}_k(\nu)\frac{\bar{q}_k(\nu)}{1 + \bar{q}_k(\nu)}}, \tag{3.87}$$

where $\bar{e}_k(\nu)$ is the original time-averaged energy value of the k-th filter bank channel, while ν designates the considered envelope. The magnitude of the sinusoids $\bar{s}_k(\nu)$ is calculated in an analogous manner, i. e.

$$\bar{s}_k(\nu) = \sqrt{\bar{e}_k(\nu)\frac{1}{1 + \bar{q}_k(\nu)}}. \tag{3.88}$$

With all the required estimates at hand, the energy of the newly generated SBR subband signals is adjusted to the desired envelope shape. The gain $\bar{a}_k(\nu)$ that is applied to retain the proper envelope depends on the particular case as explained below.

Case 1. k lies within the bounds of a frequency band, for which a missing harmonic was detected. In this case the gain is given by

$$\bar{a}_k(\nu) = \sqrt{\frac{\bar{e}_k(\nu)}{\bar{e}'_k(\nu)} \frac{\bar{q}_k(\nu)}{1+\bar{q}_k(\nu)}} = \frac{\bar{n}_k(\nu)}{\sqrt{\bar{e}'_k(\nu)}}. \tag{3.89}$$

Case 2. The frequency band that includes k does not miss a harmonic, and ν is either the transient envelope or the decay envelope. In case of the latter, ν must be equal to one, i. e. the respective envelope must be the very first envelope in the current frame. Is this condition fulfilled, then the gain is calculated as

$$\bar{a}_k(\nu) = \sqrt{\frac{\bar{e}_k(\nu)}{\bar{e}'_k(\nu)}}. \tag{3.90}$$

Case 3. If the frequency band that includes k does not miss a harmonic, and none of the above holds true, the gain accounts to

$$\bar{a}_k(\nu) = \sqrt{\frac{\bar{e}_k(\nu)}{\bar{e}'_k(\nu)} \frac{1}{1+\bar{q}_k(\nu)}} = \frac{\bar{s}_k(\nu)}{\sqrt{\bar{e}'_k(\nu)}}. \tag{3.91}$$

Noise substitution limiting [40] is conducted complementarily to the adaptive noise floor addition. By introducing a noise level similar to the original in the recreated high band, the noise floor addition also contributes to the reduction of the effect of frequency band shutdown, which usually occurs under low bit rate conditions in natural coding systems [40]. Frequency band shutdown denotes spectral vacancies that may appear in an arbitrary fashion over the coded frequency range and eventually cause audible artifacts. However, supposed that the original signal has great amount of energy in a frequency band and the transposed signal shows little or no energy within the same band, noise or other unwanted signal components will be amplified to the energy level of the original signal in that particular frequency band. This is referred to as unwanted noise substitution. To avoid unwanted noise substitution, the amplification factors $\bar{a}_k(\nu)$ need to be limited.

In a first step, the maximum gain values $\bar{a}_{\kappa,\max}(\nu)$ are derived from the original envelope $\bar{e}_k(\nu)$ and the generated envelope $\bar{e}'_k(\nu)$, according to

$$\bar{a}_{\kappa,\max}(\nu) = \sqrt{\frac{2\sum_{k=k_1(\kappa)}^{k_2(\kappa)} \bar{e}_k(\nu)}{\sum_{k=k_1(\kappa)}^{k_2(\kappa)} \bar{e}'_k(\nu)}}, \tag{3.92}$$

using the frequency borders $k_i(\kappa)$ from the limiter band table. Thereafter, these grouped gain factors are mapped to the frequency resolution of the filter bank and capped, as said by

$$\bar{a}_{\kappa\mapsto k,\max}(\nu) = \min\left\{\bar{a}_{\kappa,\max}(\nu), a_{\max}\right\}, \tag{3.93}$$

where $f : \kappa \mapsto f(\kappa) := \{k \mid k_1(\kappa) \leqslant k \leqslant k_2(\kappa)\}$. In conclusion, the gain values $\bar{a}_k(\nu)$ are compared to the ones in Eq. (3.93), yielding the limited amplification factors

$$\check{a}_k(\nu) = \min\left\{\bar{a}_k(\nu), \bar{a}_{k,\max}(\nu)\right\}. \tag{3.94}$$

Similarly the noise levels $\bar{n}_k(\nu)$ are capped proportional to the energy loss due to the limitation of the gain values, resulting in

$$\check{n}_k(\nu) = \min\left\{\bar{n}_k(\nu), \bar{n}_k(\nu)\frac{\bar{a}_{k,\max}(\nu)}{\bar{a}_k(\nu)}\right\}. \tag{3.95}$$

To ensure that the mean signal energy $\bar{e}_\kappa(\nu)$ within a limiter band κ is maintained after the application of the limited gain values $\check{a}_k(\nu)$, corrective boost factors $\hat{a}_k(\nu)$, which compensate for the limiter-imposed energy loss, are calculated. The required compensation is defined as

$$\hat{a}_\kappa(\nu) = \sqrt{\frac{\sum_{k=k_1(\kappa)}^{k_2(\kappa)} \bar{e}_k(\nu)}{\sum_{k=k_1(\kappa)}^{k_2(\kappa)} \check{a}_k^2(\nu)\bar{e}'_k(\nu) + \check{n}_k^2(\nu) + \bar{s}_k^2(\nu)}}, \tag{3.96}$$

whereas the noise term $\check{n}_k(\nu)$ is neglected in case there is a harmonic missing in subband k, or if ν coincides with either the transient envelope or the decay envelope. The final boost factors $\hat{a}_k(\nu)$ are obtained through frequency mapping $f : \kappa \mapsto f(\kappa) :=$

$\{k \mid k_1(\kappa) \leqslant k \leqslant k_2(\kappa)\}$, while capping the mapped values to an upper-bound limit $\hat{a}_{\max}$, yielding

$$\hat{a}_{\kappa \mapsto k}(\nu) = \min\left\{\hat{a}_\kappa(\nu), \hat{a}_{\max}\right\}. \tag{3.97}$$

The boost factors $\hat{a}_k(\nu)$ are then applied to the limited gain values $\check{a}_k(\nu)$, the respective noise floor levels $\check{n}_k(\nu)$, and the levels of additional sinusoids $\bar{s}_k(\nu)$ according to

$$\begin{aligned} \tilde{a}_k(\nu) &= \hat{a}_k(\nu)\check{a}_k(\nu), \\ \tilde{n}_k(\nu) &= \hat{a}_k(\nu)\check{n}_k(\nu), \\ \tilde{s}_k(\nu) &= \hat{a}_k(\nu)\bar{s}_k(\nu), \end{aligned} \tag{3.98}$$

$\forall\ k, \nu$.

The amplification factors from Eq. (3.98), which are temporally spanned over several subsamples, are further remapped to the time resolution of the underlying filter bank analogous to the mapping of, e. g., the boost factors to a higher frequency resolution as shown in Eq. (3.97). That is to say

$$\begin{aligned} a_k(\nu \mapsto n) &= \tilde{a}_k(\nu), \\ n_k(\nu \mapsto n) &= \tilde{n}_k(\nu), \\ s_k(\nu \mapsto n) &= \tilde{s}_k(\nu), \end{aligned} \tag{3.99}$$

where $f : \nu \mapsto f(\nu) := \{n \mid n_1(\nu) \leqslant n \leqslant n_2(\nu)\}$. In the very last step these levels are respectively applied to the generated highband sequences $x'_k(n)$, the white Gaussian noise with zero mean and unit variance $\eta_k(n)$, and the synthesized sinusoids $\xi_k(n)$, yielding the envelope adjusted highband signals

$$y_k(n) = a_k(n)x'_k(n) + n_k(n)\eta_k(n) + s_k(n)\xi_k(n). \tag{3.100}$$

3.5.6 Subband synthesis

An M-channel synthesis CPQMF bank transforms the concatenated wideband signal $\{z_k(n)\}_{k\in[0,k_{\max}]}$, which is the union of the lowband signal $\{x_k(n)\}_{k\in[0,k_c-1]}$ and the reconstructed highband signal $\{y_k(n)\}_{k\in[k_c,k_{\max}]}$, where k_c is the crossover subband, from the subband domain back to the time domain. The sampling rate of the SBR-processed signals $z_k(n)$ is twice the sampling rate of the decoded core signals $x_k(n)$ and equal to the sampling rate of the original broadband signal reduced by a factor of M.

To obtain the reconstructed time-domain signal $z(n)$, the subsequences $z_k(n)$ are first expanded to

$$z_k^{(M)}(n) = \sum_{\mu=0}^{2M-1} z_k(n)\delta(n-\mu M), \tag{3.101}$$

which increases the sampling rate by a factor of M, and subsequently interpolated by the bandpass synthesis filters $g_k(n)$ removing all the frequency-scaled images of the underlying continuous-frequency spectrum $X(\omega)$ except at integer multiples of 2π. The upsampled output sequence $z(n)$ is therefore given by

$$z(n) = \sum_{k=0}^{M-1} g_k(n) * z_k^{(M)}(n) = \sum_{k=0}^{M-1}\sum_{\nu=0}^{N}\sum_{\mu=0}^{2M-1} g_k(n-\nu)z_k(\nu)\delta(\nu-\mu M), \tag{3.102}$$

where the synthesis filters $g_k(n)$ are identical to the analysis filters due to Eq. (3.5).

Eq. (3.102) can be efficiently computed by drawing a parallel between the CPQMF synthesis and the *inverse* DFT (IDFT), where the modulation terms $e^{\jmath\Omega_k(n+\frac{1}{2})}$ from the analysis filters in Eq. (3.17) are interpreted as the kernel of a finite-length forward transform

$$U(k) = \sum_{n=0}^{N-1} u(n)e^{\jmath\Omega_k\left(n+\frac{1}{2}\right)}. \tag{3.103}$$

To obtain the periodically extended sequence $u(n)$ from the spectral coefficients $U(k)$, the orthogonality of the set of complex exponential sequences is exploited. At an arbitrary time instant m, we multiply both sides of Eq. (3.103) by $e^{-\jmath\Omega_k\left(m+\frac{1}{2}\right)}$ and sum from $k=0$ to $k=M-1$. As a result, after interchanging the summation order on the right-hand side, we get

$$\begin{aligned}\sum_{k=0}^{M-1} U(k)e^{-\jmath\Omega_k\left(m+\frac{1}{2}\right)} &= \sum_{k=0}^{M-1}\sum_{n=0}^{N-1} u(n)e^{\jmath\Omega_k(n-m)} \\ &= \sum_{n=0}^{N-1} u(n)\sum_{k=0}^{M-1} e^{\jmath\Omega_k(n-m)}.\end{aligned} \tag{3.104}$$

Then we apply the identity

$$\sum_{k=0}^{M-1} e^{\jmath\Omega_k(n-m)} = M\delta(n-m), \tag{3.105}$$

which expresses the orthogonality of the complex exponentials, to Eq. (3.104) and obtain

$$\sum_{k=0}^{M-1} U(k)e^{-\jmath\Omega_k\left(m+\frac{1}{2}\right)} = M\sum_{n=0}^{N-1} u(n)\delta(n-m) = Mu(m). \tag{3.106}$$

Hence, the time sequence $u(n)$ in Eq. (3.103) is gained from $U(k)$ by the relation

$$u(n) = \frac{1}{M} \sum_{k=0}^{M-1} U(k) e^{-\jmath \Omega_k \left(n + \frac{1}{2}\right)}. \tag{3.107}$$

This DFT like approach can be utilized to carry out block convolution using the CPQMF bank more efficiently, e. g., based on the classical *overlap-add method* (OLA) [34].

3.6 Algorithmic delay

The estimation of the algorithmic delay is conducted based on the the block diagram of the SBR superstructure, which is depicted in Fig. 3.8. In this schematic every block

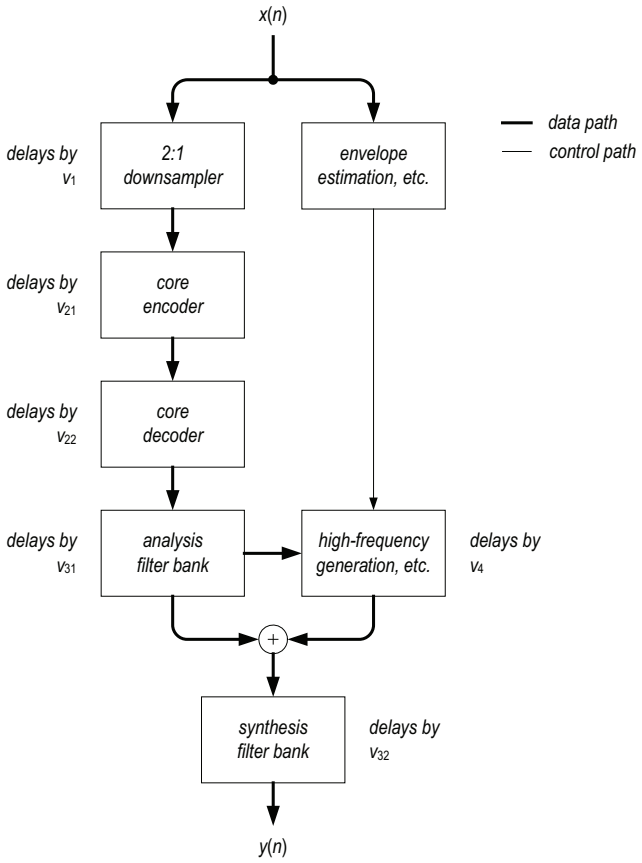

Fig. 3.8: *Algorithmic delay along the signal path*

along the signal or data path, which contributes to the delay of the system is marked with the respective delay value counted in time samples. The overall system delay is therefore given by the sum of all delay values, i. e., the delay of the downsampling

filter ν_1, the delay of the core codec chain $\nu_2 = \nu_{21} + \nu_{22}$, the delay of the CPQMF bank $\nu_3 = \nu_{31} + \nu_{32}$, and finally the delay entailed by the high-frequency generation algorithm ν_4. The output from the synthesis filter bank,

$$y(n) = \hat{x}\,(n - \nu)\,, \tag{3.108}$$

is thus the reconstructed version $\hat{x}(n)$ of the wideband input signal $x(n)$, delayed by

$$\nu = \sum_{i=1}^{4} \nu_i \tag{3.109}$$

samples.

The delay of the SBR extension is for the larger part caused by the L-length periodic version of the prototype filter function $p_0(n)$. Not including the blocking delay of the M-length transformation core, the delay of the CPQMF bank is determined by the number of samples that the prototype window overlaps towards the future.[3] Hence, due to the type-I symmetry of the prototype filter $p_0(n)$, both the analysis and the synthesis jointly introduce a delay of $L - M$ samples. Fig. 3.9 depicts the analysis windowing scheme used in [32] pointing out the overlap delay. The input sequence to be transformed is pre-padded with $L - M$ zeros representing the past, where M is likewise the downsampling factor. The delay of the HFR unit is determined by the size

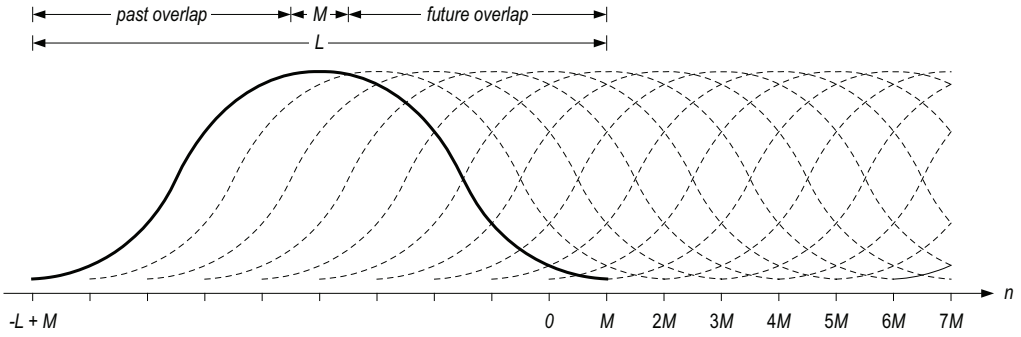

Fig. 3.9: *"Jumping" window sequence*

of the look-ahead buffer, which is a quarter of the block size N.[4] Presuming that the

[3] In a complete coding scheme the blocking delay of $M - 1$ samples is covered by the core encoder.

[4] Due to further optimization, particularly by reducing the temporal resolution of envelopes by a factor of two, the size of the SBR look-ahead buffer in [32] is downsized to $N/4 - 2M = 384$ samples.

downsampling filter has a low order, and hence has a negligible delay, the total SBR delay in milliseconds amounts to

$$\tau = \frac{1}{f_s}\left(L - M + \frac{N}{4}\right), \tag{3.110}$$

where f_s is the sampling frequency.[5] Tab. 3.1 lists the SBR delay for the three most used sampling frequencies, with the parameters L, M, and N taken from the HE-AAC scheme. It should be noted that the algorithmic delay is rather a theoretical value. In

Sampling rate	SBR delay	Optimized SBR delay
32 kHz	34 ms	30 ms
44.1 kHz	24.7 ms	21.8 ms
48 kHz	22.7 ms	20 ms

Tab. 3.1: *SBR delay for* $L = 640$*,* $M = 64$*, and* $N = 2048$

a hardware implementation the system would require unlimited processing power and infinite transmission velocity to achieve it.

[5] The approximation is surely appropriate, if the downsampling filter possesses the *infinite-impulse-response* (IIR) property.

4. MPEG-4 LOW-DELAY SPECTRAL BAND REPLICATION

4.1 Introduction

Low-Delay Spectral Band Replication (LD-SBR) [24] is a derivative of the standard SBR tool, which as a bandwidth extension coder complements the waveform coding low-delay core, forming a low-delay codec for high-quality bidirectional communication at low bit rates. Its algorithmic delay is reduced to a minimum, such that the overall system delay does not significantly exceed the algorithmic delay of widely used speech codecs, which is round about 20 ms. The two major modifications to the standard SBR technique, which address the delay sources assessed in Sec. 3.6, are now discussed.

4.2 Complex low-delay filter bank

First and foremost, LD-SBR utilizes a delay-optimized *complex low-delay filter* (CLDF) bank for the subband analysis-synthesis filtering replacing the CPQMF bank in the decoder. The modulated filter bank that features perfect reconstruction was designed based on the framework formulated in [46], [47], [48], [49], and finally in [50]. To maintain compatibility with the algorithms that bear on the output from the filter bank, the number of subbands M as well as the length L of the prototype window was retained unchanged. The modulation kernel was derived from the type-IV DCT (DST).

Fig. 4.1 shows the impulse response of the LD-SBR analysis prototype filter $p_a(n)$ and the corresponding magnitude response. A substantial delay reduction is achieved by shifting the modulation core consisting of M samples to the right, and, in doing so, shortening the overlap towards the future. The analysis filters $h_k(n)$ of the CLDF bank are obtained by complex-exponential modulating the asymmetric lowpass analysis

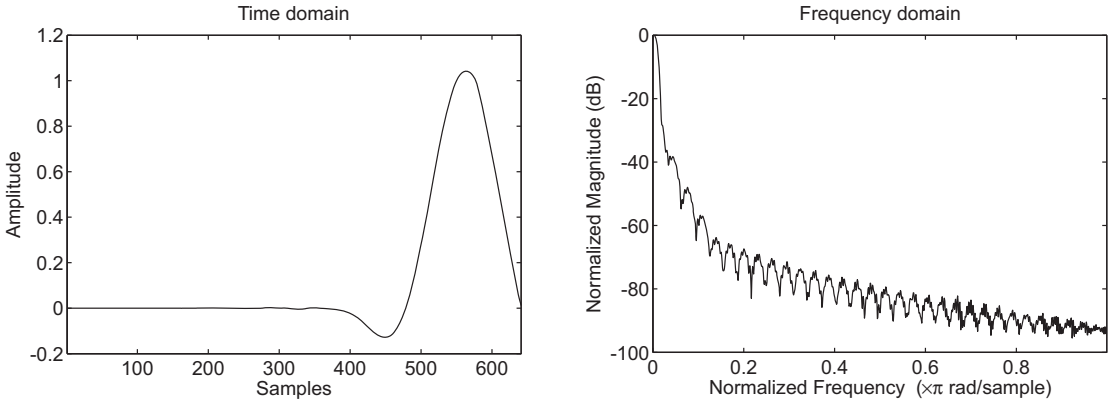

Fig. 4.1: *Time and frequency domain plot of the* 640*-length analysis window. The relative sidelobe attenuation is* -36.3 *dB. The mainlobe width* (-3 *dB) is* 0.018555.

prototype $p_a(n)$ according to

$$h_k(n) = 2p_a(2n)e^{\jmath\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(2n+\frac{1}{2}-\frac{M}{2}-M+\frac{1}{2}\right)}, \tag{4.1}$$

where $k = 0, 1, \ldots, M/2-1$, M is the number of synthesis channels, and $n = 0, 1, \ldots, L/2-1$, where L is the length of the analysis prototype window. The synthesis filters $g_k(n)$ are given by

$$g_k(n) = p_s(n)e^{\jmath\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}-\frac{M}{2}\right)}, \tag{4.2}$$

$k = 0, 1, \ldots, M-1$, $n = 0, 1, \ldots, L-1$, where the synthesis prototype window $p_s(n)$ is a time-reversed version of the analysis window, i. e.

$$p_s(n) = p_a(L-1-n), \tag{4.3}$$

$0 \leqslant n < L$.

4.3 Frame-locked time-frequency grid selection

LD-SBR employs a reduced set of frame classes when selecting the time-frequency grid for the considered signal segment. The two classes used are: `FIXFIX` and `LD_TRAN`, where the `FIXFIX` class is inherited from standard SBR, see Sec. 3.3.4. The new frame class `LD_TRAN` is intended to account for the presence of transients within the frame, however, without introducing an additional delay. This is accomplished by

placing the leading and the trailing time borders at the nominal frame boundaries, while selectively adopting the envelope distribution inside the frame to the position of the detected transient. Apparently, the second source of delay is extinguished, as the look-ahead into the future frame is not required anymore. There aren't any restrictions on LD-SBR frame class transitions, allowing any permutations of the two classes.

4.4 Algorithmic delay

In the case of LD-SBR, the system delay is solely determined by the delay of the CLDF bank, or rather the shape of the analysis prototype window $p_a(n)$, see Fig. 4.1, and the shape of the synthesis prototype window $p_s(n)$, respectively. The L-length analysis prototype $p_a(n)$ has as few as $M/2$ samples overlapping to the right, i. e. towards the future, and thus causing delay. Correspondingly, as a result of Eq. (4.3), the overlap to the left of the temporally mirrored synthesis prototype $p_s(n)$ delays by the same number of samples, such that the overlap delay of the CLDF bank constitutes a total of M samples. As the longer tail of the analysis (synthesis) window refers to the past, it does not add further delay. Therefore, neglecting the delay of the downsampling filter and the blocking delay, the overall LD-SBR algorithmic delay is given by

$$\tau = \frac{M}{f_s} \tag{4.4}$$

irrespective of the window length, where f_s is the sampling frequency of the LD-SBR superstructure. Tab. 4.1 provides a listing of the LD-SBR delay for the three commonly used sampling frequencies with respect to AAC-ELD. Further considerations regarding the delay of the CLDF bank are given in [51].

Sampling rate	LD-SBR delay
32 kHz	2 ms
44.1 kHz	1.5 ms
48 kHz	1.3 ms

Tab. 4.1: *LD-SBR delay for* $L = 640$, $M = 64$

5. PROPOSED LOW-DELAY BANDWIDTH EXTENSION

5.1 Introduction

The herein proposed low-delay bandwidth extension is rooted in the SBR method, with two-way real-time communication being its aspired field of application. Just like the original, the presented scheme operates on subband samples of an arbitrary acoustic signal in both the time and the frequency dimension using a modulated filter bank in the analysis stage. Contrary to the time-domain prediction method, the two-dimensional decomposition allows a high degree of flexibility with respect to the choice of the crossover frequency between the low and the high band, which is a highly anticipated feature for any HFR system. Since [46] at the latest it is also known that the algorithmic delay of a modulated filter bank can be explicitly controlled and thus reduced to a minimum irrespective of the length of the impulse response of the prototype filter used. In other words, the filter bank can be designed such that the overall system delay does not exceed the framing delay, i. e. the time it takes to fill a frame. Another argument for the filter bank approach is the fact that there exist computationally efficient transform algorithms for the modulation core. The *fast* Fourier *transform* (FFT) for instance can be used to implement the DCT or the DST, respectively. Hence, having specified the latency constraint, the end-to-end delay issue is shifted away from the filter bank towards the parametric description of the truncated high band.

In the following, the most significant modifications that have been made to the standard SBR technique are discussed. To distinguish the proposed descendant from MPEG's own low-delay counterpart, the acronym LD-BWE is consistently used throughout the rest of the thesis.

5.2 Delay considerations

There are basically two ways of implementing the bandwidth extension on top of a waveform codec in a dual rate system. The one method, which is drawn on in HE-AAC and also in AAC-ELD, increases the frequency resolution of the filter bank, by keeping the number of subbands constant, while reducing the signal bandwidth by the downsampling factor at the same time. Transform based codecs in particular profit from this technique, as the irrelevancy and the redundancy reduction both can be accomplished more accurately leading to a higher coding gain. The main drawback here is that the filter or window length is implicitly augmented and so is the delay. In a system, where the core codec operates at half the sampling rate of the audio signal, its algorithmic delay at the end of the processing chain is therefore doubled. Alternatively, the window function can be decimated as well, if a higher spectral resolution is not essential for the coding performance. This is certainly the case for a coding scheme like the ULD, where the irrelevancy reduction and the redundancy reduction are separated from each other.

The Fraunhofer implementation of the ULD coder incorporates a psychoacoustic model based on a 256-point DFT with 50 % overlap bearing 128 frequency bins in the positive range for irrelevancy reduction. This number of subbands has been found to be sufficient to approximate the masking threshold. In this particular case, it is therefore not necessary to further increase the spectral resolution, so that the DFT filters should be decimated on a par with the signal sequence for the delay not to grow. Yet in spite of the above considerations, LD-BWE is designed and developed under the constraint that the core codec should not be affected by the modifications performed on SBR. The chosen frame length in samples is thus 256, which is consistent with the 128-length ULD frames.

5.3 System overview

The structure of LD-BWE is symbolically depicted in Figs. 5.1 and 5.2. The first step towards a parametric representation of the high band is to branch the digital audio signal into a sufficient number of subbands. From there on, the transient detector, the

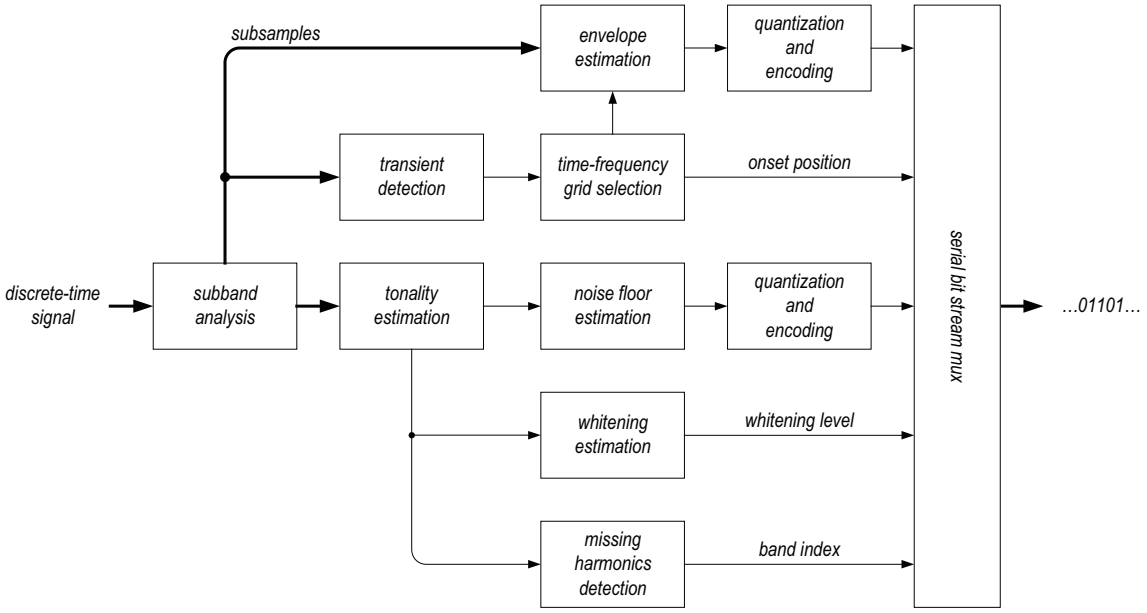

Fig. 5.1: *Block diagram of the LD-BWE encoder*

envelope estimator, and the tonality estimator, all operate on subsamples from each of these subbands. The output from the transient detector, i. e. the onset position of a plosive sound for example, is used for the selection of a proper time-frequency grid for the analyzed time segment. Based on this grid, one or two spectral envelopes per frame are calculated and encoded. The tonality estimator provides one tonality value for each subband. By comparing the tonality between the original highband signal and the a priori known copy that will be generated in the LD-BWE decoder, additional control parameters are derived. They include the noise floor level, the level of whitening that needs to be applied to the copy of the high band, and the frequency bands, in which a missing tonal component was detected. The gathered information is multiplexed into a serial bit stream and transmitted or stored.

On the decoder side, the serial bit stream is demultiplexed and the envelope data comprising the spectral envelope of the high band and the respective noise floor level is decoded. The narrowband signal from the local waveform decoder is decomposed into a reduced number of subbands and upsampled. Pursuant to a prescribed patching rule, the upper part of the spectrum is then filled with the frequency content coming from the lower subbands. The so acquired highband spectrum is simultaneously flattened according to the signaled whitening level. The spectral envelope is interpolated, gain

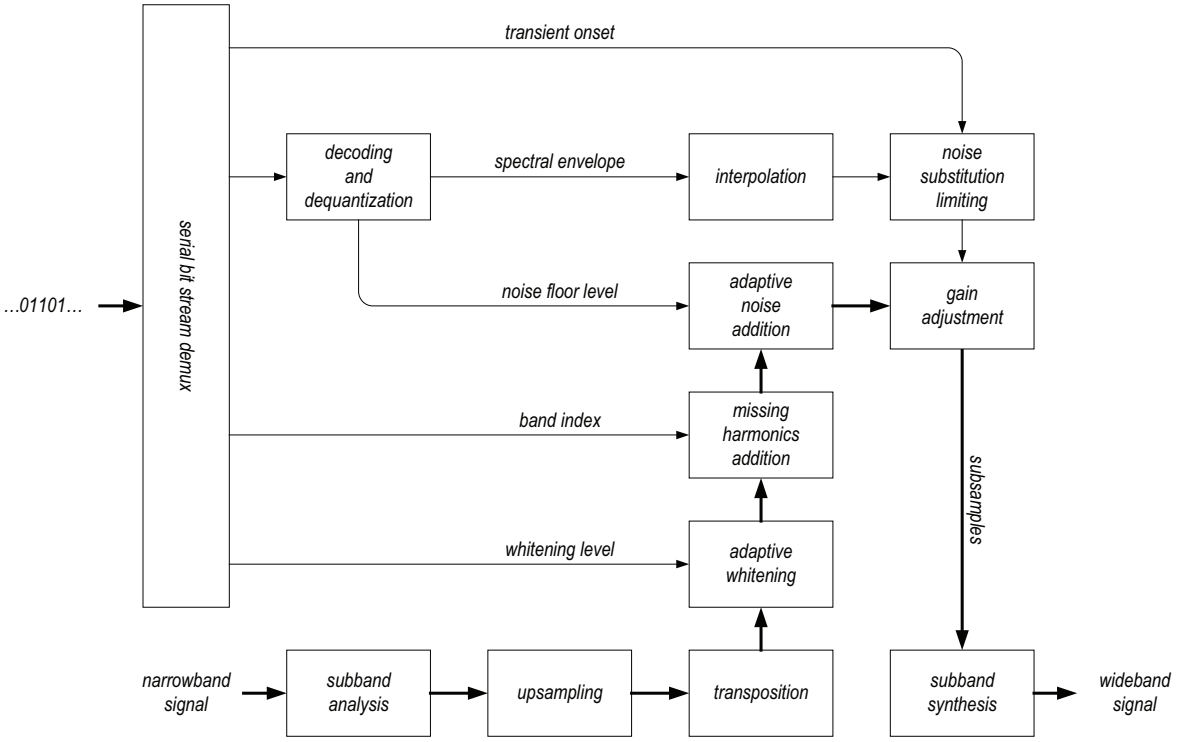

Fig. 5.2: *Block diagram of the LD-BWE decoder*

adjusted, and finally applied to the reconstructed highband signal inclusive of random noise and synthetic sinusoids. The wideband signal is obtained by transforming the frequency-extended lowband signal from the subband domain back to the time domain.

5.4 Filter bank

The present LD-BWE implementation makes use of the CLDF bank that has been introduced earlier in Sec. 4.2 in connection with LD-SBR. The filter bank on the decoder side is completely defined by the analysis and the synthesis filters, $h_k(n)$ and $g_k(n)$, given by Eqs. (4.1) and (4.2). The number of filter bank channels at full rate is 64. The length of the downsampled analysis prototype window is half the filter length of the synthesis prototype, which is 640. The synthesis window is the time-reversed version of the full-rate analysis window. The overlap delay of the CLDF bank amounts to 64 samples.

To synchronize the envelope data from the LD-BWE encoder with the output from the waveform coding core, the input signal to the LD-BWE encoder is to be time

delayed by the number of samples that corresponds to the algorithmic delay of the underlying codec including the downsampling filter. This can be done in either the time or the subband domain. Attention should be paid when the time delay is not an integer multiple of the frame length. In this case, the proper sync delay may be adjusted by distributing the delay value over the two domains. The analysis CPQMF bank on the encoder side is also to be replaced by the analysis CLDF bank with the asymmetric window function. Since all the processing on the decoder side is carried out in the subband domain, no additional syncing between the lowband signal and the recreated high band is necessary.

The analysis subband filtering in the LD-BWE encoder comprises framing, folding, and block convolution, altogether performed on a discrete time-domain input signal $x(n)$ stored in a buffer

$$\mathbf{x} = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-L+1) \end{bmatrix}^{\mathrm{T}}, \tag{5.1}$$

$\mathbf{x} \in \mathbb{R}$, where the size of the buffer is determined by the window length L. The current LD-BWE implementation utilizes a 640-length buffer on the encoder side and a 320-length analysis buffer in the decoder. The filtering steps are illustrated below.

Step 1. Shift the samples in the input vector $\mathbf{x}$ by 64 towards the past. Discard the oldest 64 samples and store the newest 64 at the beginning of the buffer.

Step 2. Multiply the data vector $\mathbf{x}$ element-wise by the window coefficients $\mathbf{p}$ to obtain the windowed vector $\mathbf{z}$.

$$z_i = x_i p_i, \tag{5.2}$$

where $i = 0, 1, \ldots, 639$.

Step 3. Fold the $\mathbf{z}$ vector to create the 128-element vector $\mathbf{u}$.

$$u_i = \sum_{j=0}^{4} (-1)^j z_{i+128 \cdot j}, \tag{5.3}$$

where $i = 0, 1, \ldots, 127$.

Step 4. Calculate 64 new subband samples by modulating the $\mathbf{u}$ vector.

$$y_k(n) = \sum_{i=0}^{127} u_i e^{\jmath \frac{\pi}{64}(k+0.5)(i-95)}, \tag{5.4}$$

$k = 0, 1, \ldots, 63$, where n is the subsample index. Begin with $n = 0$.

Step 5. Repeat *Step 1* to *Step 4* for $n = 1, 2, 3$.

In the synthesis stage, the subband samples run through the same steps in reverse order, i. e. from the bottom up, so that for every time slot n, $n \in [0, 3]$, 64 complex-valued subsamples $\{y_k(n)\}$ from each of the filter bank channels k, $k = 0, 1, \ldots, 63$, are transformed into a series of 64 real-valued time-domain samples $y(n)$, totaling in 256 samples for the entire frame.

The suitability of the 32-channel MPEG-1 Layer II CPQMF bank with 512-length prototype filters has also been tested and compared against the 64-channel CPQMF bank. Despite the fact that the filter bank performed well at a crossover frequency around 8 kHz, it should fail when the signal bandwidth is further limited. The most obvious reason is that the critical bandwidth along the cochlea decreases towards the low frequency end. More extensive tests are yet necessary to prove this assumption.

5.5 Transient detection

The detection algorithm used in LD-BWE represents to a great extent a linearized version of ROSENFELD's operator known from image processing [52]. Similar to the discrete approximation of the first-order derivative in a given direction as one of the simplest methods, the ROSENFELD operator is based on differencing. The general idea is that of computing differences between non-overlapping averages of so-called neighborhoods that meet at the same point. In audio processing these neighborhoods correspond to a series of samples before and after a time instant. The processing steps are as follows:

Step 1. For each subband k and subsample n compute the energy of the neighborhood centered at that time instant without averaging.

$$e_k^{(d)}(n) = \sum_{m=n-d}^{n+d} |y_k(m)|^2, \tag{5.5}$$

where the integer d is initially set to zero.

Step 2. For each subband k and subsample n compute the difference between two adjacent neighborhoods that merge at that time instant leaving out the absolute value operator. Consider positive differences only.

$$\Delta e_k^{(d)}(n) = \max\left\{e_k^{(d)}(n+d) - e_k^{(d)}(n-d-1), 0\right\}. \tag{5.6}$$

Step 3. Repeat *Step 1* and *Step 2* for $d = 1$.

Step 4. For each subsample n compute the total energy difference accumulated over M filter bank channels.

$$\Delta e(n) = \sum_{k=0}^{M-1} \sum_{d=0}^{1} \Delta e_k^{(d)}(n). \tag{5.7}$$

Following the standard paradigm for edge detection, the transient candidates are determined through a comparison of the positive differences $\Delta e_k^{(d)}(n)$ against a dynamic threshold $\tilde{t}_i(k)$. The threshold value is calculated in line with Eq. (3.31) on a frame-by-frame basis. For the position of the transient onset n_t the same decision criterion as in Sec. 3.3.3 is used, whereas the value of $s_{\min}$ is slightly lowered accounting for the new settings. The transient detection range is limited by the nominal frame boundaries, so that $n_t \in \{0, 1, 2, 3\}$. Simulations have shown that the modified algorithm provides

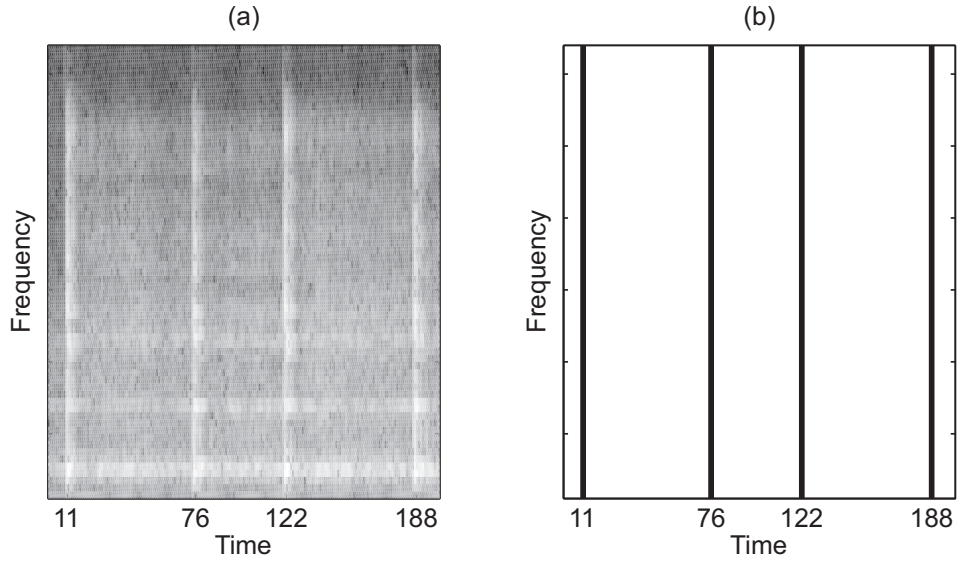

Fig. 5.3: *Modified edge detection: (a) spectrogram, (b) detected transients*

more accurate results in comparison with ROSENFELD's edge enhancement method, see Fig. 5.3.

5.6 Time-frequency resolution

LD-BWE employs two frame classes to discriminate between (quasi-) stationary and transient excerpts from the analyzed signal. A transient frame is further characterized by the onset position of the transient sound, which is determined by the transient detector. The time-frequency resolution is chosen upon the class of the processed frame and the transient position as illustrated in Tab. 5.1. A transient frame is temporally

Onset position	Onset envelope	Time borders	Frequency resolution
0	–	0, 4	`low`
1	second	0, 1, 4	`high`, `low`
2	second	0, 2, 4	`high`, `low`
3	second	0, 3, 4	`high`, `low`

Tab. 5.1: *Time-frequency resolution of a transient frame*

divided into one or two spectral envelopes, whereas a stationary frame always consists of one envelope. Both the start and the stop time borders of a stationary frame equal the nominal frame boundaries and the frequency resolution is constantly `high`, unless the preceding frame has been classified as holding a transient. In this case, the envelope resolution is kept on `low` for another envelope to account for the decay phase of the transient. Stationary and transient frames may alternate in arbitrary order. The time resolution of the noise floor is identical to the time resolution of a stationary frame, and its spectral resolution is given by the noise floor frequency band table.

5.7 Tonality estimation

It was shown in Sec. 3.3.6 that a tonality measure can be derived using the covariance method based on an N-th order prediction error model. As a damped sine wave follows a second-order difference equation, a second-order prediction error model can be used to approximate the autocorrelation structure. For the prediction coefficients to be consistently estimated, the covariance method requires a data vector of at least $2N+1$ samples. In case of LD-BWE, a 256-length frame is decomposed into 64 subbands

with 4 subsamples per subband, which is one subsample less than the required five for a second-order predictor. Hence, without changing the model order, a different polynomial predictor has been tried out [53].

To get an estimate of the tonality, the modulus r and the argument or phase φ of two successive subsamples are used to predict the modulus $\hat{r}$ and phase $\hat{\varphi}$ of the frequency component in subband k at time n according to

$$\hat{r}_k(n) = r_k(n-1) + [r_k(n-1) - r_k(n-2)] \tag{5.8}$$

and

$$\hat{\varphi}_k(n) = \varphi_k(n-1) + [\varphi_k(n-1) - \varphi_k(n-2)]\,. \tag{5.9}$$

The EUCLIDEAN metric or distance $d_k(n)$ between the predicted and the true values,

$$d_k(n) = \sqrt{\hat{r}_k^2(n) + r_k^2(n) - 2\hat{r}_k(n) r_k(n) \cos\{\hat{\varphi}_k(n) - \varphi_k(n)\}}, \tag{5.10}$$

is then mapped to a measure of unpredictability, alias the *chaos measure* $c_k(n)$,

$$c_k(n) = \frac{d_k(n)}{r_k(n) + |\hat{r}_k(n)|}. \tag{5.11}$$

The scale of the chaos measure is further limited to 0.05 at the lower end for highly tonal spectral components and 0.5 at the upper end for noise-like signals. A tonality index $v_k(n)$ that allows weighting towards one extreme or the other is computed from the chaos measure $c_k(n)$ via

$$v_k(n) = -0.43 \log_{10} c_k(n) - 0.299. \tag{5.12}$$

The presented tonality measuring scheme is commonly used in perceptual audio coding to derive an estimate of the masking threshold of the human auditory system.

Nonetheless, what works for a perceptual model does not necessarily work for an HFR system. The tonality measure from above did not prove to be an equivalent substitute for the covariance method, being unable to identify distinctively tonal components within the given time range, see Fig. 5.4. Therefore, the original approach was fetched back and adapted in the following manner: The series of four subsamples have been extended to the past, reaching by one subsample into the preceding frame. Though this method may cause a marginal smearing of the extracted parameters, it does not

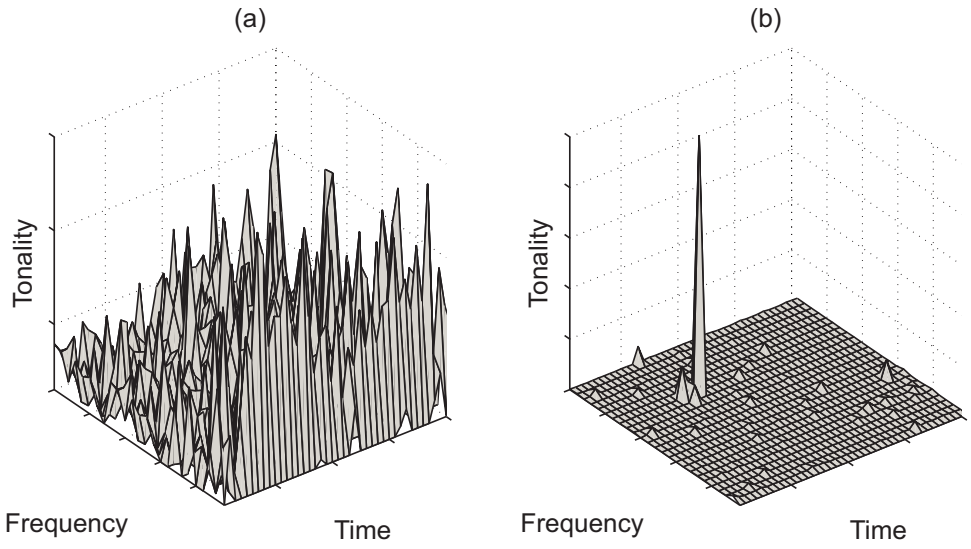

Fig. 5.4: *Estimated tonality: (a) chaos measure, (b) covariance method*

introduce further delay, which is the most important factor here. In consequence, the prediction coefficients, which are also needed on the decoder side to spectrally whiten the transposed lowband signal, are obtained using Eq. (3.51) based on a sample of now five time slot values.

5.8 System delay

The question to be answered in this section is what would be the overall system delay, if LD-BWE was incorporated into the ULD codec. In order to give an answer, a closer look at the specifics of the ULD scheme needs to be taken. As already noted in Sec. 5.2, the irrelevancy reduction in the ULD encoder is conducted based on an estimate of the masking threshold. The masking threshold itself is deduced from the time-dependent DFT spectrum by considering the magnitude level of tonal and noise maskers. An N-point DFT can be viewed as a bank of N orthogonal[1] filters, all being a modulated version of an N-length rectangular window. The system delay of a DFT filter bank corresponds to the length of the window minus one, which is the number of samples that has to be waited on, before a block of N samples is FOURIER transformed. Fraunhofer's ULD coding scheme uses a $\frac{\pi}{2}$-shifted 256-length sine window,

[1] The polyphase matrix of the DFT is unitary

so that the blocking delay on the encoder side accounts to 127 samples. For a smooth transition between the frames, the time-varying pre-filter coefficients, which represent the inverse of the masking threshold, are linearly interpolated. As a consequence of this, 128 more samples appear in the delay calculation. After the pre-filtering step, the remaining signal redundancy may be reduced by either a forward- or backward-adaptive predictor without further increasing the coding delay. On the decoder side the redundancy-minimized residual signal is decoded and shaped by the post-filter. Since the authors of ULD do not deem these operations delay causing, the overall system delay of the ULD codec in samples is quoted with 255.

Apart from the non-causal overlap delay of 64 future samples, the CLDF bank in LD-BWE has a blocking delay of 63 samples, which is yet accounted for by the larger blocking delay of the ULD encoder. Moreover, the temporal envelope borders do not cross the nominal frame boundaries, so that no additional look-ahead delay is produced in the time-frequency grid generator. Hence, the extra delay that is due to LD-BWE corresponds to 64 samples. The overall algorithmic delay of the ULD plus LD-BWE scheme ultimately adds up to 575 samples in the standard scenario, where the ULD codec runs at half the sampling rate with non-decimated analysis filters.

Sampling rate	ULD delay		LD-BWE delay	Total delay	
32 kHz	8 ms	*4* ms	2 ms	18 ms	*10* ms
44.1 kHz	5.8 ms	*2.9* ms	1.5 ms	13 ms	*7.2* ms
48 kHz	5.3 ms	*2.6* ms	1.3 ms	12 ms	*6.6* ms

Tab. 5.2: *System delay for a combination of the ULD codec with the LD-BWE tool. Possible values are italicized.*

Tab. 5.2 depicts the delay values that are achieved with the current implementation of LD-BWE at different sampling frequencies. Values on the right-hand side of a column may well be put into practice, if the DFT filters were decimated by a factor of two and the ULD frame size was cut down to 128 samples. In this case, however, the LD-BWE tool would have to be adapted to the shorter frame size as well. Another option for delay reduction is to drop the interpolation of the masking threshold in the

ULD encoder without changing the frame size.

5.9 Bit rate & computational complexity

Aside from the spectral envelope and the noise floor level, the parametric description of the high band, which is transmitted as ancillary information packed into a stream of bits to the LD-BWE decoder, comprises the following control data:

Frame class. One bit indicating whether there was a transient detected or not.

`0` : frame is stationary
`1` : frame contains a transient

Transient onset position. In the case of a transient frame, two more bits specifying the subsample index. Given the onset position, the number of spectral envelopes, the temporal borders, and the associated frequency band tables are known from Tab. 5.1

Whitening level. Two bits for each noise floor frequency band representing how strongly the transposed lowband signal is to be flattened.

`0` : no whitening
`1` : low-level whitening
`2` : medium-level whitening
`3` : high-level whitening

Missing harmonic flag. One bit that indicates whether missing tonal components were detected in the transposed lowband signal or not.

Add harmonic. One bit for each high-resolution frequency band to indicate that a sinusoid should be inserted in the respective band.

`0` : do not add a sinusoid to the frequency band
`1` : add a sinusoid to the frequency band

At 48 kHz sampling rate the frequency range of the spectral envelope is divided into either fourteen non-uniform frequency bands in case of a stationary time segment or seven frequency bands otherwise. The number of noise floor frequency bands is held constant at two. Not including the envelope data, the number of required bits for the additional control parameters per frame is hence at the least 6 and the most 23, which corresponds to a bit rate of 1.1 to 4.3 kbps.

To give a first impression of the coding gain that can be expected from the current implementation of LD-BWE and to compare it with SBR, the envelope data consisting of envelope and noise floor scale factors has been delta encoded. These delta encoded scale factors were then drawn on in a further redundancy minimization analysis using variable-length HUFFMAN codes. Fig. 5.5 shows the payload size in bits plotted against

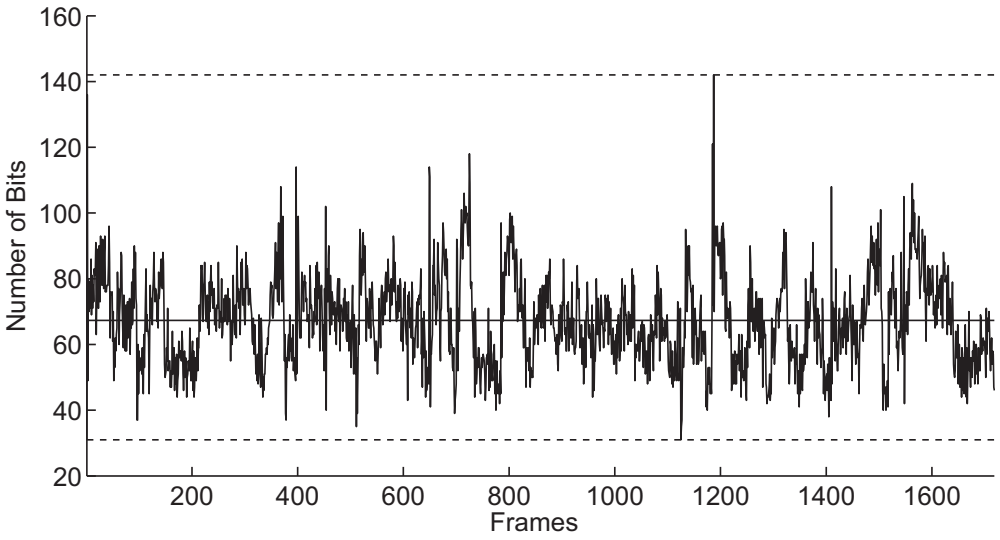

Fig. 5.5: *Bit rate curve for an excerpt from an a cappella piece*

the frame index of an arbitrarily chosen audio signal. It can be observed that at the minimum 31 bits are necessary to code the LD-BWE data, even if the data does not carry any information. Conversely, as many as 111 bits more in the peak value of 142 bits have to be allocated, when high dynamics with respect to signal characteristics occur. The measured mean of 68 bits corresponds to a bit rate of 12.8 kbps. With reference to the figures given in [54], the current LD-BWE implementation would thus bring about a threefold increment of the SBR bit rate. Nevertheless, comparing a ULD-coded mono signal at a data rate of 64 kbps with 32 kbps allotted to the highband

spectrum, LD-BWE would still afford a coding gain of approximately 30 %.

The computational complexity of LD-BWE integrated in ULD can be deduced from the complexity estimates for HE-AAC presented in [54], [19]. Due to the fact that both SBR and LD-BWE employ a 64-channel complex exponential modulated filter bank with prototype filters of equal length, the number of operation within a coeval time segment should not differ significantly between the two. Over and above, the complexity of the ULD coder is considerably higher than the complexity of a transform coder like AAC. Putting one and one together, one can conclude that, if the complexity of HE-AAC is in the region of AAC's complexity, the combination of ULD with LD-BWE in a dual-rate operation mode should not require more processing power compared to ULD alone. In addition, replacing the complex-valued low-delay filter bank in the LD-BWE decoder by a real-valued low-delay filter bank with the same characteristics, the computational demand of the analysis and the synthesis is nearly halved, whereas the high-frequency generation and the envelope adjustment is simplified as well. The occurring aliasing from mirror images is minimized at low computational cost, so that the audio quality is not significantly worsened. Concrete figures are not yet available, though.

6. SUBJECTIVE QUALITY ASSESSMENT

6.1 Introduction

Subjective listening tests are widely recognized as the most reliable way of measuring the quality of audio systems, and there are well described and proven methods for the assessment of audio quality at the top and the bottom quality range under controlled and repeatable conditions [55]. Recommendation ITU-R BS.1116 [56], e. g., is used for the evaluation of high-quality audio systems exhibiting small impairments. For low bit rate communication systems with an algorithmic delay below 10 ms, however, lower-quality audio is acceptable or sometimes unavoidable due to the limited capacity of the transmitting channel. The test method described in [56] is truly not suitable for evaluating these systems, because it is poor at discriminating between small differences in quality at the bottom of the scale. Therefore, the evaluation of the proposed HFR technique, which enhances the subjective performance of the collateral band-limited core system, is conducted by using the *MUlti Stimulus test with Hidden Reference and Anchor* (MUSHRA) method [57]. This method, which has been successfully tested in [58], provides a reliable and repeatable measure of systems with intermediate audio quality, which is the case here. MUSHRA is a double-blind multi-stimulus test method with a hidden reference and at least one hidden anchor.

6.2 Test method

The conducted MUSHRA test uses the original unprocessed audio material with full bandwidth as the (hidden) reference signal and one hidden anchor. The set of processed signals consists of all the signals under test and one additional signal being the the lowpass filtered version of the unprocessed signal. The bandwidth of this additional

signal representing the anchor is chosen in such a way that it equals the bandwidth of the lowband signal, which is likewise the effective bandwidth of the core coding system. The recommended 3.5 kHz anchor is excluded from the test, as the value of such an anchor is put into question regarding the context of the thesis. The anchor would compress the subjective scale, and thus worsen the resolution of the experiment, since all the systems under test are expected to deliver comparable sound quality.

During the test, the assessor or *subject* can switch in any order between the reference signal and any of the signals under test using a computer-controlled replay system. The test itself comprises a sequence of multiple trials, in which the listener detects and assesses any perceptible annoyance of artifacts, which may occur in the versions of each of the reference signals that were processed by the systems under test. The hidden reference signal is meant to help the assessor to detect the artifacts. This method benefits from a paired comparison, so that the subject can more easily detect differences between the impaired signals and grade them accordingly. A high degree of resolution in the grades awarded to the systems is therefore possible.

The listeners score the stimuli according to the *continuous quality scale* (CQS) ranging from 0 to 100, by judging their degree of preference for one type of artifact versus some other type. The CQS is divided into five equal intervals labeled as *bad*, *poor*, *fair*, *good*, and *excellent*. The order of presentation of stimuli is distributed randomly, both within and between sessions. Compared to [56], the MUSHRA method has the advantage of presenting all stimuli at the same time, so that the assessor is able to carry out a direct comparison between one particular system and the reference signal or any of the other systems in each trial as well. The results are more consistent, which leads to smaller confidence intervals after the statistical analysis.

6.3 Test material

The audio sequences used in the subjective test of systems that provide bandwidth extension are a subset of the test signals for MPEG-4 audio.[1] These test sequences, which are known to be critical to perceptual audio codecs, have previously been used

[1] Date back to June 24, 1997

in similar experiments [59], [60]. They are expected to reveal differences among the systems under test. Tab. 6.1 gives the exact list of the items that were used in the MUSHRA test. The names of the test items are coded as follows: The first letter

Test item	Type of content	Duration
es02_48m	Male speech (German)	9 s
es03_48m	Female speech (English)	8 s
sc02_48m	Orchestral piece	13 s
sc03_48m	Contemporary pop music	12 s
si01_48m	Harpsichord	9 s
si02_48m	Castanets	7 s
si03_48m	Pitch pipe	29 s
sm01_48m	Bagpipes	11 s
sm02_48m	Glockenspiel	10 s
sm03_48m	Plucked strings	13 s

Tab. 6.1: *Test items that were selected for the listening test*

identifies the excerpt as speech (e) or sound (s) and music respectively. The second letter in case of speech indicates a single speaker (s), whereas for a sound and music likewise it has the following meaning: i stand for a single instrument, m for a simple sound mixture, and c for a complex mixture of sounds, accordingly. All signals are sampled at 48 kHz in mono (m).

6.4 Listening conditions

The MUSHRA test was conducted in the listening room of the Ilmenau University of Technology, an environment which is fully compliant with the acoustic requirements of [56]. STAX open air type electrostatic earspeakers were used in the test for sound reproduction. The gain of the amplifier was adjusted to a reference listening level of -18 dB below the clipping level of a digital tape recording.

6.5 Statistical analysis

The fundamental aim of the statistical analysis of test results is to accurately identify the average performance of each of the systems under test and the reliability of any differences among those average performance figures. The latter aspect requires an estimation of the variability of the results, typically based on an an *analysis of variance* (ANOVA) model. The absolute scores for the assessments of each system under test or *test condition* are derived from the normalized scores in the range of 0 to 100, where 0 corresponds to the bottom of the CQS. The calculation of the averages of the normalized scores of all listeners yields the *mean subjective scores* (MSS).

The first step in the analysis of the results is the calculation of the mean score $\bar{x}_{jk}$ for each of the presentations, according to

$$\bar{x}_{jk} = \frac{1}{N} \sum_{i=1}^{N} x_{ijk}, \tag{6.1}$$

where x_{ijk} is the score of assessor i for a given test condition j and audio sequence k, and N is the number of assessors. Similarly, overall mean scores $\bar{x}_j$ and $\bar{x}_k$ are calculated for each test condition and each test sequence.

The second step calculates the stochastic endpoints $\bar{x}_{jk} - \delta_{jk}$ and $\bar{x}_{jk} + \delta_{jk}$ of the 95 % confidence interval for the mean score $\bar{x}_{jk}$, where δ_{jk} is defined as

$$\delta_{jk} = t_{0.05} \frac{\sigma_{jk}}{\sqrt{N}}. \tag{6.2}$$

In Eq. (6.2), $t_{0.05}$ represents the quantile of the STUDENT's t-distribution with $N - 1$ degrees of freedom for the significance level 0.05. The standard error $\sigma_{jk}/\sqrt{N}$ is derived from the the number of listeners N and the standard deviation

$$\sigma_{jk} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_{ijk} - \bar{x}_{jk})^2}. \tag{6.3}$$

The standard deviations σ_j are determined in an analogous manner.

The confidence level allows the interpretation that in 95 % of the cases the true mean score μ_{jk} will lie between the endpoints calculated from the experimental mean $\bar{x}_{jk}$, on condition that the individual scores x_{ijk} are normally distributed. In 5 % of the cases, however, it will not.

6.6 Test results

Seventeen evaluators, with some few experienced listeners among them, have taken part in the test. To obtain meaningful and reliable results within the realms of possibility, the unexperienced listeners were exposed to the full range and nature of the occurring impairments and all of the audio sequences before the actual test. The training phase should have sufficiently sensitized the non-experts for the various types of artifacts encountered during the test. Nevertheless, four assessors were excluded in the post-screening phase, as they were found to be not able to discriminate between the artifacts with sufficient accuracy in the provided context.

The three HFR systems that were compared against each other in one session are listed in Tab. 6.2. The stand-alone version of the SBR tool was provided by the

HFR scheme	Label	Crossover frequency
Low-Delay Bandwidth Extension	LD-BWE	7.5 kHz
MPEG-4 Spectral Band Replication	SBR	7.5 kHz
Lowband-to-Highband LSF Mapping	CBM 8000	8 kHz

Tab. 6.2: *HFR systems under test*

Fraunhofer Institute for Integrated Circuits (IIS[2]). It is identical with the one in the HE-AAC profile. The lowermost system, labeled CBM 8000, recovers the wideband signal based on *codebook mapping* (CBM) of *vector quantized* (VQ) *line spectral frequencies* (LSF) used to represent the short-term spectral envelope of the missing high band. The scheme was developed in the Department of Applied Media Systems for audio coding applications that demand a low system delay and a low bit rate, and is hence the main competitor for the proposed technique. A detailed description of the system is given in [59] and [60]. Keeping with the SBR tunings for 48 kHz-sampled mono signals coded in perceptually high quality at a nominal bit rate of 40 kbps, the crossover frequency for LD-BWE was accordingly set to 7.5 kHz. The bandwidth of the CBM 8000 system constitutes one third of the NYQUIST frequency and is determined by the sampling rate. The broadband test sequences were internally downsampled by all of

[2] Institut für Integrierte Schaltungen

the HFR systems, and then reconstructed from the narrowband PCM signal based on a parametric description of how the high band correlates with the provided low band. The experiment was hence conducted without a core codec to better differentiate between the coding artifacts of the *bandwidth extension* (BWE) schemes under test.

Fig. 6.1 illustrates the mean and the 95 % confidence interval of the statistical distribution of the subjective scores for the hidden reference, the anchor, and the three test conditions. The overall mean score for the test items covering pure speech, musical

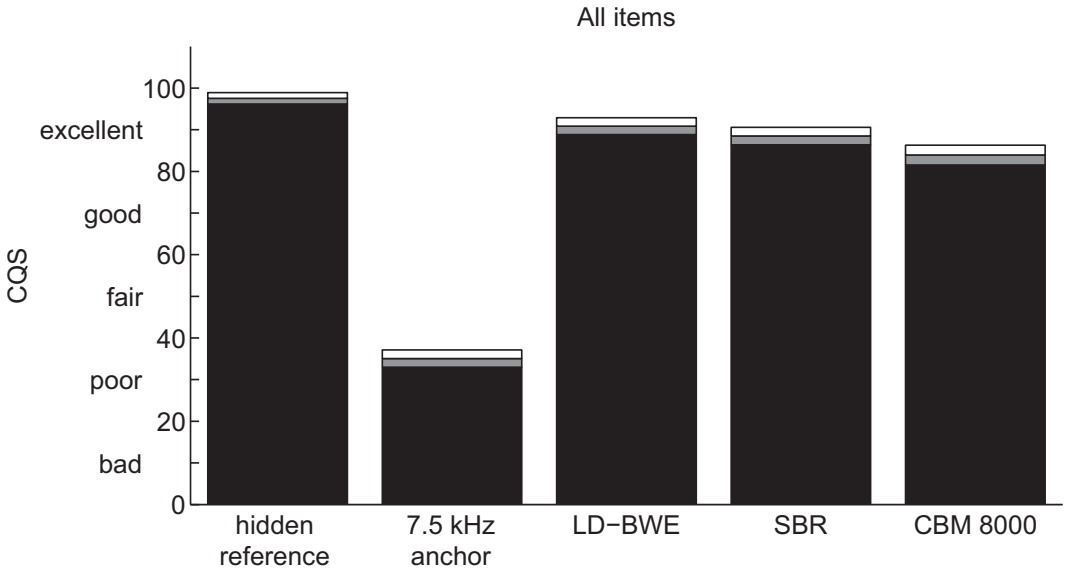

Fig. 6.1: *Mean scores and* 95 *% confidence intervals for the tested HFR systems*

instruments, and music can be seen in B. Quite clearly, with a mean score of 90.9, the proposed LD-BWE scheme outperformed the LPC-based CBM 8000 system, which ended up in third place with a mean of 83.9. SBR was assessed second best scoring 88.5 on average. At this point, I would like to draw the reader's attention to the fact that these results conflict with the results of previous listening tests presented in [60], where the codebook mapping technique was declared the clear winner over SBR. This statement is to be strongly doubted, particularly looking at the system performance for the following test items: speech, contemporary pop music, castantes, and glockenspiel. The inability to control the tonal-to-noise ratio in the reconstructed high band often results in a hollow or metallic sounding copy of the original wave, which is perceived as annoying. When overtones get lost, the harmonic structure of

the sound can become audibly defective. Distinctively chordal sound pieces may even suffer from dissonance, if the missing tones are not synthetically recovered. As none of these effects is sufficiently handled in CBM 8000, its superiority over SBR is highly questionable. The relatively high bandwidth and the good temporal resolution due to short processing blocks might have affected the listeners' judgment in favor of the system. With the other test sequences all of the tested systems performed comparably well, whereas for speech and percussive sounds, in particular, LD-BWE stood out from the other two. More distinct differences between the systems are to be expected for lower crossover frequencies, since artifacts should become more audible and hence easier to detect due to the growing impairment in quality in comparison to the unprocessed signal.

7. SUMMARY

A low-delay bandwidth extension method, namely LD-BWE, has been proposed in this thesis. LD-BWE is a delay-optimized version of MPEG-4 SBR, designed in such a manner that it can be used together with a low-delay speech and audio coder like the ULD. The presented approach is based on a short-time subband representation of an acoustic signal of natural or artificial origin, and as such it utilizes a complex-exponential modulated low-delay filter bank for the extraction and the manipulation of sound characteristics. A further delay reduction has been achieved by analyzing the statistical properties of a finite number of subband samples within the range of one ULD frame, so that the necessity for a look-ahead could be avoided. The overall system delay for a combination of the ULD coder with the LD-BWE tool adds up to 12 ms at a sampling rate of 48 kHz, which is less than two-fifths of AAC-ELD's delay. At the present stage, LD-BWE generates a subjectively confirmed excellent-quality highband replica at a simulated mean data rate of 12.8 kbps. Hence, all of the aforementioned allows the conclusion that the major goal of the thesis, which was to elaborate a low-delay version of SBR for ULD, has been accomplished.

The algorithmic delay of the proposed bandwidth extension technique comes for the most part from upsampling the time-delayed lowband signal, so that a further delay reduction would call for an updated version of the ULD coder. As is explained in Secs. 5.2 and 5.8, a combination of ULD and LD-BWE would exhibit a cumulative system delay of only 6.6 ms at 48 kHz sampling rate, if the frame size was halved and the DFT window was respectively decimated by a factor of two. On the other hand, this would require further modifications in the LD-BWE tool. The transient detection, e. g., would have to be carried out in the time domain or might even be completely removed from the LD-BWE encoder. As an alternative to the current approach, the noise floor level measure could be computed as the difference between the peak and

dip follower functions as outlined in [40]. Other tonality-based estimators would need a matching substitute as well. One should bear in mind, though, that shorter frames inevitably cause losses in coding efficiency.

The other objective of importance, to preserve the perceptual performance of SBR, has also been achieved. The MUSHRA test has moreover confirmed that LD-BWE performs better than SBR in the case of unsteady or transient sounds, reducing the effect of audible pre-echoes. Nonetheless, the subjective character of the MUSHRA method has definitely shone through in the form of fairly large confidence intervals for certain sound pieces.

The simulated average bit rate of 12.8 kbps is roughly four times the bit rate of SBR. A more efficient redundancy coding scheme must hence be found, in order to attain a higher compression ratio. Delta encoding of spectral data along the frequency axis should be complemented by data differencing between sequential envelopes, e. g. In addition to that, the spectral envelope data may also be vector quantized. A further data reduction can be achieved by estimating the sinusoidal model parameters of high-frequency components on the decoder side, so as to circumvent the transmission of the frequency location of missing harmonics. It is also conceivable that by considering the threshold in quiet and the masking threshold, respectively, the irrelevant part of the envelope data, and thus the overall data rate, could be minimized without any degradation in perceived sound quality.

The following points can be made with regard to the complexity of LD-BWE alone and LD-BWE used in conjunction with ULD: Exchanging the CLDF bank for its real-valued counter part would lower the computational load, and a fast implementation of the DCT could be used for the analysis and the synthesis. The CLDF bank in the LD-BWE encoder might also be replaced by a delay-optimized *modified* DCT (MDCT) [61] filter bank, so that one signal spectrum could be shared between the LD-BWE encoder and the ULD encoder. The *modified* DST (MDST) would have to be implemented on the ULD side as well, so as not to lose important phase information. The computational complexity of the MDCT (MDST) might then be minimized by means of frequency warping, i. e., by partitioning the frequency range of interest in a non-uniform fashion corresponding to the Bark scale, in order to reduce the number of subbands.

BIBLIOGRAPHY

[1] Homer W. Dudley. *System for the Artificial Production of Vocal or Other Sounds*, June 1938. US 2,121,142.

[2] Hugo Fastl and Eberhard Zwicker. *Psychoacoustics – Facts and Models*. Springer, Berlin, 3rd edition, 2007.

[3] Wai C. Chu. *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. John Wiley & Sons, 2003.

[4] Bernd Edler and Gerald Schuller. Audio coding using a psychoacoustic pre- and post-filter. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000)*, pages 881–884, Istanbul, Turkey, June 2000.

[5] Bernd Edler, Christof Faller, and Gerald Schuller. Perceptual audio coding using a time-varying linear pre- and post-filter. In *109th AES Convention*, Los Angeles, CA, USA, September 2000.

[6] Gerald Schuller and Aki Härmä. Low delay audio compression using predictive coding. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, pages 1853–1856, Orlando, FL, USA, May 2002.

[7] Gerald D. T. Schuller, Bin Yu, Dawei Huang, and Bernd Edler. Perceptual audio coding using adaptive pre- and post-filters and lossless compression. *IEEE Transactions on Speech and Audio Processing*, pages 379–390, September 2002.

[8] Ulrich Krämer, Gerald Schuller, Stefan Wabnik, Juliane Klier, and Jens Hirschfeld. Ultra Low Delay audio coding with constant bit rate. In *117th AES Convention*, San Francisco, CA, USA, October 2004.

[9] Stefan Wabnik, Gerald Schuller, Krämer Ulrich, and Jens Hirschfeld. Frequency warping in low delay audio coding. In *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, pages 181–184, Philadelphia, PA, USA, March 2005.

[10] Stefan Wabnik, Gerald Schuller, Jens Hirschfeld, and Uli Krämer. Packet loss concealment in ppredictive audio coding. In *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005)*, pages 227–230, New Paltz, NY, USA, October 2005.

[11] Stefan Wabnik, Gerald Schuller, Jens Hirschfeld, and Ulrich Krämer. Reduced bit rate Ultra Low Delay audio coding. In *120th AES Convention*, Paris, France, May 2006.

[12] Manfred Lutzky, Gerald Schuller, Marc Gayer, Ulrich Krämer, and Stefan Wabnik. A guideline to audio codec delay. In *116th AES Convention*, Berlin, Germany, May 2004.

[13] ISO/IEC. *Information technology – Coding of audio-visual objects – Part 3: Audio, Subpart 4*, 2005. ISO/IEC 14496-3:2005.

[14] Karlheinz Brandenburg. Low bitrate audio coding – state-of-the-art, challenges and future directions. *Fraunhofer Publica [http://publica.fraunhofer.de/oai.har] (Germany)*, 2000.

[15] Karlheinz Brandenburg, Oliver Kunz, and Akihiko Sugiyama. MPEG-4 natural audio coding. *Signal Processing: Image Communication*, pages 423–444, January 2000.

[16] Eric Allamanche, Ralf Geiger, Jürgen Herre, and Thomas Sporer. MPEG-4 low delay audio coding based on the AAC codec. In *106th AES Convention*, Munich, Germany, May 1999.

[17] Andreas Spanias, Ted Painter, and Venkatraman Atti. *Audio Signal Processing and Coding*. John Wiley & Sons, 2007.

[18] Martin Dietz, Lars Liljeryd, Kristofer Kjörling, and Oliver Kunz. Spectral Band Replication, a novel approach in audio coding. In *112th AES Convention*, Munich, Germany, May 2002.

[19] Martin Wolters, Kristofer Körling, Daniel Homm, and Heiko Purnhagen. A closer look into MPEG-4 High Efficiency AAC. In *115th AES Convention*, New York, NY, USA, October 2003.

[20] Stefan Meltzer, Reinhold Böhm, and Fredrik Henn. SBR enhanced audio codecs for digital broadcasting such as "Digital Radio Mondiale" (DRM). In *112th AES Convention*, Munich, Germany, May 2002.

[21] Alexander Gröschel, Michael Schug, Michael Beer, and Fredrik Henn. Enhancing audio coding efficiency of MPEG Layer-2 with Spectral Band Replication for digital radio (DAB) in a backwards compatible way. In *114th AES Convention*, Amsterdam, The Netherlands, March 2003.

[22] Andreas Ehret and Michael Schug. aacPlus for ring tones. In *AES 12th Regional Convention*, Tokyo, Japan, July 2005.

[23] Erik Larsen and Ronald M. Aarts. *Audio Bandwidth Extension.* John Wiley & Sons, 2004.

[24] ISO/IEC. *Information technology – Generic coding of moving pictures and associated audio information – Part 3: Advanced Audio Coding (AAC), AMENDMENT 9: Enhanced low delay AAC*, April 2007. ISO/IEC 14496-3:2005/FPDAM 9.

[25] Curtis Roads, Stephen Travis Pope, Aldo Picialli, and Giovanni De Poli, editors. *Musical Signal Processing*, chapter Musical Sound Modeling with Sinusoids plus Noise. Swets & Zeitlinger, 1997.

[26] Albertus C. den Brinker, Jeroen Breebaart, Per Ekstrand, et al. An overview of the coding standard MPEG-4 audio amendments 1 and 2: HE-AAC, SSC, and HE-AAC v2. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[27] Xuancheng Shaoa and Steven G. Johnson. Type-II/III DCT/DST algorithms with reduced number of arithmetic operations. *Signal Processing*, pages 1553–1564, June 2008.

[28] Xuancheng Shaoa and Steven G. Johnson. Type-IV DCT, DST, and MDCT algorithms with reduced numbers of arithmetic operations. *Signal Processing*, pages 1313–1326, June 2008.

[29] Per Ekstrand. Bandwidth extension of audio signals by Spectral Band Replication. In *1st IEEE Benelux workshop on Model based Processing and Coding of Audio (MPCA-2002)*, pages 53–58, Leuven, Belgium, November 2002.

[30] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1993.

[31] ETSI. *General audio codec audio processing functions; Enhanced aacPlus general audio codec; General description (3GPP TS 26.401 version 8.0.0 Release 8)*, January 2009. ETSI TS 126 401 V8.0.0.

[32] ETSI. *General audio codec audio processing functions; Enhanced aacPlus general audio codec; Encoder specification; Spectral Band Replication (SBR) part (3GPP TS 26.404 version 8.0.0 Release 8)*, January 2009. ETSI TS 126 404 V8.0.0.

[33] ETSI. *General audio codec audio processing functions; Enhanced aacPlus general audio codec; Floating-point ANSI-C code (3GPP TS 26.410 version 8.0.0 Release 8)*, January 2009. ETSI TS 126 410 V8.0.0.

[34] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall International, 2nd edition, 1999.

[35] J. Princen and J. D. Johnston. Audio coding with signal adaptive filterbanks. In *1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1995)*, pages 3071–3074, Detroit, MI, USA, May 1995.

[36] Julius O. Smith III and Jonathan S. Abel. Bark and ERB bilinear transforms. *IEEE Transactions on Speech and Audio Processing*, pages 697–708, November 1999.

[37] Simon Haykin. *Adaptive Filter Theory*. Prentice-Hall International, 4th edition, 2001.

[38] Karl-Dirk Kammeyer and Kristian Kroschel. *Digitale Signalverarbeitung*. B. G. Teubner, 6. edition, 2006.

[39] Kristofer Kjörling, Per Ekstrand, Fredrik Henn, and Lars Villemoes. *Enhancing perceptual performance of high frequency reconstruction coding methods by adaptive filtering*, April 2004. EP 1 342 230.

[40] Lars Gustaf Liljeryd, Kristofer Kjörling, Per Ekstrand, and Fredrik Henn. *Enhancing perceptual performance of SBR and related HFR coding methods by adaptive noise-floor addition and noise substitution limiting*, September 2004. EP 1 157 374.

[41] Heiko Purnhagen, Nikolaus Meine, and Bernd Edler. Sinusoidal coding using loudness-based component selection. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, pages 1817–1820, Orlando, FL, USA, May 2002.

[42] Heiko Purnhagen. Parameter estimation and tracking for time-varying sinusoids. In *1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, pages 1817–1820, Leuven, Belgium, November 2002.

[43] David A. Huffman. A method for the construction of minimum-redundancy codes. In *Proceedings of the I.R.E.*, pages 1098–1101, Cambridge, MA, USA, September 1952.

[44] Lars Gustaf Liljeryd, Kristofer Kjörling, Per Ekstrand, and Fredrik Henn. *Efficient spectral envelope coding using variable time/frequency resolution*, July 2004. EP 1 216 474.

[45] Lars Gustaf Liljeryd, Per Ekstrand, Fredrik Henn, and Kristofer Kjörling. *Improved spectral translation/folding in the subband domain*, September 2003. EP 1 285 436.

[46] Gerald Schuller and Mark J. T. Smith. A general formulation for modulated perfect reconstruction filter banks with variable system delay. In *NJIT Symposium on Applications of Subbands and Wavelets*, Newark, NJ, USA, March 1994.

[47] Gerald Schuller and Mark J. T. Smith. Efficient low delay filter banks. In *Sixth IEEE Digital Signal Processing Workshop*, pages 231–234, Yosemite National Park, CA, USA, October 1994.

[48] Gerald D. T. Schuller and Mark J. T. Smith. New framework for modulated perfect reconstruction filter banks. *IEEE Transactions on Signal Processing*, pages 1941–1954, August 1996.

[49] Gerald Schuller. A new factorization and structure for cosine modulated filter banks with variable system delay. In *Thirtieth Asilomar Conference on Signals, Systems and Computers*, pages 1310–1314, Pacific Grove, CA, USA, November 1996.

[50] Gerald D. T. Schuller and Tanja Karp. Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case. *IEEE Transactions on Signal Processing*, pages 737–748, March 2000.

[51] Markus Schnell, Ralf Geiger, Markus Schmidt, et al. Low delay filterbanks for Enhanced Low Delay audio coding. In *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2007)*, pages 235–237, New Paltz, NY, USA, October 2007.

[52] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on Computers*, 20(5):562–569, May 1971.

[53] Mark Kahrs and Karlheinz Brandenburg. *Applications of Digital Signal Processing to Audio and Acoustics*. Springer, Berlin, 1998.

[54] Andreas Ehret, Martin Dietz, and Kristofer Kjörling. State-of-the-art audio coding for broadcasting and mobile applications. In *114th AES Convention*, Amsterdam, The Netherlands, March 2003.

[55] ITU-R. *A guide to ITU-R Recommendations for subjective assessment of sound quality*, December 2003. Recommendation ITU-R BS.1283.

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

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

[58] Gerhard Stoll and Franc Kozamernik. EBU listening tests on internet audio codecs. *EBU Technical Review*, June 2000.

[59] Tobias Friedrich. Spectral Band Replication tool for very low delay audio coding applications. Diploma thesis, Ilmenau University of Technology, Ilmenau, Germany, February 2007.

[60] Michael Werner. Weiterentwicklung eines SBR Verfahrens für Audiocodierungsanwendungen mit geringer Verzögerung. Diploma thesis, Ilmenau University of Technology, Ilmenau, Germany, September 2008.

[61] Bernd Edler. Coding of audio signals with overlapping block transform and adaptive window functions. *Frequenz*, 43(9):252–256, 1989.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

3GPP 3rd Generation Partnership Project

AAC Advanced Audio Coding

AAC-ELD Enhanced Low-Delay Advanced Audio Coding

AAC-LD Low-Delay Advanced Audio Coding

ANOVA Analysis of Variance

AR Autoregressive

BWE Bandwidth Extension

CBM Codebook Mapping

CLDF Complex Low-Delay Filter

CELP Code Excited Linear Prediction

CPQMF Complex Pseudo Quadrature Mirror Filter

CQMF Complex Quadrature Mirror Filter

CQS Continuous Quality Scale

DC Direct Current

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DIF Decimation In Frequency

DST Discrete Sine Transform

FIR Finite Impulse Response

HE-AAC High-Efficiency Advanced Audio Coding

HFR High-Frequency Reconstruction

IDFT Inverse Discrete Fourier Transform

IDMT Institute for Digital Media Technology

IIR Infinite Impulse Response

IIS Institute for Integrated Circuits (*Ger.* Institut für Integrierte Schaltungen)

ITU International Telecommunication Union

LD-BWE Low-Delay Bandwidth Extension

LD-SBR Low-Delay Spectral Band Replication

LMS Least Mean Squares

LPC Linear Prediction Coding

LSF Line Spectral Frequency

MDCT Modified Discrete Cosine Transform

MDST Modified Discrete Sine Transform

MELP Mixed Excitation Linear Prediction

MPEG Moving Picture Experts Group

MSS Mean Subjective Score

MUSHRA Multi Stimulus Test with Hidden Reference and Anchor

OLA Overlap-Add

PCM Pulse Code Modulation

PQMF Pseudo Quadrature Mirror Filter

PR Perfect Reconstruction

QMF Quadrature Mirror Filter

SBR Spectral Band Replication

TF Time-Frequency

ULD Ultra Low Delay

VQ Vector Quantization

WSS Wide-Sense Stationary

LIST OF PRINCIPAL SYMBOLS

$a_k(n)$ Envelope gain

$e_{\kappa,\max}(\nu)$ Maximum energy in κ-th frequency band and ν-th envelope

$\bar{e}_\kappa(\nu)$ Mean energy in κ-th frequency band and ν-th envelope

$e_\kappa^{\lceil}(\nu)$ Quantized envelope scale factor in κ-th frequency band and ν-th envelope

f_c Crossover frequency

f_s Sampling frequency

$g_k(n)$ k-th synthesis filter

$h_k(n)$ k-th analysis filter

$k_1(\kappa)$ First subband in κ-th frequency band

$k_2(\kappa)$ Last subband in κ-th frequency band

k_c Crossover subband

$n_1(\nu)$ First subsample in ν-th envelope

$n_2(\nu)$ Last subsample in ν-th envelope

$n_k(n)$ Noise magnitude

n_t Transient onset position

$P(z)$ Linear prediction error filter

$P(z,\beta)$ Pre-whitening filter

$p_0(n)$ Prototype filter

$p_a(n)$ Analysis prototype filter

$p_s(n)$ Synthesis prototype filter

p_n Linear prediction filter coefficients

$q_{\kappa,\max}(\nu)$ Maximum noise floor level in κ-th frequency band and ν-th envelope

$\bar{q}_\kappa(\nu)$ Mean noise floor level in κ-th frequency band and ν-th envelope

$q_\kappa^{\lceil}(\nu)$ Quantized noise floor scale factor in κ-th frequency band and ν-th envelope

$s_k(n)$ Sinusoid magnitude

$T_{\kappa,\max}(\nu)$ Maximum original tonality in κ-th frequency band and ν-th envelope

$T'_{\kappa,\max}(\nu)$ Maximum replicated tonality in κ-th frequency band and ν-th envelope

$\bar{T}_\kappa(\nu)$ Mean original tonality in κ-th frequency band and ν-th envelope

$\bar{T}'_\kappa(\nu)$ Mean replicated tonality in κ-th frequency band and ν-th envelope

$\tilde{t}_i(k)$ Threshold in i-th frame and k-th subband

$x_k(n)$ k-th subband signal

$x'_k(n)$ k-th transposed subband signal

$y_k(n)$ k-th highband signal

$z_k(n)$ k-th wideband signal

$\beta(k)$ Bandwidth expansion factor in k-th subband

$\eta_k(n)$ Additive white GAUSSIAN noise with zero mean and unit variance

$\xi_k(n)$ Synthetic sinusoid

APPENDIX

A. SPECTROGRAMS

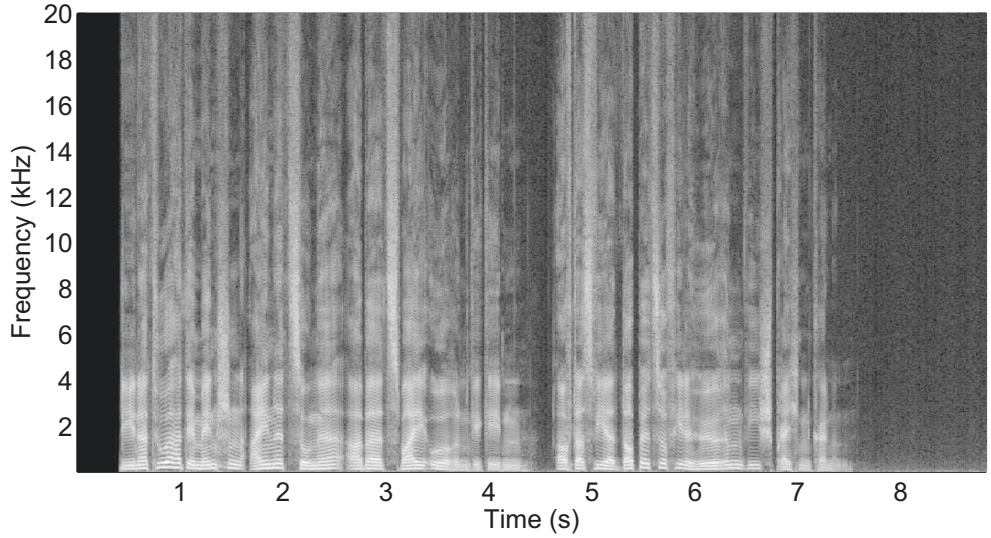

Fig. A.1: *Male speech (German), original (`es02_48m`)*

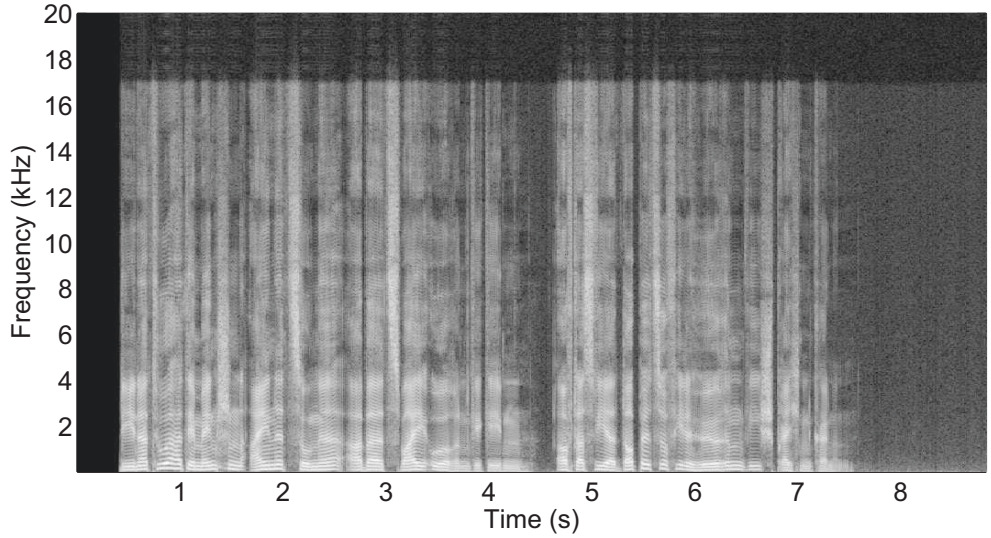

Fig. A.2: *Male speech (German), replica (`es02_48m`)*

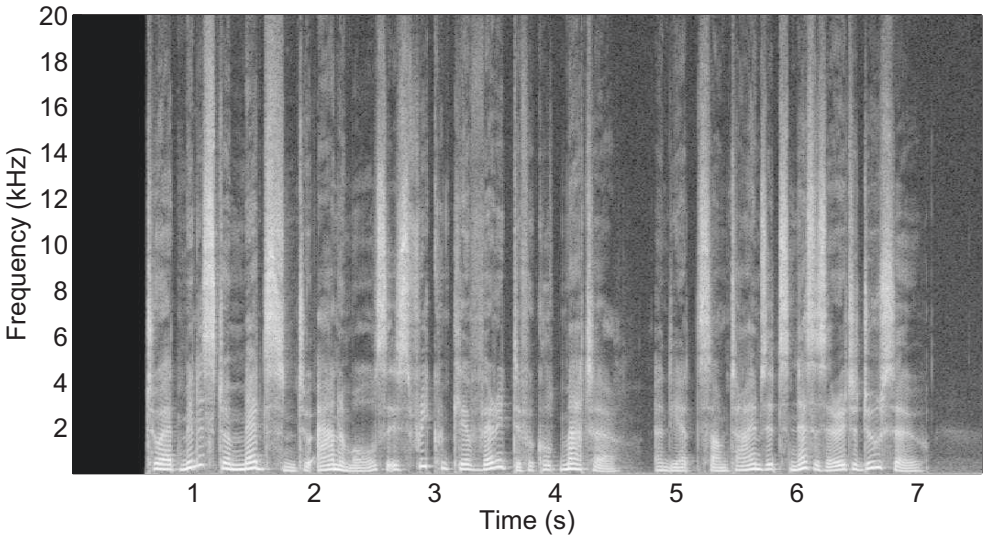

Fig. A.3: *Female speech (English), original (`es03_48m`)*

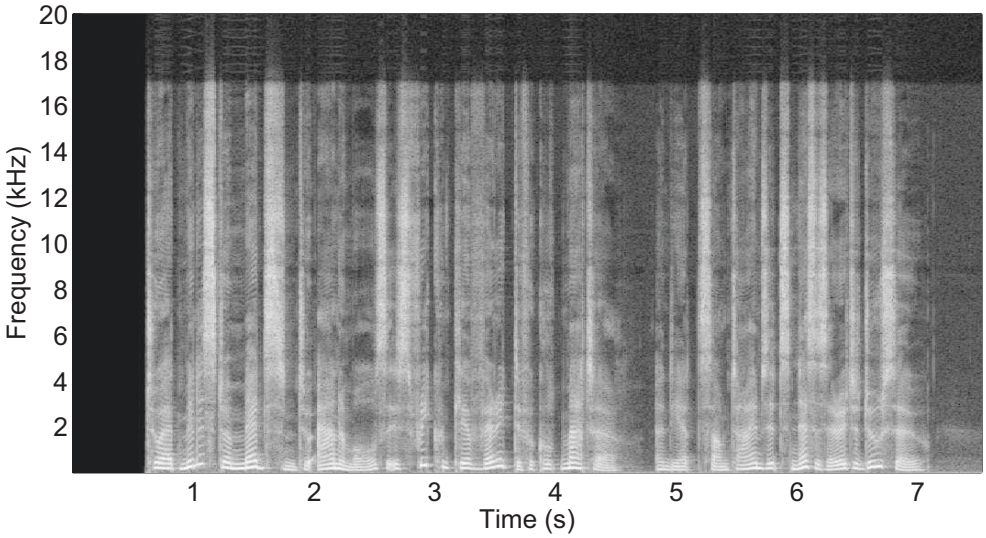

Fig. A.4: *Female speech (English), replica (`es03_48m`)*

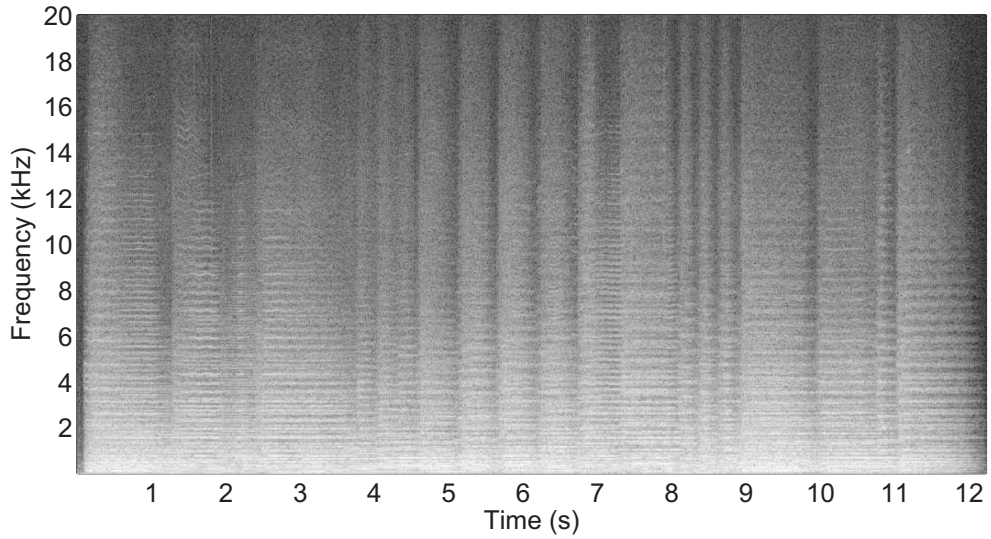

Fig. A.5: *Orchestral piece, original (`sc02_48m`)*

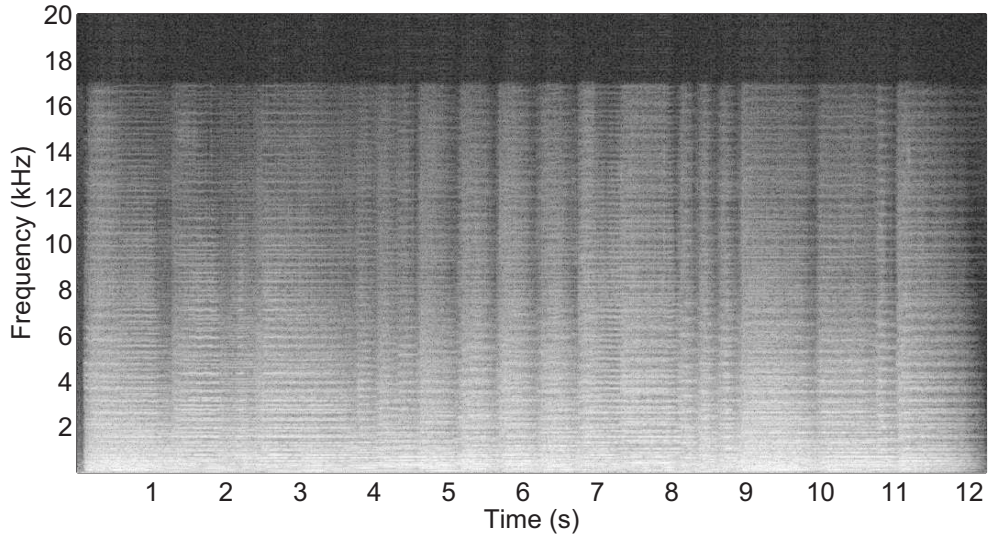

Fig. A.6: *Orchestral piece, replica (`sc02_48m`)*

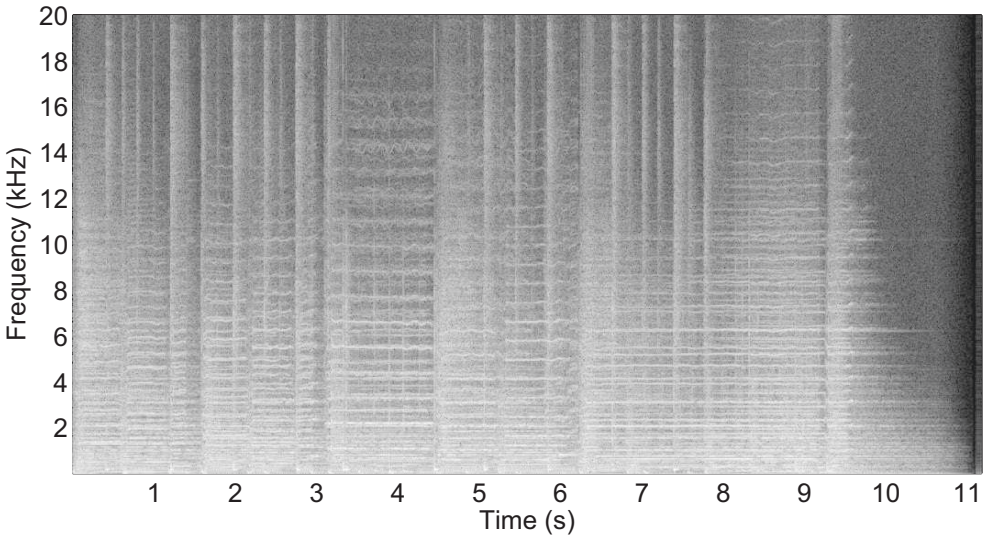

Fig. A.7: *Contemporary pop music, original (`sc03_48m`)*

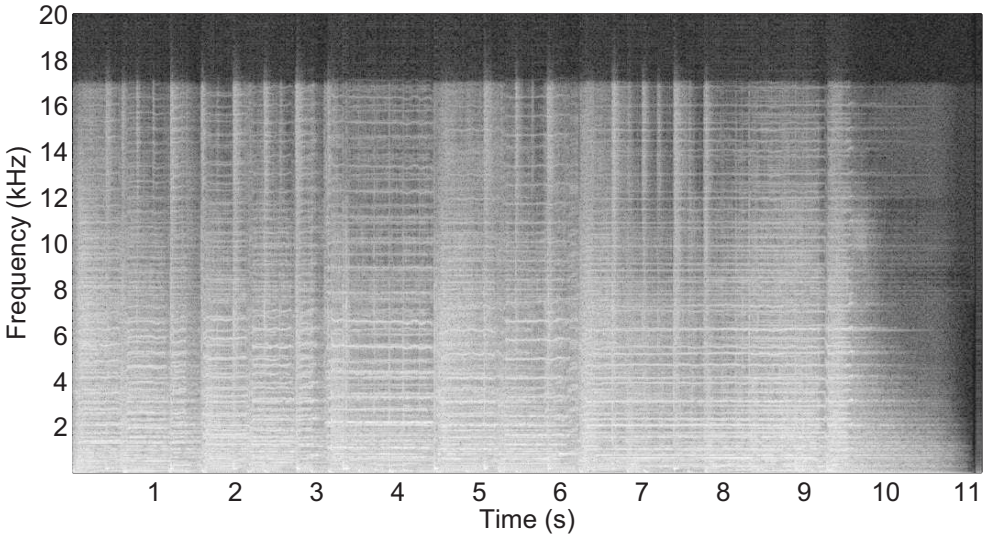

Fig. A.8: *Contemporary pop music, replica (`sc03_48m`)*

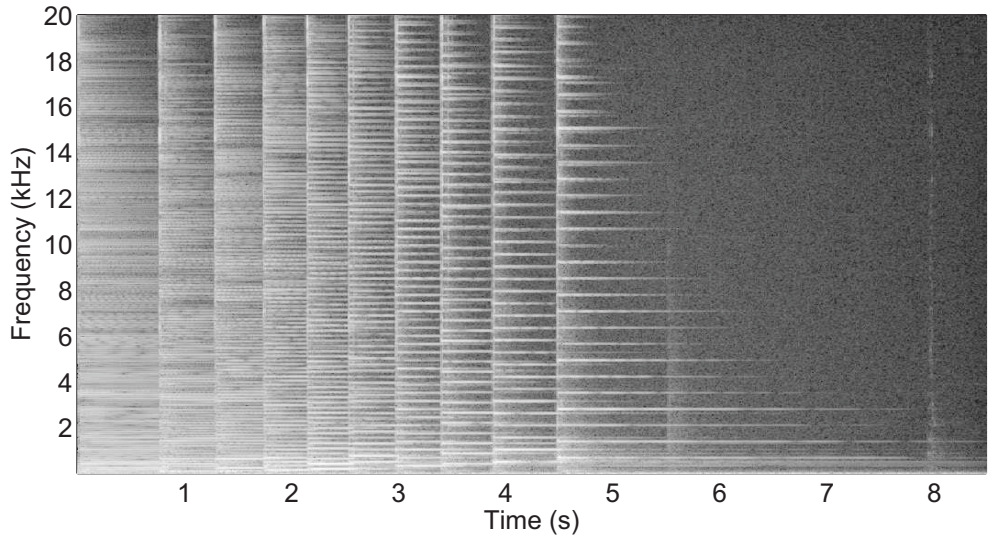

Fig. A.9: *Harpsichord, original (`si01_48m`)*

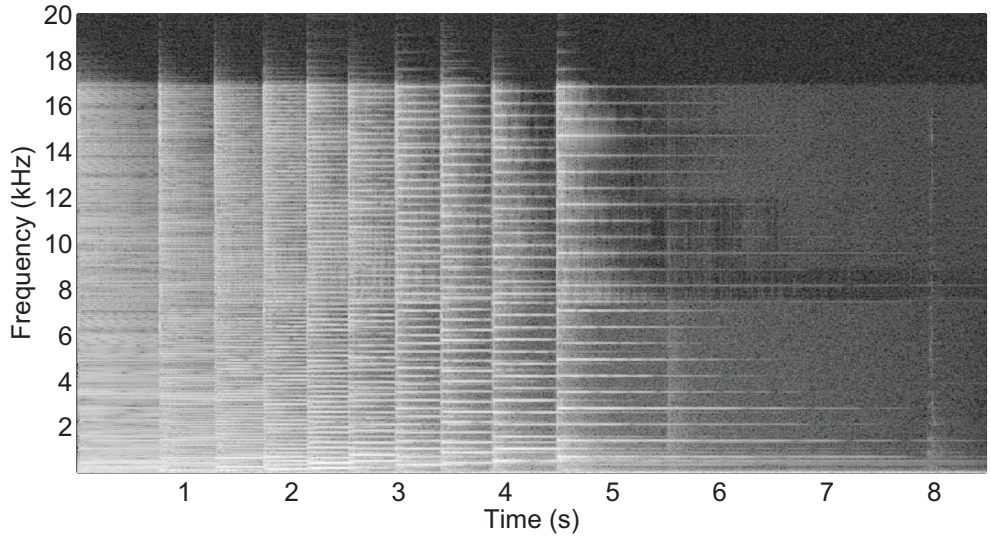

Fig. A.10: *Harpsichord, replica (`si01_48m`)*

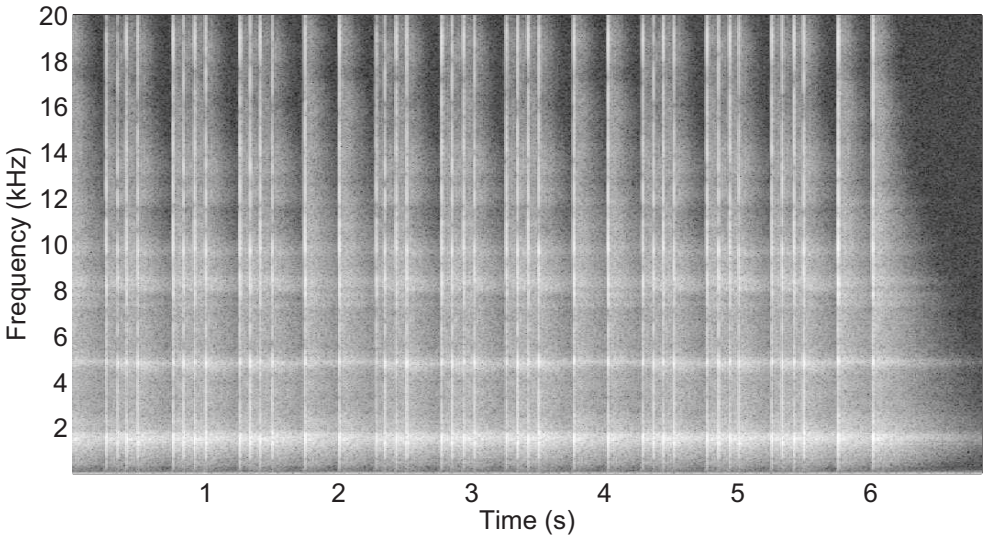

Fig. A.11: *Castanets, original (`si02_48m`)*

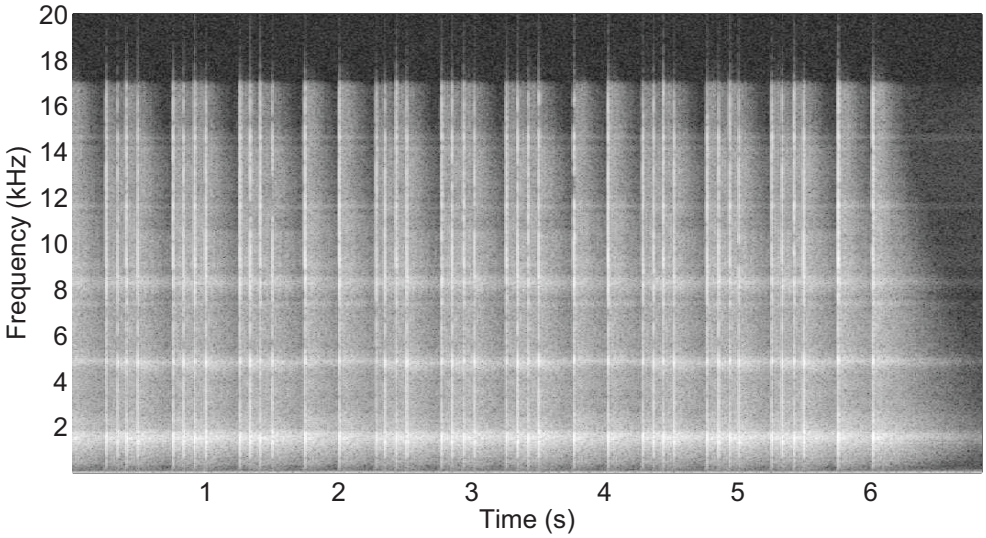

Fig. A.12: *Castanets, replica (`si02_48m`)*

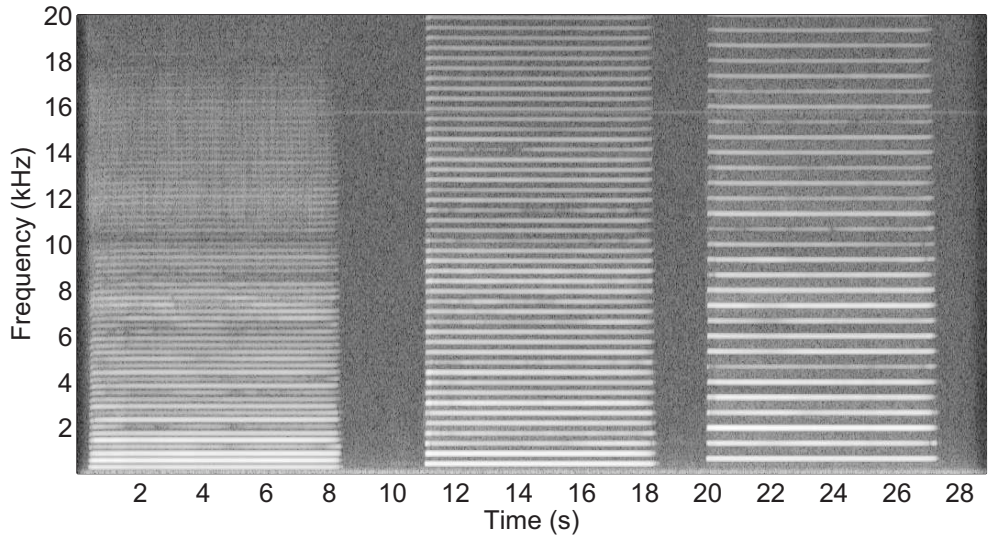

Fig. A.13: *Pitch pipe, original (`si03_48m`)*

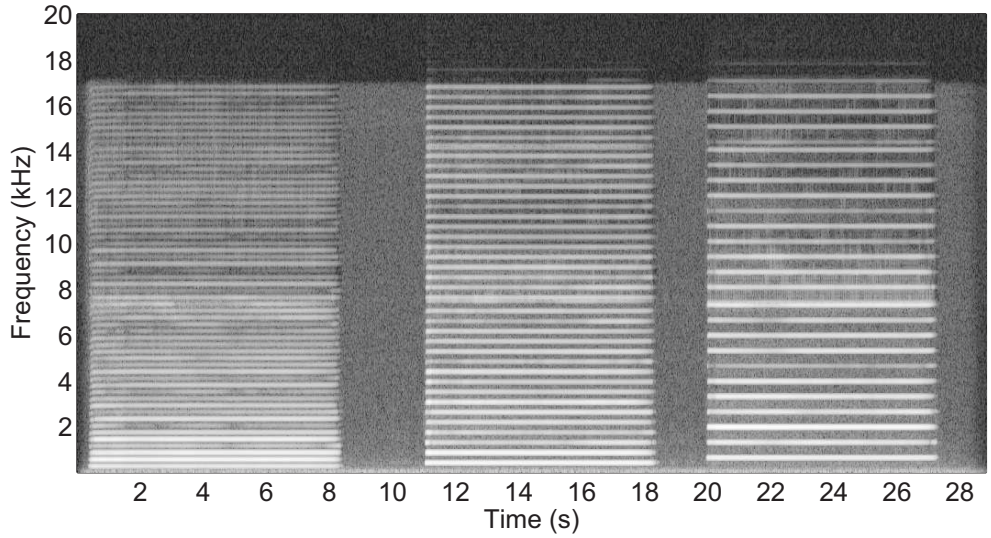

Fig. A.14: *Pitch pipe, replica (`si03_48m`)*

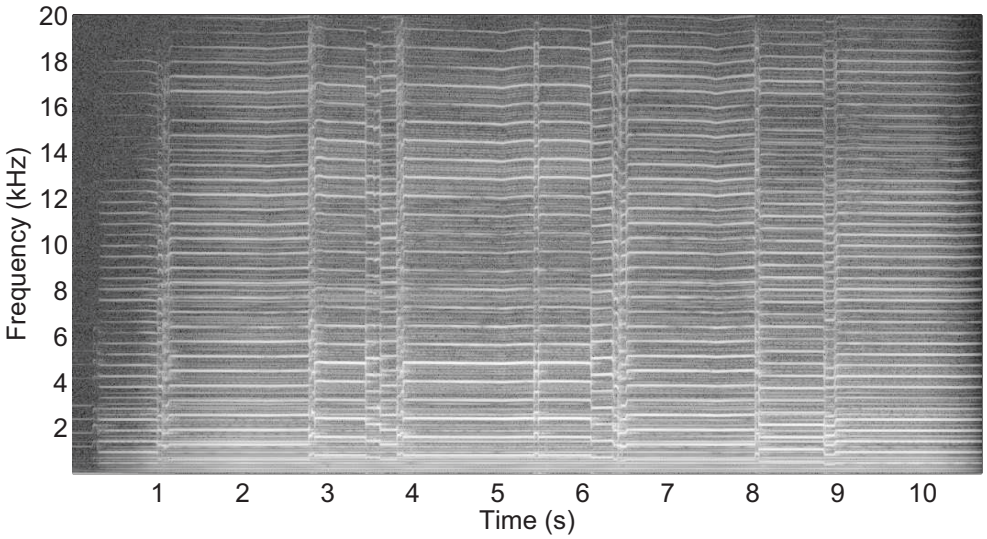

Fig. A.15: *Bagpipes, original (sm01_48m)*

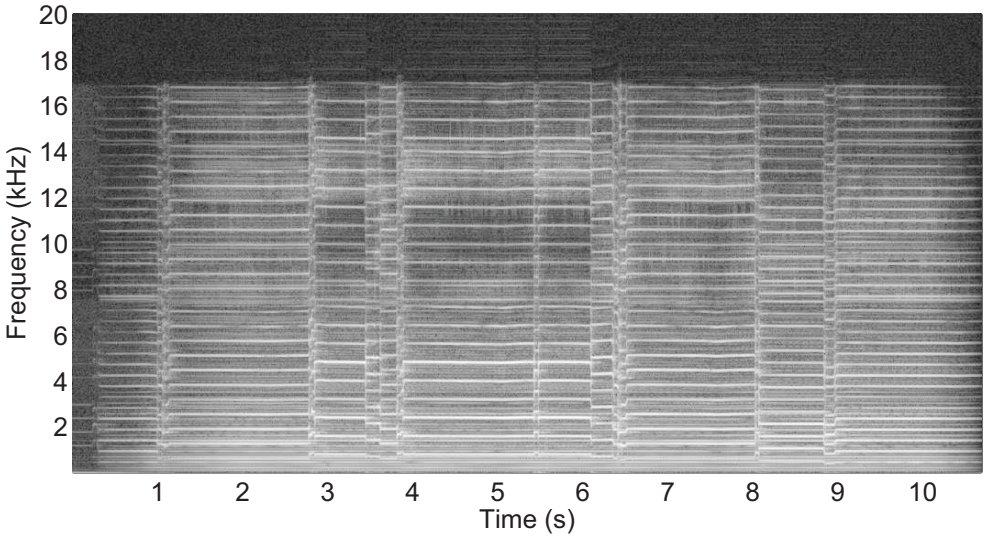

Fig. A.16: *Bagpipes, replica (sm01_48m)*

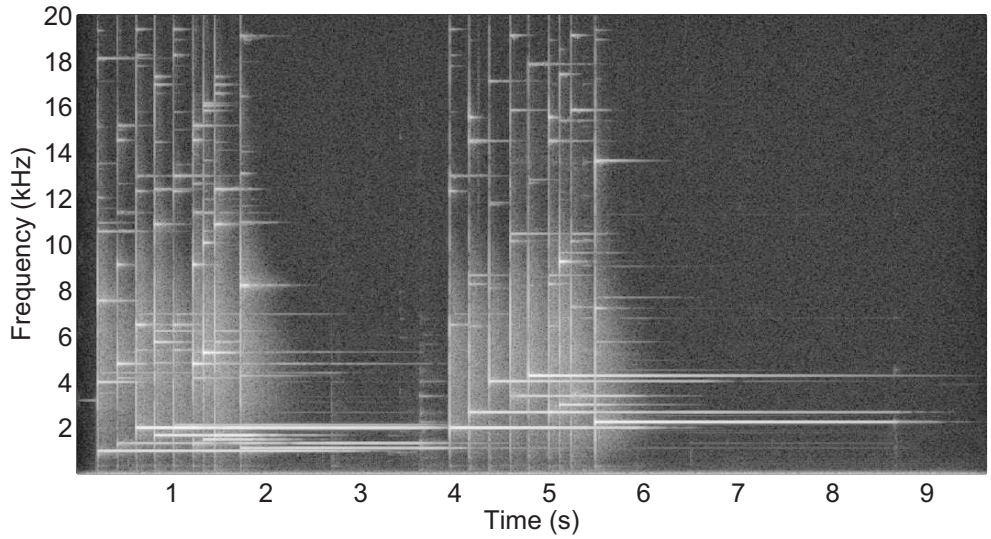

Fig. A.17: *Glockenspiel, original (*`sm02_48m`*)*

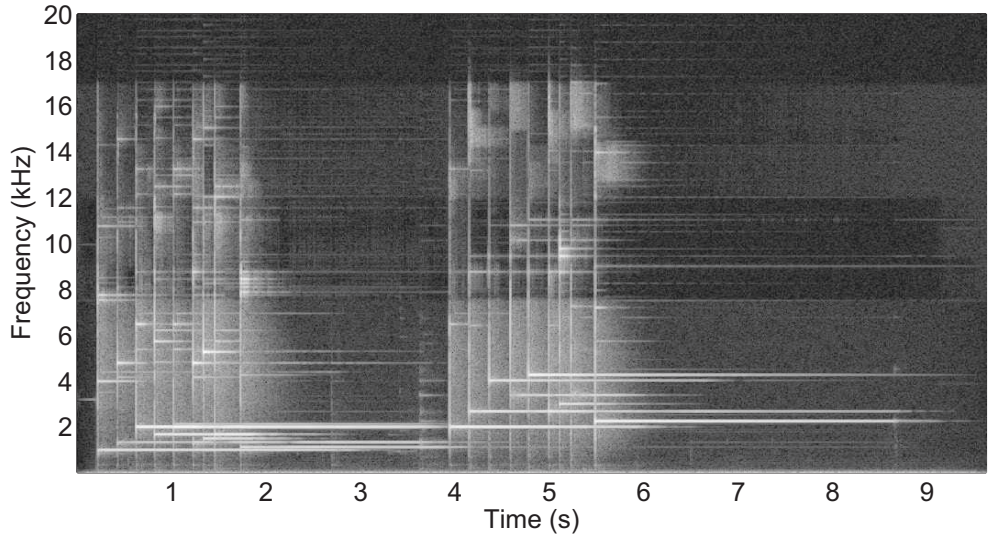

Fig. A.18: *Glockenspiel, replica (*`sm02_48m`*)*

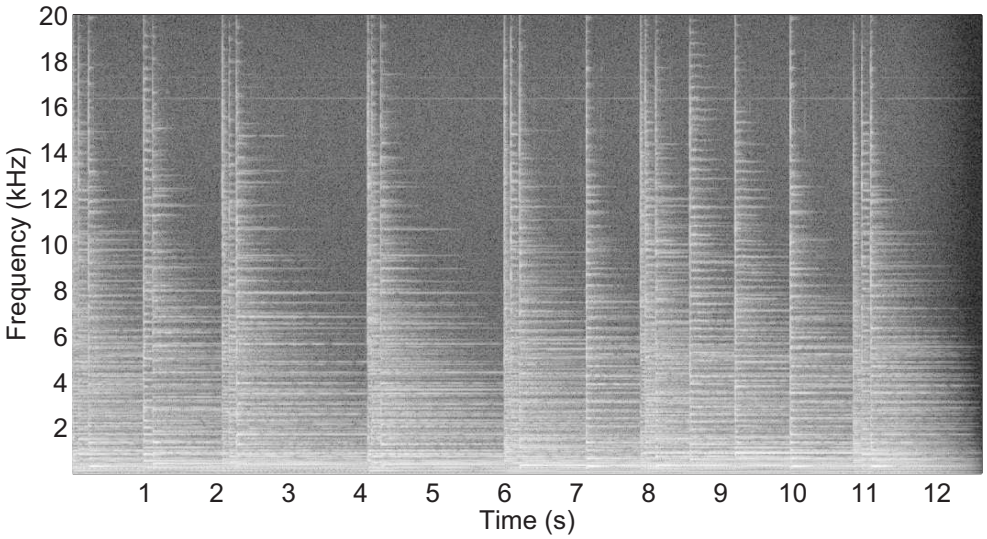

Fig. A.19: *Plucked strings, original (`sm03_48m`)*

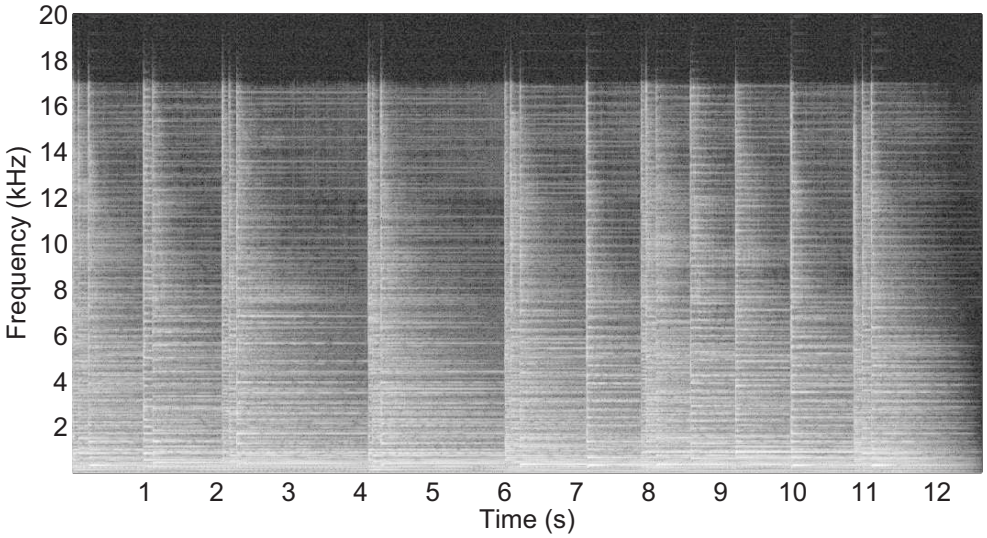

Fig. A.20: *Plucked strings, replica (`sm03_48m`)*

B. MEAN SUBJECTIVE SCORES

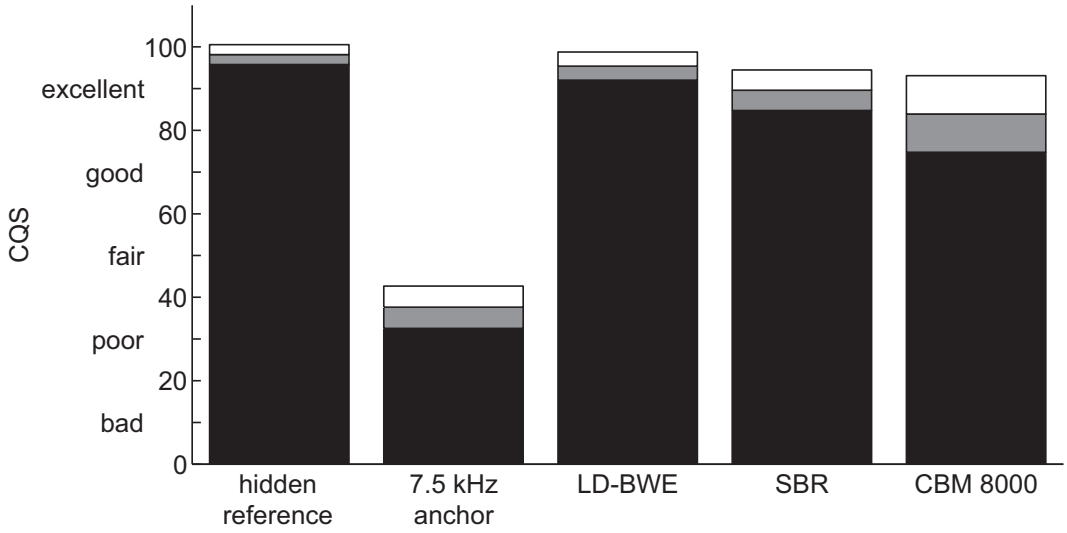

Fig. B.1: *Male speech, German (es02_48m)*

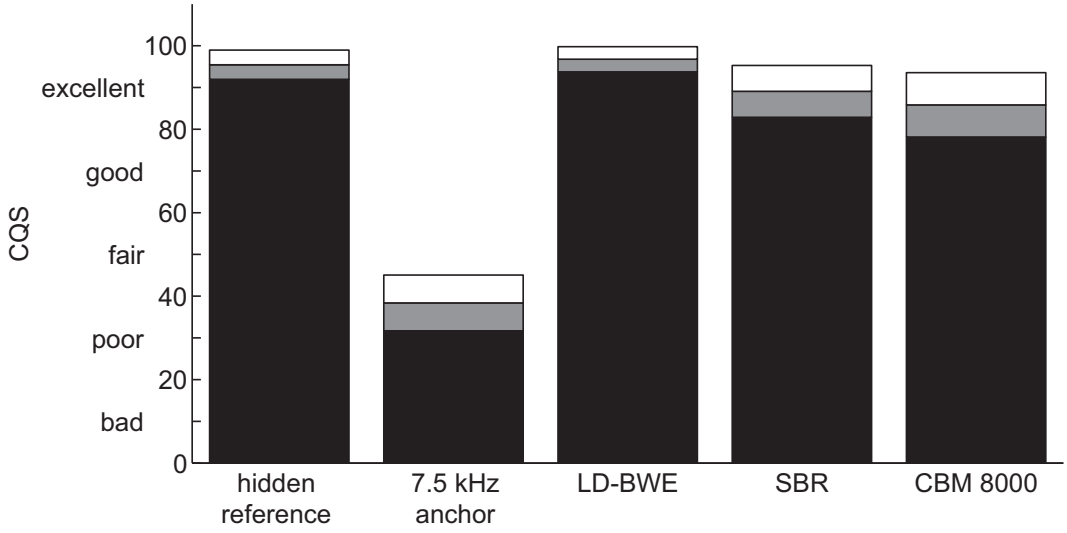

Fig. B.2: *Female speech, English (es03_48m)*

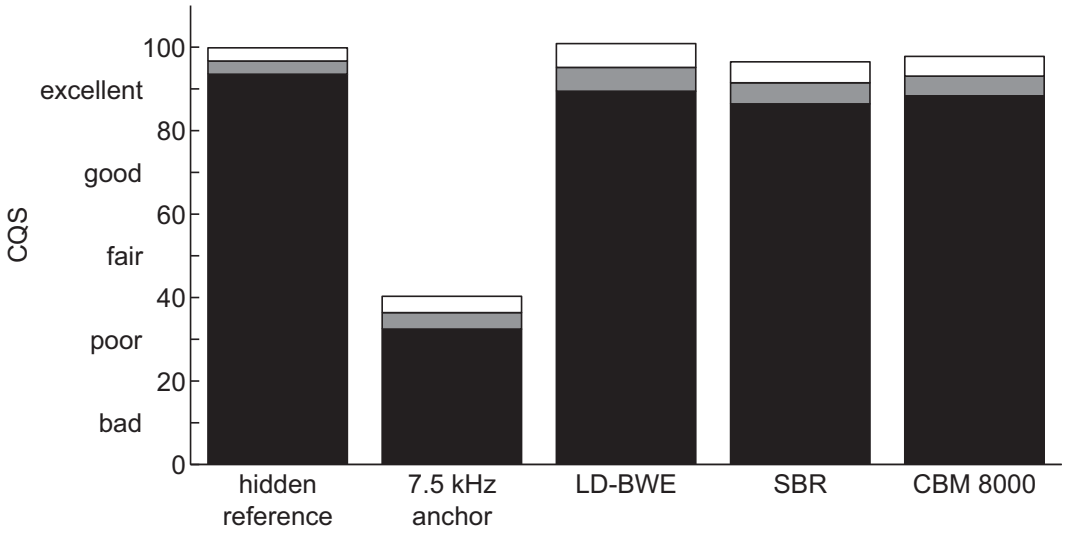

Fig. B.3: *Orchestral piece (sc02_48m)*

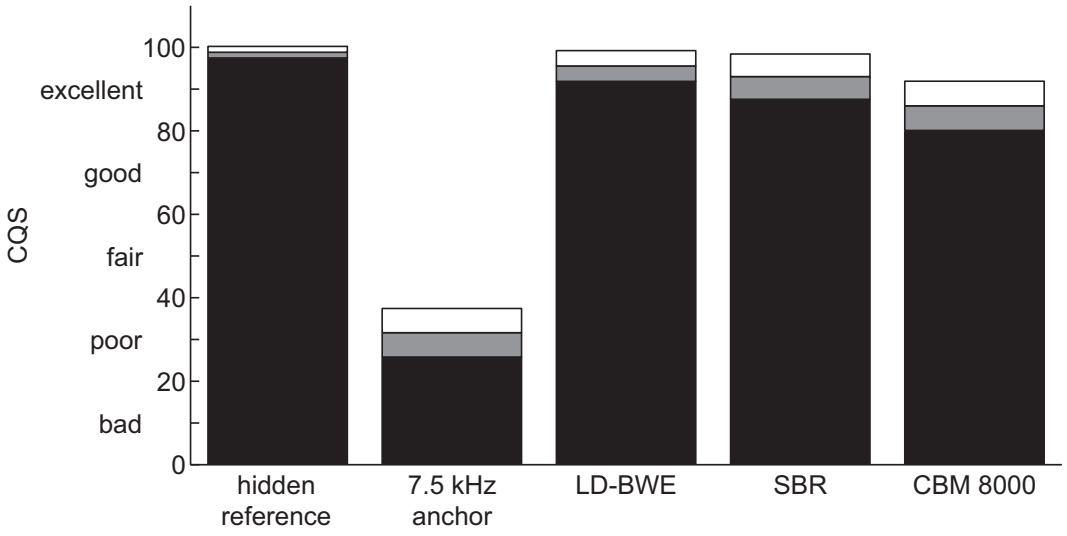

Fig. B.4: *Contemporary pop music (sc03_48m)*

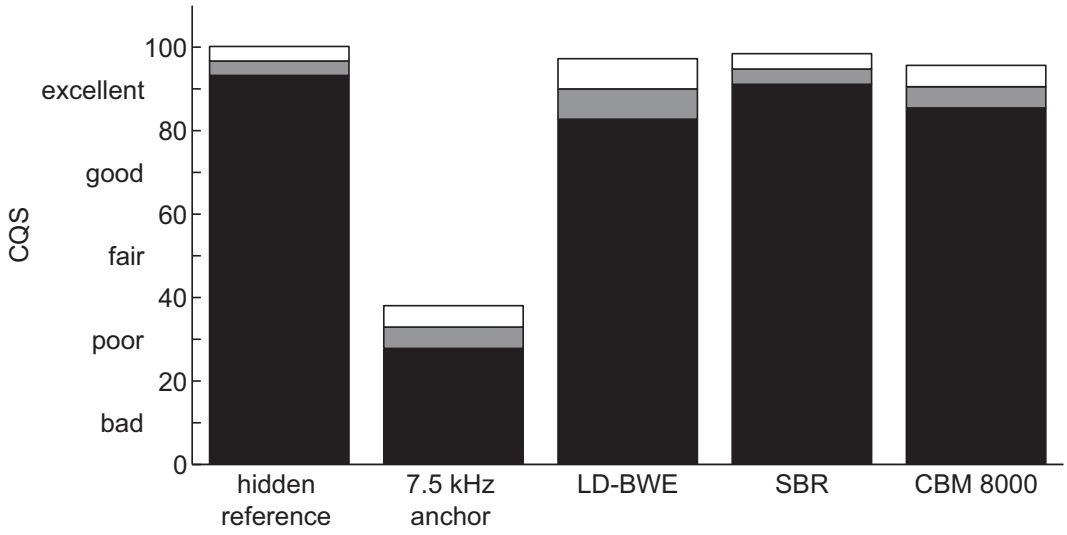

Fig. B.5: *Harpsichord (si01_48m)*

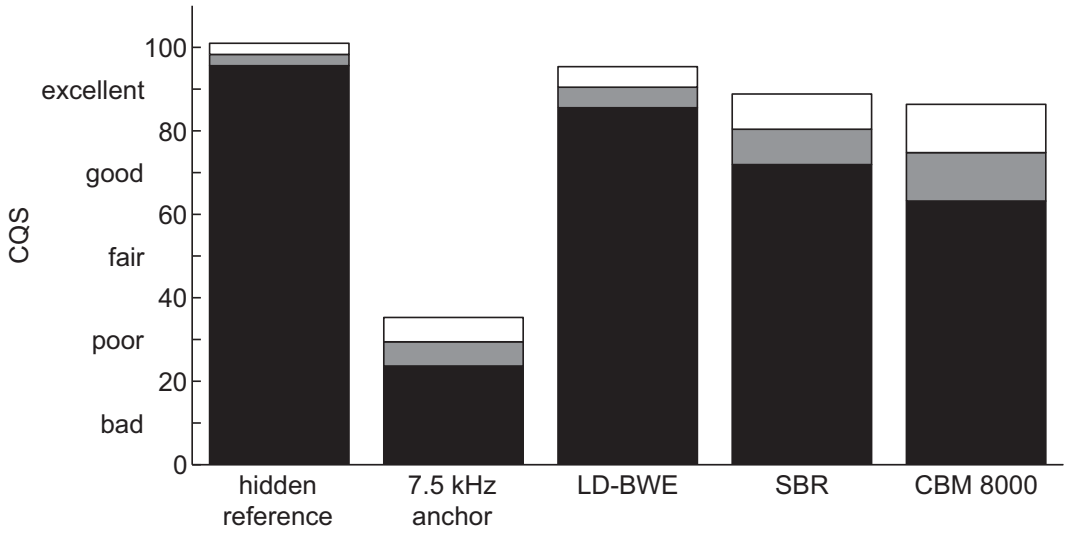

Fig. B.6: *Castanets (si02_48m)*

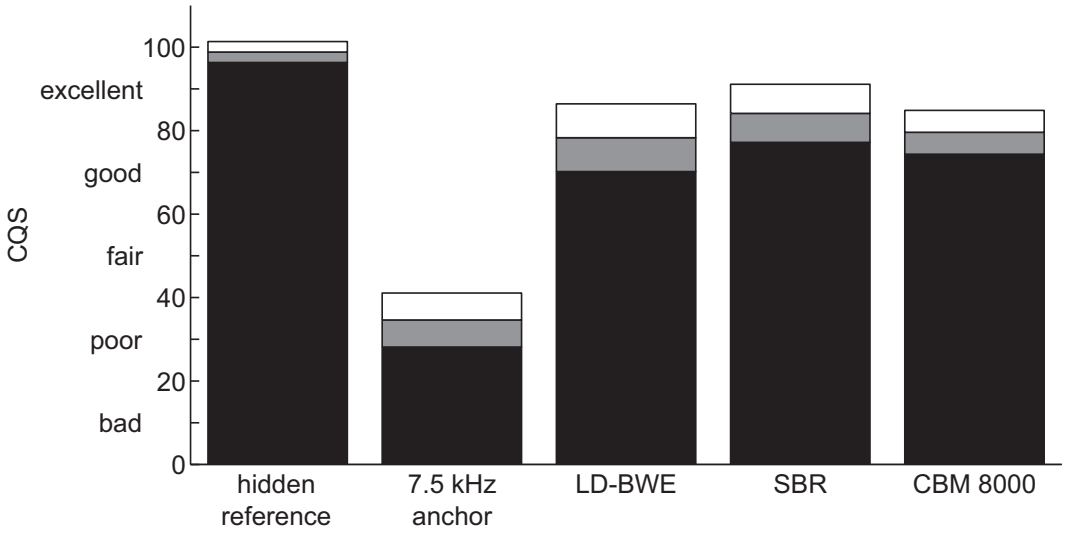

Fig. B.7: *Pitch pipe (si03_48m)*

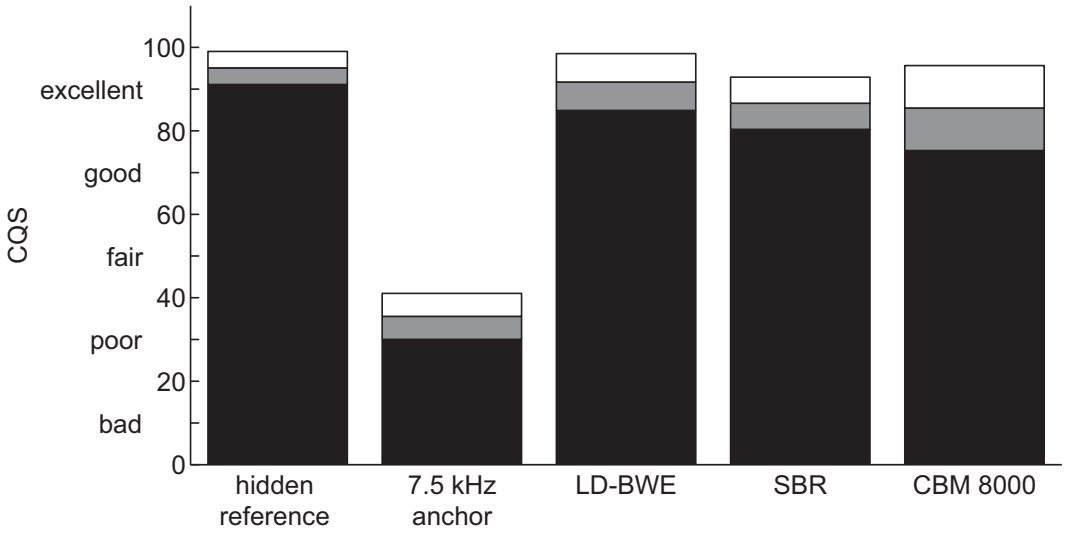

Fig. B.8: *Bagpipes (sm01_48m)*

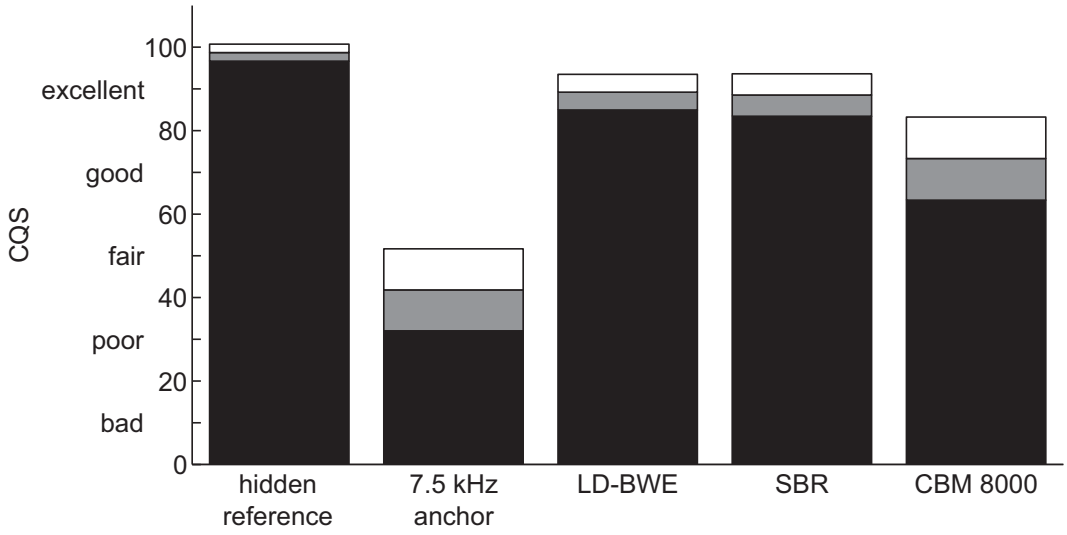

Fig. B.9: *Glockenspiel (sm02_48m)*

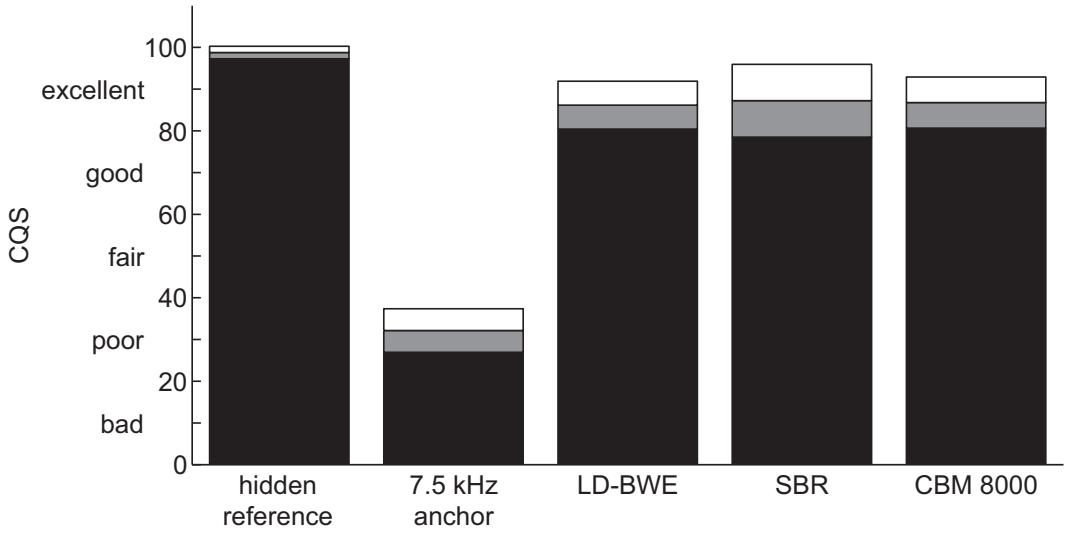

Fig. B.10: *Plucked strings (sm03_48m)*

C. SOURCE MATLAB® CODE

This annex includes an electronic attachment containing the source MATLAB® code for the floating-point implementation of the LD-BWE coder. SBR and LD-SBR modes are also supported.

You will find this appendix in the "Digitale Bibliothek Thüringen (dbt)" under:

http://nbn-resolving.de/urn:nbn:de:gbv:ilm1-2014200087

DECLARATION OF ORIGINALITY

I, Stanislaw Gorlow, declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.

THESES

1. The high-quality performance of the proposed bandwidth extension method is due to auxiliary algorithms, which beneficially contribute to its perceptual ratings.

2. The subband decomposition allows a high degree of flexibility with respect to the choice of the crossover frequency between the low band and the high band.

3. The crossover frequency is subject to the spectral resolution of the filter bank.

4. The algorithmic delay of a modulated filter bank can be explicitly controlled and thus reduced to the framing delay irrespective of the length of the prototype filter.

5. A low-delay filter bank reduces the effect of pre-echo artifacts.

6. The fast FOURIER transform can be used to efficiently implement a modulated filter bank.

7. The frequency resolution of the filter bank in the waveform coder, which is used in combination with the proposed bandwidth extension method, can either be increased by the multi-rate factor or kept constant by means of decimation.

8. The algorithmic delay of the waveform coder, which is used in combination with the proposed bandwidth extension method, increases in proportion to the multi-rate factor.

9. The frequency resolution of a uniform 32-channel filter bank is not sufficient to model the critical bands of hearing.

10. The mean bit rate of the proposed bandwidth extension method is in the region of 10 to 15 kbps, affording a coding gain of approximately 30 %.

11. The combination of ULD with the proposed bandwidth extension method in a dual-rate operation mode does not require more processing power than a stand-alone ULD coder.

12. The MUSHRA method is the most reliable way to measure the audio quality of an HFR system.