Transform-based Coding Methods for Speech and other Audio Signals

Transformationsbasiertes Codierverfahren für Sprache und andere Audiosignale

Dissertation

Der Technischen Fakultät

der Friedrich-Alexander-Universität Erlangen-Nürnberg

zur Erlangung des Doktorgrades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Goran Marković

aus

Belgrad, Serbien

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 27.4.2022

Gutachter: Prof. Dr.-Ing. Bernd Edler
Tom Bäckström, D.Sc. (Tech)

dedicated to

Nevena, Aiko and Lars

in loving memory of my mother

Iva

Abstract

The increasing need for efficient transmission of speech and other audio signals has led to development of specialized codecs. The most successful speech codecs are based on adaptive linear prediction (LP) and time domain coding of the prediction residual, while the most successful general audio codecs are based on the modified discrete cosine transform (MDCT). For optimal coding at low bitrates, recently standardized systems switch between speech and general audio codecs. The switching systems currently achieve the best perceptual quality, but require a lot of resources due to the need of two independent codecs and an additional switching decision. The switching may also introduce significant perceptual quality degradations.

This thesis aims to contribute to the ultimate goal of achieving a truly unified and universal coding scheme that does not require hard switching between paradigms for coding speech and other signal types. First, a short overview of speech and general audio coding is presented together with identification of their similarities and differences. Second, the shortcomings of the switching systems are identified.

The core of this thesis is the description of the newly proposed coding scheme, named Implicit Voice or Anything (IVA). It uses only one paradigm for coding both music and speech, making switching obsolete. IVA extends MDCT-based coding with: time domain (TD) coding of pulse-like portions of input signals, long-term prediction (LTP) and harmonic post-filtering (HPF). Further, parametric coding of spectral portions, named integral Band-wise Parametric Coding (iBPC), is tightly integrated in IVA's MDCT coding. Differences of the proposed pulse coding, LTP, HPF and iBPC to similar existing methods are presented and their advantage is discussed. The proposed MDCT-based coding scheme is the first one that includes all three common speech coding tools (TD pulse coding, LTP, HPF). It provides operation at low bitrate and low latency (< 36 ms), outputting signal at full bandwidth.

Finally, performance of IVA is compared to state-of-the-art codecs at around 24 kbps. For clean speech signals, its performance is close to the best existing LP speech codecs, far exceeding quality of other transform codecs. For music signals, its performance is for most items on par with the best high-latency transform codecs, far exceeding quality of other low-latency codecs. Overall, IVA provides the most consistent quality over different signal types. The experiments show that it is possible to attain music performance of high-latency transform codecs, even at constant low bitrate and low latency. At the same time performance for speech signals is significantly improved.

Extending MDCT-based coding, with tools known from speech coding, is a promising approach. Improvements are evident for both: speech and music signals. A unified and universal coding scheme can deliver perceptual quality on par to switched systems, even at low bitrates and latency.

Zusammenfassung

Die Nachfrage nach effizienter Übertragung von Sprache und anderen Audiosignalen führte zur Entwicklung von spezialisierten Codierverfahren. Die erfolgreichsten Codierverfahren für Sprache basieren auf linearer Prädiktion (LP) und Zeitbereichscodierung des Schätzfehlers. Die erfolgreichsten Codierverfahren für allgemeine Audiosignale basieren auf der modifizierten diskreten Kosinustransformation (MDCT). Für eine optimale Codierung bei niedrigen Bitraten schalten neuere standardisierte Systeme zwischen Sprach- und MDCT-basierten Verfahren um. Derzeit erreichen Umschaltungssysteme die beste Wahrnehmungsqualität, erfordern jedoch viele Ressourcen, da zwei unabhängige Verfahren und eine zusätzliche Umschaltentscheidung erforderlich sind. Das Umschalten kann auch eine wesentliche Qualitätsverschlechterung verursachen.

Diese Dissertation soll dazu beitragen ein tatsächlich einheitliches und universelles Codierverfahren zu erreichen, welches kein Umschalten zwischen verschiedenen Codierungsparadigmen benötigt. Zuerst wird ein kurzer Überblick über bestehende Verfahren für Sprach- und allgemeine Audiosignale sowie ihrer Ähnlichkeiten und Unterschiede vorgelegt. Danach werden die Nachteile der Umschaltungssystemen identifiziert.

Das Kernstück dieser Dissertation ist die Beschreibung des „Implicit Voice or Anything“ (IVA) Codierungsschemas. Es verwendet nur ein Paradigma um sowohl Musik als auch Sprache zu codieren, wodurch das Umschalten überflüssig wird. IVA erweitert MDCT-basierende Codierung mit einer Zeitbereichscodierung von impulsartigen Anteilen von Eingangssignalen, Langzeitvorhersage (long-term prediction, LTP) und harmonischer Nachfilterung (harmonic post-filtering, HPF). Parametrische Codierung von Spektral-komponenten „integral Band-wise Parametric Coding“ (iBPC) ist fest in IVAs MDCT-Codierung integriert. Unterschiede zwischen der vorgeschlagenen Pulscodierung, LTP, HPF und iBPC zu ähnlichen bestehenden Methoden werden vorgestellt und deren Vorteil argumentiert. Das vorgeschlagene MDCT-basierende Codierschema ist das erste, das alle drei gängigen Sprachcodierungswerkzeuge (Pulscodierung, LTP, HPF) enthält. Es ermöglicht den Einsatz bei niedriger Bitrate, niedriger Latenz (< 36 ms) und erzeugt das Signal bei voller Bandbreite.

Schließlich wird die Qualität von IVA mit modernsten Codierverfahren bei etwa 24 kbps verglichen. Bei reinen Sprachsignalen liegt seine Wahrnehmungsqualität nahe an den besten existierenden LP-Sprach-Codierverfahren und übersteigt die Qualität anderer Transformations-Codierverfahren. Bei Musik ist seine Leistung für die meisten Signale auf dem Niveau der besten Transformations-Codierverfahren mit hoher Latenz und übertrifft die Qualität anderer Codierverfahren mit niedriger Latenz bei weitem. Insgesamt bietet IVA die konsistenteste Qualität über verschiedene Signaltypen. Untersuchungen zeigen, dass es möglich ist, die Musikleistung von Transformations-Codierverfahren mit hoher Latenz auch bei niedriger und konstanter Bitrate und niedriger Latenz aufrechtzuerhalten. Gleichzeitig wird die Wahrnehmungsqualität für Sprachsignale deutlich verbessert.

Die Erweiterung der MDCT-basierten Codierung mit Sprach-Codierungsmethoden ist ein vielversprechender Ansatz. Verbesserungen sind bei Sprach- und Musiksignalen erkennbar.

Ein einheitliches und universelles Codierschema kann eine Wahrnehmungsqualität auf dem Niveau von Umschaltungasystemen selbst bei niedrigen Bitraten und Latenz erreichen.

Acknowledgments

This doctoral dissertation is a result of my research at the International Audio Laboratories Erlangen, a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg and the Fraunhofer Institute for Integrated Circuits IIS.

First of all, I want to profoundly thank Prof. Dr.-Ing. Bernd Edler for mentoring me, for fruitful and open-minded discussions and most of all for his unlimited encouragement.

Further, I am very grateful to Prof. Dsc. Tom Bäckström for accepting to be my second reviewer and for his helpful comments.

I am thankful to Prof. Dr.-Ing. Bernhard Grill and Dr. Frederik Nagel, as Fraunhofer IIS representatives, for granting financial funding and Markus Multrus, Nikolaus Rettelbach, Johannes Hilpert, Martin Dietz for their assistance in the decision.

I want to thank Dr. Guillaume Fuchs, Jan Frederik Kiene, Srikanth Korse and Dr.-Ing. Christian R. Helmrich for the thorough review of this thesis.

I am also in debt to Prof. Dr.-Ing. Jürgen Herre and Dr. Guillaume Fuchs for the discussions during my PhD research.

Many thanks to Elke Weiland, Tracy Harris, Day-See Riechmann and Stefan Turowski for helping me through the administrative issues.

Furthermore, I appreciate all colleagues at AudioLabs and Fraunhofer IIS for the inspiring conversations and nice atmosphere, making it the best place to work. My special gratitude goes to Adrian Tomasek, Andreas Niedermeier, Benjamin Schubert, Christian Helmrich, Christian Neukam, Christopher Oates, Conrad Benndorf, Elena Burdiel Pérez, Eleni Fotopoulou, Emmanuel Ravelli, Fabian Bauer, Florian Schuh, Florin Ghido, Franz Reutelhuber, Guillaume Fuchs, Jan Büthe, Jérémie Lecomte, Jouni Paulus, Jürgen Herre, Klaus Berghammer, Kristina Hensel, Manfred Lutzky, Markus Multrus, Markus Schnell, Martin Dietz, Max Neuendorf, Michael Sturm, Michael Schnabel, Nikolaus Rettelbach, Pavan Kantharaju, Peter Rettinger, Ralf Geiger, Ralph Sperschneider, Sascha Disch, Srikanth Korse, Stefan Bayer, Stefan Döhla, Tom Bäckström and Wolfgang Jaegers at Fraunhofer IIS and Adrian Herzog, Alexander Adami, Alexandra Craciun, Carlotta Anemüller, Christian Dittmar, Christof Weiß, Daniele Mirabilii, Prof. Dr. ir. Emanuël Habets, Esther Feichtner, Fabian-Robert Stöter, Prof. Dr. Frank Wefers, Frank Zalkow, Johannes Fischer, Konstantin Schmidt, Maja Taseska, Martin Strauß, María Luis Valero, Prof. Dr. Meinard Müller, Michael Krause, Niklas Winter, Nils Werner, Ning Guo, Oliver Thiergart, Pablo Delgado, Patricio López-Serrano, Richard Füg, Sascha Dick, Sebastian Rosenzweig, Sebastian Schlecht, Soumitro Chakrabarty, Stefan Balke, Thomas Prätzlich, Thorsten Kastner, Prof. Dsc. Tom Bäckström, Vlora Arifi-Müller, Wolfgang Mack, Yigitcan Özer and Youssef El Baba at AudioLabs. Many thanks also to all the participants in the listening tests, some of who are already listed above.

I owe my gratitude also to the students that I have supervised during their internship or master thesis writing, namely Amulya Veerappa, Arjola Hysneli, Arnee Niitsoo, Jan Frederik Kiene, Pablo Pérez Zarazaga, Rohan Shet, Sebastian Bilz, Sharvin Vittappan and Suraj Khan.

I use this opportunity to thank the inventors behind mp3, who have influenced me in the decision to pursue scientific research in the field of audio coding. Among many who have contributed to mp3, let me name Karlheinz Brandenburg, Ernst Eberlein, Heinz Gerhäuser, Bernhard Grill, Jürgen Herre, Harald Pop and Bernd Edler.

I am eternally grateful to my parents, Iva and Božidar, for their support throughout my life. Big thanks also to my brother Milan for all of his help.

Last but not least, I cannot thank enough my wife Nevena for being there for me all the time.

Contents

1 Introduction

Communication is indispensable for humanity. Since the end of the 19th century, advances in telecommunication have allowed us to cross borders not imaginable before [1]. Today, telecommunication technologies help us to exchange ideas, talk to our loved ones, avoid conflicts and organize businesses across the globe. Since the world resources are limited, there is a need to optimize means for the communication. In the light of this limitation, advantages of digital communication became apparent at the end of the last century, such as: increased reliability, reduced cost, easier implementation, easy encryption and multiplexing [2–4].

Digital representation of an analog signal cannot be perfect and introduces a distortion of the original. However, from limitations of human hearing it is possible to define a sufficient representation [5, 6] of analog audio signals. Digital representation allows us to precisely define the amount of data that needs to be transmitted, quantifying the amount of data as the bitrate. Additional aspects of a digital system, that have to be taken into account, are latency [7] and computational complexity. Sufficient representation can be obtained for a single channel monaural audio signal with minimal latency and complexity at the bitrate of 768 kbps using the simple pulse code modulation (PCM) with linear quantization [2, 8]. One also has to take into account that required resources, e.g. electricity, are proportional to the required bitrate [9, 10] and that in many places of Earth achieving high bitrates is not possible because of a poor infrastructure. Thus reducing bitrate is beneficial for saving the environment [11, 12] and allowing development in the whole world [13, 14]. For reducing the bitrate many speech and audio codecs have been developed [15, 16] with conflicting goals of minimal bitrate, complexity, latency and introduced distortion.

Humans primarily use speech for communication and codec development was initially concentrated on speech signals [17, 18], giving rise to highly specialized solutions [16, 19]. Other acoustic signals, including music, are also very important for conveying messages. Because of diverse content, it is harder to achieve small bitrate when coding music. The severity of the problem is probably an additional reason, besides the importance of speech, for music coding being widely approached much later [15]. The specialized speech codecs have poor performance on music [20–22] and general audio codecs took different path [15] eventually resulting in the well-known Moving Picture Experts Group (MPEG) standard

MPEG-1 Layer III (MP3) [23]. Despite further evolution of general audio codecs [15], the quality of coded speech at bitrates below 48 kbps was inferior to specialized speech codecs [24].

At the turn of the century, efforts have emerged to design codecs that are well suited for both speech and music [21, 25–30]. Finally a codec capable of achieving high quality at low bitrates for both speech and general audio was developed within the MPEG-D Unified Speech and Audio Coding (USAC) standard [22, 31–33]. USAC, also marketed as Extended High-Efficiency Advanced Audio Coding (xHE-AAC), switches between an algebraic code-excited linear prediction (ACELP) speech codec and a modified discrete cosine transform (MDCT) based general audio codec with special attention to smooth transition between the codecs. Nevertheless, the long delay of xHE-AAC [34] doesn't make it suitable for real-time communication. Consequently, the Enhanced Voice Services (EVS) codec was developed and standardized for usage in mobile communications [35–37]. In the same period, the royalty-free and low latency codec Opus was developed [38], employing switching between speech codec SILK [39] and MDCT-based Constrained Energy Lapped Transform (CELT) [40], using mono at bitrates lower than 32 kbps [41]. EVS and Opus have been compared using listening tests in [42] and [43].

1.1 Comparison of speech and general audio codecs

The most common architecture for speech coding is based on adaptive linear prediction (LP) [4, 16], including the most advanced recent codecs [37, 38]. The method is usually referred to as linear predictive coding, which can be connected to models of speech production [4].

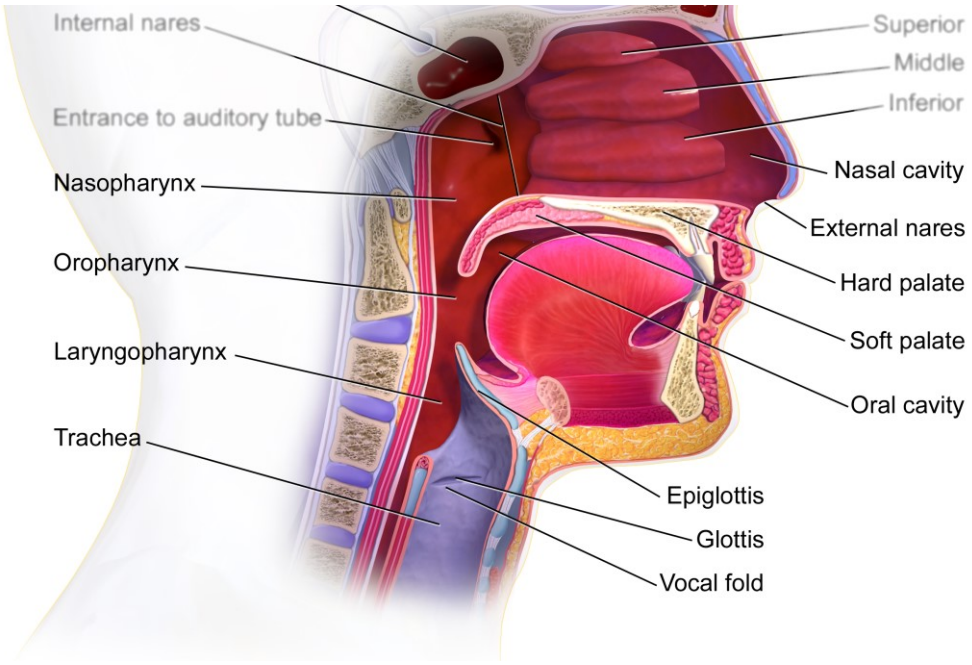

Figure 1.1 Structures of mouth and pharynx from [44], CC BY 3.0

Speech is produced by expelling air from the lungs, that then encounters obstructions at different places (e.g. palate, tongue, teeth, lips) in the vocal tract [Figure 1.1]. The vibration of

vocal folds creates quasi-periodic obstructions of the air flow at glottis, giving rise to voiced phones. For unvoiced phones, the air flows freely from the lungs to the mouth. The cavities of the vocal tract (pharynx, oral and nasal cavity) introduce frequency selective resonances.

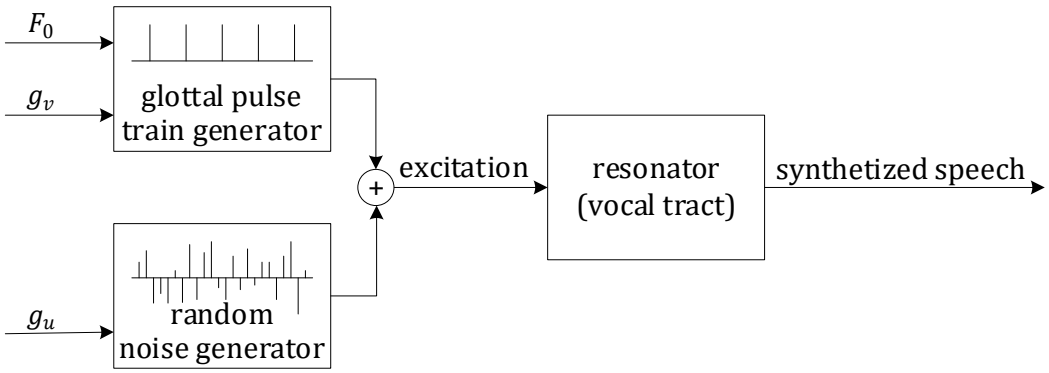

Figure 1.2 Simplified source-filter model

In a simplified source-filter model of speech synthesis [Figure 1.2], an excitation signal is passed through a frequency selective resonator representing the vocal tract. The excitation can be voiced or unvoiced or a combination of both. The voiced excitation consists of a glottal pulse train with the fundamental frequency F_0 that models the vocal fold vibrations. The unvoiced excitation is modeled by a random noise. The gains g_v and g_u model the voicing and the overall intensity.

The adaptive prediction consists of long and short term prediction. The short term predictor

$$y[n] = x[n] + \sum_{i=1}^{k} a_i y[n-i]$$

is often referred to as the linear predictor (LP) and a_i as linear prediction coefficients (LPC). The short term predictor models the resonances in the vocal tract and $x[n]$ represents the excitation [4]. Because $x[n]$ may be obtained from $y[n]$ using the inverse of the LP, it is also referred to as the LP residual. The long-term predictor (LTP)

$$x[n] = e[n] + \sum_{i=-l}^{l} b_i x[n-d+i]$$

is used for modeling repetitions of the glottal pulse train, where d is a pitch lag corresponding to the distance between two consecutive pulses expressed in number of samples. For non-integer pitch lags, interpolation of the neighboring samples is often used (e.g. weighting $2l+1$ neighboring samples with the coefficients b_i as in the LTP above).

An LP speech codec estimates the model parameters from its input – the original signal. The LTP parameters are usually updated every 4-5 milliseconds and the LP coefficients every 20 milliseconds. Codecs at bitrates above 5 kbps usually also code the prediction residual $e[n]$. The coded residual is often referred to as the innovation. It is determined, so that after processing it with the LTP and the LP, the synthetized speech $y[n]$ is as close to the original signal as possible. This process is called analysis by synthesis [45, 46]. The innovation $e[n]$ is ideally uncorrelated and thus it should be possible to efficiently code it with a scalar

quantizer. Nonetheless, because of the non-stationary nature of speech, the predictors cannot model all correlations in the natural speech. One limiting factor is that the static segmentation and windowing of the input signal are not optimally fitted for the prediction model. The correlations within the innovation lead to a need of vector quantization for increasing the performance [45, 46]. The innovation embodies the unvoiced excitation, onsets and jitters in the periodicity of the glottal pulses. The excitation $x[n]$ is generated in non-overlapping blocks. Smoothing across blocks is achieved by keeping LP filter memories and using zero input response (ZIR) of the LP filter [47].

LP basically modifies the excitation and so acts as a digital filter. The characteristic of this filter is that, when its input is a signal with flat spectrum, its output's spectral envelope is close to the spectral envelope of the original signal. The coefficients a_i of LP are found so that the associated all-pole model power spectrum approximates the original signal's power spectrum [48]. LTP acts as a comb filter [49].

Initially, the development of adaptive prediction speech codecs was concentrated on low complexity and minimization of the signal-to-noise ratio (SNR) [50], but very soon, the importance of human hearing was noticed [51]. The problem definition then became: find an optimum representation of the excitation $e[n]$, so that after filtering it with LTP and LP, the perceptually filtered error to the original signal is minimized [45, 46]. The perceptual filter shapes the error spectrum, allowing minimization according to importance for human hearing. Finding an effective representation of the excitation [52] via the concept of code-excited linear prediction (CELP) was the next significant step. A reduction in computational complexity via algebraic codebooks [53] followed. ACELP became the most successful coding scheme for speech at bitrates between 6 and 32 kbps [19, 20, 37, 42, 43]. It is further known from psychoacoustics [54, 55] that details at high frequencies are of much less importance to human hearing, yet higher bandwidth significantly improves perceived quality [42, 56].

In contrast to the initial objective of just high intelligibility for speech coding, quality acceptance criterion for music reproduction is much higher and therefore the digital representation of music in the form of the compact disc (CD) occurred only in 1982 [57]. The Compact Disc Digital Audio (CDDA) format uses simple linear PCM, sampled with 16 bits at 44.1 kHz and hence can present all signals bandlimited to 22 kHz, being principally enough for all sounds perceivable by humans [5, 6, 54, 55]. CDDA requires 705.6 kbps per channel, which was too high for a transmission over the existing networks at the time and the work on a compressed presentation of the general audio began soon after the occurrence of the CD [15]. The initial expectation of a general audio compression scheme was that it produces output close to transparency for an expert listener [58]. It is not possible to represent music using just one simple model, e.g. source-filter model for speech. Thus, general audio coding took a different path from speech coding and the driving factor became to represent all sounds that humans can perceive.

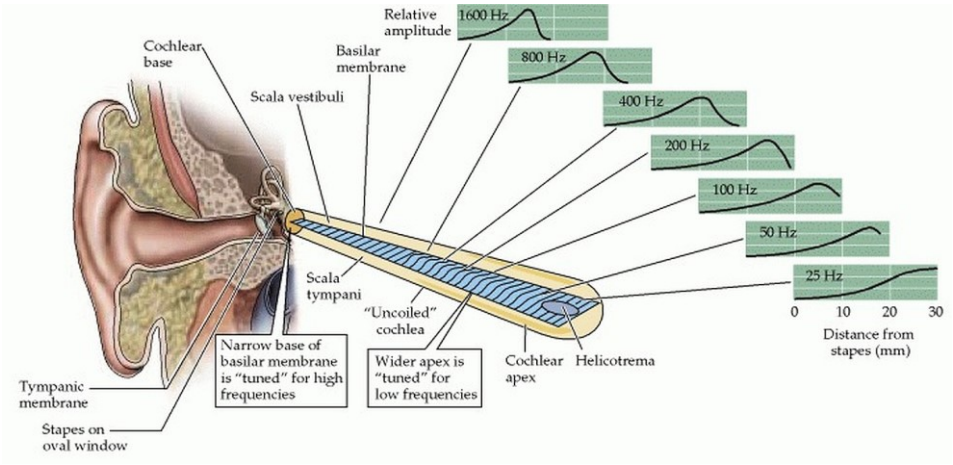

Figure 1.3 An illustration of an uncoiled cochlea © 2001, Sinauer Associates, Inc. [59]

The cochlea, located in the inner ear, acts as a time-varying frequency analyzer [60]. A sound that enters the ear produces frequency dependent displacements of the basilar membrane, as illustrated in Figure 1.3. The displacements are converted into nerve impulses that we perceive as sound. A conversion of time domain (TD) signals into a frequency domain (FD) is also an integral part of every successful general audio codec.

A transform converts a block of TD signal samples into FD values known as spectral coefficients or bins. Usage of TD to FD transforms was already investigated for speech coding [16], one example being transform-coded excitation (TCX) [27]. TD to FD transforms are efficient decorrelators [2] with the discrete cosine transform (DCT) providing close to optimum coding gain over PCM for scalar independent quantization of correlated signals with low pass characteristics, where coding gain refers to increase in the SNR of the scalar quantizer [61].

The most successful audio codecs use the MDCT. The advantage of the MDCT over other transforms is that it is critically sampled even for overlapping blocks and allows smoothing over successive blocks via windowing and overlap-and-add [62]. Many fast algorithms for the MDCT were developed; most notably it can be implemented using folding of $2N$ TD samples into N samples followed by the DCT-IV and the inverse MDCT as the DCT-IV followed by unfolding of the N samples into $2N$ aliased samples [63]. The aliasing, produced by the folding/unfolding, is handled with time domain aliasing cancelation (TDAC), first introduced in [64]. It was shown in [65–67] that the coding gain of the MDCT of block length $2N$ is even higher than for the optimum Karhunen–Loève Transform (KLT) of length N and close to the KLT of length $2N$.

Transform codecs achieve further coding gain with time and frequency varying quantizers, where the quantizer step size is derived from a separately coded spectral envelope [61]. This is comparable to the adaptive short term LP in speech codecs [2, 68]. The frequency dependent quantizer step size in EVS is for example achieved by dividing MDCT coefficients with the magnitude response of the coded LP coefficients [37]. In EVS the LPCs are just a way of coding a smooth spectral envelope. In other codecs, the spectral envelope is usually represented in terms of scale factors, which are coded directly in a FD [15].

Through the transformation and the spectral envelope dependent quantizer, general audio codecs achieve that memoryless statistical model can be efficiently used for all spectral coefficients. The statistical redundancy of the quantized spectral coefficients is then exploited through a lossless entropy coder, e.g. Huffman or arithmetic coder [69].

In transform coding of LP residual, adaptation of an FD quantizer is replaced by shaping of the quantization noise with the LP filtering. This approach was investigated in [70–73]. For these codecs, the LP filtering first reduces short term correlations and thus flattens the spectra. A TD to FD transform then further decorrelates the samples and allows scalar or short length vector quantization.

LTP did not originally play major role in the transform coding as long transform blocks are efficient for long term decorrelation. It is easy to implement LTP in the scheme of non-overlapping transform coding of the LP residual and examples can be found in [27, 70–72]. LTP is important in low delay transform coding and combining it with the overlapping MDCT will be discussed later.

Exploiting perceptual irrelevancy, coming from limitations of human hearing, is as important as statistical redundancy and was investigated early in the general audio transform coding [15, 62, 74]. As illustrated in Figure 1.3 and Figure 1.4, the frequency resolution of cochlea is non-linear and nearly inversely proportional to the frequency. A pure sinusoidal signal displaces basilar membrane maximally at a specific position, but also around it. This leads to frequency dependent hearing phenomena (e.g. masking [54]). These physiological findings were confirmed with psychoacoustic experiments [54, 55]. Additionally, it was also shown that loudness perception is non-linear. The conversion into FD allows effective modelling of these observations.

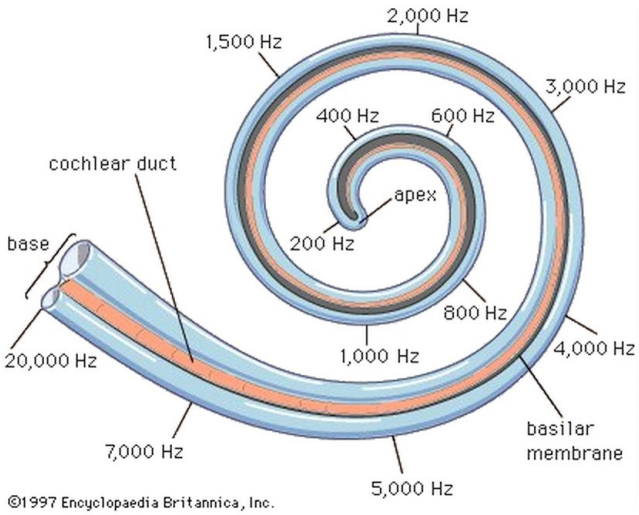

Figure 1.4 Model showing the distribution of frequencies along the basilar membrane of the cochlea. By courtesy of Encyclopædia Britannica, Inc., copyright 2009; used with permission [75]

For taking into account perceptual irrelevancies, the frequency dependent quantizer step sizes are derived from psychoacoustic models that stem from the spectral envelope. In this manner, the transform codecs jointly handle LP modelling and perceptual weighting of CELP

speech codecs. The psychoacoustic models can be elaborate as in [76, 77] or implicit and simple as in [78–80]. The modeling of perceptual irrelevancies can also be incorporated in LP models [81, 82].

As already written, transform codecs process signals in blocks. The quantization error is equally spread across the whole transform block. Simultaneous masking explains how quantization noise, present at the same time as the quantized signal, is masked by the signal, being also called the masker [54, 55]. Non-simultaneous masking is the effect that a masker signal hinders perception of a test signal, present at a different time instance than the masker [54, 55]. If the masker is present before the test signal, it is called forward masking. If the masker is present after the test signal, it is called backward masking. Backward masking is significantly weaker than forward masking. As with simultaneous masking, the just-noticeable difference (JND) of the test signal depends on the masker level. For temporal masking the JND of the test signal is also dependent on the temporal distance between the masker and the test signal. Forward masking also depends on the masker duration. Temporal masking often cannot hide the quantization noise for transients of short duration (e.g. castanet clap). A single transient event at the end of a block is especially critical, because the quantization error, spread over the whole transform block, cannot be masked by a low level signal before the transient.

Different methods have been developed in transform codecs to temporally localize the quantization noise at the location of the signal and accordingly utilize simultaneous masking. The most successful so far are window and block switching [83] and temporal noise shaping (TNS) [84]. TNS is achieved through linear prediction in the MDCT spectrum and one could argue that, besides the temporal shaping of the noise, TNS also achieves a coding gain dual to the coding gain of LP in the time domain. Another approach, specific for localizing the quantization noise in transform coding of LP residual, is to have a gain and an LP adaptation more frequent than the transform frame rate [81, 85].

Finally, we summarize similarities and main differences between CELP and MDCT coding in Table 1.1.

		CELP speech coding	MDCT audio coding
Similarities		Process input in blocks Provide methods for smoothing at block boundaries Employ a model for spectral envelope and achieve coding gain through it Utilize perceptual irrelevancies in the quantization process	
Differences	Quantization of the spectrally flattened signal	Analysis by synthesis in TD	Scalar or short-word quantizers in FD
	Modeling of temporal envelope	Short blocks	Short blocks and LP filtering in FD

Table 1.1 Similarities and main differences between CELP speech and MDCT audio coding

There are many other differences in the current CELP speech and MDCT audio codecs, but one could argue that those are tunings tailored to the target usage or that they are derived from the listed main differences.

State-of-the-art speech codecs offer better perceptual quality of coded speech at low bitrates at a reasonable complexity. However, the complexity increases significantly with the bitrate and the quality saturates, not reaching transparency. MDCT audio codecs reach excellent quality with low complexity at 48 kbps per channel for all signals [24, 42], including speech, but they are far from satisfactory for specific signals and speech at 24 kbps and lower bitrates.

Codebooks in ACELP speech codecs are tuned specifically for speech without background noises. They are not able to represent prediction residual of a polyphonic music signal. Providing codebooks to cover all types of input signals would be tedious, require lots of memory and complexity. Conversion into the MDCT domain provides decorrelation even for polyphonic music signals and therefore allows usage of scalar quantization.

At low bitrates, both approaches have enough bits only for coding part of the whole perceivable bandwidth and employ some kind of bandwidth extension. Speech codecs must also limit coded bandwidth because of the computational complexity. Hence, speech codecs utilize bandwidth extensions decoupled from the core coding, e.g. [39, 86]. Specialized speech bandwidth extension [86] can achieve high quality for speech signals, but the non-linear function used in [86] doesn't allow applying it to polyphonic signals. MDCT codecs can implement parametric coding directly in the MDCT domain and restrict it to a part of the whole spectrum, thus it can also act as a bandwidth extension. Examples are simple noise filling [87, 88], Perceptual Noise Substitution (PNS) [89] and more sophisticated Intelligent Gap Filling (IGF) [90].

1.2 Structure of the thesis

Chapter 2 provides the motivation for the research and the thesis objectives.

Chapter 3 presents the investigation of MDCT coding of LP residual and reasoning for choosing FD scale factors instead.

Chapter 4 introduces the final Implicit Voice or Anything (IVA) codec structure with a list of transmitted parameters. Additionally, it details the derivation of the pitch contour and the high frequency tonality flag. Differences in the spectral noise shaping (SNS) approach, to the previously published methods, are also described.

The following four chapters describe the newly proposed technologies employed in IVA. Chapter 5 presents the pulse extraction and coding, relating it to the existing methods for handling transients in FD audio codecs. Chapter 6 explains a new type of LTP in a combination with MDCT coding. Chapter 7 proposes integral Band-wise Parametric Coder (iBPC). On the encoder side it unifies parametric and waveform coding of the MDCT spectrum. On the decoder side it uniformly combines previous techniques known as noise filling and bandwidth extension. The relation of iBPC with existing methods and its advantage is demonstrated. Chapter 8 describes the new harmonic post-filter (HPF), which reduces noise between

harmonics by comb-filtering the decoded signal across frame borders, and explains differences to similar methods.

Objective and subjective evaluations of the IVA codec are presented in chapter 9, including listening test results that compare it to the state-of-the-art audio and speech codecs.

Chapter 10 concludes the thesis, outlines original contributions and proposes potential future research topics.

Appendix A.1 gives reasoning for choosing the MDCT-based TCX from EVS as the basis. The choice of a window function and a psychoacoustic model is further argued.

Appendix A.2 explains process of obtaining the spectrograms used in the thesis.

Appendix A.3 lists items used in the listening tests.

2 Motivation and objectives

2.1 Motivation

Several codecs have been standardized in recent years aiming to code both speech and music, even at low bit bitrates:

- AMR-WB+ [21, 88]
- xHE-AAC, also known as MPEG-D USAC [22, 31–34]
- EVS [35–37, 91]
- Opus [38]
- MPEG-H 3D audio [92, 93]
- AC-4 [94–96]

All of these codecs use two modes at low bitrates: one optimized for speech and one optimized for general audio. The general audio mode in all of them is based on the MDCT, except in Extended Adaptive Multi-Rate Wide Band (AMR-WB+), where it is based on the discrete Fourier transform (DFT). The speech mode in AMR-WB+, xHE-AAC, EVS, and MPEG-H 3D audio uses ACELP. Opus uses a specific approach for the quantization and pulse coding in its speech mode, but also based on LP coding. AC-4 is the only, among the listed codecs, that uses the MDCT in its mode optimized for speech, called Speech Spectral Front-end. LP is used in AMR-WB+ for modeling spectral envelope and noise shaping, also in its FD mode. In all other codecs, quantization noise in the FD modes is shaped by directly scaling the spectrum coefficients.

Systems which switch between specialized codecs, depending on classification of the input signal, require separate maintenance of each codec and have increased memory requirements. They may also need some type of signal classifier with its own development, additional to the codec development. Avoiding such a classifier and switching between specialized codecs would allow saving resources: memory, computational and human. Researchers and developers could put focus on the unified approach, bringing improvements to the whole system with a single modification. With specialized codecs, each improvement of one

specialized codec requires rechecking the other codecs in the system as well and also an eventual adaptation of the classifier.

Successive coding blocks may have different quantization noise spectral characteristics and require smoothing across block borders. Smoothing across blocks in LP coding is achieved by maintaining LP filter memories and using zero input response (ZIR) of the LP filter [47]. FD coding uses overlapping windows for smoothing. Difference in the quantization noise is significantly bigger when successive blocks are coded in different modes, thus requiring special attention. Switching between LP and FD coding poses a challenge because of the different approaches for smoothing at block borders. Many schemes were developed to tackle the problem [38, 97, 98]. Switching decreases coding efficiency during the transitions, increases complexity and is prone to quality problems. It also requires processing separated from the core coders for having stereo and bandwidth extension tools common to specialized modes. Theoretically both coding methods should work well enough for spectral shaping of quantization noise.

Having two modes requires a decision on which mode to choose. The decision can be implemented in a closed- or an open-loop approach. In the closed-loop decision, a signal block is coded with each of the modes and it is decided, based on a measurement, which coding mode performs better. AMR-WB+, xHE-AAC and EVS use segmental SNR for the closed-loop decision [88, 99, 100]. Running both modes in the closed loop significantly increases total computational complexity. The open-loop decisions in AMR-WB+, xHE-AAC, EVS and Opus are based on a classification of the input signal [88, 100–103]. The classification intends to decrease complexity, but it also comes at the expense of reduced overall quality [104]. A compromise can be achieved by estimating each mode's distortion [99]. However, with the approach from [99] only 24%-31% reduction in the encoder's computational complexity is achieved compared to the closed loop.

The mode decision needs to be accurate, reactive and stable [101]. Its complexity should also be low and ideally it shouldn't introduce any additional delay. These requirements are contradictory and require tradeoffs. Careful tuning including hysteresis [99] reduces occurrences of problems in a switched codec, but when switching occurs in borderline cases of a mostly stable signal, quality degradations can be significant. One example of a switching problem in EVS is presented in Figure 2.2 with the original in Figure 2.1. In the example, ACELP is used for the ranges 1.9s-2.5s, 4.3s-4.8s and 4.9s-5.3s. MDCT-based TCX is used for the rest of the presented excerpt. Different characteristics of the quantization error in the specialized codecs are obvious. These changes of spectral characteristics in the coded signal are very unpleasant, introducing a lower overall quality compared to coding the whole excerpt with only one of the two codecs.

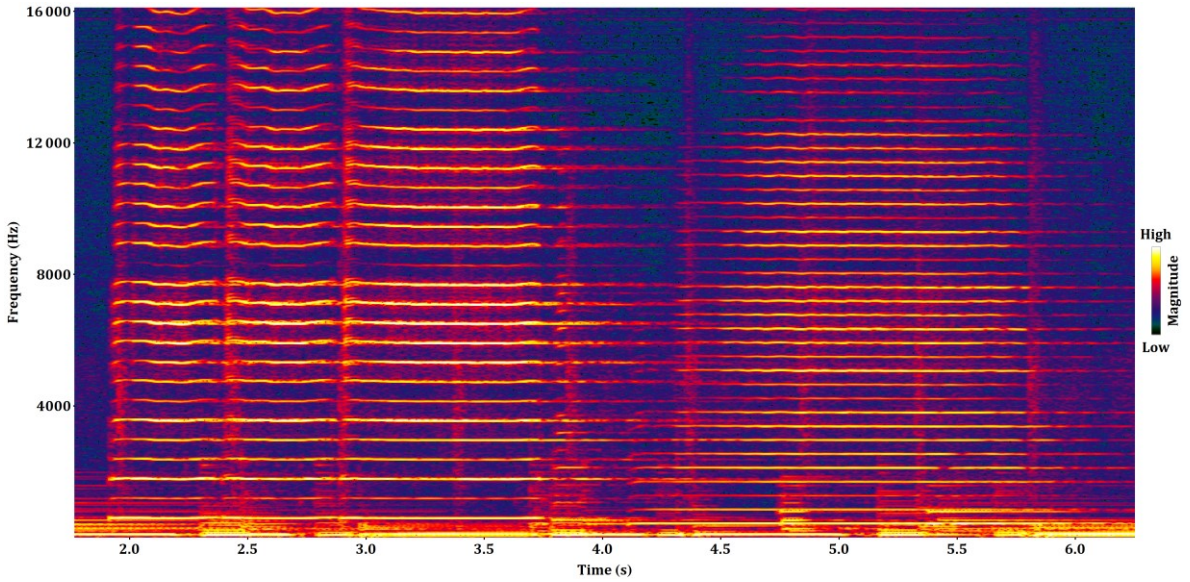

Figure 2.1 Spectrogram [A.2] of an excerpt from a trumpet solo [hanco], the original signal

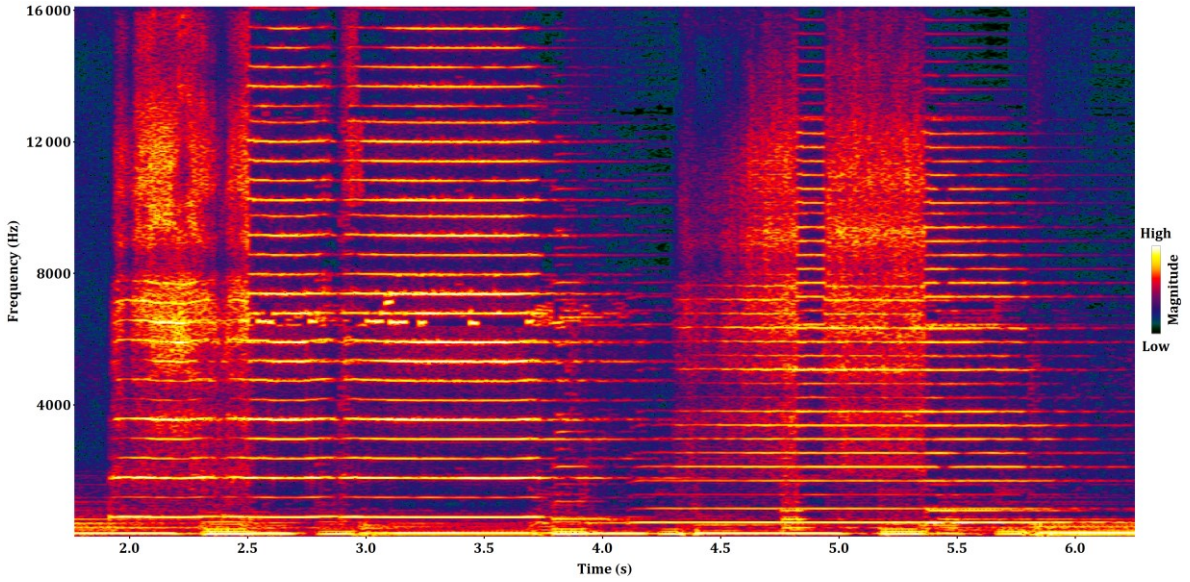

Figure 2.2 Spectrogram of an excerpt from a trumpet solo coded by EVS

With all problems explained above, it becomes clear that switching of specialized modes, requiring signal classification, should be avoided.

Using the MDCT in the speech mode of AC-4 can be regarded as an attempt to reduce switching problems, but AC-4 still requires signal classification. Another example of always coding in the MDCT domain is Multimode Transform Predictive Coder (MTPC) [29], with a difference that MTPC uses LP for quantization noise shaping. MTPC also has a classifier and three modes: speech, audio and transitional mode. MTPC is limited to wideband (WB).

A codec using psychoacoustic pre- and post-filter in [81, 85] uses only one structure for coding all signal types. The quantization noise, assumed to be additive white noise, is shaped by the post-filter controlled via a psychoacoustic model. The perceptual impact of the shaped noise is hence minimized. The pre-filter is the inverse of the post-filter. The filters have higher

resolution at lower frequencies, achieved by so-called frequency warping [105]. The pre-/post-filtering is integrated in MDCT-based Perceptual Audio Coder (PAC) [106], replacing PAC's FD noise shaping. Quality improvements are reported for speech, while maintaining performance of PAC for other types of signals. This pre-/post-filter approach is based on the short term LP, and not to be mixed with long-term pre-/post-filtering.

An early version of CELT includes LTP and is used for coding both speech and music in [107]. There are no specialized modes of coding in this early version of CELT. Convincing results for superwideband (SWB) are presented at 48 kbps and 64 kbps. Nonetheless, the authors opted for switched approach at lower bitrates in Opus and replaced LTP with a long-term pre-/post-filter approach in CELT [40]. There are also other examples of including LTP in MDCT coding [95, 108–110]. LTP is an important tool in LP speech coding and its usage within MDCT seems worth investigating.

Long-term post-filtering is a technique known from LP speech coders [37, 88, 100, 111]. It is sometimes also called bass post-filter [37]. It is named harmonic post-filter (HPF) in this thesis to distinguish it from LTP. The main idea is to attenuate quantization noise between harmonics. In FD coding, HPF is often accompanied by pre-filtering [40, 71, 112, 113]. The long-term pre-/post-filtering approach is similar to LTP, with the difference that the pre-filter uses the original signal for the prediction. Opus uses long-term pre-/post-filtering. EVS and MPEG-H use only long-term post-filtering [114] in their FD coding modes. Improvements, brought by HPF in the FD mode, were shown to be significant at 48 kbps in MPEG-H [115].

Another important part in LP speech coders is pulse coding of the LP and LTP residual [46]. Precise coding of pulses is important for voiced phones. It is more efficient to code pulses in TD than in FD. In [116], voiced segments are coded with a specialized speech codec and the decoded voiced segments are subtracted from the original input. The residual is coded in FD. The subjective listening tests didn't show any improvement with this approach. Thus, parallel coding of pulses and other parts of the input signals remains research opportunity.

Even though there was significant research on single structure for coding any type of signal, the recent standards have kept classification and switching approach. This choice is supported by listening test results [24, 42]. MPEG-H 3D audio and AC-4 target multi-channel broadcasting and there are no tests with publicly available results, at the moment, that check their quality at bitrates below 48 kbps. Nevertheless, MPEG-H 3D audio inherits the core coding from MPEG-D and adds tools known from EVS (e.g. long-term post-filter, time-domain bandwidth extension and IGF), that are expected to improve quality at low bitrates.

Traditionally, one way of reducing required bitrate was to reduce bandwidth. Reducing bandwidth is made counterproductive with the latest bandwidth extensions; fullband (FB) codecs can already at 16 kbps have quality not reachable by a WB codec [42]. Therefore, full perceivable frequency range should be coded to achieve maximum perceptual quality.

The bandwidth extension in AMR-WB+, xHE-AAC and AC-4 is implemented via quadrature mirror filterbank (QMF) as a pre- and post-processing step. This introduces additional delay and further reduces codec's structure uniformity. IGF [90] acts as a parametric codec and a bandwidth extension in EVS and MPEG-H, operating directly in the MDCT domain. IGF doesn't introduce additional delay and doesn't need an additional filterbank. However, the IGF

processing is still separated from the waveform preserving coding of the MDCT coefficients. IGF operates in EVS and MPEG-H above a specific frequency, below which simple noise filling is used. Thus, a full integration of parametric coding within the MDCT is not yet achieved.

To have pleasant and interactive communication, latency needs to be kept low [7]. The end to end latency depends on the codec delay and on the network delay [117]. As the network delay can be significant [118, 119], the codec delay must be kept low and the latency requirement was set to 32 ms in EVS [35].

The above analysis shows that there is still potential for an investigation, how to uniformly code any signal type in FB at low latency.

2.2 Objectives

This thesis aims at contributing to the ultimate goal of achieving a truly unified and universal coding scheme that does not require hard switching between paradigms for coding speech and other signal types. The MDCT allows efficient coding of every signal type and a direct integration of a bandwidth extension. A system that has the MDCT in its core is thus chosen as the starting point for the research. The objective is to study if transform-based coding can be extended to lower delays and bitrates than usually used, by adding advanced coding tools. The investigation should include coding the full or close to the full perceivable frequency range, even at low bitrates.

Computational complexity directly influences required electricity or battery life on an end device. Huge effort is invested in optimizing algorithms for deployment of standardized codecs. Such an effort is beyond scope of this thesis. Nonetheless, complexity is to be considered when making choices among different approaches and is yet one more of the reasons for choosing an MDCT-based coding.

Finally, the new transform-based coding scheme is to be compared to the state-of-the-art codecs at full bandwidth and at bitrates as low as 24 kbps.

3 Investigation on LP filtering

Frequency domain noise shaping (FDNS) is used for shaping of the quantization noise in MDCT-based TCX from EVS [37, 120]. FDNS is based on a spectral envelope. We can also consider LP filtering as a type of time domain noise shaping (TDNS), where spectral shaping of quantization noise is achieved by modifying the signal in TD. Such TDNS was already used in a combination with FD coding of the LP residual, as for an example in [121].

A decision whether to use FDNS or TDNS significantly influences structure of a codec, dictates choice of other methods and tuning of methods' parameters. It is thus important for further investigations to choose one of them as early as possible. However, scientifically sound comparison of FDNS and TDNS is hard due to the dependence of other methods on them and because of the complexity of a development of an audio codec.

The approach in the course of this thesis was to first investigate performance of a system in which FDNS is replaced with TDNS and to base the decision on the encountered advantages and problems over FDNS. This investigation is the first contribution of this thesis and is presented in this chapter.

3.1 Codec structure with LP filtering

In the first implementation of IVA, we started with MDCT-based TCX from EVS and replaced FDNS with TDNS. The structure is shown in Figure 3.1. The core structure of this implementation is similar to MDCT-based TCX from [121], where the codec in [121] is derived from TCX in [88] by replacing the DFT with the MDCT.

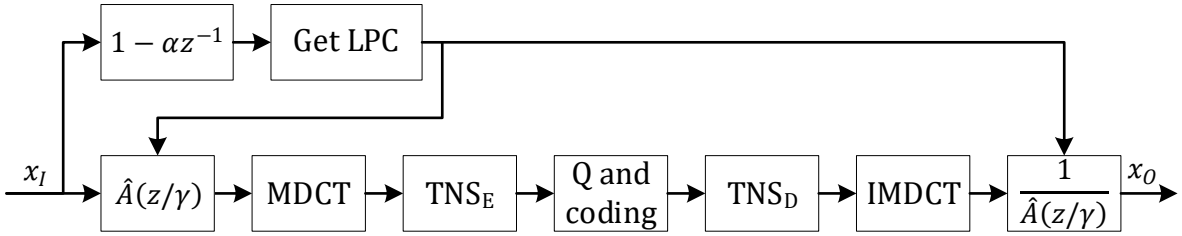

Figure 3.1 First IVA implementation with LP filtering

The concept resembles the pre/post-filter approach from [81] with a simplified psychoacoustic model. Since the codec operated only in WB, the non-linear frequency scale was not as important as in [81]. The input for obtaining LPCs is pre-emphasis filtered using $1 - \alpha z^{-1}$. The pre-emphasis acts as a spectrum tilt in the frequency response of the LPCs and so decreases quantization errors at low frequencies and increases quantization errors at high frequencies. Bandwidth expansion with factor γ, denoted as z/γ, moves the poles of the synthesis filter $1/\hat{A}$ [122] toward the origin and therefore is a simplified model of simultaneous masking. $\hat{A}(z/\gamma)$ will be referred to as the perceptual analysis and its inverse $1/\hat{A}(z/\gamma)$ as the perceptual synthesis filter. The output of the perceptual analysis is transformed with the MDCT and quantized and coded in the MDCT domain.

TNS is applied as in [37, 120] for temporal shaping of the quantization noise. TNS_E filters the MDCT coefficients so that the signal at its output has a flattened temporal envelope. TNS_D filters the decoded and dequantized MDCT coefficients, restoring the original temporal envelope and temporally shaping the quantization noise.

As in traditional speech coders, a slow change of the spectral envelope is assumed and the LPCs are updated every 20 ms. The same procedure as in EVS [120] is used for obtaining the LPCs: window the pre-emphasized signal with the 25 ms asymmetric window, calculate the autocorrelation of the windowed signal, apply adaptive lag windowing on the autocorrelation and acquire the LPs via Levinson-Durbin recursion. The autocorrelation's asymmetric window was chosen during the standardization of EVS [37, 120] and is kept in IVA. The LPCs are coded using line spectral frequencies (LSF) as in EVS and the coded LPCs $\hat{A}$ are used for the spectral shaping. The LSFs are first linearly interpolated to obtain 5 sets of LPCs, one set of LPCs for each 4 ms sub-frame. The LSFs in the sub-frame $j \in \{0,1,2,3,4\}$ are obtained using weighted arithmetic mean with $c_C[j]$ as the weight for the current frame LSFs and $1 - c_C[j]$ as the weight for the past frame LSFs, where $c_C = (0.2,0.4,0.6,0.8,1.0)$. The LPCs, derived from the interpolated LSFs, are exponentially weighted with γ to produce the perceptual filter coefficients, which are then transformed to reflection coefficients [3]. The LP filtering is implemented using a lattice structure [3]. It was found by informal listening that an additional sample-by-sample linear interpolation of the reflection coefficients within the sub-frame improves quality of coded clean speech for the codec presented in Figure 3.1. Further investigations have shown that even though the total energy of the prediction residual increases with the intra-sub-frame reflection coefficient interpolation, the energy below the fundamental frequency and between lower harmonics is decreased. Completely replacing the LSF interpolation with the reflection coefficient interpolation is not desired because LSF interpolation has smaller spectral distortion relative to LSFs obtained per sub-frame and higher prediction gain as previously investigated in [123, 124]. The perceptual filter coefficients of the two interpolated LSFs are however very similar and potential resonances are reduced with the exponential weighting, allowing well behaved reflection coefficients interpolation within the sub-frame. The conversion of LSFs into LPCs is computationally complex and a sample-by-sample LSF interpolation is thus not practical. It was also noticed in [85] that there is a need for a fine temporal interpolation of the filter coefficients. In CELP codecs, the analysis-by-synthesis coding per sub-frame takes care of the LPC changes at sub-frame borders, which is not the case for the investigated MDCT-based codec.

3.2 Adaptive interpolation of LPCs

In listening tests at 16 kHz sampling rate [9.1.1], that include the codec with the structure presented in Figure 3.1, participants' most frequent complaint for IVA is presence of subtle crackling, popping, clicking and snapping sounds. It is very hard for many of the perceived problems to find the corresponding location in a spectrogram, especially for low frequency noises. One example of an introduced transient that can be easily heard is presented in the top spectrogram in Figure 3.5. The noise burst around 1 kHz at 1.01 s is hard to spot in the spectrogram. Its existence becomes apparent, when comparing it to the spectrogram of the original signal in the bottom plot of the same figure. The noise burst is of a low level, but it happens at very sensitive frequencies for human hearing and there is no part of the original signal to mask it. The transient nature of the distortion, introduced by the coding, makes it easy to spot and hence significantly affects grading in a listening test. A sudden change is visible in signal characteristics of the original at the problematic place. Detailed investigations have revealed that the cause is a combination of the inadequate perceptual filter coefficient adaptation and limitations of the MDCT codec in presenting signal changes within the MDCT window at low bitrate.

A solution that was tried is replacing the fixed linear interpolation with coefficients c_C by an adaptive interpolation of the LPCs. The pre-emphasized signal in the current frame is windowed with 3 overlapping symmetrical cosine windows. For each window the perceptual filter coefficients $A_{M,j}(z/\gamma)$ are obtained with the same procedure as for the coefficients used in the LP filtering ($0 \leq j < 3$). The perceptual filter coefficients are also obtained for the whole past frame $\hat{A}_P(z/\gamma)$ and the whole current frame $\hat{A}_C(z/\gamma)$. Another set of perceptual filter coefficients $\hat{A}_I(z/\gamma)$ is derived from the mean of the two coded LSFs, from the previous and from the current frame. Magnitude responses ($X_{\hat{A}_P}, X_{\hat{A}_C}, X_{\hat{A}_I}, X_{A_{M,j}}$) of the perceptual filters are acquired via the DFT.

Energies corresponding to $\hat{A}_P$ and $\hat{A}_C$ are calculated:

$$E_{\hat{A}_P} = \sum\nolimits_i \left(X_{\hat{A}_P}[i]\right)^2$$

$$E_{\hat{A}_C} = \sum\nolimits_i \left(X_{\hat{A}_C}[i]\right)^2$$

Relative differences between frequency responses are:

$$R_{\hat{A}_P,\hat{A}_C} = \begin{cases} \sum_i \left(\dfrac{X_{\hat{A}_P}[i] - X_{\hat{A}_C}[i]}{\max\left(1, X_{\hat{A}_P}[i]\right)}\right)^2, E_{\hat{A}_P} < E_{\hat{A}_C} \\ \sum_i \left(\dfrac{X_{\hat{A}_P}[i] - X_{\hat{A}_C}[i]}{\max\left(1, X_{\hat{A}_C}[i]\right)}\right)^2, E_{\hat{A}_P} \geq E_{\hat{A}_C} \end{cases}$$

$$R_{\hat{A}_P,A_{M,j}} = \sum\nolimits_i \left(\frac{X_{\hat{A}_P}[i] - X_{M,j}[i]}{\max\left(0.1, X_{\hat{A}_P}[i] + X_{M,j}[i]\right)}\right)^2$$

$$R_{\hat{A}_C,A_{M,j}} = \sum_i \left(\frac{X_{\hat{A}_C}[i] - X_{M,j}[i]}{\max\left(0.1, X_{\hat{A}_C}[i] + X_{M,j}[i]\right)} \right)^2$$

$$R_{\hat{A}_I,A_{M,j}} = \sum_i \left(\frac{X_{\hat{A}_I}[i] - X_{M,j}[i]}{\max\left(0.1, X_{\hat{A}_I}[i] + X_{M,j}[i]\right)} \right)^2$$

The change position j_A is set:

$$j_A = \begin{cases} 0, \forall j \in \{0,1,2\}\, R_{\hat{A}_P,A_{M,j}} \geq \dfrac{R_{\hat{A}_C,A_{M,j}}}{2} \wedge R_{\hat{A}_I,A_{M,j}} \geq R_{\hat{A}_C,A_{M,j}} \\ \max(j+1), \exists j \in \{0,1,2\}\, R_{\hat{A}_P,A_{M,j}} < \dfrac{R_{\hat{A}_C,A_{M,j}}}{2} \vee R_{\hat{A}_I,A_{M,j}} < R_{\hat{A}_C,A_{M,j}} \end{cases}$$

LSF interpolation coefficients are chosen among sets for slow c_S and fast c_F LPCs change:

$$c_S[0] = (0.6,0.8,0.9,1.0,1.0)$$

$$c_S[1] = (0.4,0.6,0.8,0.9,1.0)$$

$$c_S[2] = (0.2,0.4,0.6,0.8,0.9)$$

$$c_S[3] = (0.0,0.2,0.4,0.6,0.8)$$

$$c_F[0] = (0.8,0.9,1.0,1.0,1.0)$$

$$c_F[1] = (0.0,0.8,0.9,1.0,1.0)$$

$$c_F[2] = (0.0,0.0,0.8,0.9,1.0)$$

$$c_F[3] = (0.0,0.0,0.0,0.8,0.9)$$

Whether slow c_S or fast c_F set is used, depends on the relative difference between frequency responses for the past frame and the current frame $R_{\hat{A}_P,\hat{A}_C}$. Which coefficients are exactly used depends on the change position j_A. In short, the LSF interpolation coefficients c_A choice is described by:

$$c_A = \begin{cases} c_S[j_A], & R_{\hat{A}_P,\hat{A}_C} < \tau_A \\ c_F[j_A], & R_{\hat{A}_P,\hat{A}_C} \geq \tau_A \end{cases}$$

The LSFs in the sub-frame $j \in \{0,1,2,3,4\}$ are then obtained using weighted arithmetic mean with $c_A[j]$ (instead of $c_C[j]$ used in the non-adaptive interpolation) as the weight for the current LSFs and $1 - c_A[j]$ as the weight for the past frame LSFs. As before the LSFs in each sub-frame is transformed to the reflection coefficients for the end of the sub-frame and the reflection coefficient interpolation in the lattice structure is used within the sub-frame.

This adaptation brings faster filter transition for a more significant signal characteristics' change and the transition is located so that the interpolated perceptual filters are closer to the local perceptual filters.

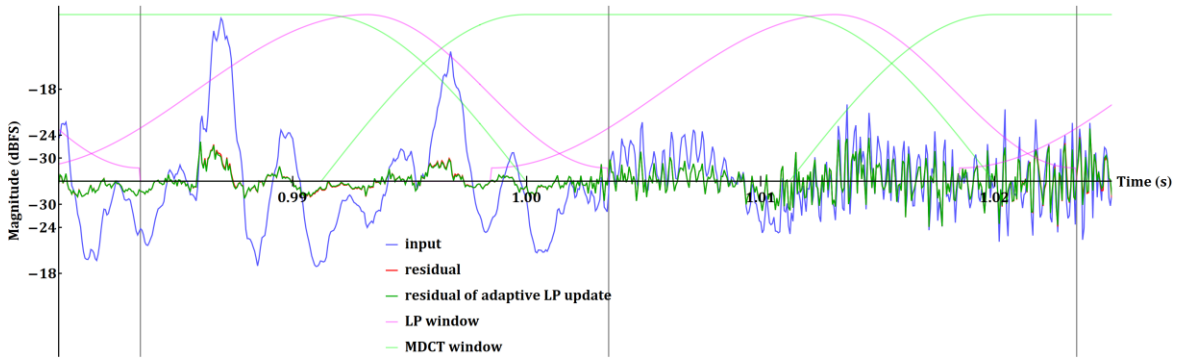

Figure 3.2 Window alignment and filtering in German male clean speech item de02

The alignment of the autocorrelation windows and the MDCT windows is depicted in Figure 3.2. The symmetric low overlap MDCT window with 8.75 ms overlap is used in the IVA version discussed in this chapter. The autocorrelation input is windowed with the asymmetric 25 ms window from EVS. The vertical lines mark the position where the new reflection coefficients, corresponding to the LSFs of the current frame, are applied. The LSFs of the current frame are acquired from the autocorrelation window whose right end is aligned with the vertical line where the corresponding reflection coefficients are in action.

The signal of 20 ms between 2 vertical lines is divided into 5 sub-frames. For each sub-frame, LSFs are interpolated from the LSFs corresponding to the autocorrelation window with the center between the 2 vertical lines and the LSFs of the previous autocorrelation window. As already mentioned c_A is used for the interpolation. The reflection coefficients at the end of each sub-frame are obtained from the interpolated LSFs, as described above, and the reflection coefficients are interpolated within each sub-frame.

As depicted in Figure 3.2, the most weight of the autocorrelation window is not centered at the location where the corresponding reflection coefficients are active. This is a consequence of the low latency objective stated in 2.2, as shifting the autocorrelation window to the right relative to the MDCT window would increase latency.

The residuals of the non-adaptive and the signal adaptive filter interpolation display almost no differences in TD [Figure 3.2]. Yet, as visible in the middle spectrogram of Figure 3.5, the transient distortion is significantly reduced with the adaptive filter interpolation. As already mentioned in 3.1, $c_C = (0.2,0.4,0.6,0.8,1.0)$ is always used for the non-adaptive interpolation. For the adaptive interpolation, as defined by the choice for c_A, $c_S[1] = (0.4,0.6,0.8,0.9,1.0)$ is used in the frame around 0.99 s and $c_S[2] = (0.2,0.4,0.6,0.8,0.9)$ in the frame around 1.01 s. This is an example where even small changes in LP filtering can have significant perceptual effects in combination with MDCT coding of residual.

Many other signals were tested with the adaptive filter interpolation. In many cases distortions were reduced, but also at many new places distortions have appeared. One such example of a new distortion is presented in Figure 3.6. The comparison of the residuals and the window alignment for this signal is shown in Figure 3.3. For the adaptive interpolation, as defined by the choice for c_A, $c_S[3] = (0.0,0.2,0.4,0.6,0.8)$ is used in the frame around 5.43 s and $c_F[2] = (0.0,0.0,0.8,0.9,1.0)$ in the frame around 5.45 s.

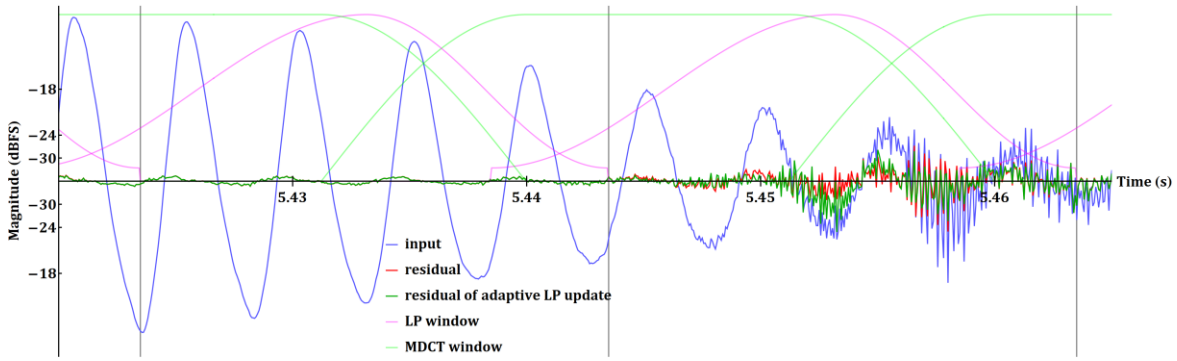

Figure 3.3 Window alignment and filtering in English female clean speech item [eng_f]

Another problem in the same English female clean speech is visible in Figure 3.7. This distortion is not easy to spot in the spectrogram, but can easily be heard in the version with the adaptive interpolation. This kind of low frequency plop is the most common disturbing distortion in the non-adaptive filter coefficient interpolation as well.

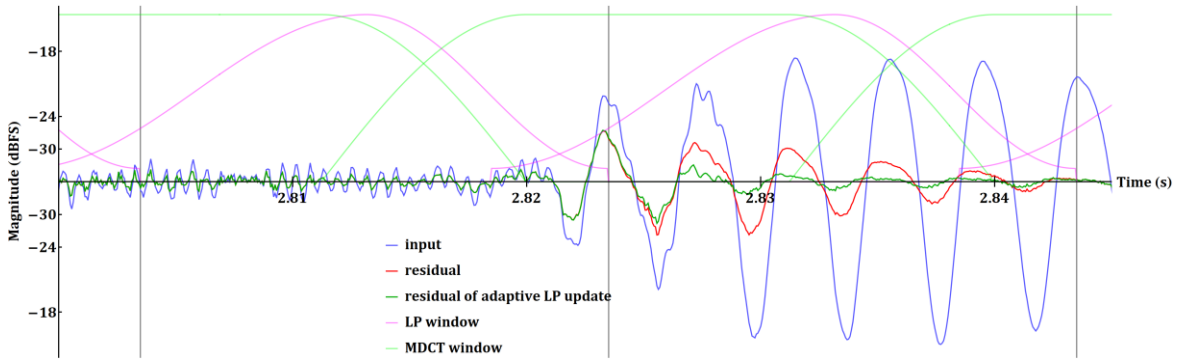

Figure 3.4 Window alignment and filtering in English female clean speech item eng_f

The residual is smaller with the adaptive interpolation [Figure 3.4], but has stronger amplitude modulation than the residual of the non-adaptive interpolation. Signals with strong amplitude modulation have flat spectrum and require more bits for presenting enough of important coefficients using scalar quantization in an FD. For the adaptive interpolation, as defined by the choice for c_A, $c_S[2] = (0.2,0.4,0.6,0.8,0.9)$ is used in the frame around 2.81 seconds and $c_F[0] = (0.8,0.9,1.0,1.0,1.0)$ in the frame around 2.83 seconds.

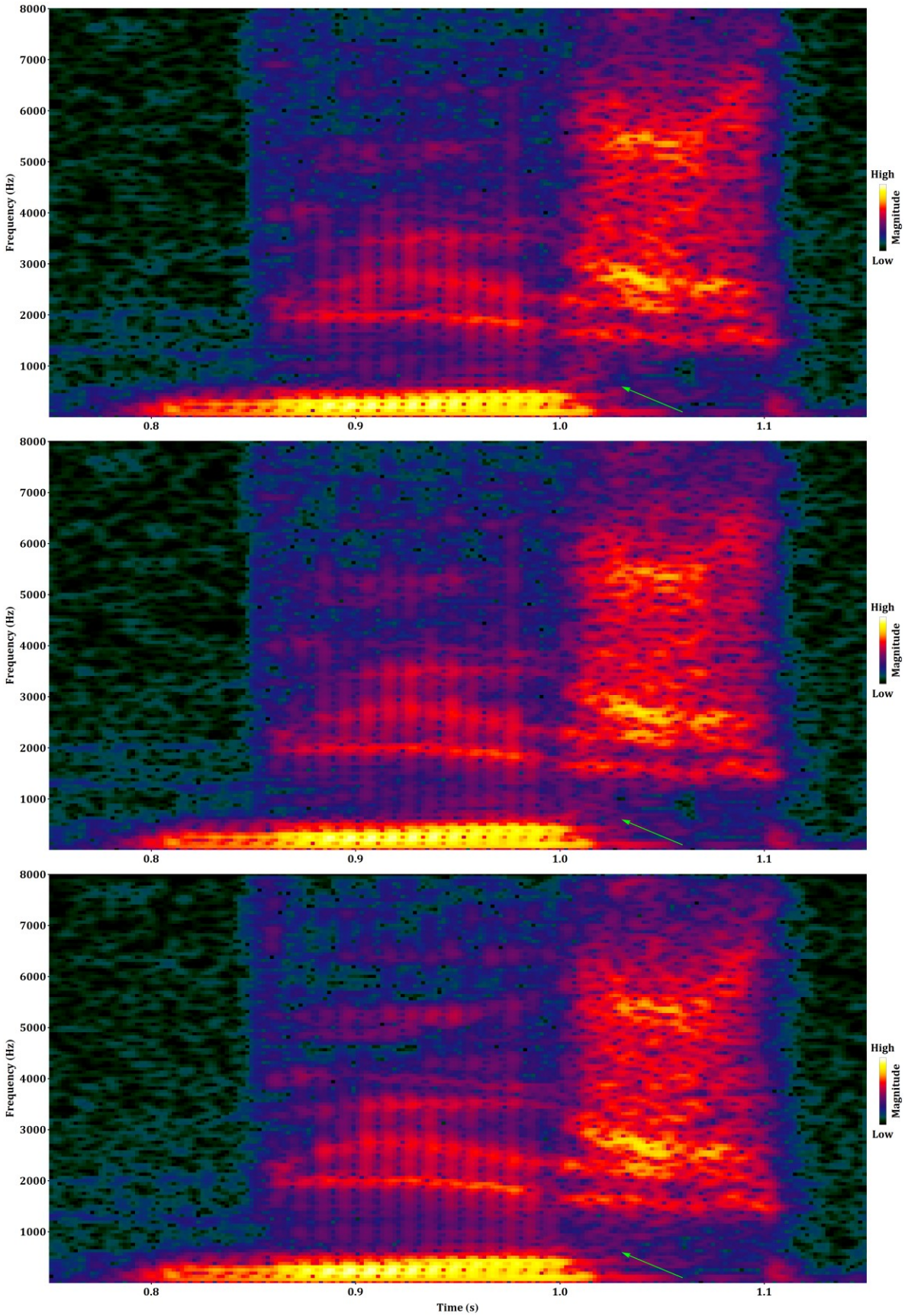

Figure 3.5 Click in German male clean speech item de02. Top – decoded, constant interpolation speed; Middle – decoded, adaptive interpolation speed; Bottom - original

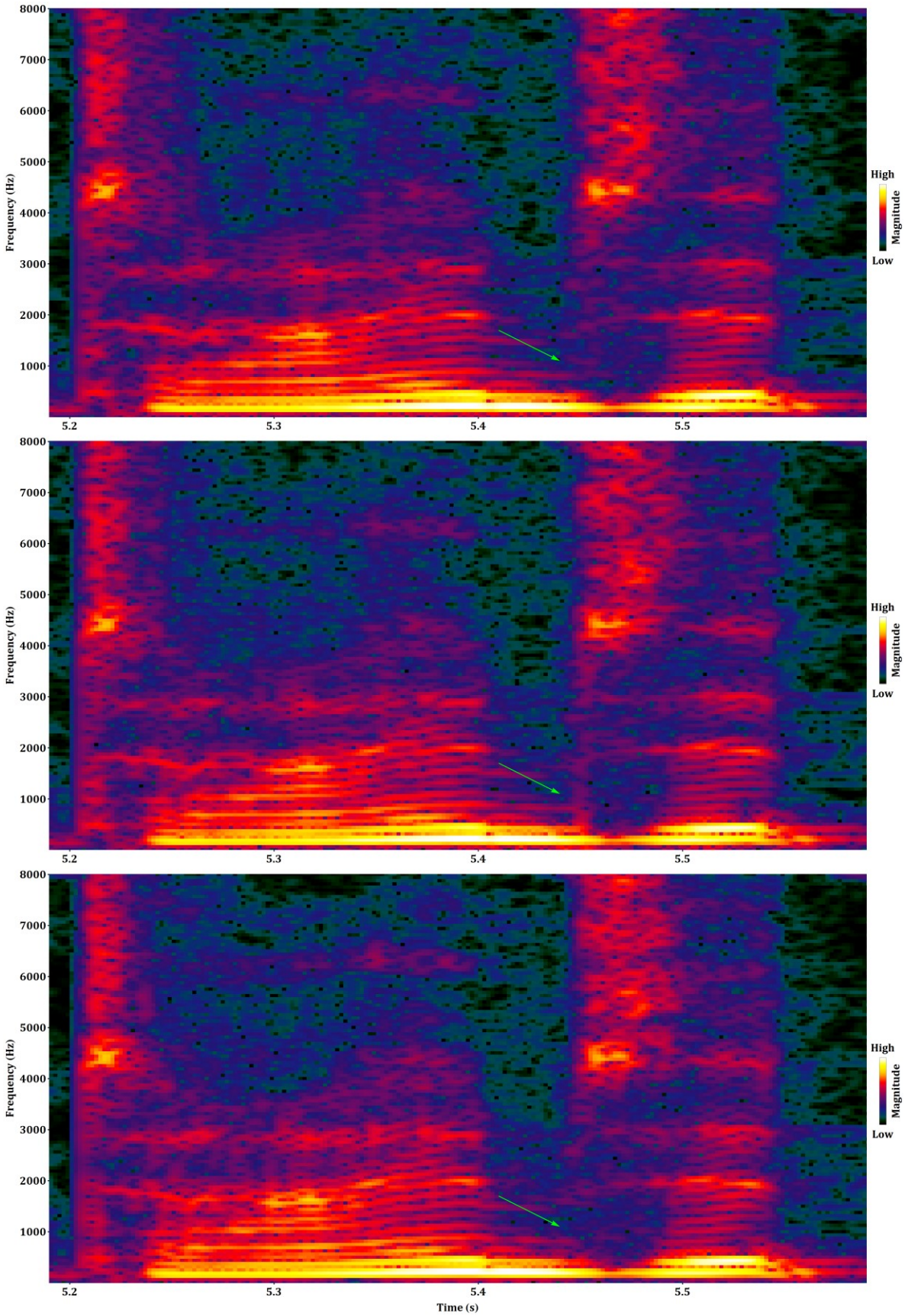

Figure 3.6 Click in English female clean speech item [eng_f]. Top – decoded, constant interpolation speed; Middle – decoded, adaptive interpolation speed; Bottom - original

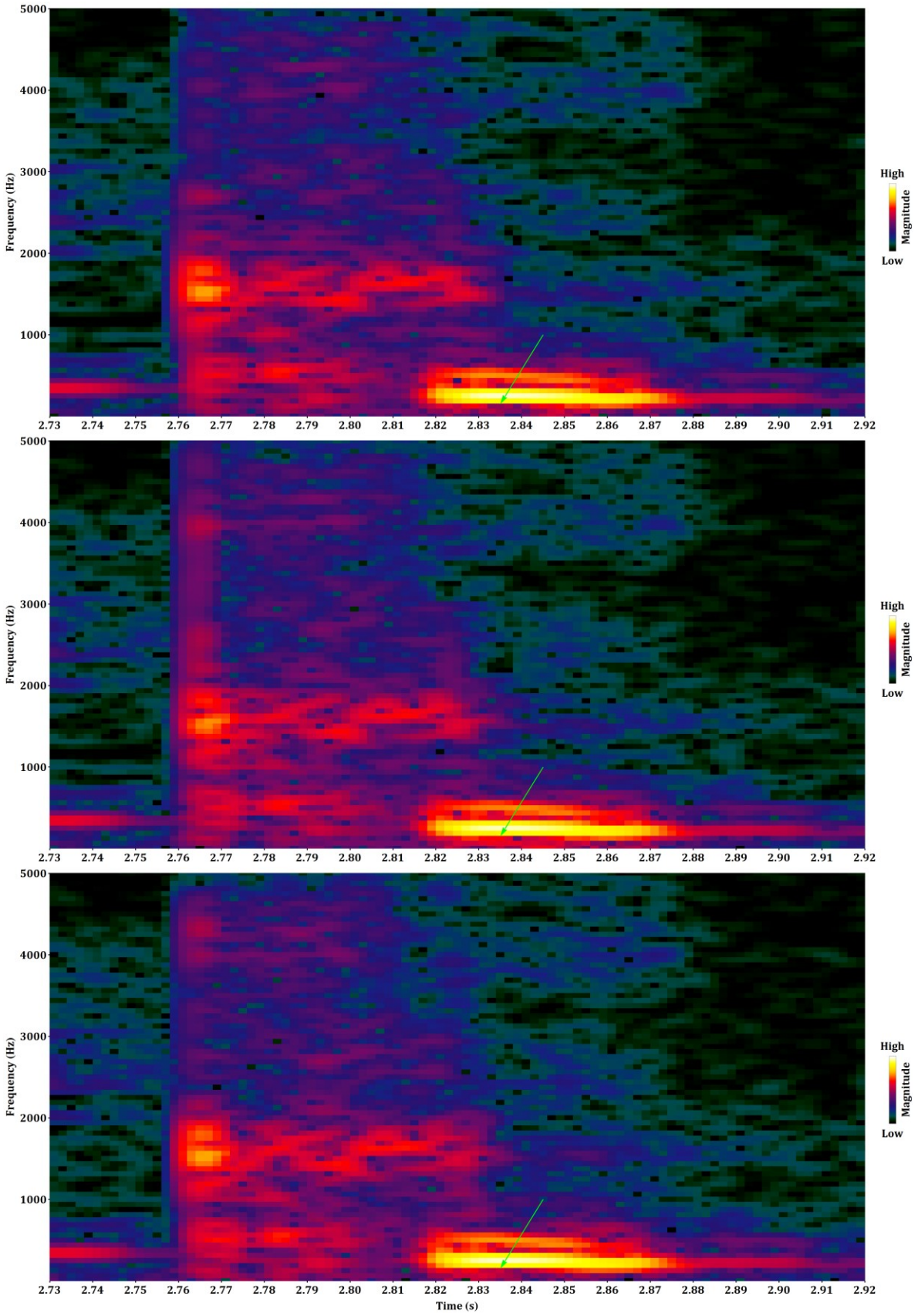

Figure 3.7 Click in English female clean speech item eng_f. Top – decoded, constant interpolation speed; Middle – decoded, adaptive interpolation speed; Bottom - original

3.3 Decision between TDNS and FDNS for FB system

The presented system [Figure 3.1] was tested for 16 kHz input, operating in WB. On one side, the time variant LP filters improve perceptual quality over the FDNS from EVS for coding clean speech. On the other side, the LP filters introduce crackling, popping, clicking and snapping distortions. These distortions are not encountered when the LP filtering is replaced with FDNS. The adaptive interpolation of the filter coefficients requires computationally complex signal analysis. Even the LP analysis and the LP coding themselves have significant computational complexity. Moving to FB, additional complexity would be introduced with frequency warping [85]. In [85] an update of the filter coefficients every 2 milliseconds was chosen to achieve high quality. It was also suggested that the number of bits for coding the filter coefficients needs to be small at low bitrates and an adaptive update of the coefficients was developed. The observation, that it is better to use fewer bits for coding a perceptual weighting at low bitrates, was also confirmed in the investigations with FDNS in [125]. Nonetheless, the same as with the above described linear frequency scale filtering, the investigations in [126] of different interpolations of the frequency warped filter coefficients from [85] showed unsatisfactory perceptual quality at low bitrates.

At the same time with the above described investigations, a new type of FDNS, named Spectral Noise Shaping (SNS), was developed [127]. SNS is a low complexity way for spectrum envelope modeling and perceptual quantization noise shaping. The SNS coefficients are derived from an MDCT or some other spectrum. Details of SNS are described in 4.5. SNS is used in the newly standardized Low Complexity Communication Codec (LC3) codec [79, 80]. Signal adaptive windowing for the MDCT [128] could be used with the SNS for achieving similar effects as with the adaptive interpolation of the LPCs. It needs to be noticed that the window overlapping provides frequency dependent smoothing between FDNS in neighboring frames.

In [81, 85] a separation of irrelevancy and redundancy reduction is proposed through the use of the pre-/post-filtering for the quantization noise shaping and the transform for the redundancy reduction. With SNS, separation of irrelevancy and redundancy reduction is not possible. Nevertheless, this separation could be questioned. First, the masking curve is derived from the spectral envelope and the perceptual pre-filtering also flattens the spectral envelope of the residual and consequently also partially decorrelates the signal, achieving redundancy reduction. Second, the quantization is the only non-preserving operation in the investigated codec and the only source of the perceived distortions. The quantization noise is of course modified with other processing, especially by TDNS or FDNS, to minimize the perception of the distortions. Yet, the quantization is done in the MDCT domain and only at very high bitrates it can be assumed to be white noise. The choice of the MDCT windowing scheme will thus also affect the irrelevancy reduction.

Except good modeling of a slow formant shifting in clean speech, no other advantage of TDNS was noticed by informal listening nor by comparing spectrograms of coded signals to the originals. Because of the annoying crackling distortions with TDNS, because of the computational complexity and because of the simplicity of SNS, we decided to use SNS for an FB implementation of IVA.

4 Codec structure

The proposed codec IVA is built upon the MDCT-based TCX from EVS [37, 120] with adaptations of some technologies from LC3 [79, 80]. It has been proven in EVS and LC3 that an MDCT-based codec with HPF can code well a wide range of mono signals, including clean speech, at bitrates of 48 kbps and bandwidth of at least 16 kHz [42, 129]. The MDCT-based codec is also the most effective in coding various signals even at lower bitrates, with few exceptions such as speech, and is thus chosen as the starting point for the presented research. FDNS is chosen over TDNS for spectral noise shaping in IVA, as the consequence of the investigation presented in 3. This codec structure was then extended, as a further contribution of this thesis, by adding new or modifying existing technologies, so that coding of a wide range of signals may be possible even at lower bitrates. Pulse extraction and coding is introduced for precise coding of pulses in voiced phones, as detailed in 5. LTP that uses decoded signal for the prediction, as described in 6, was added within the MDCT-based codec to improve coding of both tonal music and speech signals. HPF is modified to allow faster adaptation to the processed signal, as described in 8. Pulse coding, LTP and HPF are techniques known from specialized speech codecs as very important for achieving good quality at low bitrates. These techniques as proposed in this thesis are very different from their counterparts in speech codecs, but are anyhow expected to bring the same benefits in IVA. Finally, IGF and core band noise filling are replaced with the integral band-wise parametric coder (iBPC), achieving integrated approach for waveform preserving coding and bandwidth extension, as another step towards unified codec structure. The iBPC is presented in 7. Before describing the details of these technologies, additional smaller contributions are presented in 4 together with the description of the overall IVA codec structure.

4.1 Codec structure overview

The MDCT-based codec processes its input signal in frames of 20 ms, the length in samples being H_M. The other components follow the same framing and have the update rate of 50 frames per second of the coded parameters.

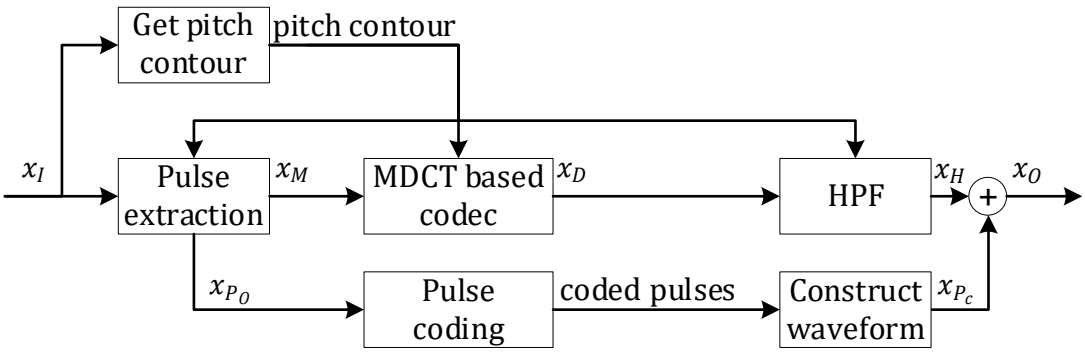

Figure 4.1 IVA codec structure overview

The input signal x_I, which is the original audio signal processed with a 20 Hz high-pass/DC rejection filter, is input to the pitch analysis and to the pulse extraction. In the pitch analysis, denoted as "Get pitch contour", a pitch contour is obtained. The pitch contour steers many parts of the codec, including pulse extraction and HPF. The pitch contour tracks pitch change over time. In this thesis, pitch is used to term the fundamental frequency or the pitch lag, where the pitch lag is also known as the fundamental period and is the inverse of the fundamental frequency. Viewing pitch as a frequency or as a period is equivalent when describing the concepts of the implemented algorithms. When pitch is used for filtering in TD it is the corresponding pitch lag, expressed in number of samples, that is used. Whenever the exact representation is important, pitch will be denoted as pitch lag or fundamental frequency or frequency bin index.

The pulse extraction extracts pulses from x_I and codes them, where a pulse is a glottal pulse or any other kind of transient. The extracted pulses x_{P_O} are subtracted from x_I. The signal without the pulses x_M is coded within an MDCT-based codec.

The decoded output x_D of the MDCT-based codec is filtered via an HPF. The HPF produces x_H, in which noise between harmonics is suppressed.

A waveform x_{P_C} constructed from the coded pulses is added to x_H to obtain the final output of the codec x_O.

4.2 Encoder

The encoder splits the input signal x_I into 20 ms frames and outputs to the bit-stream with the rate of 50 frames per second:

- static configuration, 5 bits consisting of the bitrate and the bandwidth
- pitch contour
- MDCT window choice, 2 bits
- LTP activation parameter
- coded pulses

- coded information for the spectral shaping via SNS, denoted as “sns”
- coded information for the temporal shaping via TNS, denoted as “tns”
- global gain g_Q, that is the global quantization step size for the MDCT codec
- the entropy coded quantized MDCT spectrum, denoted as “spect”
- the parametrically coded zero portions of the quantized MDCT spectrum, denoted as “zfl”

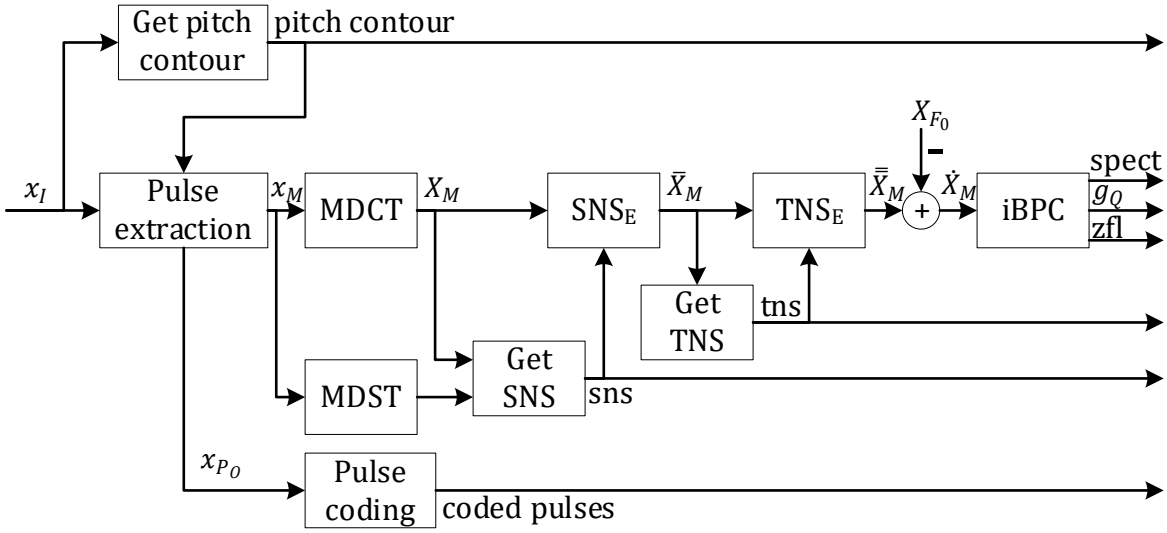

Figure 4.2 Encoder structure in IVA

The pitch contour is obtained and coded for frames with high harmonicity. For frames with low harmonicity, it is signaled with one bit that the pitch contour is not present. The pitch analysis also outputs the high frequency tonality flag ϕ_H and a flag indicating if constant parameters should be used for the HPF across the whole frame.

The pulse extraction obtains a short-time Fourier transform (STFT) of the input audio signal, and uses a log magnitude spectrogram and the phase spectrogram of the STFT to find and extract pulses. Each pulse has a waveform with high-pass characteristics and the energy concentrated in the temporal center [Figure 5.15,Figure 5.17]. The pulse residual signal x_M is obtained by subtracting a signal consisting of pulses x_{P_0} from the input audio signal x_I. The pulses are coded and output by the “Pulse coding”.

The pulse residual signal x_M is windowed and transformed via the MDCT to produce the spectrum X_M. Additionally the windowed x_M is transformed via the modified discrete sine transform (MDST). A spectrum consists of frequency coefficients, also named frequency bins or lines. The length of X_M, and all other spectra derived from X_M, is equal to the frame length H_M. The MDCT window is chosen among three windows as in EVS, with the difference that all three windows are symmetrical low overlap windows and that the overlap is longer. The low overlap window is equal to the square root of the Tukey window. The three windows used have the length of 30 ms, 25 ms and 21.25 ms with the corresponding overlap region having the length of 10 ms, 5 ms and 1.25 ms. The longer window allows better energy compaction

for tonal signals compared to EVS. The decision which window to use is basically the same as in EVS, with the difference that there is only overlap change and no splitting of the frame into sub-frames. This means that the overlap choice is made so that an increase of energy occurs in the non-overlapping part of the MDCT window, thus avoiding problems with the MDCT unfolding and TNS. In the initial versions of IVA, sub-frames of 5 ms and 10 ms, as in EVS, were used to have control over temporal allocation of the quantization noise. With the introduction of the pulse extraction and by using TNS, the signal to be quantized $\bar{\bar{X}}_M$ is mostly stationary within the frame and thus the need for splitting the frame is reduced. Avoiding the sub-frame splitting, makes the implementation of new technologies and the maintenance easier. It could be investigated in the future, if there are signals where the sub-frame splitting could still give an additional benefit.

The spectral envelope of X_M is perceptually flattened using SNS obtaining $\bar{X}_M$. SNS originates from LC3 [79, 80]. It is implemented in IVA from scratch, with some algorithm adaptations compared to LC3.

TNS provides temporal shaping of the quantization noise by filtering across frequencies in the MDCT spectrum [84]. TNS filter coefficients are obtained from the autocorrelation of the MDCT spectrum and TNS is activated if the filter provides coding gain above a threshold. Two independent filters are used, one for the range from 800 to 4500 Hz and another one above 4500 Hz. If it is active, the TNS filters $\bar{X}_M$ and produces $\bar{\bar{X}}_M$ as its output. The TNS implementation is essentially the same as in EVS, the main difference being additional tuning of the frequency range where it is used and the activation thresholds. The changes in the TNS are introduced to solve problems identified by informal listening to coded outputs. The tuning was done by manual searching for parameters that reduce the identified problems without significantly affecting other samples. TNS and the pulse coding complement each other: since the TNS filters are of relatively low order, they cannot model very sharp changes in the time envelope of the signal. As the pulse coding only models transients of short duration, TNS is still needed for modeling temporally broader transients and strong onsets of a noise or a tone.

The perceptually flattened predicted MDCT spectrum X_{F_0}, obtained from the previously decoded frames via LTP, is subtracted from $\bar{\bar{X}}_M$ and the resulting difference $\dot{X}_M$ is quantized and coded in the iBPC. The usage of the decoded signal for LTP makes it a coding gain tool and not a noise shaping tool as LTPF in EVS. A part of the decoder [Figure 4.6, Figure 6.1], required for the functionality of LTP, must also be included in the encoder, but is not shown in Figure 4.2 to keep it readable. The quantization process finds the optimal global gain g_Q and outputs besides the quantized global gain also the entropy coded quantized MDCT spectrum “spect” and energy levels “zfl” in sub-bands quantized to zero. The iBPC uses an arithmetic coder for coding both “spect” and “zfl”. The probability and context tables from EVS are kept [37, 120], while the arithmetic coder is implemented as in LC3 [79]. A pair of values, called 2-tuple, is jointly coded as in EVS and LC3. The arithmetic coder from LC3 has smaller computational complexity than the one from EVS. It also has exact bit demand estimation using just table lookup.

4.3 Pitch contour

The pitch contour is determined by $\dot{d}_{F_0}$, $\acute{d}_{F_0}$ and $\grave{d}_{F_0}$. The pitch $\dot{d}_{F_0}$ is found in the middle and the pitch $\acute{d}_{F_0}$ at the end of the current MDCT window. The pitch $\grave{d}_{F_0}$, at the start of the current window, is equal to $\acute{d}_{F_0}$ from the previous frame. The intervals where these and other pitch values are found is presented relative to the MDCT windowing and input signal framing in Figure 4.3.

The pitch values estimated by the pitch search fall between the minimum pitch lag $\check{d}_{F_0} = 2.25$ ms (corresponding to 444.4 Hz) and the maximum pitch lag $\hat{d}_{F_0} = 19.5$ ms (corresponding to 51.3 Hz). The range from $\check{d}_{F_0}$ to $\hat{d}_{F_0}$ will be referred to as "the full pitch range".

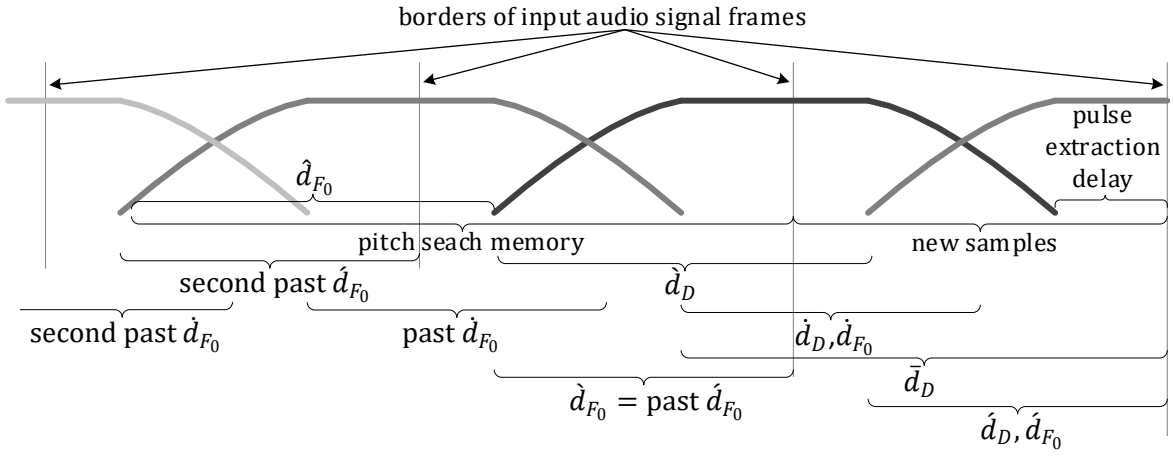

Figure 4.3 Location of the pitch contour parameters

The values of $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$ are found in multiple steps. In each step, a pitch search is executed on one of the intervals depicted in Figure 4.3. To reduce complexity, a pitch is first searched in the input signal downsampled to 8 kHz, followed by a search in the original input signal at the full input sampling rate around the value obtained on the downsampled signal.

The core of the process is the pitch search $\mathcal{F}_{F_0}$, executed in each of the multiple steps. The pitch search $\mathcal{F}_{F_0}$ is also used in the HPF.

4.3.1 Pitch search

The pitch search $\mathcal{F}_{F_0}$ calculates the sample Pearson's correlation coefficient of its input x and its delayed version in the range from $d_{\check{F}_0}$ to $d_{\hat{F}_0}$:

$$\rho_H[m] = \frac{\sum_{n=0}^{L_H} x[n]x[n-m]}{\sqrt{\left(\sum_{n=0}^{L_H} x^2[n]\right)\left(\sum_{n=0}^{L_H} x^2[n-m]\right)}}, d_{\check{F}_0} \le m \le d_{\hat{F}_0}$$

Fractional delays are obtained by interpolating ρ_H. The interpolation is done using filters from a predefined list of filters, each filter having a distinct fractional delay between 0 and 1. It will

be further considered that $\rho_H[m]$ also includes values for fractional delays m, that is, $\rho_H[m]$ is also defined when m is a rational number.

Besides $d_{\check{F}_0}$, $d_{\hat{F}_0}$ and L_H, an initial pitch candidate $d_{\tilde{F}_0}$ is also a parameter of $\mathcal{F}_{F_0}$. $\mathcal{F}_{F_0}$ returns an optimal pitch $d_{\ddot{F}_0}$ and an associated harmonicity level $\rho_{\ddot{F}_0}$:

$$\left(d_{\ddot{F}_0}, \rho_{\ddot{F}_0}\right) = \mathcal{F}_{F_0}\left(x, d_{\check{F}_0}, d_{\hat{F}_0}, L_H, d_{\tilde{F}_0}\right)$$

The value of the initial pitch candidate $d_{\tilde{F}_0}$ is, in most cases, a value of $d_{\ddot{F}_0}$ returned by a previous call to $\mathcal{F}_{F_0}$ for a temporally preceding interval or for a downsampled version of x.

The harmonicity level $\rho_{\ddot{F}_0}$ is obtained from the normalized autocorrelation ρ_H depending on $d_{\ddot{F}_0}$. The value of $\rho_{\ddot{F}_0}$ is between zero and one, zero meaning no harmonicity and one the maximum harmonicity.

The location of the absolute maximum in ρ_H is the first candidate d_{F_1} for $d_{\ddot{F}_0}$:

$$d_{F_1} = \underset{d_{\check{F}_0} \le m \le d_{\hat{F}_0}}{\operatorname{argmax}} \rho_H[m]$$

The second candidate d_{F_2} is the local maximum of ρ_H near $d_{\tilde{F}_0}$. If $d_{\tilde{F}_0}$ is near d_{F_1} the local maximum would probably coincide with d_{F_1} and thus, instead of searching for a local maximum, d_{F_2} is simply set to $d_{\tilde{F}_0}$.

The value of $d_{\ddot{F}_0}$ is chosen among d_{F_1} and d_{F_2}, with a preference for d_{F_2}:

$$d_{\ddot{F}_0} = \begin{cases} d_{F_1}, \rho_H[d_{F_1}] - \rho_H[d_{F_2}] \ge \tau_{F_0} \\ d_{F_2}, \rho_H[d_{F_1}] - \rho_H[d_{F_2}] < \tau_{F_0} \end{cases}$$

$$\tau_{F_0} = \begin{cases} 0.01, 0.75 d_{F_1} \le d_{\tilde{F}_0} \le 1.25 d_{F_1} \\ 0.02, d_{F_1} \le d_{F_2} \\ 0.03, d_{F_1} > d_{F_2} \end{cases}$$

The values of τ_{F_0} were found heuristically by listening to decoded outputs and finding values that fix identified problems with minimal impact on other samples. The threshold is bigger if there is a possibility of an octave jump, which would occur if $d_{\tilde{F}_0}$ was returned by $\mathcal{F}_{F_0}$ in the temporally preceding interval and $d_{\tilde{F}_0}$ is an integer multiple of $d_{\ddot{F}_0}$ or $d_{\ddot{F}_0}$ is an integer multiple of $d_{\tilde{F}_0}$. Preferring d_{F_2} avoids possible octave jumps and to some extent increases pitch detection accuracy in presence of noise.

4.3.2 Obtaining the pitch contour

In each of the intervals depicted in Figure 4.3, L_H in the call to $\mathcal{F}_{F_0}$ is set to the length of the interval and $x[0]$ is the first sample in the interval.

The values of $\grave{d}_D$, $\bar{d}_D$, $\dot{d}_D$ and $\acute{d}_D$ are obtained at the downsampled signal.

First, the pitch $\grave{d}_D$ and the associated $\grave{\rho}_D$ are obtained via $\mathcal{F}_{F_0}$ using the full pitch range and $d_{\tilde{F}_0} = \grave{d}_{F_0}$. Then $\bar{d}_D$ and $\bar{\rho}_D$ are obtained using again the full pitch range and $d_{\tilde{F}_0} = \grave{d}_D$. $\dot{d}_D$ and $\acute{d}_D$ are found setting $d_{\tilde{F}_0} = \bar{d}_D$, $d_{\check{F}_0} = 0.7\bar{d}_D$ and $d_{\hat{F}_0} = 1.3\bar{d}_D$.

The value of $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$ and the associated harmonicities $\dot{\rho}_{F_0}$ and $\acute{\rho}_{F_0}$ are obtained via $\mathcal{F}_{F_0}$ from the input signal at the input sampling rate, setting $d_{\bar{F}_0} = d_D$, $d_{\check{F}_0} = d_D - a$ and $d_{\hat{F}_0} = d_D + a$, where a is the ratio of the input sampling rate and the downsample rate 8 kHz (e.g. $a = 6$ for the input sampling rate of 48 kHz). $d_D = \dot{d}_D$ for finding $\dot{d}_{F_0}$ and $d_D = \acute{d}_D$ for $\acute{d}_{F_0}$.

The frame is considered to have high harmonicity if $\max(\dot{\rho}_D, \bar{\rho}_D) \geq 0.3$ and $\acute{\rho}_{F_0} \geq 0.3$ and $\dot{\rho}_{F_0} \geq 0.6$. Otherwise the frame is of low harmonicity and the pitch contour is not coded.

For frames with high harmonicity, $\acute{d}_{F_0}$ is absolutely coded and $\dot{d}_{F_0}$ is differentially coded. $\dot{d}_{F_0}$ is coded differentially to $(\grave{d}_{F_0} + \acute{d}_{F_0})/2$ using 3 bits. A predefined difference value is associated with each of the 8 codes for the differential coding of $\dot{d}_{F_0}$. The code for the difference, that minimizes the autocorrelation in the interval associated with $\dot{d}_{F_0}$, is chosen for $\dot{d}_{F_0}$.

If $\acute{\rho}_{F_0} < \dot{\rho}_{F_0}/2$, it is considered that there is an end of the harmonicity in the frame. For such frames, $\acute{d}_{F_0}$ is set to $2\dot{d}_{F_0} - \grave{d}_{F_0}$ before coding it. This allows differential coding of $\dot{d}_{F_0}$ even when there is no harmonicity at the end of the frame and the originally found $\acute{d}_{F_0}$ is unreliable.

The pitch contour d_V is obtained by an interpolation of the decoded values of $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$. The pitch contour d_V provides a value at every sample in the current MDCT window and at least in the past $\hat{d}_{F_0}$ samples. We denote the pitch at sample i as $d_V[i]$. The pitch contour in the samples before the current window is needed for LTP.

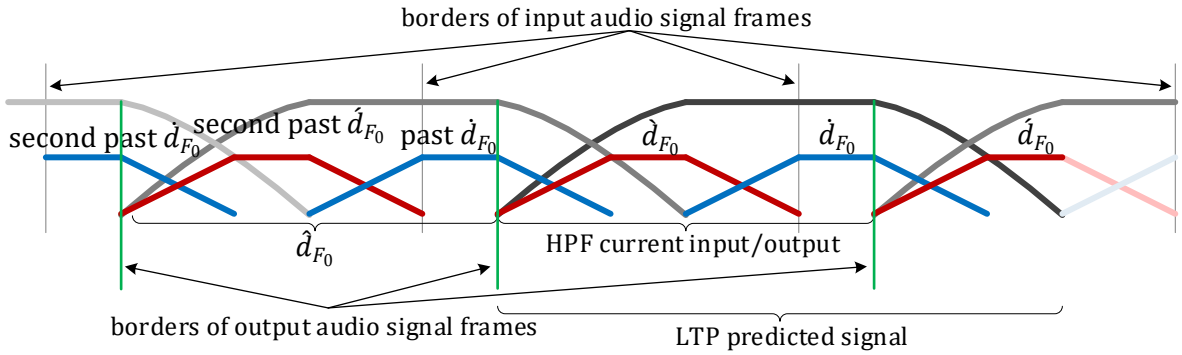

Figure 4.4 Schematic presentation of the interpolation for the pitch contour

The pitch contour consists of the constant and interpolated parts, as depicted in Figure 4.4, where the usage location of each coded pitch value is indicated in relation to the MDCT window, the input and output frame borders.

If $|\acute{d}_{F_0} - \dot{d}_{F_0}| \leq \frac{1}{24}$ ms, $|\acute{\rho}_{F_0} - \dot{\rho}_{F_0}| \leq 0.5$ and there is no significant amplitude modulation in the interval where $\acute{d}_{F_0}$ and $\dot{d}_{F_0}$ was found, it is indicated that the HPF should use constant parameters across the whole frame.

The average pitch lag $\bar{d}_{F_0}$ is calculated for each frame as the arithmetic mean of $\acute{d}_{F_0}$, $\dot{d}_{F_0}$ and $\grave{d}_{F_0}$. If any of $\acute{d}_{F_0}$, $\dot{d}_{F_0}$, $\grave{d}_{F_0}$ is zero then $\bar{d}_{F_0}$ is also set to zero, which happens when there is low harmonicity in the current or the previous frame.

4.4 Additional tonality measure

4.4.1 Per MDCT bin tonality

It is determined for each MDCT frequency bin between 4.8 kHz and 16 kHz, as in 5.3.3.2.5 in the EVS algorithmic description [37], if the frequency bin belongs to a tone or contains mostly noise. The bin tonality is used, as in EVS, for the adaptive changing of the dead-zone in the MDCT spectrum quantizer.

4.4.2 High frequency tonality flag

In a bandwidth extension, such as IGF, it is desired to reproduce the same amount of tonality in the generated spectrum portions as in the original signal. For this purpose, here presented decision was developed to signal a tonality to IVA decoder. The computational complexity of the decision is low and in informal listening no need for more advanced processing was noticed.

The total number of tonal frequency bins $\dot{N}_T$ is calculated in the current frame and smoothed over time: $N_T = 0.5N_T + \dot{N}_T$.

Normalized correlation ρ_{H_F} is calculated on a high-pass filtered x_M between the samples in the current MDCT window and the samples at the $\bar{d}_{F_0}$ delay, where x_M is filtered using a symmetric 5 tap finite impulse response (FIR) filter with the -3 dB cutoff at 6 kHz.

The high frequency tonality flag ϕ_H is set to 1 if TNS is inactive, the pitch contour is present and there is a tonality in high frequencies, where the tonality exists in high frequencies if $\rho_{H_F} > 0.7$ or $N_T > 1$.

The high frequency tonality flag ϕ_H is used in the decoder to decide how to fill MDCT coefficients quantized to zero.

4.5 SNS

The SNS scale factors are obtained from smoothed and spectrally tilted energies in 64 frequency sub-bands having increasing widths. The sub-band energies are transformed to a logarithmic domain after adding a noise floor. The sub-band energies are then downsampled to 16 values in the logarithmic domain, the arithmetic mean is removed and a fixed scaling is applied. The 16 values are then quantized and coded, the coded values being denoted as “sns”. The 16 quantized values are interpolated and transformed back to the linear domain. The interpolated scale factors are applied on the MDCT spectrum, where in $\mathrm{SNS_E}$ the MDCT spectrum is divided by the scale factors and in $\mathrm{SNS_D}$ multiplied. This process is the same as in LC3 [79, 80].

Different to the SNS implementation in LC3 is that the 64 sub-band energies are obtained from a squared magnitude spectrum, where the squared magnitude spectrum is obtained from the modulated complex lapped transform (MCLT) [130] consisting of the MDCT and MDST. Using the magnitude instead of the MDCT avoids the problem of the MDCT temporal variation for constant amplitude signals.

Another difference is that the sub-band energies are not normalized by the sub-band width. By not normalizing the sub-band energies, a similar effect is obtained as by the fixed spectral tilt in LC3 SNS. Instead of LC3's strong fixed spectral tilt ($10^{i/21}$ for 48 kHz), smaller adaptive tilt is used, proportional to the harmonicity of the input signal (maximum of $10^{i/63}$). Via the stronger tilt for harmonic signals, harmonic components at low frequencies are given more importance and vice versa. Weaker tilt for applause signals achieves more even distribution of quantization steps across the whole bandwidth and allows perceptually efficient coding of wideband transients.

An additional difference is that the SNS scale factors are coded using a two stage full search vector quantizer without a split, having 9 bits in the first and 6 bits in the second stage. This quantization scheme requires less bits than the scheme with pyramid vector quantizer in LC3, thus leaving more bits for coding the flattened MDCT spectrum, which was shown to be beneficial in [125].

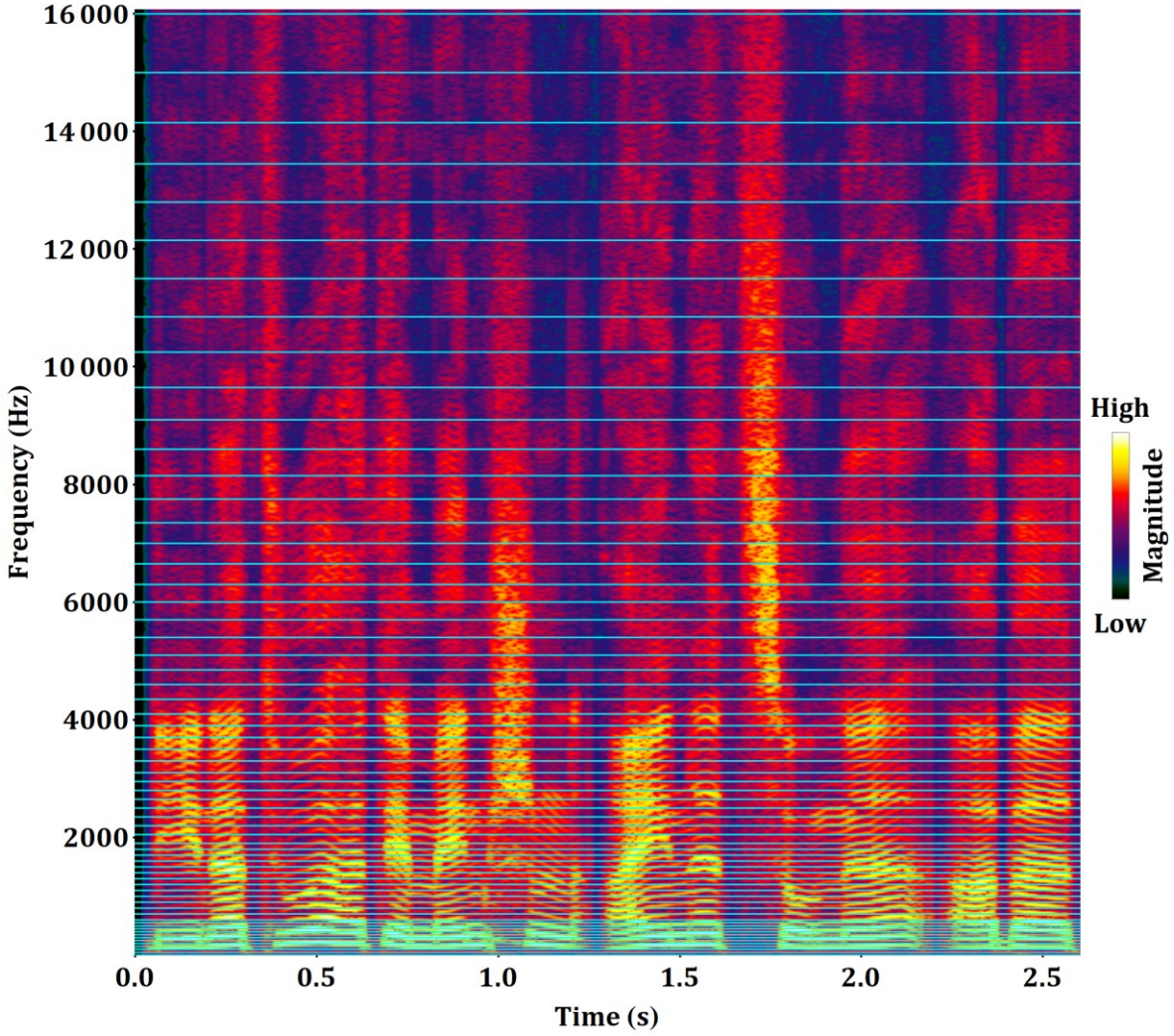

Figure 4.5 Frequency band division used in SNS

The borders of the 64 sub-bands used in IVA are shown as horizontal lines in Figure 4.5. The last sub-band from 16 to 24 kHz is not shown.

The quantized values are interpolated to 128 scale factors, thus producing a smoother envelope than the 64 interpolated scale factors in LC3. The 128 sub-bands are obtained by splitting each of the 64 sub-bands into two halves.

4.6 Decoder

The decoder processes the output of the encoder frame by frame.

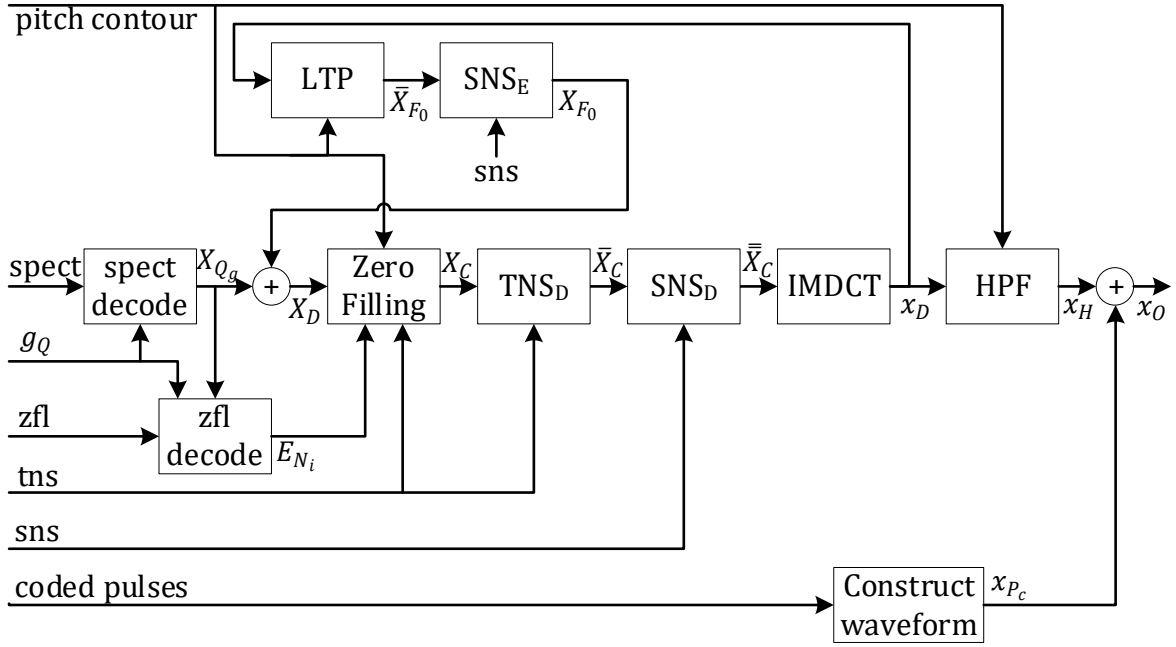

Figure 4.6 Decoder structure in IVA

The coded "spect" is decoded to obtain the quantized MDCT spectrum X_Q. The energies E_{N_i} are decoded from "zfl" for sub-bands that are completely zero in X_Q. The quantized MDCT spectrum X_Q is scaled with the global gain g_Q to produce X_{Q_g}. Since the zeros in X_Q and in X_{Q_g} are at the same position, each of them can be used for determining the location of the zeros.

The perceptually flattened predicted MDCT spectrum X_{F_0}, obtained from the previously decoded frames via LTP, is added to X_{Q_g}, producing X_D. For spectral coefficients that are zero in X_{Q_g}, values are generated in the Zero Filling and added to X_D, producing X_C. The Zero Filling is generated depending on the high frequency tonality ϕ_H flag, "pitch contour", "tns", E_{N_i} and the content of X_D. The Zero Filling includes copying from the low frequency regions into the high frequency regions, known as the copy-up. The Zero Filling is a part of the iBPC and is described in 7.4.

If TNS is active, the temporal envelope is restored in "TNS$_D$" producing $\bar{X}_C$. Further the spectral envelope is restored in "SNS$_D$" to produce $\bar{\bar{X}}_C$. By restoring the temporal and spectral envelopes, the quantization noise is shaped accordingly. Since it is expected that the quantization noise is uniformly distributed and since SNS is perceptually motivated, the quantization noise in $\bar{\bar{X}}_C$ should be perceptually minimized for the given bitrate constraint. As

discussed in previous chapter, the uniform distribution of the quantization noise cannot be truly satisfied at low bitrates, but we assume that it is at least somewhat achieved. To truly minimize the perceived distortions of the decoded output, all components can be tuned in an iterative process to account for their interaction. This is a disadvantage over the analysis by synthesis approach, but allows developing a low computational complexity system that doesn't rely on a psychoacoustic model which should be tuned also in accordance to the system.

The reshaped spectrum $\bar{\bar{X}}_C$ is a quantized version of the original spectrum X_M. It is converted to the time domain signal x_D via the inverse MDCT, windowing and adding of the overlapping parts, thus also achieving TDAC. The joint process of the inverse MDCT, windowing and the adding of the overlapping parts is presented in diagrams using single block named "IMDCT".

LTP generates a time domain prediction from x_D depending on the pitch contour and the LTP activation parameter, windows the time domain prediction and converts the windowed signal to the predicted MDCT spectrum $\bar{X}_{F_0}$. The spectrum $\bar{X}_{F_0}$ is perceptually flattened using the "$\mathrm{SNS_E}$". The perceptually flattened predicted MDCT spectrum X_{F_0} is in the same domain as $\dot{X}_M$ and X_{Q_g} and can be used to reduce the range of values in the spectrum to be coded.

As already mentioned, x_H is produced by suppressing noise between harmonics of x_D via HPF, driven by the pitch contour. Finally x_{P_C}, constructed from the coded pulses, is added to x_H to obtain the final output of the codec x_O.

5 Pulse extraction and coding

Audio signal components can roughly be divided into three categories: tones, noise and transients. Transients are characterized by a short duration and by higher energy than its temporal surrounding. Since duration is relative to the resolution of a time-frequency analysis, it will be generally considered here that a transient is a signal component shorter than the MDCT window. Transients can also be periodically repeating. This is the case in low-pitched speech or singing, where the signal may be considered as filtered train of glottal pulses. Transients have a compact time representation, but a broad spectral representation and thus are problematic for coding in the MDCT domain.

Traditional MDCT-based codecs (e.g. MP3 [77], AAC [76]) use switching to short blocks [83] and TNS [84] for handling transient signals. Nevertheless, there are problems with these techniques. Time domain aliasing in the MDCT significantly limits TNS. TNS operates as a linear prediction filter in the MDCT domain. It temporarily flattens a transient signal in the encoder and then reintroduces the original temporal envelope in the decoder. The TNS filtering doesn't affect perfect reconstruction in absence of quantization noise. However, at low bitrates there is usually high level of quantization noise. Frames are independently quantized and coded and consequently the noise is uncorrelated between the frames. Quantization noise in the overlap region, that is shaped by TNS, cannot be canceled out by TDAC. The unfolding of the inverse MDCT duplicates the temporally shaped noise burst and creates transients not existing in the original signal. The addition of originally non-existing transients has severe effect on the perceived coding quality. Adapting the overlap, so that a transient is located in non-overlapping region [128], reduces the problem of the unfolding and TNS. Yet shorter overlap degrades performance for coding tonal parts of the signal. Short blocks even more deteriorate signals that are both harmonic and transient. Because of this, both methods are very limited for modelling a train of glottal pulses in low-pitched speech.

We make a distinction between a pulse and an impulse. An example of a pulse is a glottal pulse [16, 51, 68, 131]. Any other kind of transient is also regarded as a pulse. Each pulse has a characteristic waveform, which will be called pulse waveform [Figure 5.17], and it has an associated spectrum or spectrogram [Figure 5.15]. An impulse is the unit sample or the unit impulse [3] together with its sign. This terminology is not followed when describing the state of the art; instead the terminology from the referenced state of the art is used. In the

terminology of ACELP, pulse is often used to refer to the unit impulse used in the algebraic codebook [20]. In [132] the terms pulse and impulse are interchangeably and inconsistently used for unit impulses in ACELP and for an impulse-like portion.

The existing methods are first analyzed, followed by a description of the used algorithm with new or modified methods in the pulse extraction and coding. The pulse coding is combined with an MDCT codec and its advantage is demonstrated for both speech and music signals.

5.1 State of the art

An algorithm for the detection and extraction of transient signal components is presented in [133]. The magnitude spectrogram is obtained via the MCLT with short blocks, forming the basis for the time frequency analysis. A temporal envelope is generated for each band in the magnitude spectrogram, using only the magnitudes of the band. Within a so-called time viewport of one time frequency tile, the onset start and end tiles are found, hence also defining the onset duration. Additionally, a weighting factor is calculated for the tile, using the time envelope. Using the onset durations and the weighting factors of the tile and its 14 neighboring tiles at the same time instance, a relative onset duration is calculated for the tile. The tiles between the onset start and the end, for which the relative onset duration is below a threshold, are marked as transient tiles. The transient tiles of the MDCT part of the MCLT spectrogram are combined into a separate signal. The extraction of the transients is achieved by multiplying the MDCT coefficients with cross fade factors, where the cross fade factors are calculated based on the temporal envelope and the transient decision. The coding of the transients is done in the MDCT domain. This saves the additional inverse MDCT to calculate the transient time signal. The encoded transient signal is decoded and the resulting time domain signal is subtracted from the original signal. The residual is coded with a transform-based audio coder, using only long blocks. Masking curve is shared between the transient and residual coder, while each coder has its own global quantization step size. No solution for the bitrate distribution between the transient and the residual coder was provided by the authors.

Since only onsets are considered in the algorithm from [133], transient events like glottal pulses would not be detected or would be inefficiently coded. By using linear magnitude spectrum and by using separate envelopes for each band, broad-band transients may be missed in a presence of background noise/signals.

In [132] an audio encoder includes an impulse extractor for extracting an impulse-like portion from an audio signal. A residual signal is derived from the original audio signal so that the impulse-like portion is reduced or eliminated in the residual audio signal. The impulse-like portion and the residual signal are encoded separately and both are transmitted to the decoder where they are separately decoded and combined. Open- and closed-loop approaches are presented. In the open-loop approach, the impulse-like portion and the residual are coded separately. In the closed-loop approach, the impulse-like portion is first coded, for example using a modified ACELP, and the decoded impulse-like portion is removed from the original audio signal. The expected advantage is that the quantization error of the impulse encoder is compensated by the residual encoder. In the open-loop approach, the quantization errors are independent and add up in the decoder. The advantage is that the impulse-like portion

quantization error does not affect the residual signal. Peaks that determine the impulse-like portion are found in the time domain or in an LP residual. As usual in patents, [132] tries to achieve wide scope of protection and lacks many algorithm details. For an example, it is written in [132] that "the determination, whether the audio signal is stationary or impulse-like, can also be performed frequency-selective so that a certain frequency band or several certain frequency bands are considered to be stationary and other frequency bands are considered to be impulse-like. In this case, a certain time portion of the audio signal might include an impulse-like portion and a stationary portion". We could not find further details in [132] on how to perform it the signal classification frequency-selectively, which in our opinion is important for the pulse extraction. It is further written in [132] that "an impulse, starting from some samples to the left of the peak and ending at some samples to the right of the peak, is picked out from the signal" which means that a hard cut off is used for extracting the impulse-like portion. Additionally, an impulse characteristic enhancement may process the peaks so that each peak has the same height and shape. The residual signal is obtained, so that it is stationary and that it does not have temporal holes. The holes may be produced because of the hard cut off for extracting the impulse-portion. For removing the holes, time-variant scaling or interpolation of the time domain signal at the location of the holes is proposed. This synthetized signal at the hole location may be then subtracted from the impulse-like portion.

In [132] any error introduced in the closed-loop approach or by performing the impulse characteristic enhancement is accounted for in the residual coder. Since the impulse characteristic enhancement processes the peaks so that each peak has the same height and shape, this leads to the error containing differences between the impulses and these differences have transient characteristics. Such errors with transient characteristics are not well suited for the residual coder, which expects stationary signal and it is not clear how exactly are such remaining transient characteristics removed from the original audio signal. Considering a signal consisting of a superposition of a strong stationary signal and a small transient, since all samples at the location of the peak are kept and all samples between peaks are removed, it means that the impulse will contain the small transient and a time-limited part of the strong stationary signal. The residual will have a discontinuity at the location of the transient that is not possible to remove by the proposed scaling nor interpolation in the time domain. Subtraction of such synthetized signal from the impulse-like portion also doesn't produce a well-defined transient signal. For such signal neither the "impulse-like" signal is suited for the impulse coder nor is the "stationary residual" suited for the residual coder.

The closed-loop approach from [132] was further investigated in [116]. In [116] the input signal is high-pass filtered with 50 Hz cutoff to remove frequencies not relevant for speech. Further, the natural tilt of speech is removed via a pre-emphasis filter. The pre-emphasized signal is filtered through an LP analysis filter, therefore spectrally flattening the signal and producing a residual signal, that models a glottal excitation. Pitch lag is estimated in segments of 8 ms. Depending on the pitch estimation, the segment is classified as voiced or unvoiced. For voiced segments, pulse positions are found by peak picking, followed by adapting the peak positions to the detected pitch using dynamic programming. There are no details on the peak picking, but based on the provided examples it seems that peak picking is simple finding of local maxima in the residual TD signal. Bounds of pitch cycles are derived from the pulse positions. Waveform of each pitch cycle is cut between the detected pulses positions. The

pitch cycle waveforms are energy normalized and upsampled to 16 ms length. The last pitch cycle in the frame is the prototype waveform. Only the prototype waveform, being the last pitch cycle in the frame, is coded and transmitted. The coded and transmitted parameters are last pitch cycle's original length, position, energy and an FD representation of the normalized and upsampled waveform. A continuous speech signal is constructed using interpolation of two consecutive prototype waveform parameters. One additional frame delay is needed for constructing the end of the current output frame. The original input signal is time warped to align it with the constructed speech signal. The constructed continuous speech signal is subtracted from the time warped original signal to obtain the residual. The residual signal is coded with an FD codec. To further improve quality the prototype waveforms are low-pass filtered by coding only the first 4 kHz of the spectrum. Again, as in [132], any quantization error in the coding of voiced segments will affect the performance of the FD residual coding. Additionally, the warping further deteriorates the coding quality. The approach in [116] describes only how to code voiced segments in speech based on the pulse detection. Because of the classification of segments as voiced depending on pitch, onsets of voiced segments can be missed or additional delay is needed. Further limitation is that the complete signal in a voiced segment is coded via the prototype waveforms; there is no explicit consideration for signals where the pulse and stationary portion occur at the same time instance.

In [134] High Resolution Envelope Processing (HREP) is proposed that works as a pre-processor that temporally flattens the signal for high frequencies. At the decoder side, it works as a post-processor that temporally shapes the signal for high frequencies using the side information.

In [135] the original and the coded signal are decomposed into semantic components (i.e., distinct transient clap events and more noise-like background) and their energies are measured in several frequency bands before and after coding. Correction gains derived from the energy differences are used to restore the energy relations of the original signal by post-processing via scaling of the separated transient clap events and noise-like background signal for band-pass regions. Pre-determined restauration profiles are used for the post-processing.

The methods in [134] and [135] don't consider separately coding transient events and thus don't use any advantage that a specialized codec for transients and a specialized codec for residual/stationary signals could have.

In [136] a harmonic-percussive-residual separation using structure tensor on log spectrogram is presented. Nonetheless, the paper doesn't consider either audio or speech coding.

How to efficiently code pulses in the open-loop approach could not be found in the state of the art. Speech specialized methods, such as CELP, are not suited as they don't take into account that the decoded pulses and the residual need to stay in sync.

5.2 Principles of the implemented pulse extraction and coding

In Figure 5.1 a spectrogram of the sentence "Die Natur hat dem Menschen eine Zunge, aber zwei Ohren gegeben" spoken by a German male speaker is shown. The spectrogram has high frequency resolution with 23.4 Hz width of the frequency bins. A harmonic structure,

characterized by horizontal lines, is visible for vowels. This harmonic structure requires many bits for coding in the MDCT domain, because of its low fundamental frequency and its variations. Because of psychoacoustic principles, an MDCT codec would mostly code only low frequencies, eventually filling high frequencies with noise.

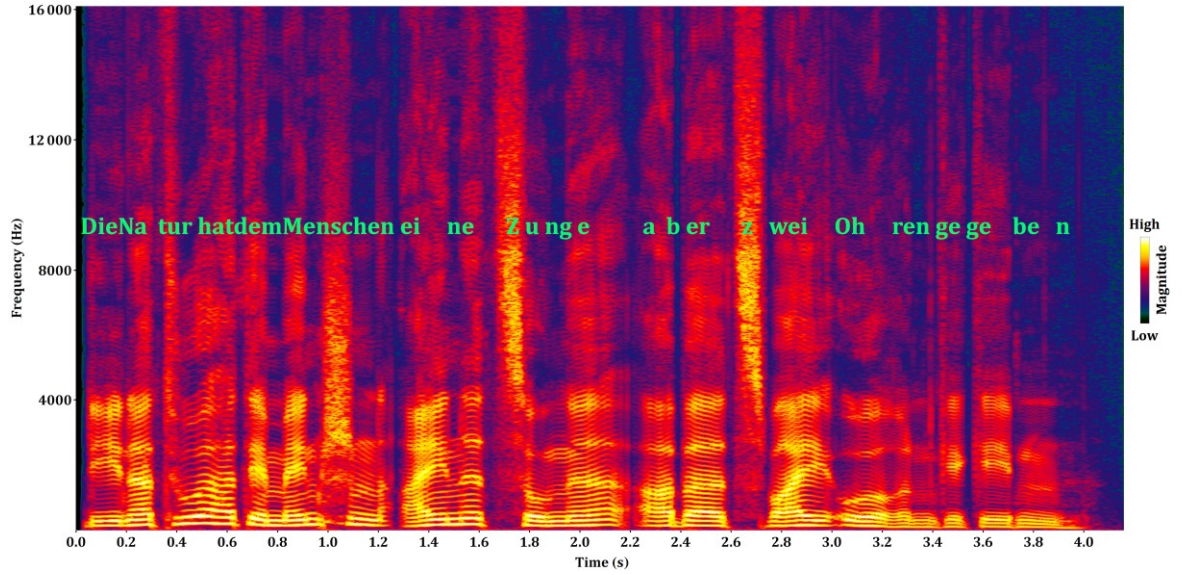

Figure 5.1 Spectrogram of a German male speaker recording [ger_m] at 23.4 Hz frequency resolution

Increasing the time resolution, it becomes visible that the harmonic portions consist of train of pulses. This is shown in the spectrogram of a portion of the sentence in Figure 5.2, where the signal is high-passed. The resonance of the vocal tract is strong at low frequencies and the high-pass makes the pulse structure easier to notice across the whole bandwidth.

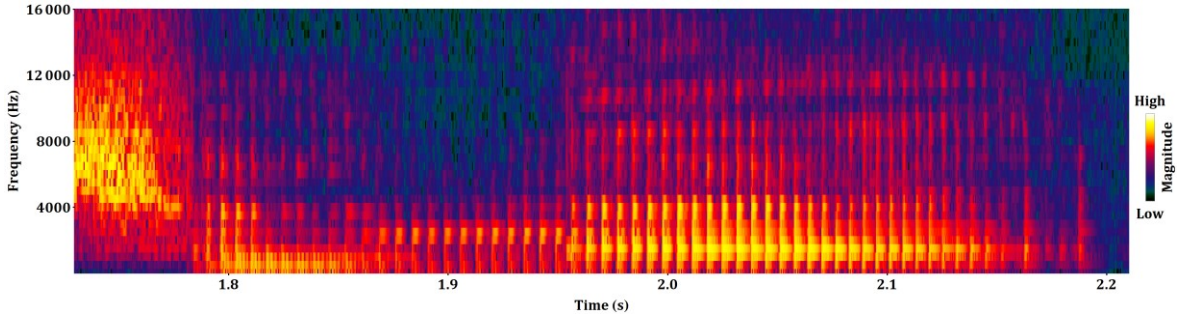

Figure 5.2 Spectrogram of the German male speaker recording at 0.5 ms time resolution

It would be good here to notice that there is a lot of variability in the glottal pulse structure and that its transient nature is not visible at all frequencies. Looking only at one frequency band it would be hard to detect this train of pulses. The coherence of the pulses across the frequencies give rise to the comodulation masking release [55] and are probably the reason why humans are sensitive to quantization noise at the location of vowels. Based on these observations a pulse extraction was developed.

Pulses are found and extracted from a high-pass filtered input signal. A waveform consisting of extracted non-quantized pulses is subtracted from the original TD signal before processing the residual of the subtraction with the MDCT.

The extracted pulses are semi-parametrically coded and the waveform constructed from the coded pulses is added to the decoded residual signal. The pulse coding includes a predictive coding.

The pulse extraction and coding follows the MDCT codec framing of 20 ms. Up to 8 pulses per frame are extracted and coded. Up to 3 pulses ($N_{P_P} \leq 3$) from the previous frames (called past pulses) are kept in a memory and used in the extraction and the predictive coding.

5.3 Pulse extraction

The Figure 5.3 shows the process of separating the input audio signal into the pulses and the residual. The details are described in the following sub-chapters.

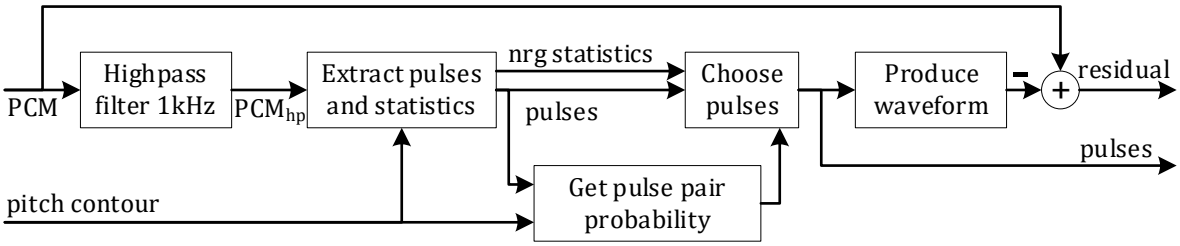

Figure 5.3 Pulse extraction

5.3.1 Detecting pulse candidates

The input TD signal is high-pass filtered using a symmetric FIR filter with -6 dB cutoff at 1 kHz. Time-frequency analysis via the STFT is used for finding and extracting pulses. The high-pass filtering may also be implemented in the STFT as the element-wise product. The high-passed TD signal is windowed using 2 ms long Hann window (also known as the raised sine window) with 75 % overlap and transformed via the DFT into the frequency domain. This windowing scheme offers better time resolution than 50 % overlap sine window and still fulfills the least squares error criteria [137]. It also provides smoother borders for waveforms synthetized from extracted pulses and this is required as, in real life signals, the borders of pulses are often not clearly defined. The high-pass cutoff and the STFT configuration were chosen based on observations of speech spectrograms.

With this configuration there are 40 points for each STFT frequency band in each frame of 20 ms, each point consisting of a magnitude and a phase. Each frequency band is 500 Hz wide and we are considering only 49 bands for the sampling rate $F_S = 48$ kHz, because the remaining 47 bands may be constructed via the symmetric extension. Thus, there are 49 points in each time instance of the STFT and $40 \cdot 49$ points in the time-frequency plane of a frame. The STFT hop size is $H_P = 0.0005 F_S$.

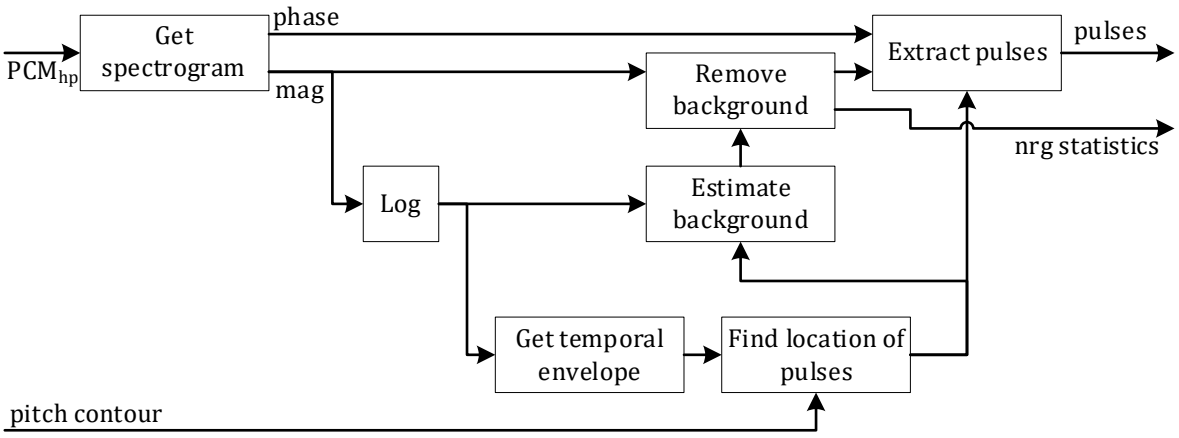

Figure 5.4 Details of the block "Extract pulses and statistics" from Figure 5.3

Extraction of pulse candidates and associated statistics is done in the block "Extract pulses and statistics" shown in Figure 5.3 and detailed in Figure 5.4. Temporal envelope is obtained from the log magnitude spectrogram by integration across the frequency axis. In other words, log magnitudes are summed up for each time instance of the STFT to obtain one sample of the temporal envelope. The temporal envelope may be hence considered a TD signal with the sampling rate of 2 kHz. Smoothed temporal envelope is a low-pass filtered version of the temporal envelope using a 4th order symmetrical low-pass FIR filter.

The sample Pearson's correlation coefficient ρ_{e_T} of the temporal envelope and its delayed version for $4 \leq m \leq 12$ is calculated together with its thresholded peak $\hat{\rho}_{e_T}$:

$$\rho_{e_T}[m] = \frac{\sum_{n=0}^{40} e_T[n] e_T[n-m]}{\sqrt{(\sum_{n=0}^{40} e_T[n] e_T[n])(\sum_{n=0}^{40} e_T[n-m] e_T[n-m])}}$$

$$\hat{\rho}_{e_T} = \begin{cases} \max\limits_{5 \leq m \leq 11} \rho_{e_T}[m] & , \max\limits_{5 \leq m \leq 11} \rho_{e_T}[m] > 0.65 \\ 0 & , \max\limits_{5 \leq m \leq 11} \rho_{e_T}[m] \leq 0.65 \end{cases}$$

where e_T is the temporal envelope after mean removal. The exact delay for the maximum $D_{\rho_{e_T}}$ is estimated using Lagrange polynomial of 3 points forming the peak $\hat{\rho}_{e_T}$ in the normalized autocorrelation. The threshold 0.65 was chosen based on temporal envelope plots [Figure 5.5] of few speech samples, so that expected accuracy of the pulse extraction is increased.

Expected average pulse distance $\widetilde{D}_P$ is estimated from the normalized autocorrelation of the temporal envelope and the average pitch lag $\bar{d}_{F_0}$ in the frame:

$$\widetilde{D}_P = \begin{cases} D_{\rho_{e_T}} & , \hat{\rho}_{e_T} > 0 \\ \min\left(\frac{\bar{d}_{F_0}}{H_P}, 13\right) & , \hat{\rho}_{e_T} = 0 \wedge \bar{d}_{F_0} > 0 \\ 13 & , \hat{\rho}_{e_T} = 0 \wedge \bar{d}_{F_0} = 0 \end{cases}$$

where $\widetilde{D}_P$ is set to 13 (corresponding to 6.5 ms) for the frames with low harmonicity.

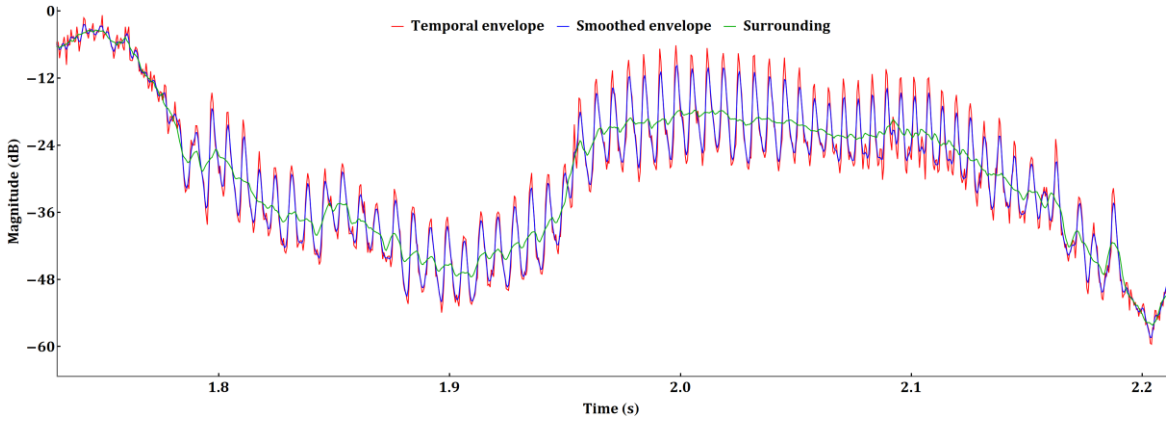

Figure 5.5 Temporal envelope at the observed segment of the German male speaker recording

Positions of the pulses are local peaks in the smoothed temporal envelope with the requirement that the peaks are above their surroundings. The surrounding is defined as the low-pass filtered version of the temporal envelope using a simple moving average filter with adaptive length; the length of the filter is set to the half of the expected average pulse distance $\widetilde{D}_P$. An example of the temporal envelope, the smoothed temporal envelope and the surrounding, for the segment of the German male speaker recording shown in Figure 5.2, is given in Figure 5.5. The limiting value 13 for $\widetilde{D}_P$ was chosen as it provides good results for low pitched male voice and single pulses, while still limiting the required delay for the moving average filter. The exact pulse position $\dot{t}_{P_i}$ is estimated using Lagrange polynomial of 3 points forming the peak in the smoothed temporal envelope. The pulse center position t_{P_i} is the exact position rounded to the STFT time instances and so the distance between the center positions of pulses is a multiple of 0.5 ms. It is considered that each pulse extends 2 time instances to the left and 2 to the right from its center position and therefore a waveform of 4 ms is associated with each pulse.

Up to 8 pulses per 20 ms are found; if more pulses are detected then smaller pulses are disregarded. Noise (e.g. sibilants) could be misdetected as pulses. The number of found pulses is denoted as N_{P_X}. The i^{th} pulse is denoted as P_i. To discard misdetected pulses two steps follow. In the first step the pulse candidates are extracted from the STFT, so that a stationary portion (including stationary noise) is reduced in the extracted pulses. In the second step features describing relation between pulses and relation of pulses to their surrounding are calculated. Finally, true pulses are chosen based on the features, as described in 5.3.2.

Magnitudes are enhanced based on the pulse positions so that the enhanced STFT consists only of the pulses [Figure 5.7]. Background of a pulse, which is the stationary portion at the location of the pulse, is estimated as the linear interpolation of the left and the right background, where the left and the right background are mean of the 3rd to 5th time instance away from the center position. The background of a pulse is estimated in the log magnitude domain and removed by subtracting it in the linear magnitude domain. Magnitudes in the enhanced STFT are in the linear scale. The background removal is done within each STFT frequency band. The phase is not modified. All magnitudes in the time instances not belonging to a pulse (at least 3 time instances away from the pulse center) are set to zero.

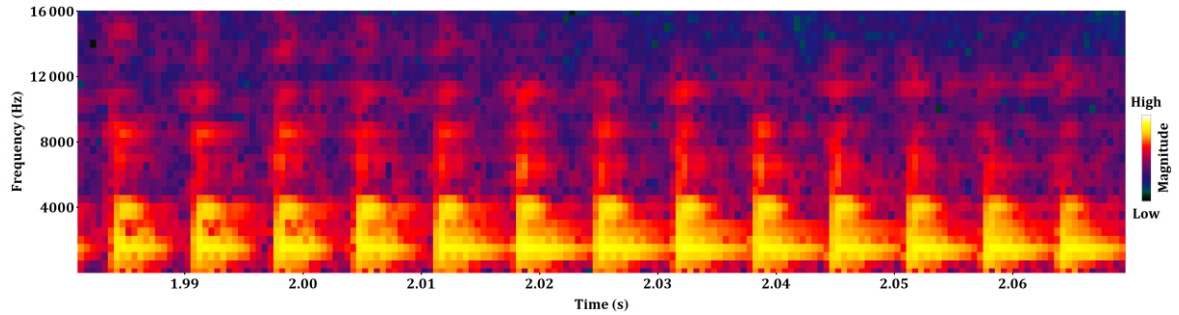

Figure 5.6 The vowel “e” within the German male speaker recording

The background is estimated only within each STFT frequency band, there is no averaging across different frequency bands, to avoid reducing already faint pulse structure at high frequencies. The noisy nature of pulses at high frequencies can be seen in Figure 5.6. The spectrograms in sections 5.3 and 5.4 are obtained from the STFT produced within IVA, as explained in 5.3.1.

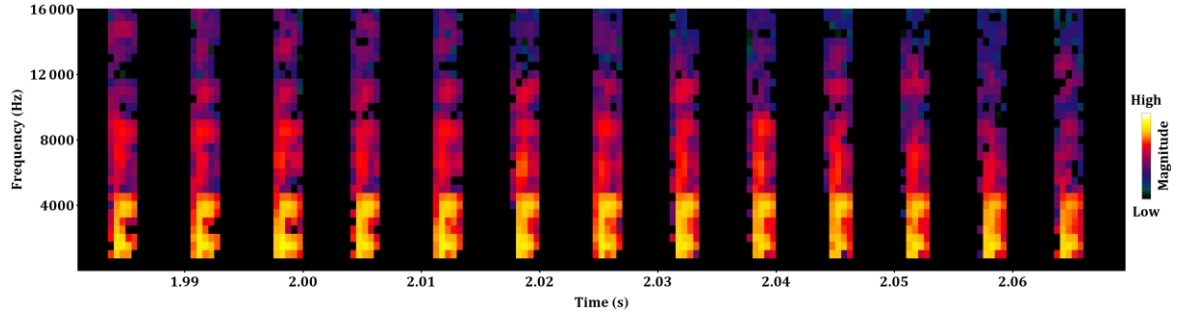

Figure 5.7 Enhanced spectrogram of the vowel "e"

Pulses appear also in audio signals produced by musical instruments (even wind instruments). An example is shown in Figure 5.8.

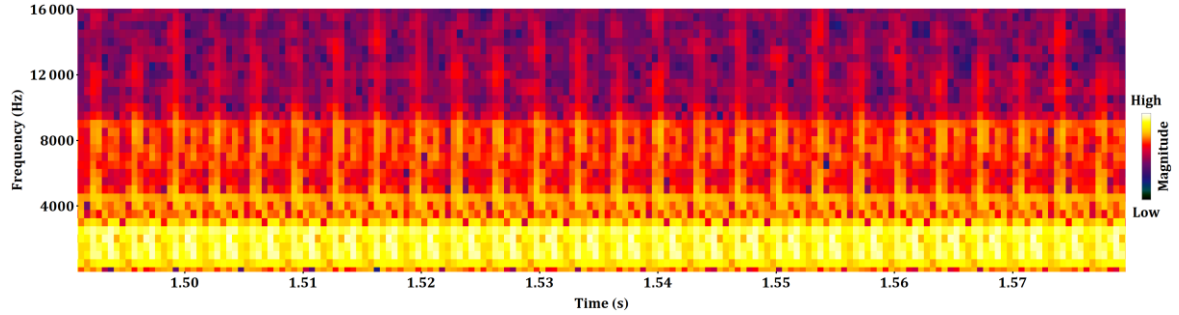

Figure 5.8 Note D4 played by pitch pipe

As it can be seen in the example, signals with higher pitch exhibit pulse structure at higher frequencies. This was encountered in many other examples. One also needs to consider that sparsity of a harmonic signal spectral presentation is proportional to its fundamental frequency. The MDCT codec is able to efficiently code spectra for higher pitched signals, as the total energy is concentrated in a small number of coefficients. To take all this into account, frequencies below a starting frequency, proportional to a pulse distance, are zeroed out in the enhanced STFT. The calculation of the starting frequency f_{P_i} is proposed in the following text.

The average pulse distance is defined as:

$$\bar{D}_P = \begin{cases} \widetilde{D}_P & ,\hat{\rho}_{e_T} > 0 \vee \bar{d}_{F_0} > 0 \\ \min\left(\frac{40}{N_{P_X}}, 13\right) & ,\hat{\rho}_{e_T} = 0 \wedge \bar{d}_{F_0} = 0 \end{cases}$$

where the expected average pulse distance $\widetilde{D}_P$ is used if it can be reliably calculated and otherwise the number of found pulses N_{P_X} is used.

The start frequency of a pulse is proportional to the inverse of the average pulse distance in the frame, but limited between 750 Hz and 7250 Hz:

$$f_{P_i} = \min\left(\left\lfloor 2\left(\frac{13}{\bar{D}_P}\right)^2 + 0.5 \right\rfloor, 15\right)$$

The start frequency f_{P_i} is expressed as the index of an STFT band.

The change of the starting frequency in consecutive pulses is limited to 500 Hz (one STFT band). Magnitudes below the starting frequency are set to zero.

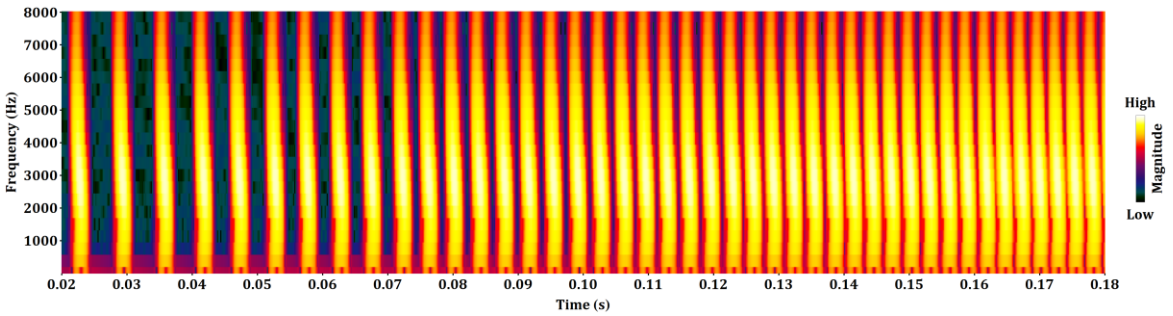

Figure 5.9 An input signal with pulses having decreasing distance

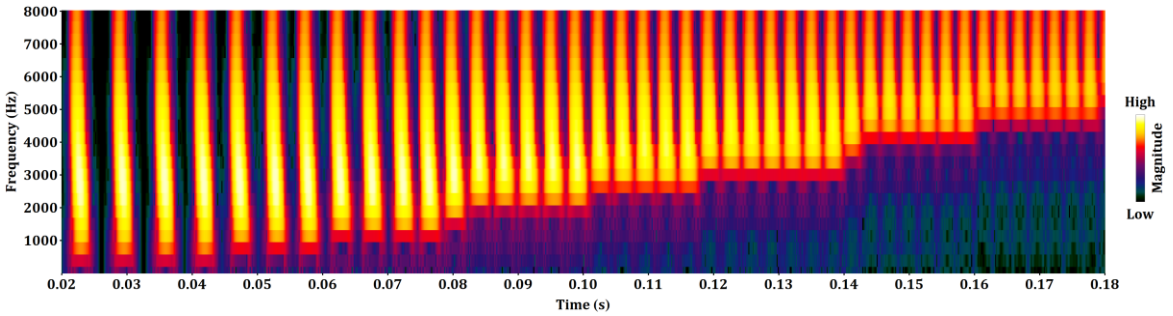

Figure 5.10 The extracted pulses from the original signal in Figure 5.9

The dependence of the starting frequency on the pulse distance is demonstrated in the figures above. Figure 5.9 shows an input signal for the pulse extraction algorithm and Figure 5.10 shows pulses extracted by the algorithm. The distance between the pulses in this example is 6.5 ms at the beginning of the signal and decreases to 2.75 ms. The value of $\bar{D}_P$ is 13 and f_{P_i} is 2 for the distance of 6.5 ms. The value of $\bar{D}_P$ is 5.5 and f_{P_i} is 11 for the distance of 2.75 ms. The starting frequency in a spectrum of the enhanced STFT is 750 Hz when f_{P_i} is 2, as can be seen at 0.02s-0.04s in Figure 5.10. The starting frequency is 5250 Hz when f_{P_i} is 11, as can be seen at 0.16s-0.18s in Figure 5.10.

The waveform of each pulse is obtained from the enhanced STFT. The pulse waveform is non-zero in 4 ms around its center and the pulse length is $L_{W_P} = 0.004F_S$ (the sampling rate of the pulse waveform is equal to the sampling rate of the input signal F_S). The symbol x_{P_i} represents the waveform of the i^{th} pulse. Negative pulse indexes point to the past pulse (the N_{P_P} pulses kept in the memory).

Each pulse P_i is uniquely determined by the center position t_{P_i} and the pulse waveform x_{P_i}. The pulses are aligned to the STFT grid.

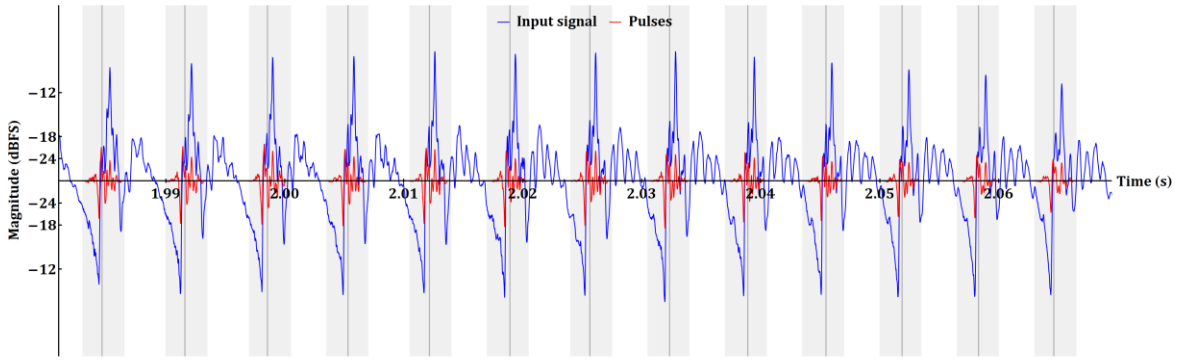

Figure 5.11 Waveform of the vowel "e" and the extracted pulse waveforms

In Figure 5.11, the non-zero samples of a pulse are marked with gray bars. The exact positions $\dot{t}_{P_i}$ are marked with vertical lines.

5.3.2 Pulse selection refinement

At this point there may be a significant number of pulse misdetections. The idea is that transients with significant percentage of the frame energy should be detected, as well as pulses that are correlated to each other. The pulse correlation is very important for coding train of pulses, as significant reduction in bit demand is achieved through the pulse prediction used in the coding. To remove wrongly detected pulses, a number of features are obtained.

Following features are calculated for each pulse:

- percentage of the local energy in the pulse p_{E_L,P_i}
- percentage of the frame energy in the pulse p_{E_F,P_i}
- percentage of bands with the pulse energy above the half of the local energy within the STFT frequency band p_{N_E,P_i}
- correlation ρ_{P_i,P_j} and distance d_{P_i,P_j} between each pulse pair (among the pulses in the current frame and the N_{P_P} last coded pulses from the past)
- pitch lag at the exact location of the pulse d_{P_i}

The local energy is calculated from the 11 time instances around the pulse center in the original STFT, obtained using the hop size of 0.5 ms. All values used for the features are calculated only above the start frequency.

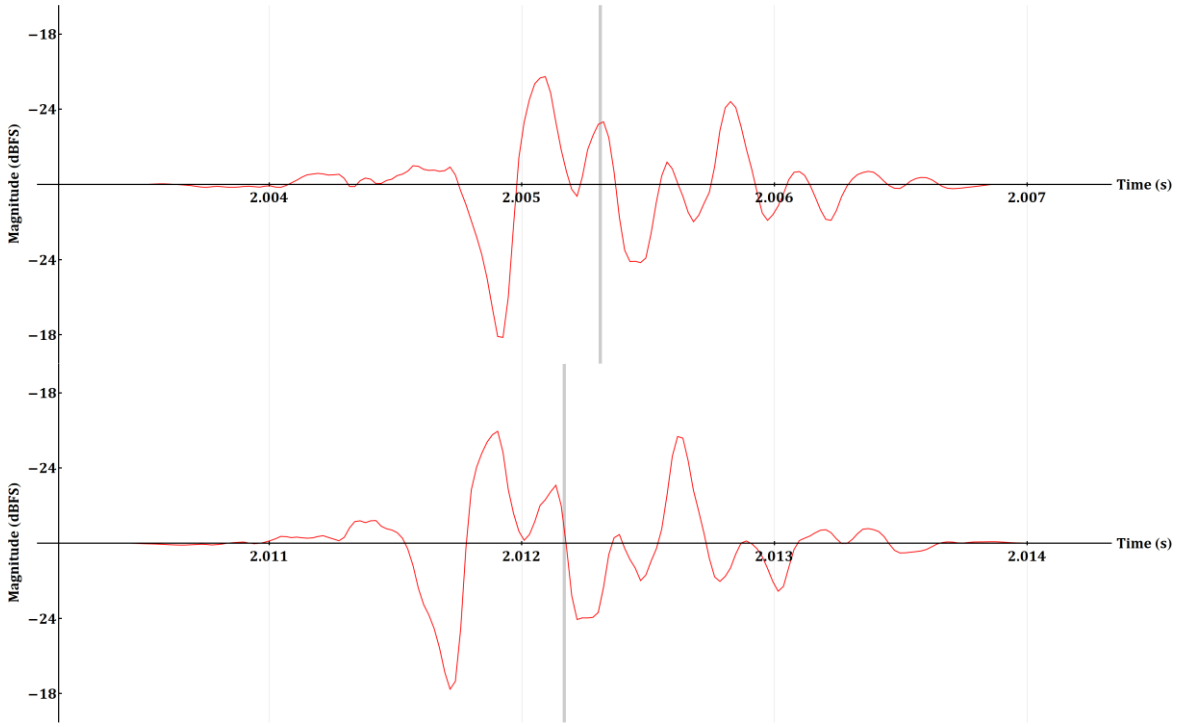

Figure 5.12 Waveforms of two consecutive pulses in the vowel “e”

The exact pulse position $\dot{t}_{P_i}$ is just an estimate and prone to errors due to noise in the input signal (see Figure 5.12, where $\dot{t}_{P_i}$ are marked with vertical lines). Thus, it is not adequate to use it for calculating the distance between pulses. Instead, the distance between a pulse pair d_{P_j,P_i} is derived from the location of the maximum cross-correlation between pulses. The cross-correlation is windowed with the 2 ms long rectangular window and normalized by the L^2 norm of the pulses (windowed with the same window). The pulse correlation ρ_{P_j,P_i} is the maximum of the sample Pearson’s correlation coefficient $(x_{P_i} * x_{P_j})$ for $0 \leq m < L_{W_P}/4$:

$$\left(x_{P_i} * x_{P_j}\right)[m] = \frac{\sum_{n=l}^{L_{W_P}-l} x_{P_i}[n]x_{P_j}[n+m]}{\sqrt{\left(\sum_{n=l}^{L_{W_P}-l} x_{P_i}[n]x_{P_i}[n]\right)\left(\sum_{n=l}^{L_{W_P}-l} x_{P_j}[n+m]x_{P_j}[n+m]\right)}}$$

$$\rho_{P_j,P_i} = \begin{cases} \max\limits_{-l\leq m\leq l}\left(x_{P_i} * x_{P_j}\right)[m], i < j \\ \max\limits_{-l\leq m\leq l}\left(x_{P_j} * x_{P_i}\right)[m], i > j \\ 0, i = j \end{cases}$$

$$\Delta_{\rho_{P_j,P_i}} = \begin{cases} \operatorname{argmax}\limits_{-l\leq m\leq l}\left(x_{P_i} * x_{P_j}\right)[m], i < j \\ -\operatorname{argmax}\limits_{-l\leq m\leq l}\left(x_{P_j} * x_{P_i}\right)[m], i > j \\ 0, i = j \end{cases}$$

$$d_{P_j,P_i} = \left|t_{P_j} - t_{P_i} + \Delta_{\rho_{P_j,P_i}}\right| = \left|t_{P_i} - t_{P_j} + \Delta_{\rho_{P_i,P_j}}\right|$$

$$l = \frac{L_{W_P}}{4}$$

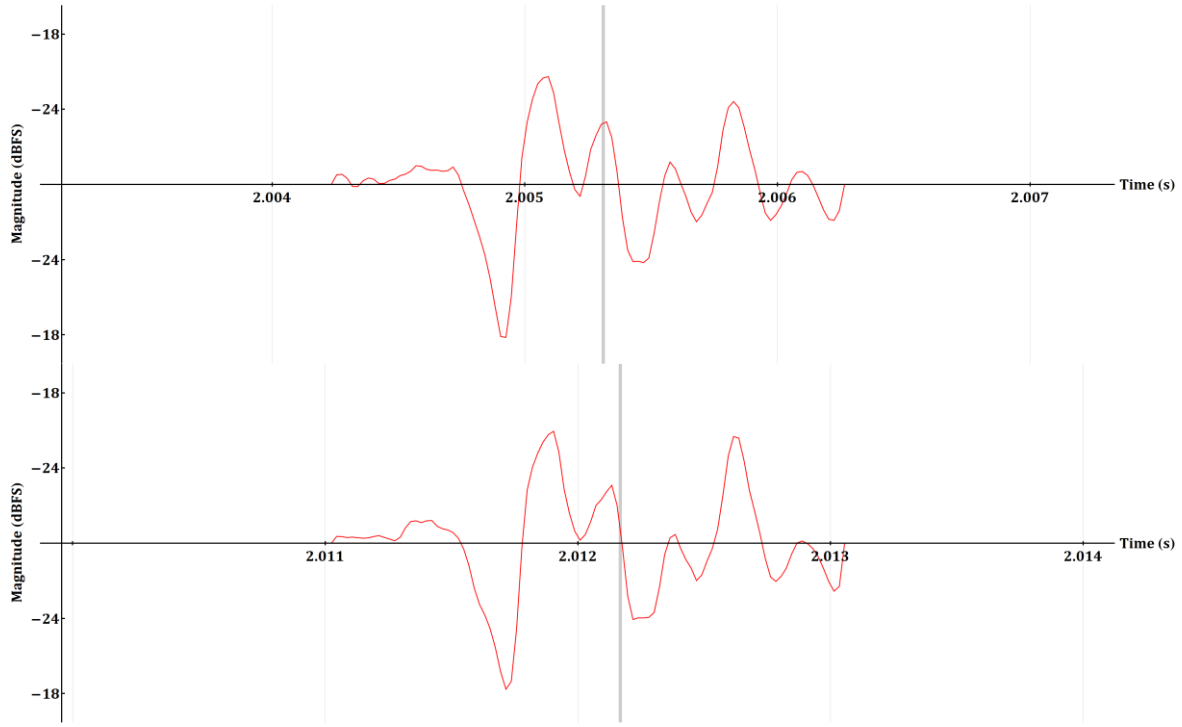

Figure 5.13 Waveforms of the two consecutive pulses in the vowel "e" aligned based on d_{P_j,P_i}

In Figure 5.13, the plot of the second pulse j is shifted for $\Delta_{\rho_{P_j,P_i}} = -0.00021F_S$ and the pulses are windowed with the 2 ms rectangular window. It is obvious that the exact pulse positions $\dot{t}_{P_i}$ (shown as vertical lines) are not aligned for the optimal pulse offset.

Error between the pitch and the pulse distance ϵ_{P_i,P_j} is calculated as:

$$\epsilon_{P_i,P_j} = \epsilon_{P_j,P_i} = \min\left(\min_{1\le k\le 6}\frac{\left|k\cdot d_{P_j,P_i} - d_{P_j}\right|}{H_P}, \min_{1\le k\le j-i}\frac{\left|d_{P_j,P_i} - k\cdot d_{P_j}\right|}{H_P}\right), i<j$$

Introducing multiple of the pulse distance ($k \cdot d_{P_j,P_i}$) in the formula for ϵ_{P_i,P_j}, octave errors in the pitch estimation are taken into account. Introducing multiples of the pitch lag ($k \cdot d_{P_j}$) solves missed pulses coming from imperfections in pulse trains: if a pulse in the train is distorted or there is a transient not belonging to the pulse train, that inhibits detection of a pulse belonging to the train.

If $\epsilon_{P_i,P_j} = 1$, it means that there is a discrepancy of 0.5 ms between the pulse distance and the pitch lag. For a signal with the fundamental frequency of 200 Hz, $\epsilon_{P_i,P_j} = 1$ means that there is an error of 10%.

Pulse pair probability p_{P_i,P_j} is a probability that the i^{th} and the j^{th} pulse belong to a train of pulses. It is calculated based on the pulse correlation ρ_{P_j,P_i} and the error between the pitch and the pulse distance ϵ_{P_i,P_j}:

$$p_{P_i,P_j} = p_{P_j,P_i} = \begin{cases} \min\left(1, \dfrac{{\rho_{P_j,P_i}}^2}{\sqrt[10]{\max\left(0.2, \epsilon_{P_i,P_j}\right)}}\right) & , -N_{P_P} \le j < 0 \le i < N_{P_X} \\ \min\left(1, \dfrac{\rho_{P_j,P_i}}{2 \cdot \sqrt{\max\left(0.1, \epsilon_{P_i,P_j}\right)}}\right) & , 0 \le i < j < N_{P_X} \end{cases}$$

Probability $\dot{p}_{P_i}$ of a pulse with a relation only to the already coded past pulses is defined, based on the pulse pair probability p_{P_j,P_i} and the percentage of the frame energy in the pulse p_{E_F,P_i}, as:

$$\dot{p}_{P_i} = \left(1 + \max_{-N_{P_P} \le j < 0} p_{P_j,P_i}\right) p_{E_F,P_i}$$

Probability of a pulse p_{P_i} to be true pulse, i.e. not a misdetection, is iteratively found:

1. All pulse probabilities p_{P_i} $(0 \le i < N_{P_X})$ are set to 1
2. In the increasing order of pulses, for each pulse that is still probable ($p_{P_i} > 0$):
    a. Probability $\ddot{p}_{P_i}$ of the i^{th} pulse belonging to a train of the pulses in the current frame is calculated from the pulse pair probabilities p_{P_j,P_i} and the percentage of the frame energy in the pulse p_{E_F,P_i}:

$$\ddot{p}_{P_i} = \left(\sum_{j=0}^{i-1} p_{P_j} \cdot p_{P_j,P_i} + \sum_{j=i+1}^{N_{P_X}-1} p_{P_j} \cdot p_{P_j,P_i}\right) p_{E_F,P_i}$$

    b. The initial probability that it is truly a pulse is then:

$$p_{P_i} = \dot{p}_{P_i} + \ddot{p}_{P_i}$$

    c. The probability is increased for pulses with the energy in many STFT frequency bands above the half of the local energy within the band p_{N_E,P_i}:

$$p_{P_i} = \max\left(p_{P_i}, \min\left(p_{N_E,P_i}, 1.5 \cdot p_{P_i}\right)\right)$$

    d. The probability is limited by the peak temporal envelope correlation $\hat{\rho}_{e_T}$ and the percentage of the local energy in the pulse p_{E_L,P_i}:

$$p_{P_i} = \min\left(p_{P_i}, \left(1 + 0.4 \cdot \hat{\rho}_{e_T}\right) p_{E_L,P_i}\right)$$

    e. If the pulse probability is below a threshold, then its probability is set to zero and it is not considered anymore:

$$p_{P_i} = \begin{cases} 1 & , p_{P_i} \ge 0.15 \\ 0 & , p_{P_i} < 0.15 \end{cases}$$

3. The step 2 is repeated as long as there is at least one p_{P_i} set to zero in the current iteration or until all p_{P_i} are set to zero.

The constants and the formulas were heuristically found by an examination of a large dataset, consisting of both speech and music.

At the end of this procedure, there are N_{P_C} true pulses with p_{P_i} equal to one. All true pulses are coded. The pulses for which p_{P_i} is not equal to one are discarded. Among the true N_{P_C} pulses up to three last pulses are kept in memory for calculating ρ_{P_i,P_j} and d_{P_i,P_j} in the following frames. If there are less than three true pulses in the current frame, the newest pulses already in memory are kept. In total up to three pulses are kept in the memory. If the memory already contains three pulses, the oldest stored pulses are replaced by newly found ones. In other words, the number of past pulses N_{P_P} kept in memory is increased at the beginning of processing until $N_{P_P} = 3$ and is kept at 3 afterwards.

5.4 Pulse coding

The pulses are coded using:

- number of pulses in the frame N_{P_C}
- position within the frame t_{P_i}
- pulse starting frequency f_{P_i}
- pulse spectral envelope
- prediction gain $g_{P_{P_i}}$ and if $g_{P_{P_i}}$ is not zero:
    - index of the prediction source $i_{P_{P_i}}$
    - prediction offset $\Delta_{P_{P_i}}$
- innovation gain $g_{I_{P_i}}$
- innovation consisting of up to 4 impulses, each impulse coded by its position and sign

The number of pulses is encoded with Huffman codes.

The first pulse position t_{P_0} is coded absolutely using Huffman codes. For the following pulses the position deltas $\Delta_{P_i} = t_{P_i} - t_{P_{i-1}}$ are coded with Huffman codes. Depending on the number of pulses in the frame and the first pulse position, different Huffman codebooks are used.

The first pulse starting frequency f_{P_0} is coded absolutely using Huffman codes. The start frequencies of the following pulses is differentially coded. If there is a zero difference then all following differences are also zero, so the number of non-zero differences is coded. All the differences have the same sign, hence the sign of the differences can be coded with a single bit per frame. In most cases the absolute difference is at most one, thus a single bit is used for coding if the maximum absolute difference is one or bigger. At the end, only if the maximum absolute difference is bigger than one, all non-zero absolute differences need to be coded and they are unary coded.

For the remaining pulse parameters, a spectral envelope is determined and the pulses are spectrally flattened using the envelope. Coding in TD of spectrally flattened pulses follow. The process is depicted in Figure 5.14.

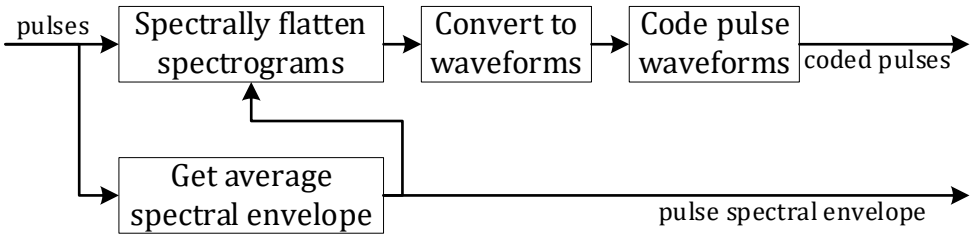

Figure 5.14 Pulse coding

5.4.1 Spectral flattening of pulses

All pulses in the frame use the same spectral envelope consisting of eight bands. Band border frequencies are: 1 kHz, 1.5 kHz, 2.5 kHz, 3.5 kHz, 4.5 kHz, 6 kHz, 8.5 kHz, 11.5 kHz, 16 kHz. Spectral content above 16 kHz is not explicitly coded. Spectral envelope in each time instance of a pulse is obtained by summing up the magnitudes in the envelope bands [Figure 5.15].

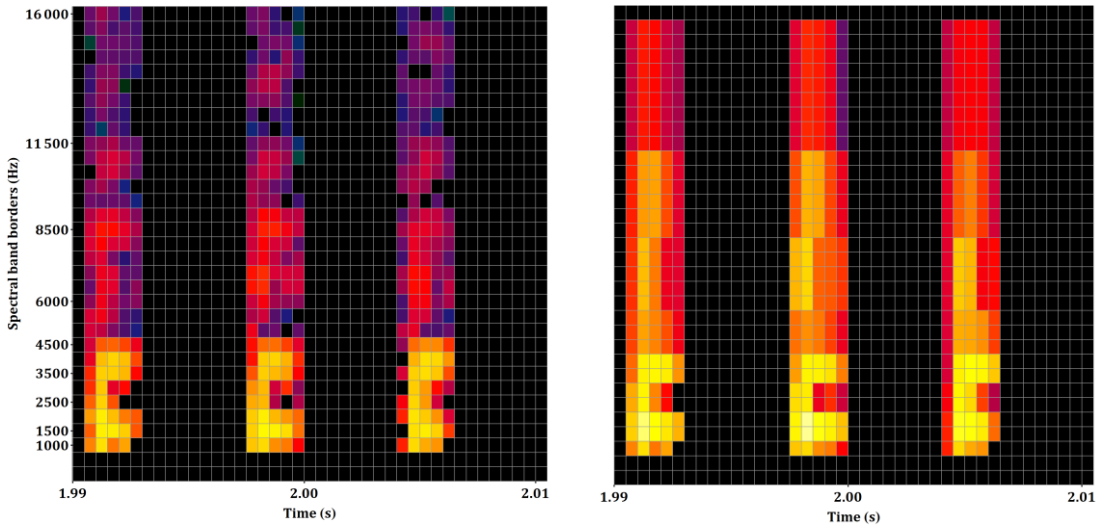

Figure 5.15 Pulses in a frame and envelopes in each time instance

The envelopes are averaged across all pulses in the frame. Points between the pulses in the time-frequency plane are not taken into account. The values are compressed using fourth root and the envelopes are vector quantized. The vector quantizer has 2 stages and the 2nd stage is split in 2 halves. Different codebooks exist for frames with $\bar{d}_{F_0} \neq 0$ [Table 5.1] and $\bar{d}_{F_0} = 0$ [Table 5.2] and for the values of N_{P_C} and f_{P_i}. Different codebooks require different number of bits. The first stage needs up to 7 bits and each split in the second stage up to 6 bits, in total up to 19 bits. For some combinations of $\bar{d}_{F_0}$, N_{P_C} and f_{P_i} no data was encountered on the training set.

$\bar{d}_{F_0} \neq 0$		f_{P_i}(kHz)													
		1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5
N_{P_C}	1	19	19	19	19	19	19	7	7	6	6	5	5	6	12
	2	19	19	19	19	19	7	6	5	5	5	4	4	3	5
	3	19	19	19	19	19	17	6	6	5	4	2	3	3	2
	4	19	19	19	19	19	19	7	6	5	3	3	2	3	3
	5	19	19	19	19	19	19	11	6	6	5	4	4	4	5
	6	7	5	3	7	19	19	19	13	7	5	4	4	4	5
	7	1	0	0	0	0	3	7	13	13	7	4	4	4	5
	8	0	0	0	0	0	0	0	0	3	5	3	3	5	5

Table 5.1 Number of bits for coding spectral envelope in frames with high harmonicity

$\bar{d}_{F_0} = 0$		f_{P_i}(kHz)													
		1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5
N_{P_C}	1	19	19	5	7	3	2	1	0	2	0	1	0	0	2
	2	19	19	5	5	3	1	1	1	0	1	0	0	0	0
	3	19	19	4	5	2	1	2	1	0	1	0	0	0	0
	4	6	7	4	5	2	3	1	0	0	0	0	0	0	0
	5	4	1	1	5	1	0	2	0	0	0	0	0	0	0
	6	1	0	0	0	2	2	0	1	2	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 5.2 Number of bits for coding spectral envelope in frames with low harmonicity

The quantized envelope is smoothed using linear interpolation. The spectrograms of the pulse are flattened using the smoothed envelope. The flattening is achieved by division of the magnitudes with the envelope, which is equivalent to subtraction in the logarithmic magnitude domain displayed in the spectrogram figures [Figure 5.16]. Phase values are not changed.

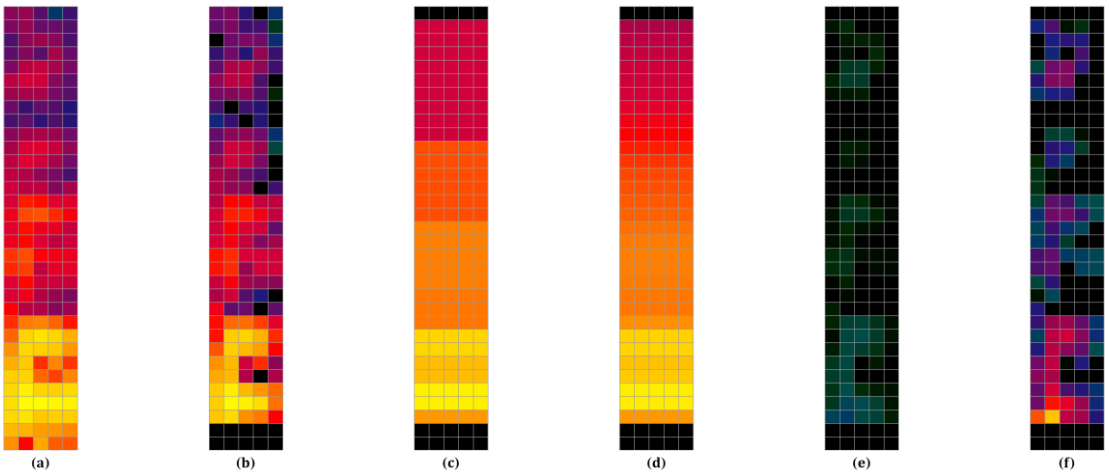

Figure 5.16 (a) Original spectrogram (b) Pulse spectrogram (c) Pulse envelope (d) Smoothed envelope (e) Flattened pulse (f) Normalized flattened pulse

Waveform of the spectrally flattened pulse y_{P_i} is obtained from the flattened STFT via the inverse DFT, windowing and overlap and add. An example of the pulse waveform obtained from the pulse spectrogram before and after the flattening is shown in Figure 5.17. The original pulse waveform in Figure 5.17 is obtained from the spectrogram (b) and the flattened original from (e) in Figure 5.16.

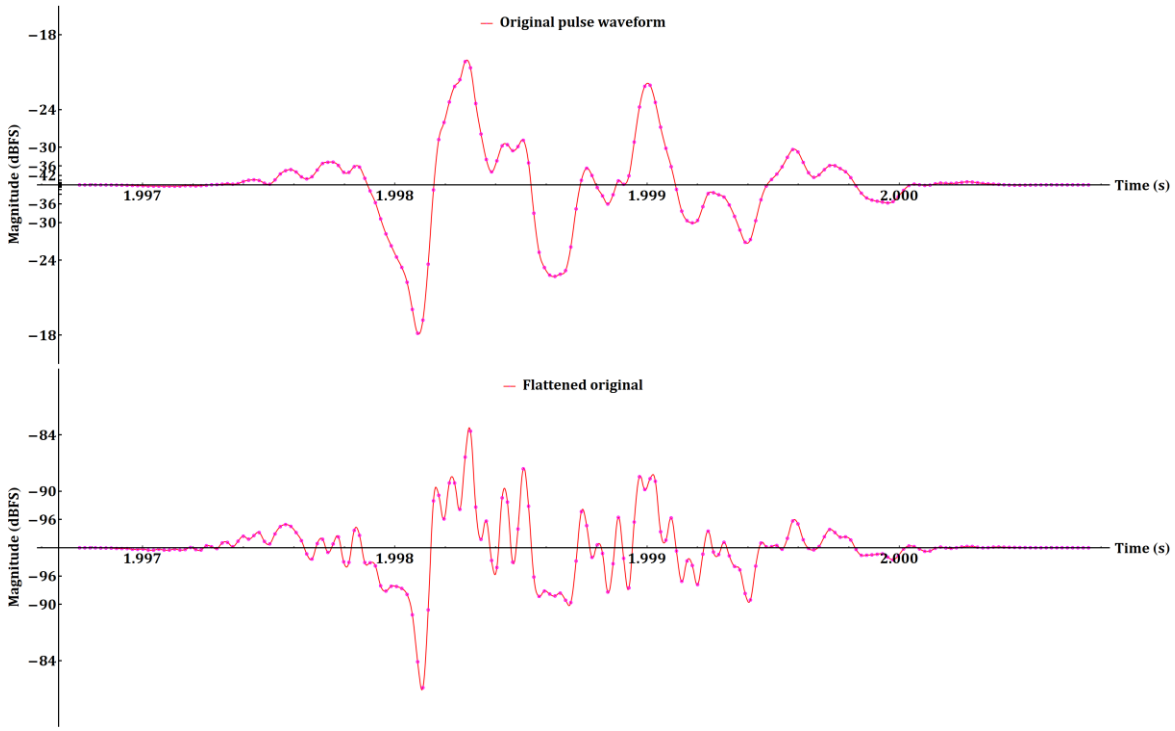

Figure 5.17 An example of the pulse waveform before and after the spectral flattening

5.4.2 Pulse prediction

The most similar previously quantized pulse is found among already quantized pulses from the current frame and N_{P_P} pulses from the previous frames. This search requires construction of the pulse waveforms from their coded representations, as shown in Figure 5.18.

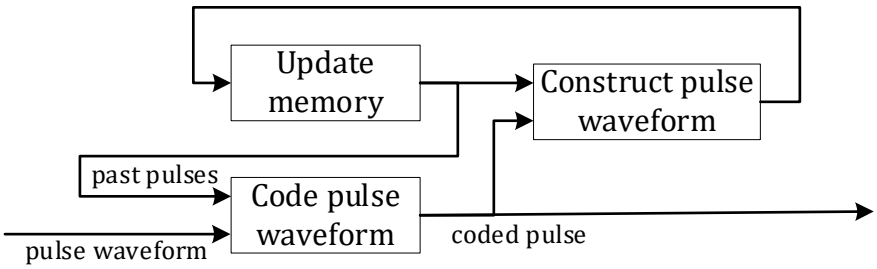

Figure 5.18 Coding of spectrally flattened pulse waveform

The correlation ρ_{P_i,P_j}, as defined in 5.3.2, is used for choosing the most similar pulse. If differences in the correlation are below 0.05, the temporally closer pulse is chosen. The most similar previous pulse is the source of the prediction ($\tilde{z}_{P_i}$) and its index $i_{P_{P_i}}$, relative to the currently coded pulse, is used in the pulse coding. Up to four relative prediction source

indexes $i_{P_{P_i}}$ are grouped and coded with Huffman codes. The grouping and the Huffman codes are dependent on N_{P_C} and whether $\bar{d}_{F_0} = 0$ or $\bar{d}_{F_0} \neq 0$.

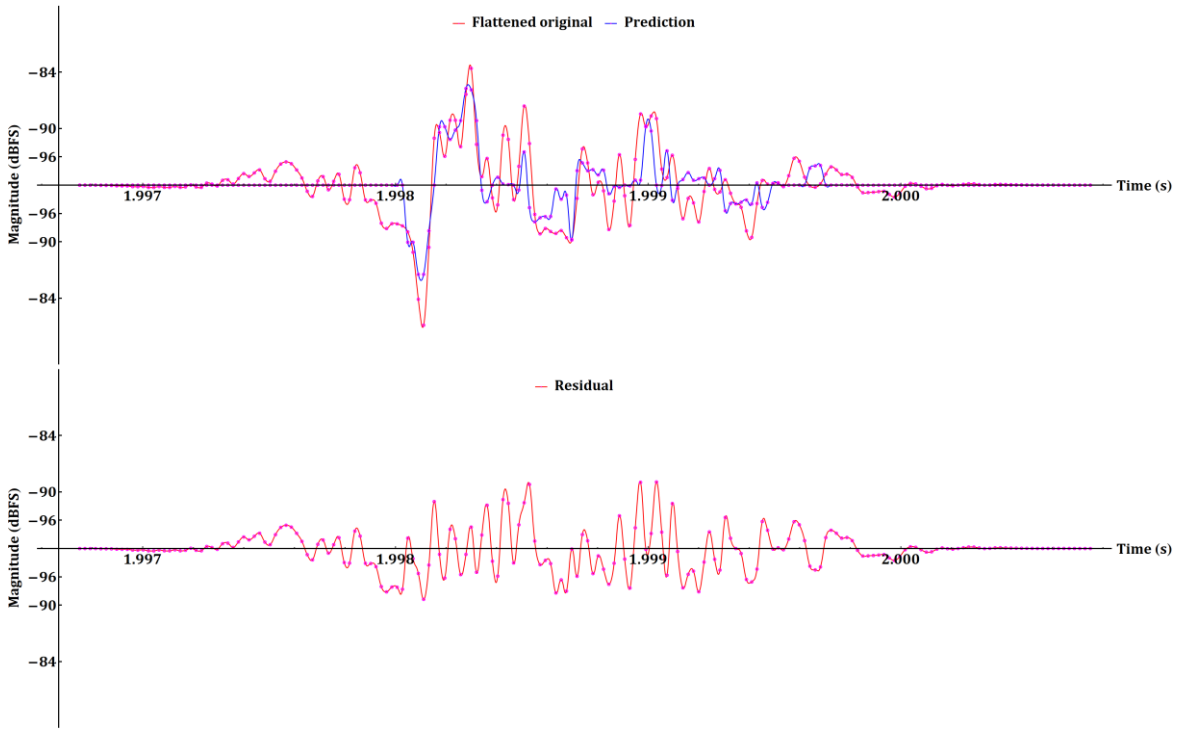

Figure 5.19 Pulse prediction and its residual

The offset for the maximum correlation is the pulse prediction offset $\Delta_{P_{P_i}}$. It is coded absolutely, differentially or relatively to the estimated value, where the estimation is calculated from the pitch lag d_{P_i} at the pulse center position. The number of bits needed for each type of coding is calculated and the one with minimum bits is chosen.

Gain $\acute{g}_{P_{P_i}}$ that maximizes the SNR is used for scaling the prediction $\tilde{z}_{P_i}$. The prediction gain is non-uniformly quantized with 3 to 4 bits. If the energy of the prediction residual is not at least 5% smaller than the energy of the pulse, the prediction is not used and $\acute{g}_{P_{P_i}}$ is set to zero.

5.4.3 Impulse quantization

The prediction residual is quantized using up to four impulses. An example is shown in Figure 5.20.

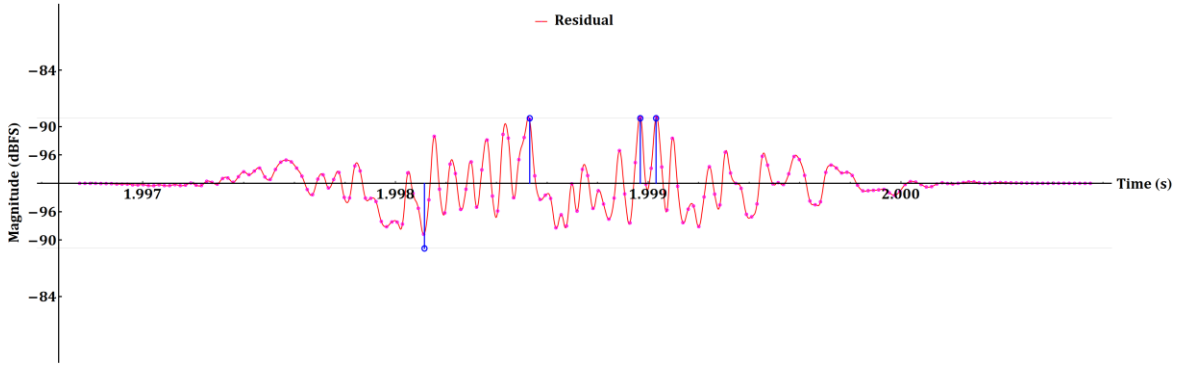

Figure 5.20 Pulse prediction residual and its 4 biggest impulses

The quantized residual consisting of impulses is named innovation $\dot{z}_{P_i}$. To save bits, the number of impulses is reduced by one for each pulse predicted from a pulse in this frame. In other words: if the prediction gain is zero or if the source of the prediction is a pulse from previous frames then four impulses are quantized, otherwise the number of impulses decreases compared to the prediction source.

For finding and coding the impulses the following algorithm is used:

1. Absolute pulse waveform $|x|_{P_i}$ is constructed using full-wave rectification:

$$|x|_{P_i}[n] = \left|x_{P_i}[n]\right|, 0 \leq n < L_{W_P}$$

2. Vector with the number of impulses at each location $\lfloor x \rfloor_{P_i}$ is initialized with zeros

3. Location of the maximum in $|x|_{P_i}$ is found:

$$\hat{n}_x = \underset{0 \leq m < L_{W_P}}{\operatorname{argmax}} |x|_{P_i}[m]$$

4. Vector with the number of impulses is increased for one at the location of the found maximum:

$$\lfloor x \rfloor_{P_i}[\hat{n}_x] = \lfloor x \rfloor_{P_i}[\hat{n}_x] + 1$$

5. The maximum in $|x|_{P_i}$ is reduced:

$$|x|_{P_i}[\hat{n}_x] = \frac{\left|x_{P_i}[\hat{n}_x]\right|}{1 + \lfloor x \rfloor_{P_i}[\hat{n}_x]}$$

6. The steps 3-5 are repeated until the required number of impulses are found

Notice that the impulses may have the same location. Locations of the impulses are ordered by their distance from the pulse center. The location of the first impulse is absolutely coded. The locations of the following impulses are differentially coded with probabilities dependent on the position of the previous impulse. Coding with Huffman codes is used for the impulse location. The sign of each impulse is also coded. If multiple impulses share the same location then the sign is coded only once.

Gain $\acute{g}_{I_{P_i}}$ that maximizes the SNR is used for scaling the innovation $\dot{z}_{P_i}$ consisting of the impulses. The innovation gain is non-uniformly quantized with 2 to 4 bits, depending on the number of pulses N_{P_C}.

The first estimate for the quantization of the flattened pulse waveform $\acute{z}_{P_i}$ is then a weighted sum of the prediction and the innovation:

$$\acute{z}_{P_i} = Q\left(\acute{g}_{P_{P_i}}\right)\tilde{z}_{P_i} + Q\left(\acute{g}_{I_{P_i}}\right)\dot{z}_{P_i}$$

5.4.4 Energy correction

Because the gains are found by maximizing the SNR, the energy of $\acute{z}_{P_i}$ can be much lower than the energy of the original target y_{P_i}. To compensate the energy reduction, a correction factor c_g is calculated:

$$c_g = \max\left(1, \left(\frac{\sum_{n=0}^{L_{W_P}}\left(y_{P_i}[n]\right)^2}{\sum_{n=0}^{L_{W_P}}\left(\acute{z}_{P_i}[n]\right)^2}\right)^{0.25}\right)$$

The final gains are then:

$$g_{P_{P_i}} = \begin{cases} c_g\acute{g}_{P_{P_i}} & , Q\left(\acute{g}_{P_{P_i}}\right) > 0 \\ 0 & , Q\left(\acute{g}_{P_{P_i}}\right) = 0 \end{cases}$$

$$g_{I_{P_i}} = c_g\acute{g}_{I_{P_i}}$$

The complete quantization process, with the energy correction, is shown in Figure 5.21. The spectrally flattened pulse is coded using the relative prediction source index $i_{P_{P_i}}$, the pulse prediction offset $\Delta_{P_{P_i}}$, the quantized prediction gain $Q\left(g_{P_{P_i}}\right)$, the quantized innovation gain $Q\left(g_{I_{P_i}}\right)$ and the locations and signs of the impulses.

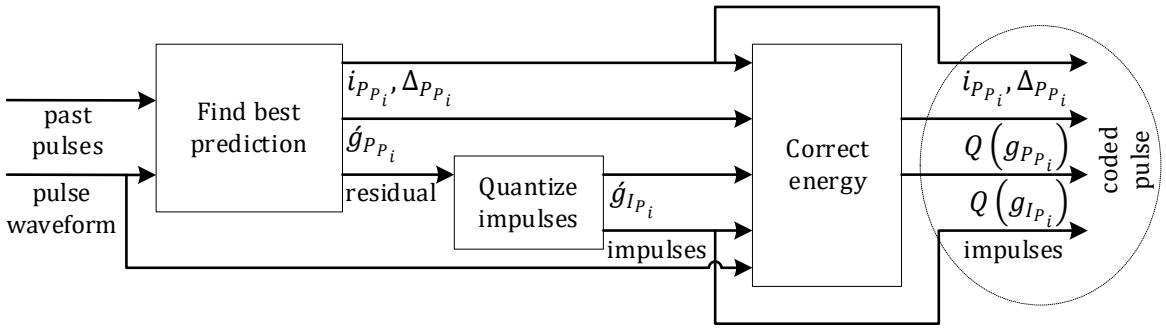

Figure 5.21 Quantization of the spectrally flattened pulse with the energy correction

5.4.5 Memory update

The memory for the prediction (see Figure 5.18) is updated using the quantized flattened pulse waveform z_{P_i}:

$$z_{P_i} = Q\left(g_{P_{P_i}}\right)\tilde{z}_{P_i} + Q\left(g_{I_{P_i}}\right)\dot{z}_{P_i}$$

At the end of the coding, up to three quantized flattened pulse waveforms are kept in the memory for the pulse prediction in the following frames.

5.4.6 Training of the entropy coders

For training the entropy coders in the pulse coding, 70 minutes of speech and 60 minutes of music/audio was used. The dataset consists of very diverse samples, 4 s to 20 s long. The vector codebooks were constructed in Wolfram Mathematica [138], mostly using simple centroid-based k-means algorithm.

5.5 Reconstructing pulses

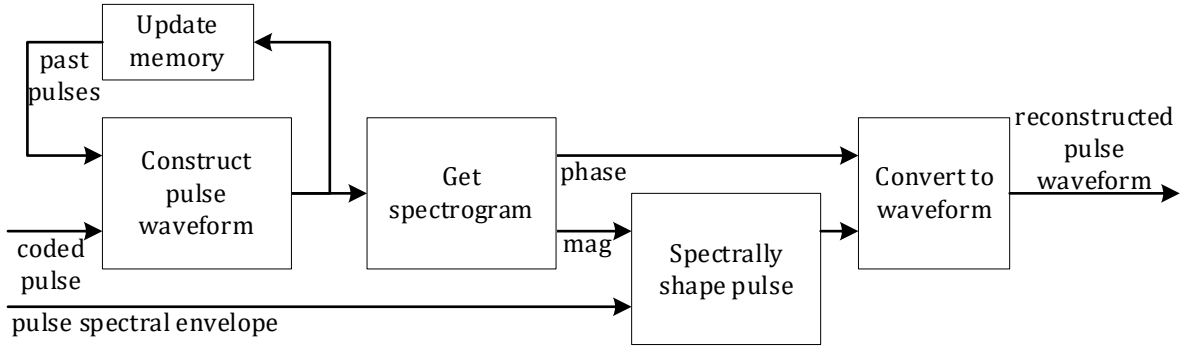

Figure 5.22 Reconstructing pulse waveform

For reconstructing the pulses on the decoder side, the quantized flattened pulse waveforms are constructed after decoding the gains $g_{P_{P_i}}$ and $g_{I_{P_i}}$, the impulses/innovation, the prediction source $i_{P_{P_i}}$ and the prediction offset $\Delta_{P_{P_i}}$. The memory for the prediction is updated in the same way as in the encoder (described in 5.4.5). The STFT is then obtained for each pulse waveform. The same 2 ms long Hann window with 75 % overlap is used as in the pulse extraction. The magnitudes of the STFT are reshaped using the decoded and smoothed spectral envelope and zeroed out below the pulse starting frequency f_{P_i}. Simple multiplication of magnitudes with the envelope is used for shaping the STFT. The phases are not modified. Reconstructed waveform of the pulse is obtained from the STFT via the inverse DFT, windowing and the overlap and add. The process of reconstructing the pulse waveform is shown in Figure 5.22. Alternatively the envelope can be shaped via an FIR filter, avoiding the STFT. Throughout the whole process of the extraction, the quantization and the reconstruction, processing with constant phase delay is used. The pulses and the residual are not necessarily clearly separated. Having constant phase delay is the safest approach as it keeps them synchronized.

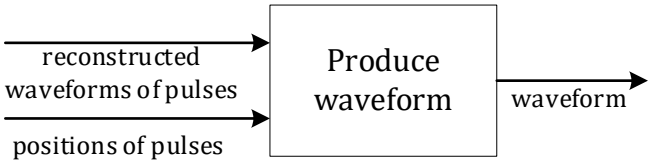

Figure 5.23 Concatenation of pulse waveforms

The reconstructed pulse waveforms are concatenated based on the decoded pulse center positions t_{P_i}, inserting zeros between the pulses [Figure 5.23]. No zeros are inserted if the pulse waveforms overlap. The concatenated waveform is added to the decoded signal. In the same manner the original pulse waveforms x_{P_i} are concatenated and subtracted from the input of the MDCT-based codec [Figure 5.3].

5.6 Problems with subtracting quantized pulses

Removing original instead of quantized (coded and reconstructed) pulses is important. The quantized pulses, because of the bitrate constraints, are not perfect representations of the original pulses. Removing the quantized pulses still leaves some transient structure of the original in the residual. As transient signals cannot be well presented with an MDCT codec, noise spread across whole frame would be present and the advantage of separately coding the pulses would be reduced.

Both strategies were investigated in [132], named as open- and closed-loop, where it was already noticed that it is advantageous if the pulse quantization error doesn't affect the residual. Yet, much more description in [132] is devoted to the closed-loop approach with a removal of the quantized pulses from the original, the quantization of the pulses coming from an ACELP like codec.

The spectrum copy-up in the Zero Filling is activated whenever TNS is applied. The repetitions of the spectrum, via the copy-up, can provoke resonance in the TNS filter. If there are original pulse remnants, because the quantized pulses are removed from the original to produce the residual, the TNS filter is activated more often and it leads, together with the copy-up, to an increase of the pulse structure in the residual. The increased pulse structure in the residual provides significant transient noise and is perceived as unpleasant. This is another reason why in IVA, the original pulses are removed from the original.

The problem of subtracting the quantized pulses will be shown on the diphthong phoneme /ai/ from the already used example of the German male speaker recording [ger_m]. The spectrograms are obtained from TD signals using 5.33 ms window with 93.75% overlap.

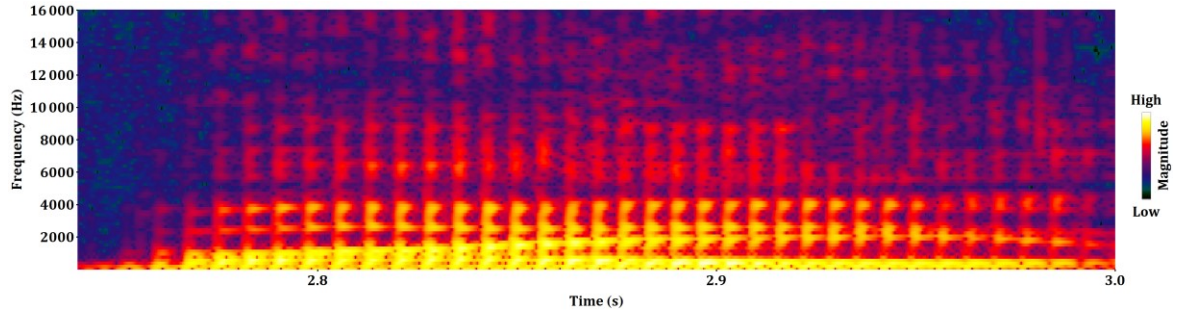

Figure 5.24 The original speech signal for /ai/

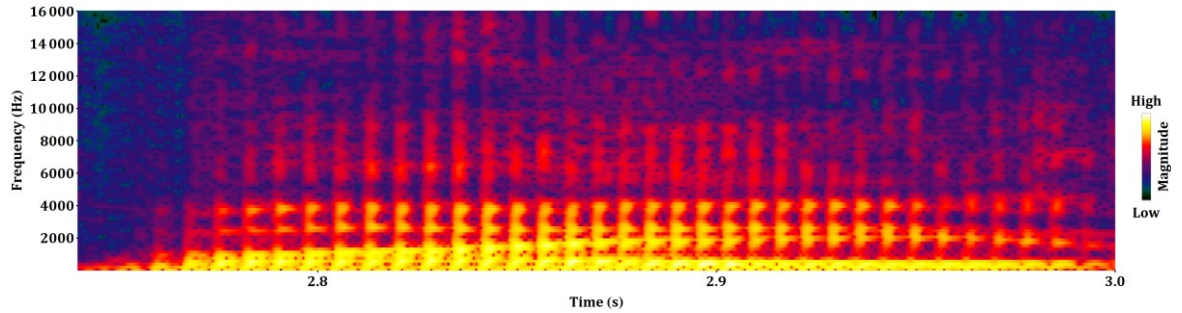

Figure 5.25 The decoded speech signal with the original pulses removed and added back

The signal in Figure 5.25 is obtained by extracting the pulses, coding and decoding the residual and adding back the original extracted pulses to the decoded residual. Comparing the original signal in Figure 5.24 and the decoded signal in Figure 5.25, it is possible to see that most of the glottal pulses are detected. Only the glottal pulses at the onset and the transient noise, between glottal pulses near the end, are not detected.

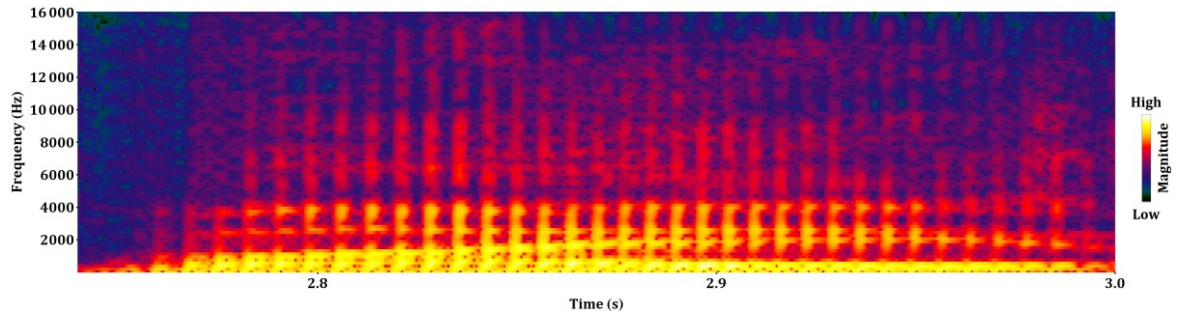

Figure 5.26 The decoded speech signal with the original pulses removed and the coded pulses added back

The signal in Figure 5.26 is obtained by extracting the pulses, coding and decoding the residual and adding back the quantized pulses to the decoded residual. This is the output of IVA. The pulses are very well coded in this case, as can be seen in comparison to Figure 5.25.

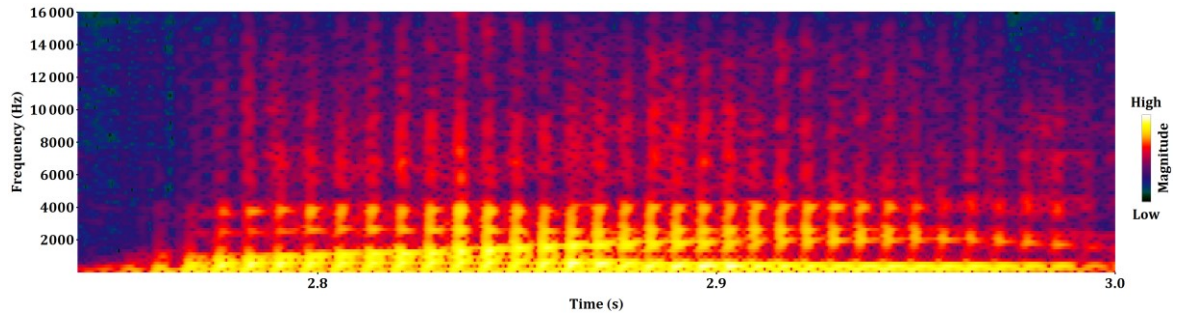

Figure 5.27 The decoded speech signal with the coded pulses removed and added back

The signal in Figure 5.27 is obtained by extracting the pulses, coding and decoding the pulses, obtaining a residual by subtracting the decoded pulses from the original signal, coding and decoding the residual and adding back the quantized pulses to the decoded residual. It is clear that the MDCT codec cannot handle remnants of the original pulses in the residual. There is much more noise between the pulses in Figure 5.27 than in Figure 5.26.

The pulses are not perfectly coded, as can be seen when comparing Figure 5.28 and Figure 5.29, but the differences are small. Yet, even these small difference are problematic for coding in the MDCT domain as shown in Figure 5.27. And there is additional burden coming from the pulse coding bits taken away from the MDCT codec.

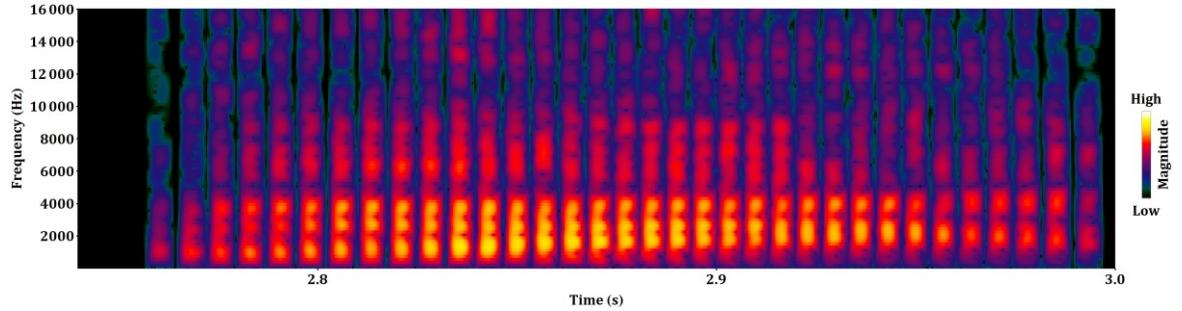

Figure 5.28 The original pulses

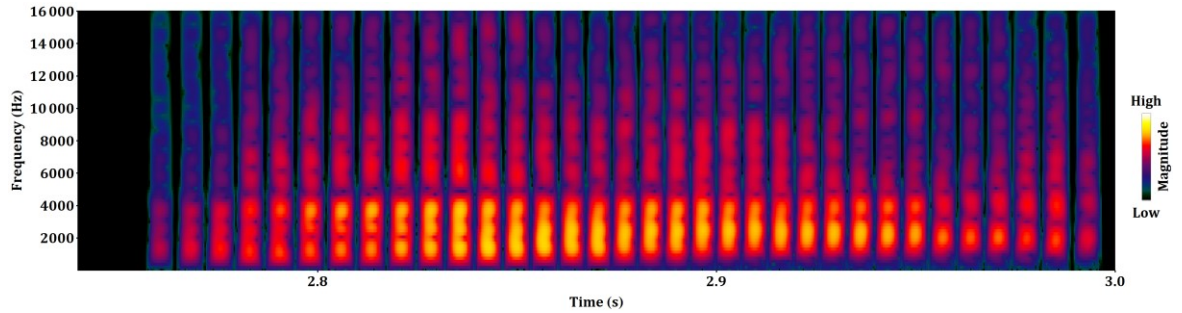

Figure 5.29 The decoded pulses

5.7 Advantages of the new contributions

The proposed approach for the pulse detection and extraction does not classify signals. It handles percussive sounds (e.g. castanets), very tonal instruments (e.g. pitch pipe) and vowels in the same manner. This differentiates it from state-of-the-art methods. Source separation methods (e.g. [136]) are splitting signals in components that belong to categories harmonic, percussive and noise. The approach in [133] is considering only transient onsets and uses the MCLT length too long even for detecting glottal pulses. In [134, 135] only dense transient events, such as applause claps, are considered. Complete voiced segments are coded with a dedicated speech codec in [116] and coding of transient sounds or tonal signals is not considered. Even in [132] it is written: “feeding tone-like signal components into the impulse coder will lead to distortions which cannot be compensated easily by the filterbank-based coder”.

Different to the state of the art, the transient detection has many transformation steps. This may seem overcomplicated, but is needed for reliable detection. Obtaining the magnitude spectrogram allows removing the stationary portions consisting of signals that are continuous within a frequency band. Because of this and due to obtaining the temporal envelope by integration of the magnitudes in the logarithmic domain over the large bandwidth, dispersed transients (including glottal pulses) can be detected even in the presence of stationary signals or noise.

By removing the stationary parts from the magnitude spectrogram of the pulses, almost only parts that are not suited for an MDCT coder are removed from the input signal.

The effectiveness of the detection and the extraction can be seen in the following examples for the diphthong phoneme /ai/ (already used example for showing the advantage of subtracting original and not quantized pulses), a castanets and a pitch pipe sample. The spectrograms are obtained from TD signals using 5.33 ms window with 93.75% overlap.

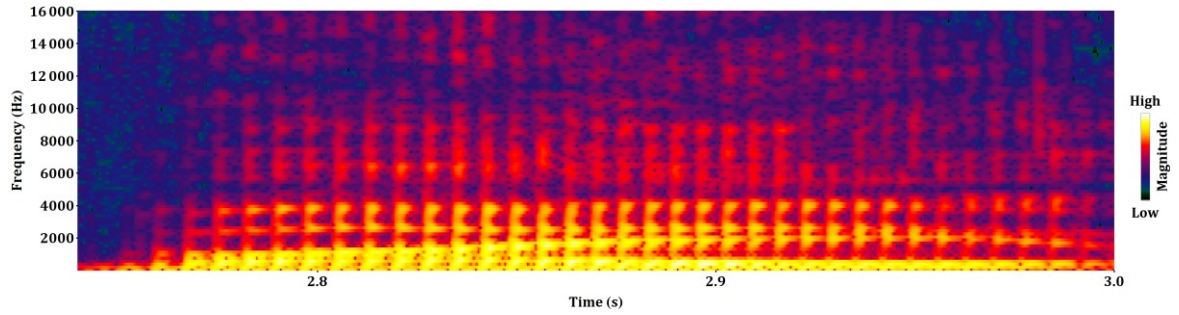

Figure 5.30 Speech original signal

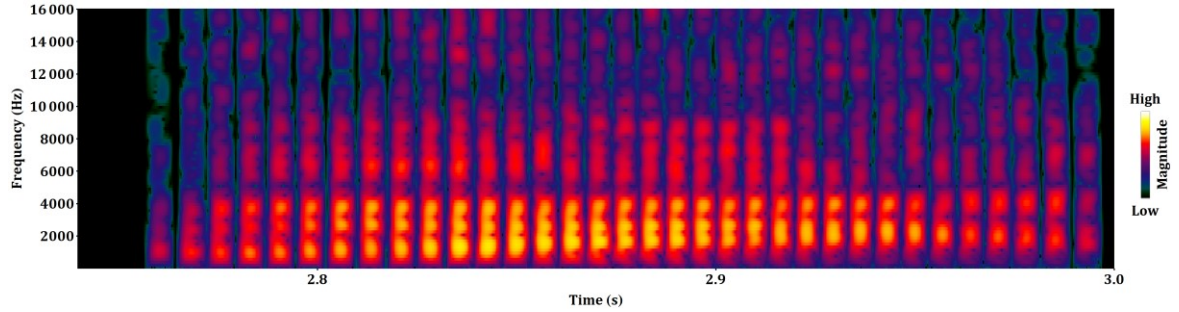

Figure 5.31 Speech pulses

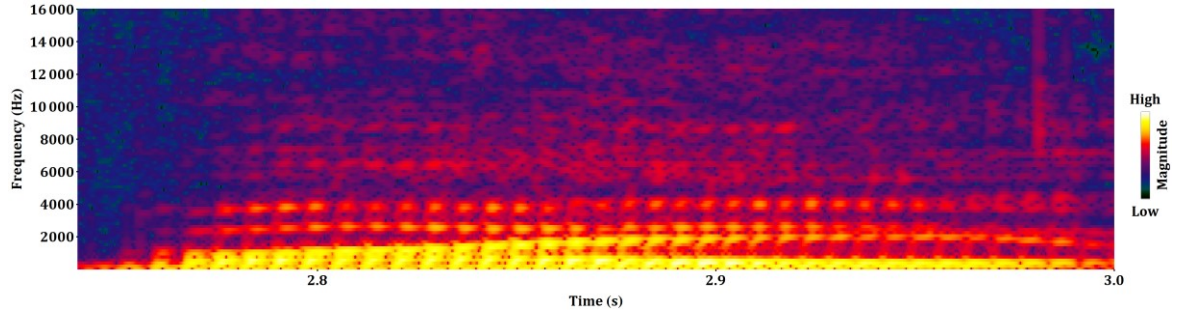

Figure 5.32 Speech residual

The pulse detection works on premises that the residual MDCT codec includes TNS or some other mechanism for handling moderate temporal modulations. This gives it advantage over the existing methods, as it doesn't need to detect transient events that can be well coded within the MDCT codec through the use of TNS. This is illustrated in the example of a castanet sound in Figure 5.33, Figure 5.34 and Figure 5.35.

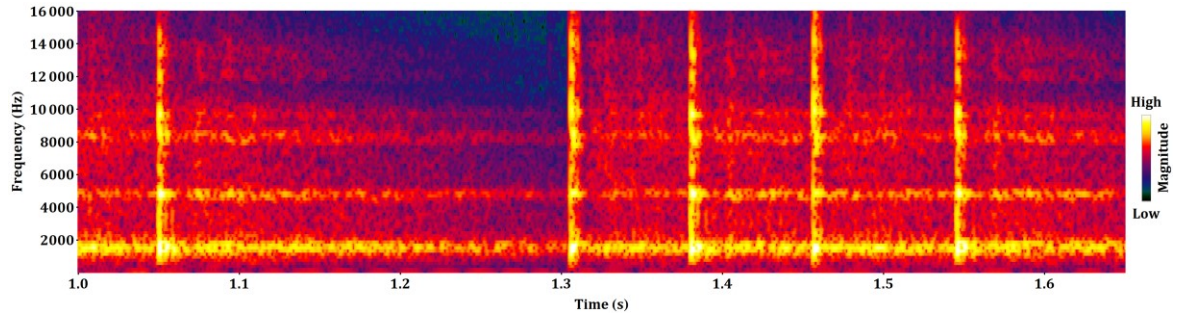

Figure 5.33 Castanets original signal

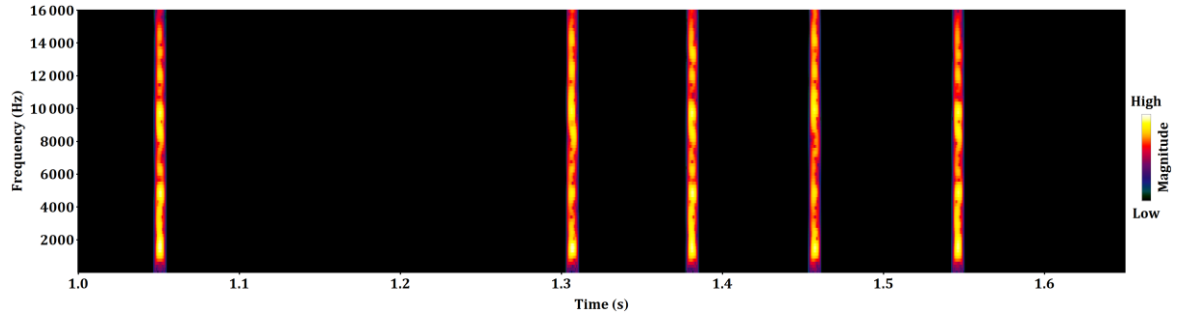

Figure 5.34 Castanets extracted pulses

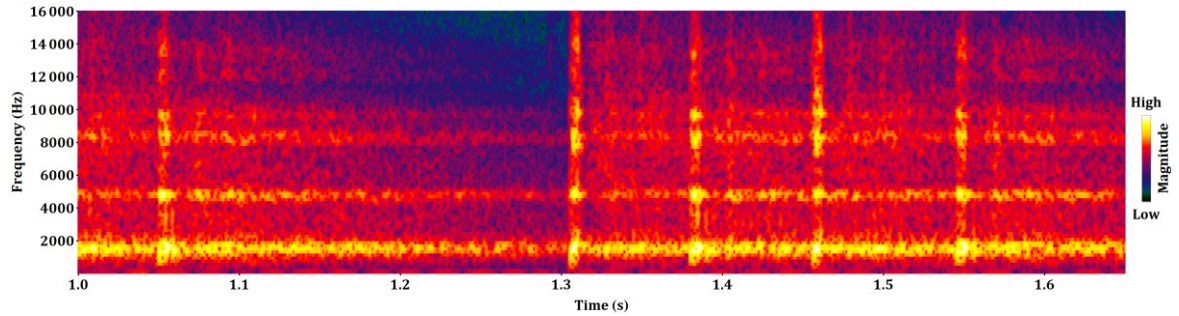

Figure 5.35 Castanets residual

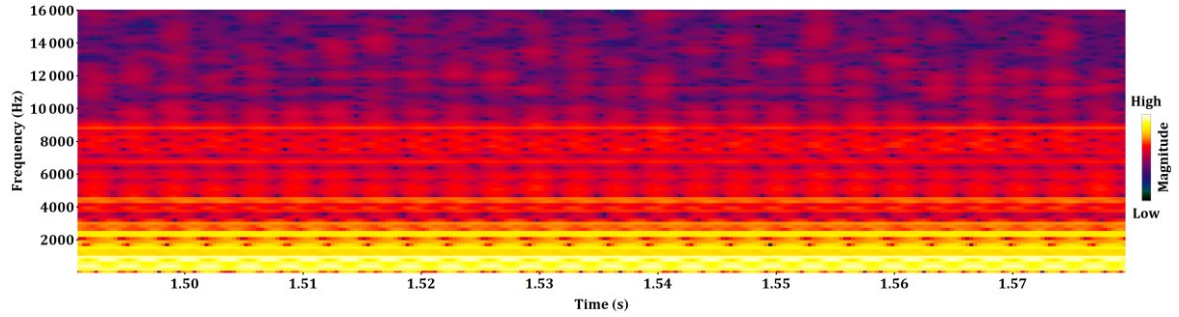

Figure 5.36 Pitch pipe original signal

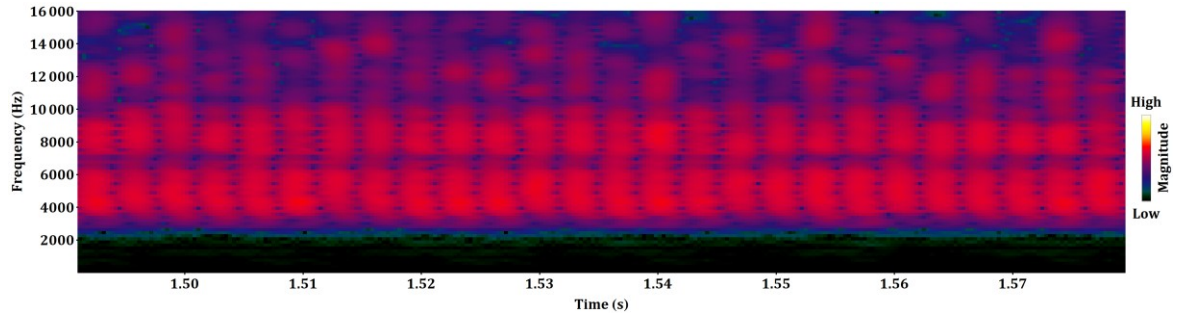

Figure 5.37 Pitch pipe extracted pulses

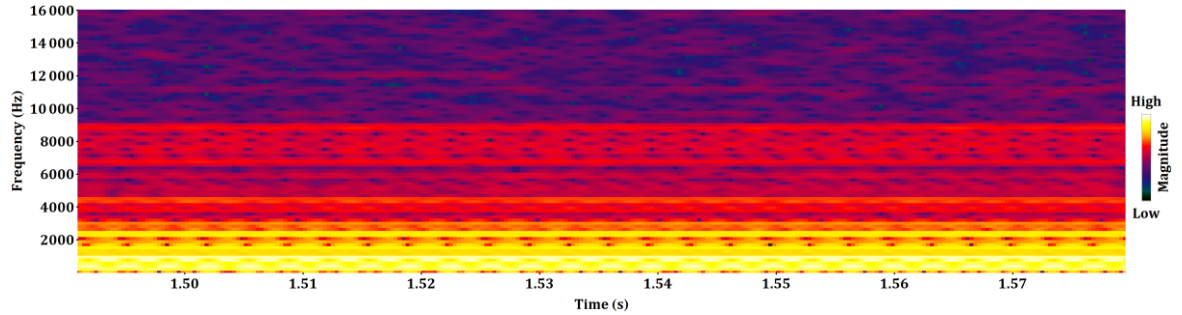

Figure 5.38 Pitch pipe residual

Signals with shorter distance between pulses of a pulse train have higher pitch and bigger distance between the harmonics, thus coding them with the MDCT coder is efficient. Such signals also exhibit less masking of broad-band transients. Different to the state of the art, the frequency region covered by the pulse coding is adaptive depending on the distance between pulses. By increasing the starting frequency for closer pulses, errors in the extraction or coding of the pulses is made less disturbing. The difference in the pulse starting frequency is visible in Figure 5.31, Figure 5.34 and Figure 5.37.

Using correlation between the pulse waveforms in the pulse choice makes sure that the pulses that can be efficiently coded are extracted. Using the ratio of the pulse energy to the local energy in the pulse choice allows that strong transients, not belonging to a pulse train, are extracted. Therefore, any kind of transients, including glottal pulses, that cannot be efficiently coded in the MDCT are removed from the input signal. It needs to be noticed that the complete pulse waveforms are obtained, where the pulse waveforms have high-pass characteristics and energy is concentrated towards its temporal center. Only after extracting such pulse waveforms, not suitable for the MDCT coding, it is decided based on correlation if they are suited for the pulse coding.

A speech signal is separated into pulse portion and residual, where even the residual contains a part of the speech signal. The residual even contains the most significant parts of vowels (see Figure 5.32), that can be efficiently coded with the MDCT codec.

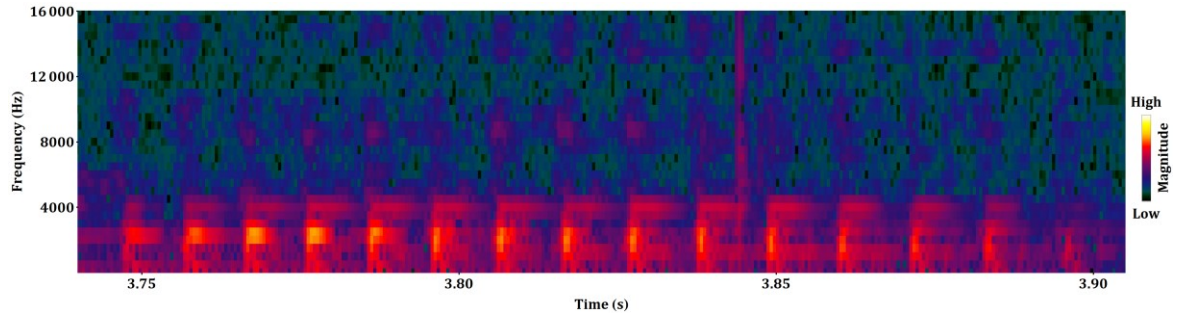

Figure 5.39 A background noise transient within the German male speaker recording

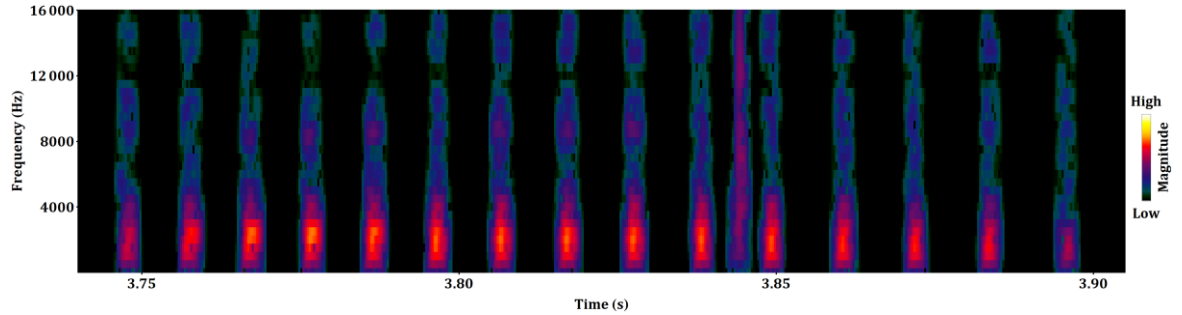

Figure 5.40 Extracted train of glottal pulses and the background transient

Using the prediction from a single pulse waveform to a single pulse waveform, coding of pulses is made efficient. The distance between neighboring glottal pulses in a clean speech recording is equal to the pitch period length. However, it may happen that second last pulse is more similar to the currently coded pulse. The proposed pulse prediction allows individual choice of the prediction source for each pulse, also using a previous pulse at distance different from the pitch period length. It may happen even in a clean speech recording, as can be seen in Figure 5.39, that there are transients not belonging to the train of glottal pulses. Again, the proposed approach will effectively handle the case and skip the background transient in the prediction process.

By the spectral flattening, changes in the spectral envelope of the pulses waveforms are ignored and the usage of the prediction is increased. The spectral flattening in the STFT doesn't change phases. It follows that the pulses and the residual are kept in sync.

Separating LTP [6.2] of the MDCT codec from the pulse prediction is advantageous for mixed signals with dominant harmonic part and the train of pulses having different fundamental frequency. Another advantage is that the pulse portion and the residual may use different prediction gains.

6 LTP

MDCT domain codecs are well suited for coding music signals as the MDCT provides decorrelation and compaction of the harmonic components commonly produced by instruments and singing voice. This MDCT property deteriorates if short MDCT windows are used or if harmonic components are frequency or amplitude modulated. By exhibiting significant frequency and amplitude modulations, vowels in speech signals are especially challenging for MDCT codecs. One method for improving the coding of the harmonic component is long-term prediction (LTP).

LTP constructs in this context a predicted MDCT spectrum from a decoded signal. The decoded signal needs to be available at both the encoder and the decoder. A coding gain may be achieved by subtracting the predicted MDCT spectrum from an MDCT spectrum that is to be coded, thus producing the LTP residual. The coding gain may be high for predictable signals consisting of single instruments, singing or speech. In the absence of quantization, LTP provides perfect reconstruction.

6.1 State of the art

LTP is a common method used in time domain speech codecs [16, 37, 39, 139]. LTP is essential for high quality at low bitrate TD coding of harmonic signals, as there is no transform to provide decorrelation of harmonic components.

In [108] a combination of LTP and the MDCT is proposed. A similar LTP method to [108] was proposed in [110, 140, 141]. In this method, pitch is determined and a prediction signal is constructed via LTP using the pitch and low-pass filtered decoded samples from past frames. A third order predictor, whose coefficients are found in a closed loop, is used for handling fractional pitch. The prediction signal is transformed via the MDCT and subtracted from the MDCT of the input signal, hence obtaining the prediction residual. The prediction residual is coded and shaped using a transmitted masking curve. Only the low-frequency coefficients where LTP coding gain is high are subtracted from the input MDCT. The prediction signal is added back to the decoded MDCT. It is stated that the pitch may be searched in sub-frames, but without any details how to apply LTP in the sub-frames. The only logical conclusion could be that the authors had in mind block switching [83] and applying the pitch search and LTP

for each sub-frame MDCT window separately as in [29]. A difference in [29] is that the MDCT is applied on the LTP residual, that is LTP operates fully in the time domain. Another combination of LTP and the MDCT was proposed in [107], which finds a block of decoded samples that best matches the current block and uses the MDCT of the found block and frequency-dependent gains for LTP. The method in [107] does not introduce anything new in regards to LTP from [29, 108, 110, 140, 141], but adapts it to the specific normalization of frequency bands used in the proposed CELT codec.

Methods that directly produce a predicted MDCT spectrum, without a need for the inverse MDCT, were presented in [95, 109, 142, 143]. The main advantage of these MDCT-only methods is to avoid additional MDCTs and an additional inverse MDCT.

In all of the above referenced LTP methods for an MDCT-domain codec, constant pitch throughout the MDCT window is assumed. This assumption significantly reduces the LTP coding gain for signals with changing pitch, the most important example being speech. An approach to handle changing pitch in speech was to use shorter MDCT windows [29, 95, 108]. Shorter windows however have degraded efficiency for coding harmonic signals.

A new method is proposed below that addresses the prediction problem of signals with varying pitch in MDCT codecs.

6.2 LTP integration

Figure 6.1 shows a part of the codec, where LTP is integrated. It includes the encoder and the decoder components. The shown decoder part is also required in the encoder and is called internal decoder. The encoder components are not needed for the LTP functioning in the decoder.

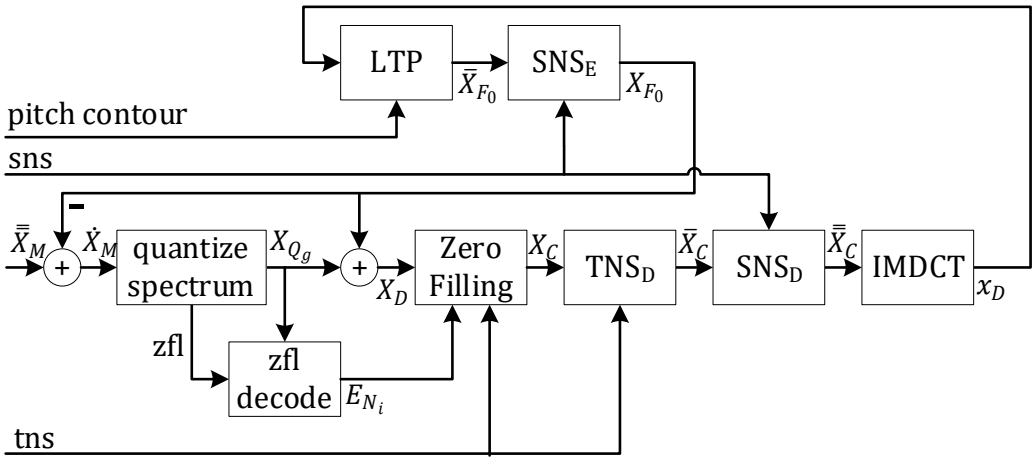

Figure 6.1 Integration of LTP

LTP updates its buffer with the output x_D of IMDCT. Driven by the coded pitch [4.3] and the LTP activation parameter, LTP constructs predicted spectrum $\bar{X}_{F_0}$, that is perceptually flattened via SNS to produce X_{F_0}. The perceptually flattened predicted spectrum X_{F_0} is

subtracted from the spectrum to be coded $\bar{\bar{X}}_M$ and added to the quantized and decoded spectrum X_{Q_g}.

The copy-up process in the Zero Filling requires that the prediction X_{F_0} is added before the start of the copy-up. Consequently, X_{F_0} needs to be in the same domain as X_{Q_g}, that is the spectral envelope of X_{F_0} needs to be perceptually flattened via SNS. An alternative would be that LTP uses IMDCT of $\bar{X}_C$ for its input, but this would require additional IMDCT and so an increase of complexity. One could even consider using IMDCT of X_D as input for LTP, but this would skip TNS which can shape a train of glottal pulses in speech and the LTP input would not be optimal for the speech prediction. Another possibility is to have a separate branch with TNS, SNS and IMDCT that doesn't include the Zero Filling. This way noise introduced in the Zero Filling would not be used in the prediction. This also would not be optimal as it was found that TNS doesn't perform well if many MDCT coefficients are zero. Separate glottal pulse train coding could avoid such limitations and its interaction with LTP and TNS offers possibilities for future research. Nevertheless, the structure as it is provides a low complexity implementation of LTP for signals with changing pitch compared to having an additional IMDCT.

6.3 LTP buffer handling

The buffer size is equal to twice of the maximum pitch lag $\hat{d}_{F_0}$, that is 39 ms. For the full overlap of 10 ms, the IMDCT output used for updating the LTP buffer coincides with the output signal frame [Figure 6.2].

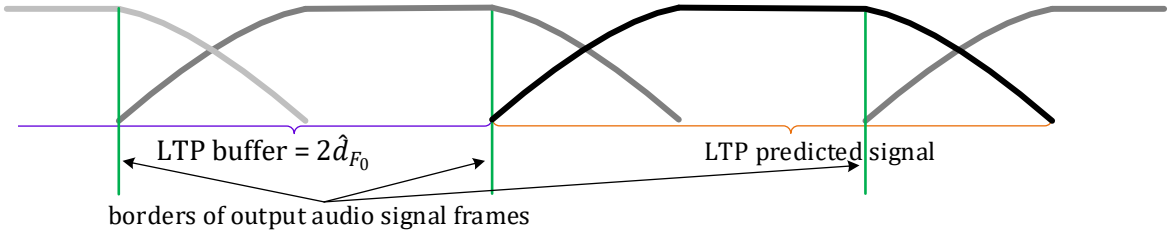

Figure 6.2 LTP buffering for the longest windows

The complete non-overlapping, hence non-aliased part, of the inverse MDCT is used for updating the LTP buffer. This means that the IMDCT output of the following frame is also used for updating the LTP buffer for a shorter overlap [Figure 6.3].

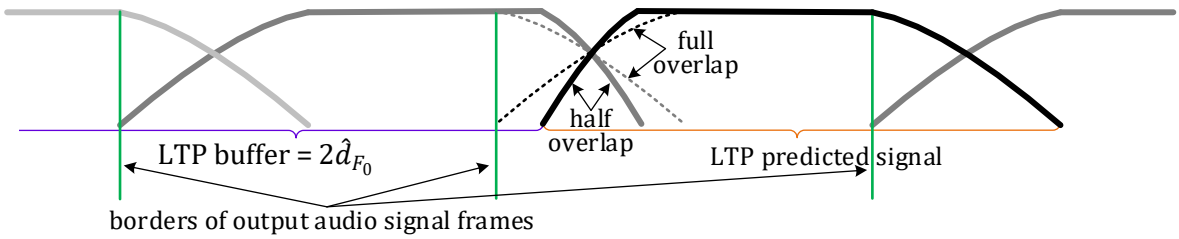

Figure 6.3 LTP buffering for windows with shortened overlap

This way, the most recent samples are used for prediction, which is especially advantageous for signals with changing pitch.

The overlap could be adaptively reduced depending on the amount of pitch change. Reducing the overlap would deteriorate the efficiency of the MDCT, so a compromise would be needed. An advantage of adapting overlap length for varying pitch couldn't be found in short experiments. Not being able to check the effect in the following frame, due to the latency constraints, is a limiting factor for this adaptation of the overlap length.

6.4 Half pitch lag correction

The LTP buffer, which is available in both the encoder and the decoder, is used to check if the pitch lag of the input signal is below the minimum codable pitch lag $\check{d}_{F_0}$. The detection if the pitch lag is below $\check{d}_{F_0}$ is called "half pitch lag detection" and if it is detected it is said that "half pitch lag is detected". The pitch lag values $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$ are coded and transmitted in the range from $\check{d}_{F_0}$ to $\hat{d}_{F_0}$. If half pitch lag is detected, it is expected that the coded pitch lag values will have a value close to an integer multiple n_{F_C} of the true pitch lag values. Equivalently the input signal's fundamental frequency is near an integer multiple n_{F_C} of the coded pitch. To extend the pitch lag range beyond the codable range, corrected pitch lag values $\dot{d}_{F_C}$ and $\acute{d}_{F_C}$ are used. The corrected pitch lag values are equal to the coded pitch lag values if the true pitch lag values are in the codable range.

Half pitch lag detection is run only if pitch is considered constant in the current window and $\bar{d}_{F_0} < \hat{n}_{F_C} \cdot \check{d}_{F_0}$, where $\bar{d}_{F_0}$ is the average pitch lag and $\hat{n}_{F_C}$ is the maximum integer multiple of the pitch lag to be checked. For signals with a changing pitch, half pitch lag detection is less reliable and thus avoided. The pitch is considered constant in the current window if $\max(|\dot{d}_{F_0} - \acute{d}_{F_0}|, |\dot{d}_{F_0} - \grave{d}_{F_0}|) < \tau_{F_C}$, where τ_{F_C} corresponds to $1/F_S$. In half pitch lag detection, for each $n_{F_C} \in \{1,2,\dots,\hat{n}_{F_C}\}$ the pitch search $\mathcal{F}_{F_0}$, the same pitch search as used for finding the pitch contour [4.3.1], is executed using $L_H = \bar{d}_{F_0}$, $d_{\tilde{F}_0} = \bar{d}_{F_0}/n_{F_C}$, $d_{\check{F}_0} = d_{\tilde{F}_0} - 3$ and $d_{\hat{F}_0} = d_{\tilde{F}_0} + 3$. The value of $\dot{n}_{F_C}$ is set to n_{F_C} that maximizes the normalized correlation $\rho_{\ddot{F}_0}$ returned by the pitch search $\mathcal{F}_{F_0}$. It is considered that half pitch lag is detected if $\dot{n}_{F_C} > 1$ and $\rho_{\ddot{F}_0}$ returned by $\mathcal{F}_{F_0}$ for $\dot{n}_{F_C}$ is at least 0.8 and at least 0.02 above $\rho_{\ddot{F}_0}$ for $n_{F_C} = 1$. It happens rarely that a signal has the fundamental frequency above 440 Hz and no signal from music recordings could be encountered that has a fundamental frequency above 880 Hz. Because the minimum codable fundamental frequency is 444 Hz, $\hat{n}_{F_C}$ was set to 2.

If half pitch lag is detected then the corrected pitch lag values $\dot{d}_{F_C}$ and $\acute{d}_{F_C}$ take the value returned by $\mathcal{F}_{F_0}$ for $n_{F_C} = \dot{n}_{F_C}$, otherwise $\dot{d}_{F_C}$ and $\acute{d}_{F_C}$ are set to the coded pitch lag values $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$ respectively. Notice that $\dot{d}_{F_C} = \acute{d}_{F_C}$ if half pitch lag is detected, which is not a limiting factor as it is considered that the pitch is constant in the current window.

The average corrected pitch lag $\bar{d}_{F_C}$ is calculated as an average of $\grave{d}_{F_0}$, $\dot{d}_{F_C}$ and $\acute{d}_{F_C}$ after correcting an eventual octave jump in $\grave{d}_{F_0}$.

The advantage of having half pitch lag correction will be explained in the context of the modification of the predicted spectrum.

6.5 Constructing the predicted spectrum

The predicted signal for the whole MDCT window is produced from the LTP buffer. The interval of the window length is split into N_{F_0} overlapping sub-intervals of length L_{F_0} with the hop size $H_{F_0} = L_{F_0}/2$, where the hop size H_{F_0} is the distance between starting point of consecutive overlapping sub-intervals. An example of splitting the window into sub-intervals is shown in Figure 6.4. L_{F_0} is chosen so that no significant pitch change is expected within the sub-intervals. Additionally, it is requested that the frame length H_M is divisible by H_{F_0} ($H_M = N_{F_0} H_{F_0}$). Therefore, H_{F_0} is chosen so that it is smaller than $\bar{d}_{F_0}/2$ and that H_M is divisible by H_{F_0} and L_{F_0} is set to $2H_{F_0}$. The value of H_{F_0} could also be limited by the half of the minimum expected pitch within the MDCT window interval, but it was found that limiting it with $\bar{d}_{F_0}/2$ is enough. The constraint that H_M is divisible by H_{F_0} makes the processing simpler, but is not necessary.

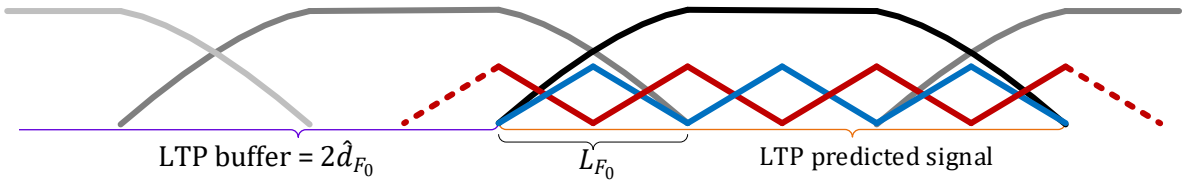

Figure 6.4 An example of splitting the MDCT window into sub-intervals for LTP

In each sub-interval the predicted signal is constructed using the LTP buffer, zero input, the sub-interval pitch lag $d_L[i]$ and a filter with the transfer function $H_L(z)$, where:

$$H_L(z) = \frac{1}{1 - B(z, \{d_L[i]\}) z^{-\lfloor d_L[i] \rfloor}}$$

In principle, zeros of length equal to the window length are appended to the LTP buffer and the filtering starts at the position of the first appended zero.

$B(z, \{d_L[i]\})$ is a transfer function of a filter with a fractional delay that has low-pass characteristics. For example at $F_S = 48$ kHz, two such filters are defined:

$$B(z, 0/2) = 0.0000z^{-2} + 0.2413z^{-1} + 0.5188z^{0} + 0.2413z^{1}$$

$$B(z, 1/2) = 0.0643z^{-2} + 0.4350z^{-1} + 0.4350z^{0} + 0.0643z^{1}$$

The filter $B(z, d)$ is chosen so that $\{d_L[i]\}$ is closest to d from a list of fractional values, where a filter $B(z, d)$ is predefined for each fractional value d from the list.

For each sub-interval, a pitch lag is obtained from the pitch contour d_V at the center of the sub-interval $i_{F_0} = i \cdot H_{F_0}, 0 \leq i < N_{F_0}$. In the first step, the sub-interval pitch lag $d_L[i]$ is set to $d_V[i_{F_0}]$. As long as the distance of the sub-interval end to the MDCT window start is bigger

than $d_L[i]$, $d_L[i]$ is increased for the value of the pitch lag at position $d_L[i]$ to the left of the sub-interval center, that is $d_L[i] = d_L[i] + d_V \left[i_{F_0} - d_L[i] \right]$ until $i_{F_0} + H_{F_0} < d_L[i]$.

This way chosen sub-interval pitch lag $d_L[i]$ allows using only samples from the LTP buffer when constructing the sub-interval. This is advantageous because for $d_L[i] \leq i_{F_0} + H_{F_0}$, constructed prediction samples from previous sub-frame in the window would be reused and the prediction signal would have increasingly low-pass characteristics closer to the right end of the window.

The predicted sub-interval signals are cross-faded in the overlap regions of the sub-intervals, as depicted in Figure 6.4. The cross-fading can also be implemented using a varying coefficient to fade-in and fade-out $B(z, \{d_L[i]\})$ and the filters in the consecutive sub-intervals can be cascaded [144].

The predicted signal is windowed, with the same window as the window used to produce X_M, and transformed via MDCT to obtain $\bar{X}_{F_0}$.

6.6 Modifying and using the predicted spectrum

The magnitudes of the predicted MDCT coefficients at least 10 bins away from the harmonics in $\bar{X}_{F_0}$ are set to zero. It is considered that the harmonics in $\bar{X}_{F_0}$ are at bin locations that are integer multiples of $f_{F_0} = 2H_M/\bar{d}_{F_C}$. Notice that in general f_{F_0} is not an integer and that the average corrected pitch lag $\bar{d}_{F_C}$ is used. The location of the harmonics are $\lfloor n \cdot f_{F_0} \rfloor$. Setting coefficient away from harmonics to zero removes noise, especially when half pitch lag is detected. This is because $\dot{d}_{F_0}$ and $\acute{d}_{F_0}$, without the half lag correction, are used for constructing the predicted spectrum. Using twice the pitch lag of the actual pitch lag, creates non existing harmonics in the middle between the actually existing harmonics. Since only the magnitudes 10 bins away from the harmonics will be modified, this means that for the 20 ms frames only signals with at least 500 Hz fundamental frequency will be affected. Modifying coefficients 8 and less bins away from harmonics was producing perceptible degradations, which indicates that this modification works well only when using double pitch lag in the construction of the prediction. Nonetheless, using very short lags for constructing the prediction produces problems because of an error accumulation coming from the limited precision of the fractional delay filter $B(z, \{d_L[i]\})$. The above described technique of using twice the pitch lag followed by the cleaning between the true harmonics solves this problem.

In music signals and especially in speech, clear harmonic structure is present only at low frequencies, while high frequencies are noisy. At low bitrates, high frequencies of the decoded spectrum consists mostly of zero or noisy portions. The predicted spectrum X_{F_0} consists of clear harmonic structure at all frequencies. Adding the complete X_{F_0} to X_Q would in many cases introduce more harmonicity at high frequency in the decoded signal than in the original signal. For this reason, only a lower part of the predicted spectrum X_{F_0} is used. A harmonic occupies $\lfloor f_{F_0} + 0.5 \rfloor$ frequency bins in X_{F_0}. Of course, because of the leakage coming from the windowing, harmonics are overlapping, but we can say that a harmonic is prominent in $\lfloor f_{F_0} + 0.5 \rfloor$ frequency bins around the frequency bin with the frequency closest to the

harmonic's frequency. Cutting X_{F_0} could produce a problem that only a part of bins belonging to a harmonic are used. To avoid the problem, X_{F_0} is cut exactly between two harmonics using the number of predictable harmonics N_L. Since N_L is signal dependent, it is determined using X_{F_0}, $\bar{\bar{X}}_M$ and $\bar{d}_{F_C}$. N_L is transmitted in the bit-stream as the LTP activation parameter.

Up to $\widehat{N}_L = 8$ harmonics may be predicted. X_{F_0} and $\bar{\bar{X}}_M$ are divided into $\widehat{N}_L$ bands of length $\lfloor f_{F_0} + 0.5 \rfloor$, each band starting at $\lfloor (n - 0.5) f_{F_0} \rfloor$, $n \in \{1, \ldots, \widehat{N}_L\}$. N_L is chosen so that for all bands $n \leq N_L$ the ratio of the energy of $\bar{\bar{X}}_M - X_{F_0}$ and $\bar{\bar{X}}_M$ is below 0.7. If there is no such n, then $N_L = 0$ and LTP is not active in the current frame. If LTP is active then the first $\lfloor (N_L + 0.5) f_{F_0} \rfloor$ coefficients of X_{F_0}, except the zeroth coefficient, are subtracted from $\bar{\bar{X}}_M$ to produce $\dot{X}_M$. The zeroth and the coefficients above $\lfloor (N_L + 0.5) f_{F_0} \rfloor$ are copied from $\bar{\bar{X}}_M$ to $\dot{X}_M$. In the similar way, if LTP is active then the first $\lfloor (N_L + 0.5) f_{F_0} \rfloor$ coefficients of X_{F_0}, except the zeroth coefficient, are added to X_{Q_g} to produce X_D. The zeroth and the coefficients above $\lfloor (N_L + 0.5) f_{F_0} \rfloor$ are copied from X_{Q_g} to X_D. To point out: using bands that consist of single harmonics has advantage over bands with fixed borders. In bands with fixed borders, a harmonic may be split across two neighboring bands. Such split harmonic will be only partially predicted, producing potentially even decreased gain for the non-coded harmonic part.

6.7 Advantages of the new contributions

All previously existing methods assume constant pitch throughout the MDCT window. They use shorter windows for handling changing pitch, thus exactly deteriorating coding of harmonic signals for which LTP should be advantageous.

With the proposed varying filter, the pitch change is decoupled from the MDCT windowing and from the frame update rate. Either keeping long windows or having an adaptive block switching independently of LTP is possible.

One could argue that short windows allow faster adaptation for the changing pitch. Having in mind that all kind of signals should be coded with IVA, adaptive block switching depending on pitch change would be needed. Block switching is dangerous for harmonic signals, similarly as already shown in example for switching in EVS [Figure 2.2]. The already presented pulse extraction and coding offers, without the block switching, a fast adaptation to pitch changes in speech. Together with it, the varying LTP allows coding efficiency maximization by keeping the long windows.

Using twice the actual pitch lag and reducing the magnitudes between the true harmonics, improvement for predicting signals with constant and high pitch is achieved. This approach works without any additional side information.

Partial prediction of a harmonic is avoided by using bands consisting of single harmonics. By using the number of harmonics and the already available pitch information, very small number of bits is needed for signaling which part of the spectrum has high enough LTP coding gain.

7 Integral Band-wise Parametric Coder

Modern audio and speech coders at low bitrates usually employ some kind of parametric coding for at least a part of its spectral bandwidth. The parametric coding is either separated from a waveform preserving coder (called core coder with a bandwidth extension in this case) or is very simple (e.g. noise filling). The name bandwidth extension comes from its use in generating high frequencies (HF), usually from low frequencies (LF). Sometimes even both, a noise filling at LF and a sophisticated bandwidth extension, are used in parallel [31, 88, 145].

A bandwidth extension may be implemented independent of a spectrum coder as a pre- and post-processing step [146] or directly working on the spectrum of the core coder [90]. The pre-/post-processing by Spectral Band Replication (SBR) [146] and similar bandwidth extensions introduce additional delay and complexity. With independent bandwidth extensions it is not possible to use the same rate and distortion criteria as used in the core coder and it is very complicated to adaptively chose which part of spectra to code parametrically. Because of these disadvantages, only bandwidth extensions operating on the core coder spectrum will be considered.

A new principle of integrating band-wise parametric coding within a waveform preserving coder is proposed. It is termed integral Band-wise Parametric Coder (iBPC). On the encoder side it consists of joint coding of band-wise parameters and the MDCT spectrum. On the decoder side it consists of an adaptive filling of spectral holes in the decoded MDCT spectrum – uniformly combining the noise filling and the bandwidth extension in the Zero Filling.

Some of the existing and related bandwidth extension and noise filling methods are first listed and then analyzed, followed by a description of the new iBPC approach.

7.1 State of the art

In AMR-WB+ [88], a comfort noise is used for frequencies between $F_s/8$ and $F_s/4$, where F_s is the sampling rate of the coded signal. The comfort noise of a magnitude derived from the transmitted noise fill-in level is inserted in sub-vectors rounded to zero. AMR-WB+ also has an independent bandwidth extension for frequencies between $F_s/4$ and $F_s/2$, that creates HF from the core coder excitation using sub-frame gains (acting as a temporal envelope) and LP synthesis filter (to restore its spectral envelope).

In [147] a noise filling in FD coder is proposed, where zero-quantized lines are replaced with a random noise shaped depending on a tonality and the location of the non-zero-quantized lines, the level of the inserted noise set based on a transmitted global noise level.

The noise filling in [88, 147] and similar methods provide substitution of spectral lines quantized to zero, but with very low spectral resolution, usually just using a single level for the whole bandwidth. Bitrate constraints are fulfilled by using constant number of bits for representing the noise level.

In Perceptual Noise Substitution (PNS) [89] noise-like components are detected on a coder frequency band basis in the encoder. The spectral coefficients in a scale factor bands containing noise-like components are omitted from the quantization/coding and only a noise substitution flag and the total power of the substituted bands are transmitted. In the decoder random vectors with the desired total power are inserted for the substituted spectral coefficients.

In [148] noise level calculation and noise substitution detection in the encoder comprise:

- Detect and mark spectral bands that can be reproduced perceptually equivalent in the decoder by noise substitution. For example, a tonality or a spectral flatness measure may be checked for this purpose;
- Calculate and quantize the mean quantization error (which may be calculated over a plurality or over all scale factor bands not quantized to zero); and
- Calculate scale factor for band quantized to zero such that the (decoder) introduced noise matches the original energy.

In [148] noise is introduced into spectral lines quantized to zero starting from a “noise filling start line”, where the magnitudes of the introduced noise is dependent on the mean quantization error and the introduced noise is per band scaled with the scale factors.

The methods in [89, 148] decide before the quantization, just depending on tonality, which sub-bands to zero out. There is also no consideration how to fulfill bitrate constraints after modifying scale factors for bands quantized to zero, in the usual cases of the entropy dependent coding of scale factors.

The methods in [88, 89, 147, 148] use just simple random noise replacement and cannot recreate tonal signals in the zero-quantized spectrum. The approaches that will be next presented, may recreate tonality in the zero portions using the LF content.

In [149] a bandwidth extension method operating in the TD that avoids inharmonicity is proposed. The autocorrelation function of the magnitude spectrum is calculated, where the magnitude spectrum is obtained from the decoded time domain signal. By using the magnitude spectrum autocorrelation, an estimation of the fundamental frequency is avoided. The analytical signal of the LF part is generated by Hilbert transformation and multiplied with a modulator to produce the bandwidth extension. The offset parameters of the modulator are derived from the location of the maximum magnitude spectrum autocorrelation and thus the harmonicity of the decoded signal is ensured. Spectral envelope shaping and noise addition is done by SBR. Even though the method in [149] belongs to the group of independent

bandwidth extensions, it is considered because of its harmonicity adaptation and it was a starting point for [150]. Besides the disadvantages of an additional delay and complexity, only the characteristics of the autocorrelation of the magnitude spectrum and predefined constants are used for choosing the offset used in the modulator and just one offset is found for the whole spectrum bandwidth.

In [150] the complete core band is copied into the HF region and afterwards shifted so that the highest harmonic of the core matches with the lowest harmonic of the replicated spectrum. The shifts finer than one frequency bin are achieved via modulation in a complex spectrum obtained from the MDCT. Finally the spectral envelope is reconstructed. The frequency shift, also named the modulation frequency, is calculated based on the fundamental frequency that can be calculated on the encoder side using the full spectrum or on the decoder side using only the core band. Only one modulation frequency for the whole bandwidth is used for the frequency shift and the modulation frequency is calculated only on the basis of the fundamental frequency. The need for the complex spectrum requires additional one frame delay. Even though the complex spectrum is derived from the MDCT, the method actually belongs to the group of independent bandwidth extensions.

In [90, 151–157] a semi-parametric coding technique, named Intelligent Gap Filling (IGF), is proposed that fills spectral holes in the HF region using synthetic spectrum generated out of LF content and post-processing by parametric side information consisting of the HF spectral and temporal envelope. The IGF range is determined by a user-defined IGF start and a stop frequency. Waveforms, which are deemed necessary to be coded in a waveform preserving way by the core coder, e.g. prominent tones, may also be located above the IGF start frequency. The encoder codes the spectral envelope in the IGF range and afterwards quantizes the MDCT spectrum. The decoder uses traditional noise filling below the IGF start frequency. A tabulated user-defined partitioning of the spectrum bandwidth is used with a possible signal adaptive choice of the source partition (tile) and with a post-processing of the tiles (e.g. cross-fading) for reducing problems related to tones at tile borders. In [154] an automated selection of source-target tile mapping and whitening level in IGF is proposed, based on a psychoacoustic model.

IGF operates on the core coder spectrum, but it is separated and independent from the core coder quantizer. Even though IGF allows preservation of spectral lines in the whole bandwidth, it requires a spectral analyzer operating before the spectral domain core encoder and it is not possible to have a choice, which parts of the spectrum to code parametrically depending on the result of the spectral domain core encoder.

IGF has predefined sub-band partitioning and the spectral envelope is transmitted for the complete IGF range, without a possibility to adaptively transmit the spectral envelope only for some sub-bands.

In [154] only predefined source tiles below the IGF start frequency are used to fill the IGF target range, where the target range is above the start frequency. The tile choice is dictated by the adaptive encoding and needs to be coded in the bit-stream. The proposed brute force approach has high computational complexity.

In IGF a source tile is obtained below the IGF start frequency and does not use the waveform preserving core coded prominent tones located above the IGF start frequency for the copy-up. There is also no mention of using combined low-frequency content and the waveform-preserving core coded prominent tones located above the IGF start frequency as a source tile. This shows that IGF is a tool that is an addition to a core coder and not an integral part of a core coder.

The following state of the art [158–161] describes how to parametrically code single spectral lines or just how to choose them. They are not directly related to the proposed iBPC and even the method from [158] is used in IVA parallel to iBPC, yet they are presented for the completeness.

In the tone filling from [159] the encoder finds extremum coefficients in a spectrum, modifies the extremum coefficient or its neighboring coefficients and generates side information, so that pseudo coefficients are indicated by the modified spectrum and the side information. The pseudo coefficients are determined in the decoded spectrum and set to a predefined value in the spectrum to obtain a modified spectrum. A time-domain signal is generated by an oscillator controlled by the spectral location and value of the pseudo coefficients. The generated time-domain signal is mixed with the time-domain signal obtained from the modified spectrum. In [160] pseudo coefficients are determined in the decoded spectrum and replaced by a stationary tone pattern or a frequency sweep pattern.

In [159] only parametric coding of single tonal components is considered. It is decided before the quantizer, which spectral lines to code parametrically and only simple maxima determination is used for the decision. The result of the quantizer is not used for determining which spectral lines to code parametrically. Non-zero pseudo coefficients need to be coded in the spectrum and coding non-zero coefficients is in almost all cases more expensive than coding zero coefficients. On top of coding the pseudo coefficients, a side information is required to distinguish pseudo coefficients from the waveform preserving spectral coefficients. A lot of information needs to be transmitted in order to generate a signal with many tonal components. The method also does not propose any solution for non-tonal parts of a signal. In addition, the computational complexity for generating signals containing many tonal components coded parametrically is very high.

In [160] the high computation complexity is reduced compared to [159], by using spectral patterns instead of time-domain generator. Yet, only predetermined patterns or their modifications are used for replacing the pseudo coefficients, either requiring a lot of storage or limiting the range of the possible tones that can be generated. The other drawbacks from [159] remain in [160].

In [158, 161] quantizers use a dead-zone that is adapted depending on the input signal characteristics. The dead-zone makes sure that low-level spectral coefficients, potentially noisy coefficients, are quantized to zero. The value range of spectral coefficients that should be set to zero is estimated. As they are not using the actual output of the quantization, they are prone to errors in the estimation.

Basically all of the state of the art, zeroes out noise bands before the core coder quantization rate loop or requires a priory distribution of bits between the bandwidth extension and the non-parametric spectrum coding.

7.2 iBPC Overview

The spectrum $\dot{X}_M$, which spectral envelope is perceptually flattened, is scalar quantized to X_Q using single quantization step size g_Q across the whole coded bandwidth and entropy coded with the context based arithmetic coder. Because of the perceptual flattening, usually more spectrum lines are quantized to zero at HF than at LF. The coded bandwidth of the MDCT is divided into sub-bands B_i of increasing width L_{B_i}. This sub-band division is independent of SNS. However, the SNS division is also used for iBPC because of simplicity.

Depending on a relation between $\dot{X}_M$ and X_Q, it is decided by the Adaptive band zeroing which sub-bands are zeroed. In addition to the sub-bands explicitly set to zero, there are usually many sub-bands zeroed out by the scalar quantization. Energies in $\dot{X}_M$ are represented for every zero sub-band by the Zero Filling Level (ZFL) E_{N_i} and entropy coded to "zfl".

The quantization step size g_Q, also called global gain, is found beforehand. iBPC in the encoder consists of the search for g_Q and the final quantization and coding (Figure 7.1).

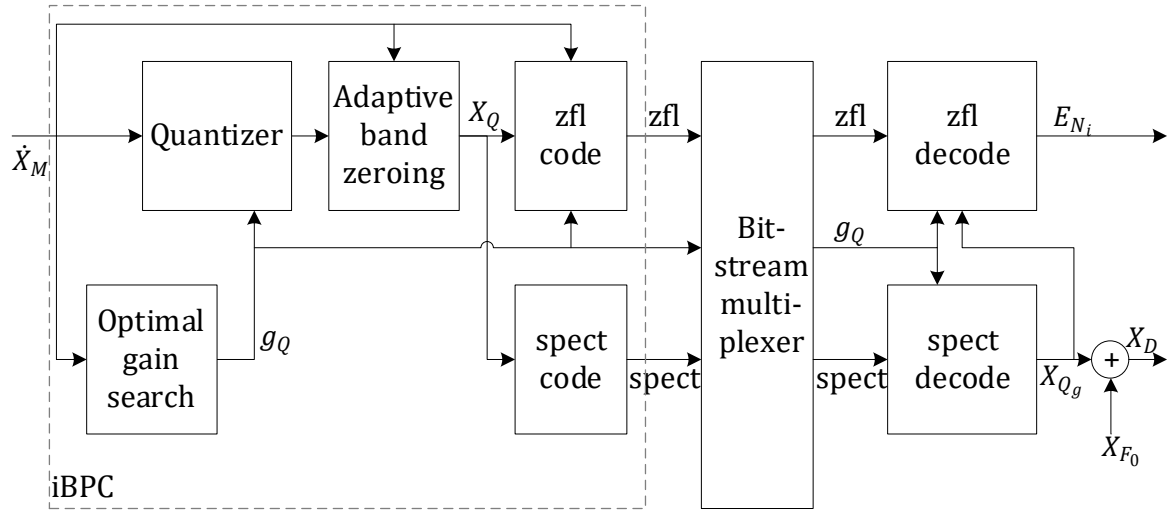

Figure 7.1 iBPC – encoding and decoding overview

At the decoder side, iBPC consists of decoding the ZFLs E_{N_i} and the Zero Filling [Figure 7.3]. First, the quantized spectrum is decoded from the entropy coded "spect" and scaled with g_Q, producing X_{Q_g}. The ZFLs are then decoded from "zfl" for all sub-bands completely zero in X_{Q_g}. The location of zeros in X_{Q_g} is the same as in X_Q as they differ only in the scaling. The number of zero bands and their locations is adaptive and completely determined by X_Q and hence not explicitly transmitted. The ZFLs for non-zero sub-bands are estimated.

The Zero Filling generates portions of spectra that are added to X_D, where X_D is a sum of X_{Q_g} and the LTP spectrum X_{F_0}. Even though the ZFLs are calculated on $\dot{X}_M$, that doesn't contain the

LTP contribution, the addition of X_{F_0} is required before using copy-up from the LF to HF portions.

The optimal gain search and the Zero Filling will be described in more details. The absolute values, e.g. the range where the Adaptive band zeroing is active, are given for coding at 24.4 kbps at 48 kHz.

7.3 Rate-distortion loop and the Adaptive band zeroing

The optimal quantization step size g_Q is iteratively found as shown in the algorithm in Figure 7.2.

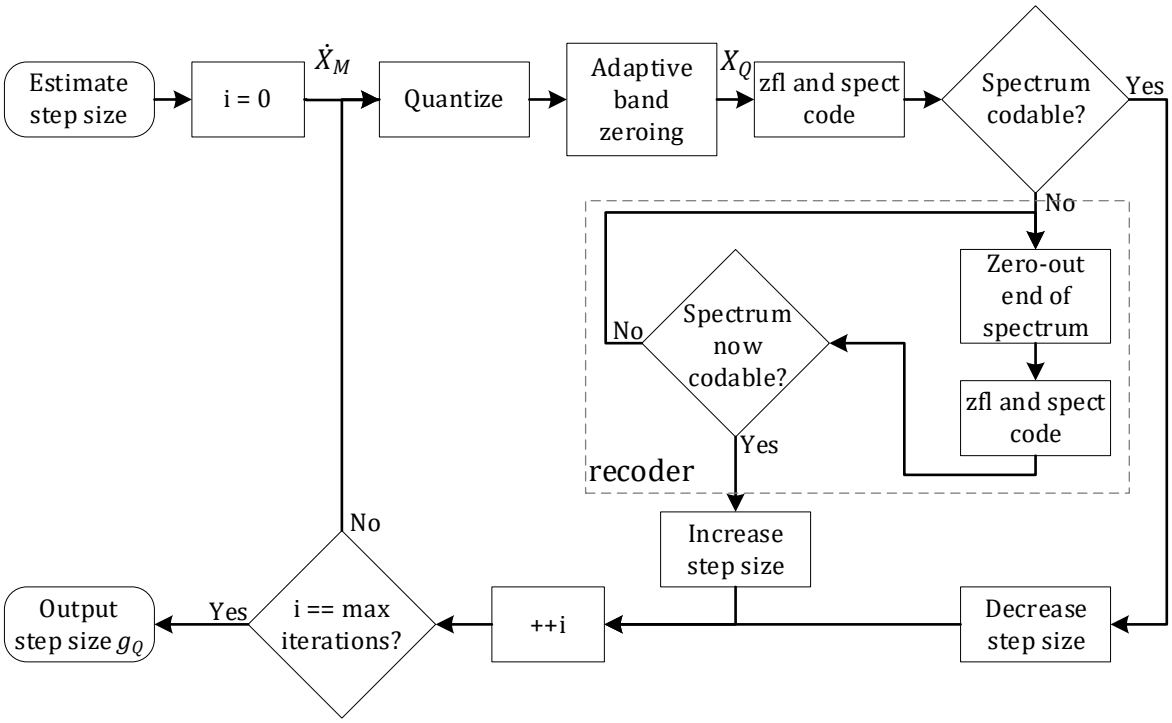

Figure 7.2 Rate-distortion loop

In each iteration the MDCT spectrum $\dot{X}_M$ is quantized to X_Q. In the Adaptive band zeroing, the ratio of the energy of the zero quantized lines and the original energy is calculated in the sub-bands B_i of $\dot{X}_M$ and if the energy ratio is above adaptive thresholds τ_{B_i}, the whole sub-band in X_Q is set to zero. The thresholds are calculated based on the HF tonality ϕ_H [4.4.2] and the flags $\dot{\phi}_{N_{B_i}}$ (that are indicating if a sub-band was zeroed-out in the previous frame):

$$\tau_{B_i} = \frac{1 + \left(\frac{1}{2} - \dot{\phi}_{N_{B_i}}\right)\phi_H}{2}$$

For each zeroed-out sub-band the flag $\phi_{N_{B_i}}$ is set to one. At the end of the frame processing, $\phi_{N_{B_i}}$ are copied to $\dot{\phi}_{N_{B_i}}$. The possible values of τ_{B_i} are 0.25, 0.5 or 0.75.

The Adaptive band zeroing is used above 7000 Hz independently in each sub-band, extending it down to 700 Hz as long as the lowest sub-band is zeroed out.

The required number of bits for the entropy coding of the ZFLs and the spectral lines in X_Q is calculated. Additionally the number of spectral lines N_Q that can be explicitly coded with the available bit budget is found. As long as there is not enough bits for coding all non-zero lines, the lines in X_Q above N_Q are set to zero and the required number of bits is recalculated in the "recoder" (Figure 7.2). The recalculation is needed as the contiguous zero portions increase and thus also the bits needed for the ZFLs. The context based arithmetic coder is used for coding the ZFLs to minimize the bit demand. Coding ZFLs independently would allow fast finding of the final N_Q without the need for the bit recalculation, but the minimal bit demand approach was chosen in IVA.

For the calculation of the bits needed for coding the spectral lines, we calculate bits needed for coding each pair of lines starting from the bottom. The recalculation of the bits needed for coding the spectral lines is made efficient by storing the number of bits needed for coding n lines for each $n \leq N_Q$.

If the required number of bits exceeds the available bits, the global gain is decreased, otherwise it is increased. In each iteration the speed of the global gain change is adapted. The rate-distortion loop from EVS [37, 120] is used to iteratively modify the global gain.

Beside the spectral lines of X_Q and the ZFLs, N_Q is also coded in the bit-stream.

7.4 Zero Filling

The spectral lines quantized to zero in X_Q are replaced in the Zero Filling with random noise or with a copy of LF spectral lines or with a mixture of both. Even though the spectrum X_D may not be zero at the positions where the Zero Filling is added, X_{Q_g} is zero at these positions.

For each sub-band B_i, being the destination, a source spectrum is determined. If the copy-up is used the source spectrum is an LF continuous portion. The distance between the destination B_i and the copy-up source is adaptively determined in the decoder. There is no explicitly continuous copying-up with a constant distance across the whole bandwidth. The Zero Filling starts at 700 Hz.

Repeating a short spectrum part produces harsh temporal structure usually not existing in the original signal, sometimes even creating transients. For this reason, it is preferable to have a big copy-up distance between the copy-up source and destination. From short experiments and from the bandwidth extension configurations in the existing codecs, choice of the copy-up distance above 5500 Hz seems safe. Because a good value for the distance is higher than the Zero Filling start frequency, either random noise or shorter distance needs to be used at lower Zero Filling frequencies.

When using copy-up it is important to preserve harmonic structure, that is the distance between harmonics. For this reason the copy-up distance is adaptive and is a multiple of the distance between harmonics. The shorter copy-up distance used at lower iBPC frequencies should also be a multiple of the distance between harmonics.

Another limit known from the existing codecs is that the source of the copy-up should not start at the lowest frequencies.

Using short experiments it was also found that the copy-up is beneficial for transient signals, i.e. signals where TNS is active. For such signals it must be taken care that the temporal structure in the signal below and above starting TNS frequency is different. Consequently, only the copy-up within the TNS region should be used.

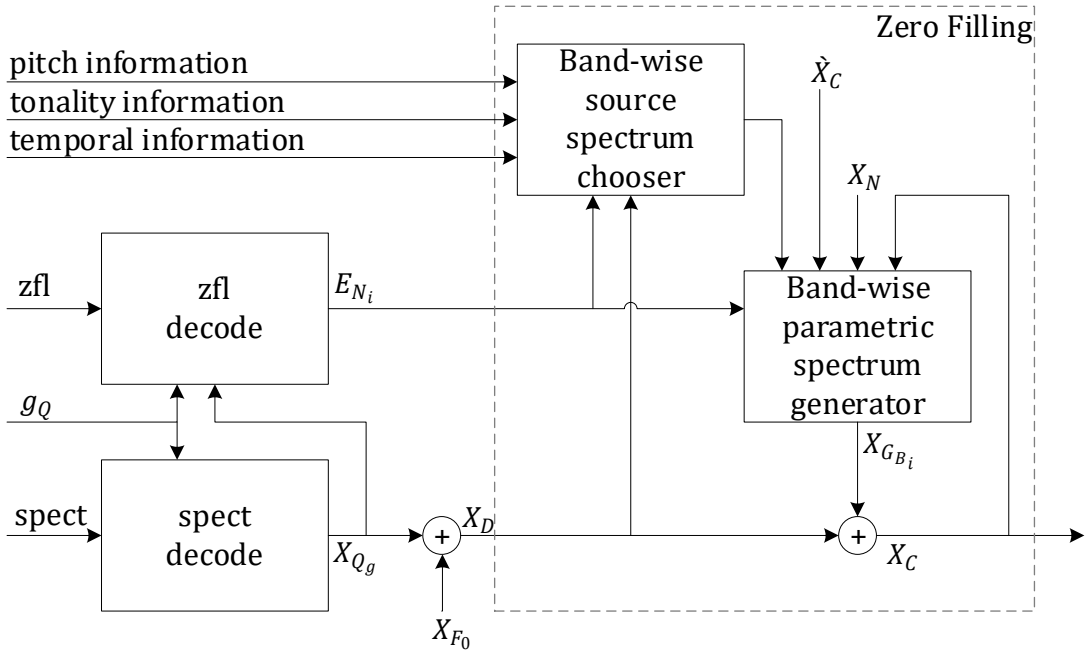

Figure 7.3 Zero Filling

Considering the above, the following parameters are adaptively found:

- optimal long copy-up distance $\dot{d}_C$
- minimum copy-up distance $\check{d}_C$
- minimum copy-up source start $\check{s}_C$
- copy-up distance shift Δ_C (that is used for modifying copy-up distance at LF)

The Zero Filling is depicted in Figure 7.3. The "pitch information" is the average pitch lag $\bar{d}_{F_0}$. The "tonality information" is the HF tonality flag ϕ_H. The "temporal information" is the information if TNS is active. The spectrum $\grave{X}_C$ is X_C from the previous frame. X_N is a random noise spectrum. The loop from X_C to the Band-wise parametric spectrum generator shows that X_C is band-wise constructed and that lower sub-bands are used for generating higher sub-bands. $X_{G_{B_i}}$ is generated for the sub-band B_i from $\grave{X}_C$, X_N and X_C, depending on the output of "Band-wise source spectrum chooser" and scaled depending on E_{N_i}.

7.4.1 Determination of the optimal copy-up distance and shift

The optimal copy-up distance $\dot{d}_C$ determines the optimum for the distance from the source to the destination spectrum when filling zero quantized lines using the spectrum copy-up. The

value of $\dot{d}_C$ is between the minimum $\dot{d}_{\check{C}}$, that is set to the equivalent of 5600 Hz, and the maximum $\dot{d}_{\hat{C}}$, that is set to the equivalent of 6225 Hz.

The distance between harmonics in the MDCT spectrum $\Delta_{X_{F_0}}$ is calculated from the average pitch lag $\bar{d}_{F_0}$. The distance between harmonics $\Delta_{X_{F_0}}$ is not necessarily an integer. If $\bar{d}_{F_0} = 0$, meaning that no harmonicity was found, then $\Delta_{X_{F_0}}$ is set to zero.

The percentual change $\Delta_{\bar{d}_{F_0}}$ of $\bar{d}_{F_0}$ between the previous frame and the current frame is also calculated. It is used in the decision for the optimal copy-up distance.

The value of $d_{C_{F_0}}$ is the minimum multiple of the harmonic distance $\Delta_{X_{F_0}}$ larger than the minimal optimal copy-up distance $\dot{d}_{\check{C}}$:

$$d_{C_{F_0}} = \left\lfloor \Delta_{X_{F_0}} \left\lceil \frac{\dot{d}_{\check{C}}}{\Delta_{X_{F_0}}} \right\rceil + 0.5 \right\rfloor$$

The value of $d_{C_{F_0}}$ is one of the possible values that will be assigned to $\dot{d}_C$. If $\Delta_{X_{F_0}}$ is zero then $d_{C_{F_0}}$ is not used.

Another candidate d_{C_ρ} for $\dot{d}_C$ is calculated from an autocorrelation of a magnitude spectrum estimate. The magnitude spectrum Z_C is estimated from the MDCT spectrum X_D:

$$Z_C[n] = \sqrt{\sum_{m=-2}^{2} (X_D[n+m])^2}$$

The starting TNS spectrum line plus the TNS order is denoted as i_T, and it is equivalent to 1000 Hz.

If TNS is inactive, i_{C_S} is set to $\left\lfloor 2.5\Delta_{X_{F_0}} \right\rfloor$. If TNS is active, i_{C_S} is set to i_T, additionally lower bound by $\left\lfloor 2.5\Delta_{X_{F_0}} \right\rfloor$ if HFs are tonal.

A normalized correlation of the estimated magnitude spectrum is calculated:

$$\rho_C[n] = \frac{\sum_{m=0}^{L_C-1} Z_C[i_{C_S}+m] Z_C[i_{C_S}+n+m]}{\sqrt{\left(\sum_{m=0}^{L_C-1} Z_C[i_{C_S}+m] Z_C[i_{C_S}+m]\right)\left(\sum_{m=0}^{L_C-1} Z_C[i_{C_S}+n+m] Z_C[i_{C_S}+n+m]\right)}}$$

$$\dot{d}_{\check{C}} \le n \le \dot{d}_{\hat{C}}$$

The length of the correlation L_C is set to the maximum value allowed by the available spectrum, but limited to the length equivalent of 5000 Hz.

Principally, we are searching for n that maximizes the correlation between the copy-up source $Z_C[i_{C_S}+m]$ and the destination $Z_C[i_{C_S}+n+m]$, where $0 \le m < L_C$. We choose d_{C_ρ} among

$\dot{d}_{\check{C}} \le n \le \dot{d}_{\hat{C}}$ where ρ_C has the first peak and is above its mean, that is: $\rho_C\left[d_{C_\rho}-1\right] \le \rho_C\left[d_{C_\rho}\right] \le \rho_C\left[d_{C_\rho}+1\right]$ and $\rho_C\left[d_{C_\rho}\right] \ge \frac{\sum \rho_C[n]}{\dot{d}_{\hat{C}}-\dot{d}_{\check{C}}}$.

If TNS is active $\dot{d}_C = d_{C_\rho}$.

If TNS is inactive $\dot{d}_C = \mathcal{F}_C\left(\rho_C, d_{C_\rho}, d_{C_{F_0}}, \grave{d}_C, \grave{\rho}_C[\grave{d}_C], \Delta_{\bar{d}_{F_0}}, \grave{\phi}_{T_C}\right)$, where $\grave{\rho}_C$ is the normalized correlation of Z_C and $\grave{d}_C$ the optimal distance in the previous frame. The flag $\grave{\phi}_{T_C}$ indicates if there was change of tonality in the previous frame that could affect copy-up. The function $\mathcal{F}_C$ was heuristically found. It returns either d_{C_ρ}, $d_{C_{F_0}}$ or $\grave{d}_C$. The decision which value to return in $\mathcal{F}_C$ is primarily based on the values $\rho_C\left[d_{C_\rho}\right]$, $\rho_C\left[d_{C_{F_0}}\right]$ and $\rho_C[\grave{d}_C]$. If the flag $\grave{\phi}_{T_C}$ is true and $\rho_C\left[d_{C_\rho}\right]$ or $\rho_C\left[d_{C_{F_0}}\right]$ are valid then $\rho_C[\grave{d}_C]$ is ignored. The values of $\grave{\rho}_C[\grave{d}_C]$ and $\Delta_{\bar{d}_{F_0}}$ are used in rare cases.

In more details $\mathcal{F}_C$ is defined with the following decisions:

- d_{C_ρ} is returned if $\rho_C\left[d_{C_\rho}\right] > \max\left(\rho_C\left[d_{C_{F_0}}\right] + \tau_{d_{C_{F_0}}}, \rho_C[\grave{d}_C] + \tau_{\grave{d}_C}\right)$, where $\tau_{d_{C_{F_0}}}$ and $\tau_{\grave{d}_C}$ are adaptive thresholds that are proportional to $\left|d_{C_\rho} - d_{C_{F_0}}\right|$ and $\left|d_{C_\rho} - \grave{d}_C\right|$ respectively
- otherwise $d_{C_{F_0}}$ is returned if $\rho_C\left[d_{C_{F_0}}\right] > \rho_C[\grave{d}_C] + 0.2$
- otherwise d_{C_ρ} is returned if $\grave{\phi}_{T_C}$ is set and $\rho_C\left[d_{C_\rho}\right] > 0$
- otherwise $d_{C_{F_0}}$ is returned if $\grave{\phi}_{T_C}$ is set and the value of $d_{C_{F_0}}$ is valid, that is if there is a meaningful pitch lag
- otherwise $d_{C_{F_0}}$ is returned if $\grave{\rho}_C[\grave{d}_C] < 0.1$ and $\Delta_{\bar{d}_{F_0}} < 0.1$ and the value of $d_{C_{F_0}}$ is valid, that is if there is a meaningful pitch lag
- otherwise $\grave{d}_C$ is returned

The flag $\grave{\phi}_{T_C}$ is set to true if TNS is active or if $\rho_C[\dot{d}_C] < 0.7$ and the tonality is low, the tonality being low if the HF tonality flag ϕ_H is false or if $\bar{d}_{F_0}$ is zero. The value set to $\grave{\phi}_{T_C}$ is used only in the following frame.

The copy-up distance shift Δ_C is set to the harmonic distance $\Delta_{X_{F_0}}$ unless the optimal copy-up distance $\dot{d}_C$ is equivalent to $\grave{d}_C$ and $\Delta_{\bar{d}_{F_0}} < 0.1$, in which case Δ_C is set to the same value as in the previous frame, making it constant over the consecutive frames. If TNS is active Δ_C is not used.

The minimum copy-up source start $\check{s}_C$ is set to i_T if TNS is active, additionally lower bound by $\left\lfloor 2.5\Delta_{X_{F_0}} \right\rfloor$ if HFs are tonal, or set to $\lfloor 2.5\Delta_C \rfloor$ if TNS is not active.

The minimum copy-up distance $\check{d}_C$ is set to $\lceil \Delta_C \rceil$ if TNS is inactive. If TNS is active $\check{d}_C$ is set to $\check{s}_C$ if HF are not tonal or to $\left\lceil \Delta_{X_{F_0}} \left\lfloor \frac{\check{s}_C}{\Delta_{X_{F_0}}} \right\rfloor \right\rceil$ if HFs are tonal.

7.4.2 Zero Filling source choice

Using $X_N[-1] = \text{short}\left(\sum_n \text{short}\left(2n\left|X_Q[n]\right|\right)\right)$ as an initial condition, a random noise spectrum X_N is constructed as $X_N[n] = \text{short}(31821X_N[n-1] + 13849)$, where the function short truncates the result to 16 bit signed integer. The random noise spectrum X_N is then set to zero at the location of non-zero quantized lines in X_Q and the portions between the non-zero lines are windowed, in order to reduce the random noise near the coded MDCT lines. This process of creating the noise spectrum is coming from EVS [37, 120, 147].

For each sub-band B_i of length L_{B_i} starting at j_{B_i} the optimal Zero Filling source $X_{S_{B_i}}$ is found.

If TNS is not active and HFs are not tonal then the random noise spectrum X_N is used as the Zero Filling source for all sub-bands, otherwise X_N is used as the source for the sub-bands where other sources are empty or for the sub-bands which start below minimal copy-up destination: $j_{B_i} < \check{s}_C + \min\left(\check{d}_C, L_{B_i}\right)$. If TNS is not active then $\check{s}_C + \check{d}_C = \lfloor 2.5\Delta_C \rfloor + \lfloor \Delta_C \rfloor \sim 3.5\Delta_{X_{F_0}}$, which means that the copy-up can start already at the fourth harmonic.

If TNS is not active and HFs are tonal, the past reconstructed spectrum $\grave{X}_C$ is used as the Zero Filling source for the sub-bands which start below $\check{s}_C + \dot{d}_C$ and for which the ZFL is at least 12 dB above neighboring ZFLs. It was heuristically determined that this is helpful for some tonal items. $\grave{X}_C$ is used instead of the predicted spectrum X_{F_0} because LTP was not active for most of these items.

For other cases distance d_C is found and $X_C[s_C + m]$ or a mixture of the $X_C[s_C + m]$ and $X_N[s_C + d_C + m]$ is used as the optimal Zero Filling source $X_{S_{B_i}}[m]$ that starts at j_{B_i}, where $s_C = j_{B_i} - d_C$ and $0 \le m < L_{B_i}$. More precisely if TNS is active, but starts only at 4500 Hz and HFs are not tonal, the mixture of the $X_C[s_C + m]$ and $X_N[s_C + d_C + m]$ is used if $\check{s}_C + \check{d}_C \le j_{B_i} < \check{s}_C + \dot{d}_C$; in other cases only $X_C[s_C + m]$ is used. It should be noticed that $X_C[s_C + m]$ may at this point be a combination of different portions of X_D, $\grave{X}_C$ and X_N. If $j_{B_i} \ge \check{s}_C + \dot{d}_C$ then $d_C = \dot{d}_C$. If TNS is active then the smallest positive integer n is found so that $j_{B_i} - \frac{\dot{d}_C}{n} \ge \check{s}_C$ and d_C is set to $\frac{\dot{d}_C}{n}$. If TNS is not active, the smallest positive integer n is found so that $j_{B_i} - \dot{d}_C + n \cdot \Delta_C \ge \check{s}_C$ and d_C is set to $\dot{d}_C - n \cdot \Delta_C$. This choice of the copy-up distance satisfies the initial observations stated at the beginning of 7.4.

For the sub-bands below the Zero Filling start frequency 700 Hz the spectrum X_D is copied to X_C, or equivalently $X_{S_{B_i}}$ is set to zero at these frequencies. The start frequency may be decreased for lower bitrates.

7.4.3 Scaling of the Zero Filling source

The ZFLs are smoothed: $E_{N_{1,i}} = \frac{E_{N_{i-1}} + 7E_{N_i}}{8}$ and $E_{N_{2,i}} = \frac{7E_{N_i} + E_{N_{i+1}}}{8}$.

The scaling factor a_{C_i} is calculated for each sub-band B_i:

$$a_{C_i} = g_Q \sqrt{\frac{L_{B_i}}{\sum_{m=0}^{L_{B_i}-1} \left(X_{S_{B_i}}[m]\right)^2}}$$

Additionally the scaling is limited with the factor b_{C_i} calculated as:

$$b_{C_i} = \frac{2}{\max\left(2, a_{C_i} \cdot E_{N_{1,i}}, a_{C_i} \cdot E_{N_{2,i}}\right)}$$

The optimal Zero Filling source $X_{S_{B_i}}[m]$ $(0 \leq m < L_{B_i})$ is split in two halves and each half is scaled, the first half with $g_{C_{1,i}} = b_{C_i} \cdot a_{C_i} \cdot E_{N_{1,i}}$ and the second with $g_{C_{2,i}} = b_{C_i} \cdot a_{C_i} \cdot E_{N_{2,i}}$, producing $X_{G_{B_i}}[m]$. The scaling factor b_{C_i} limits increase of energy in $X_{G_{B_i}}$ compared to the $X_{S_{B_i}}$. Having the limit is beneficial when the source is very sparse, to avoid introducing too strong tonality in high frequencies. In most cases $a_{C_i} \cdot E_{N_{1,i}}$ and $a_{C_i} \cdot E_{N_{2,i}}$ are smaller than 2.

The scaled source $X_{G_{B_i}}[m]$ is added to $X_D[j_{B_i} + m]$ producing $X_C[j_{B_i} + m]$ $(0 \leq m < L_{B_i})$.

7.4.4 Quantization of the ZFLs

The same way as in the random noise spectrum X_N, the coefficients in $\dot{X}_M$ at the location of the non-zero quantized lines in X_Q are set to zero and the portions between the non-zero quantized lines are windowed in $\dot{X}_M$, producing $\dot{X}_N$. An example of the process is depicted in Figure 7.4.

The ZFLs E_{N_i} are calculated from $\dot{X}_N$:

$$E_{N_i} = \frac{1}{g_Q} \sqrt{\frac{\sum_{m=j_{B_i}}^{j_{B_i}+L_{B_i}-1} \left(\dot{X}_N[m]\right)^2}{L_{B_i}}}$$

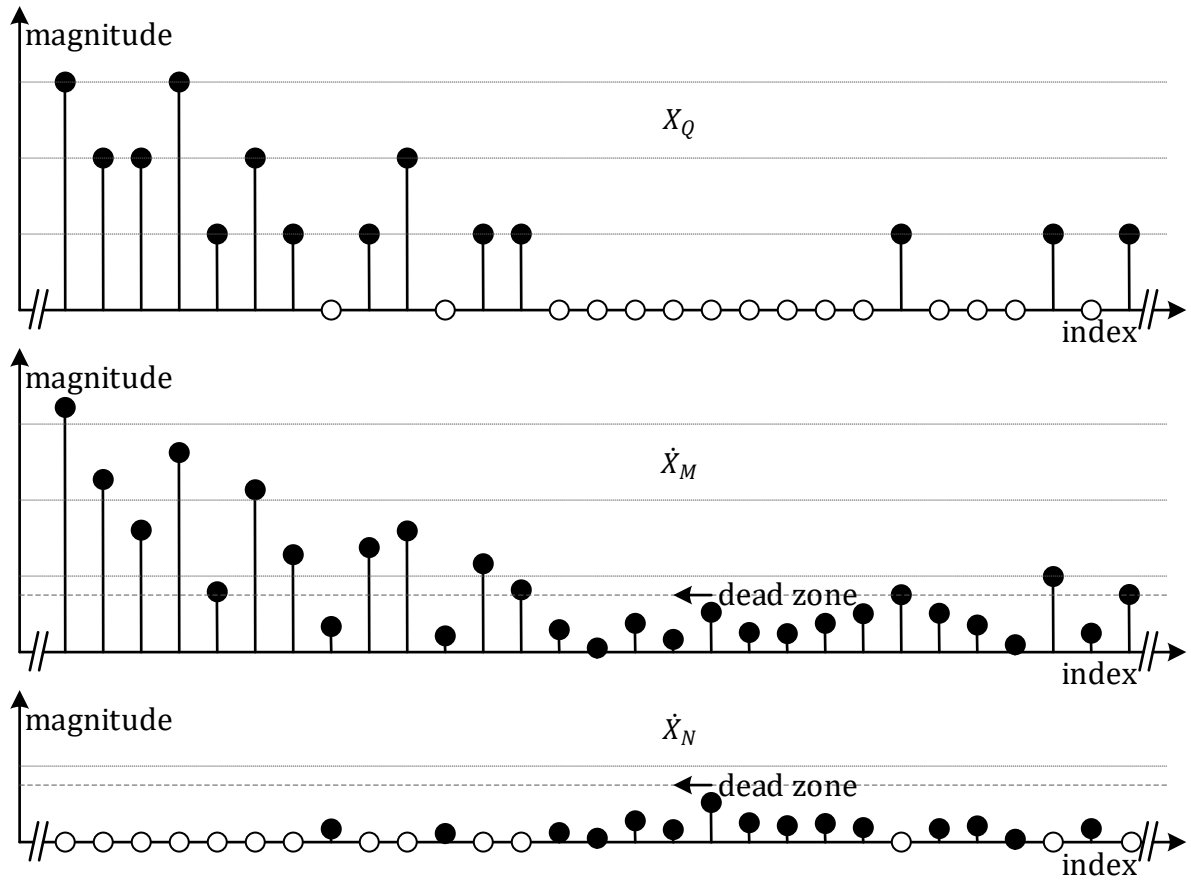

Figure 7.4 Construction of the spectrogram $\dot{X}_N$ for calculating the ZFLs E_{N_i}

The ZFLs are quantized using step size 1/8 and limited to 6/8. Separate ZFLs are coded only for the MDCT sub-bands above 3000 Hz that are completely quantized to zero. The ZFLs are transformed before arithmetically coding them. There are three transform modes:

- Absolutely

$$\acute{E}_{N_i} = \begin{cases} E_{N_i} - 2, E_{N_i} \leq 4 \\ \left(1 - 2\left(E_{N_i}\&1\right)\right)\left\lfloor \frac{E_{N_i}+1}{2} \right\rfloor \end{cases}$$

- Differentially

$$\acute{E}_{N_i} = E_{N_i} - \acute{E}_{N_{i-j}}$$

  where $\acute{E}_{N_{i-j}}$ is the last previous ZFL explicitly coded. The first zero band is differentially transformed relative to the energy of the previous non-zero band in X_Q.

- Prediction residual

$$\acute{E}_{N_i} = E_{N_i} - \grave{E}_{N_i}$$

  where $\grave{E}_{N_i}$ are the ZFLs from the previous frame, rescaled using the ratio of the global gain g_Q from the previous and the current frame and requantized.

The transform mode which requires least bits is chosen.

Additionally one ZFL E_{N_S} is calculated as the mean of all E_{N_i} from zero sub-bands below 3000 Hz and from zero sub-bands above 3000 Hz where E_{N_i} is quantized to zero. The low level E_{N_S} is quantized with the step size 1/16 and limited to 3/16. The individual E_{N_i} in the low level ZFLs are set depending either on the neighboring explicitly coded ZFLs or depending on the ZFLs in the previous frame and then scaled based on E_{N_S}. It is explicitly signaled with one bit in the bit-stream how to set the individual low level ZFLs.

The ZFLs for non-zero sub-bands is not coded explicitly, but estimated assuming value of 0.25 for zero lines that are then windowed as in the random noise spectrum.

Notice that each ZFL above 3000 Hz is individually coded; there is no downsampling as in SNS and the ZFLs have higher spectral resolution than the SNS scale factors.

7.4.5 Examples of iBPC

Examples of iBPC will be presented in the following spectrogram figures for different audio signal types. For clean speech examples, spectrograms of the coded pulses will also be shown, as it complements the Zero Filling and partially acts as a bandwidth extension for voiced segments. All spectrograms show the portions as close as possible to how they appear in the final decoded signal, meaning that they are properly scaled and shaped via SNS.

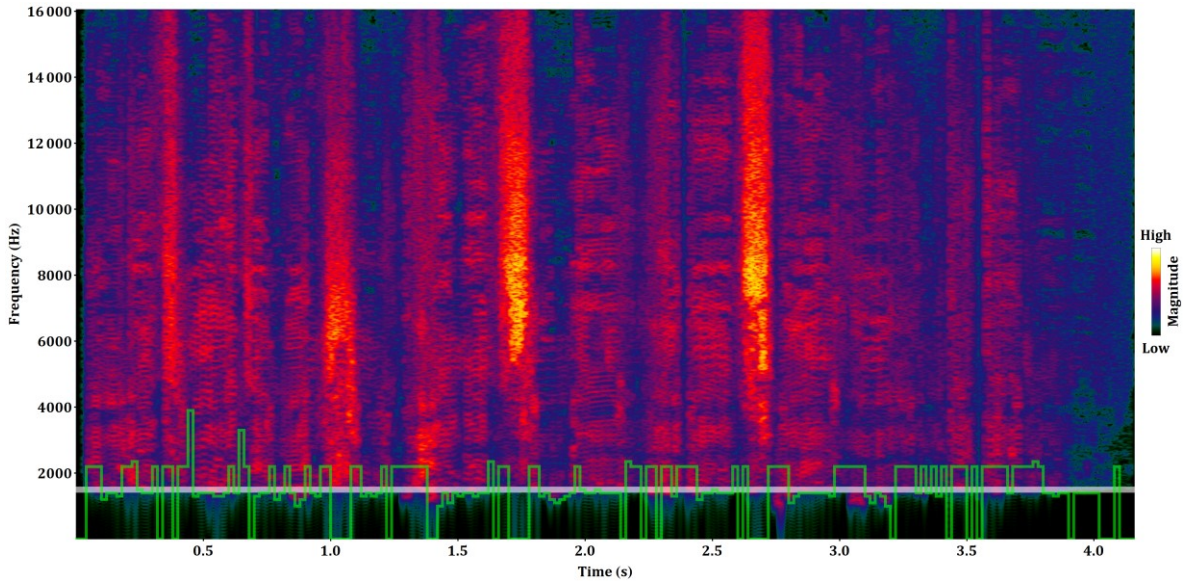

Figure 7.5 Zero Filling for German male speaker [ger_m] recording

The white line in the spectrograms is the usual start of iBPC at 1.5 kHz. iBPC can start lower, as low as 700 Hz, if there are contiguous zeros around this frequency. This adaptive lowering of the iBPC start is taken from the similar adaptation in the noise filling in EVS [37]. However, there are rarely big zero portions below 1.5 kHz at 24.4 kbps.

The green line shows where the copy-up starts in each frame. The green line at 0 means that there is no copy-up in that frame. In Figure 7.5 it can be seen that the copy-up is used for almost all voiced segments and that sibilants are generated using the random noise.

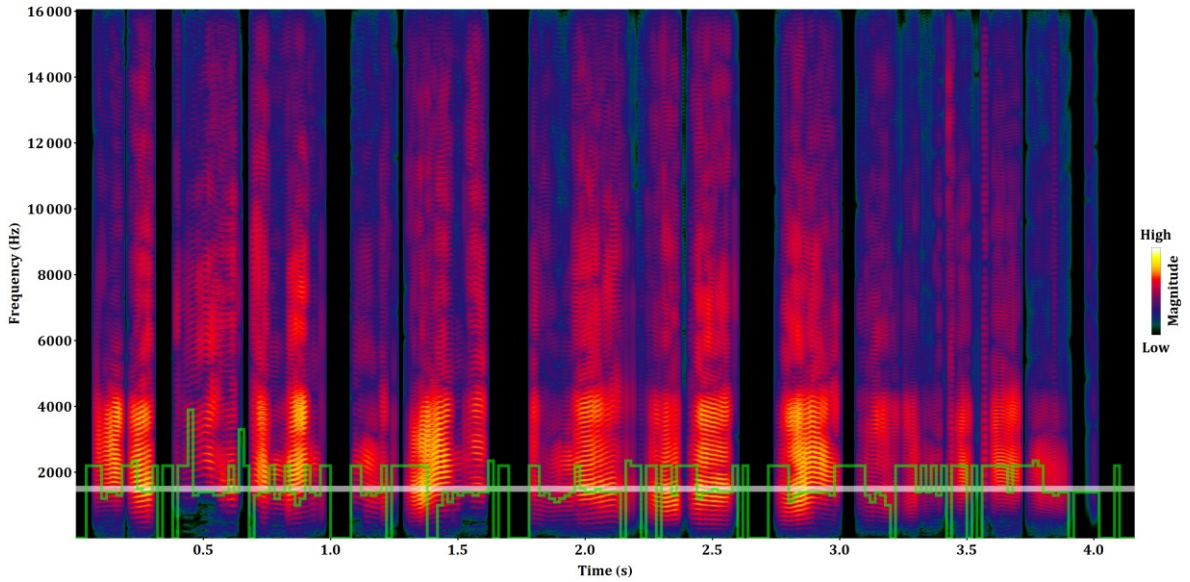

Figure 7.6 Decoded pulses for the German male speaker recording

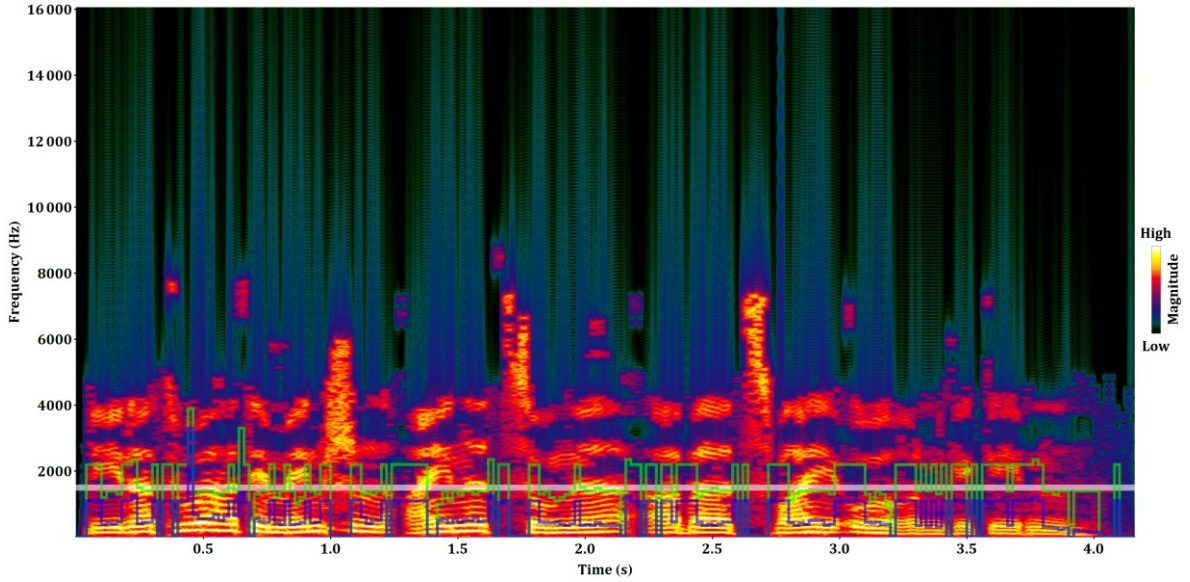

Figure 7.7 Decoded spectrum without the Zero Filling and without the pulses

The blue line is the copy-up source for the copy-up start. In other words, the Zero Filling for the sub-band starting at the green line is obtained from the portion of the spectrum starting at the blue line. As explained in 7.4.2, the source for each sub-band is adaptively determined and the sources may overlap or one source may be a subset of another one. The copy-up source is shown only in some spectrograms.

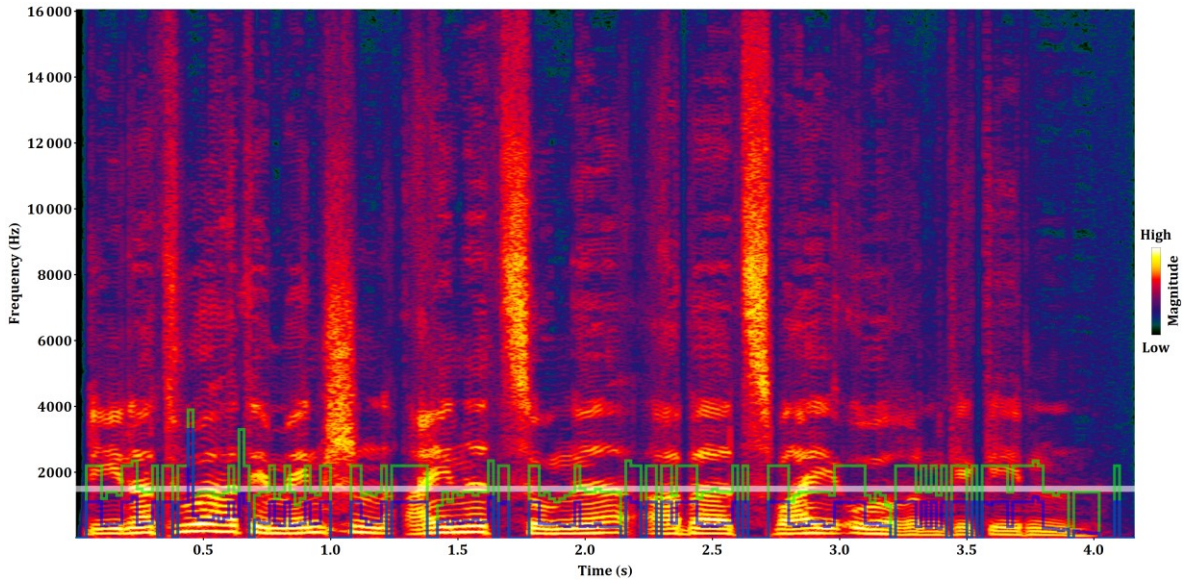

Figure 7.8 Decoded spectrum with the Zero Filling but without the pulses

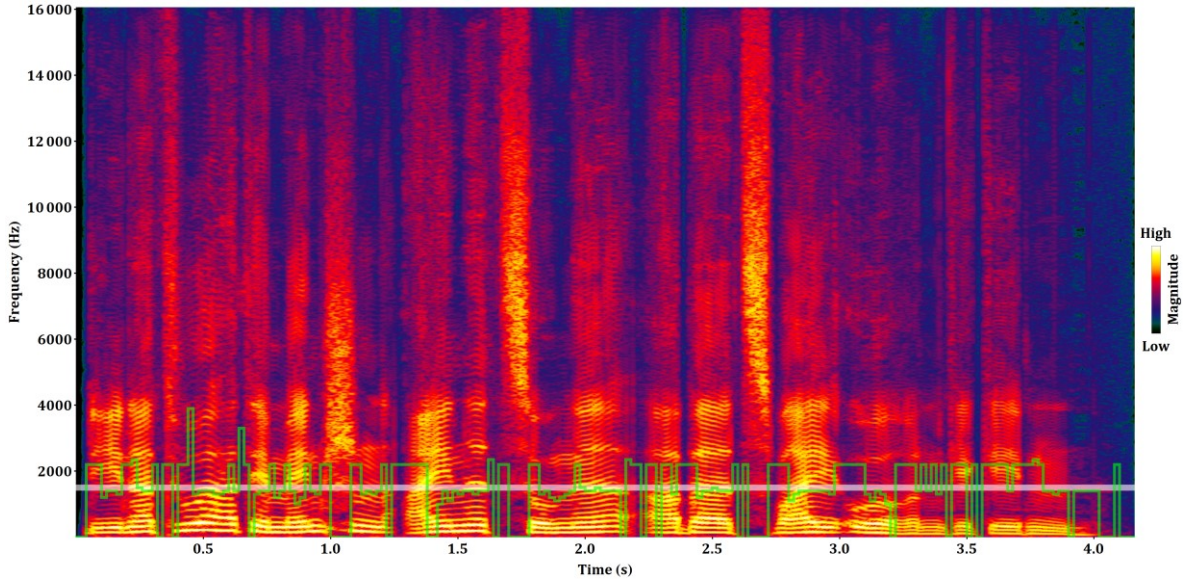

Figure 7.9 Completely decoded signal with the Zero Filling and with the pulses

It is evident from the above figures that the Zero Filling and the pulse coding are complementing each other for speech signals, similarly as the non-linear function and the noise modulation in the TD bandwidth extension from [86].

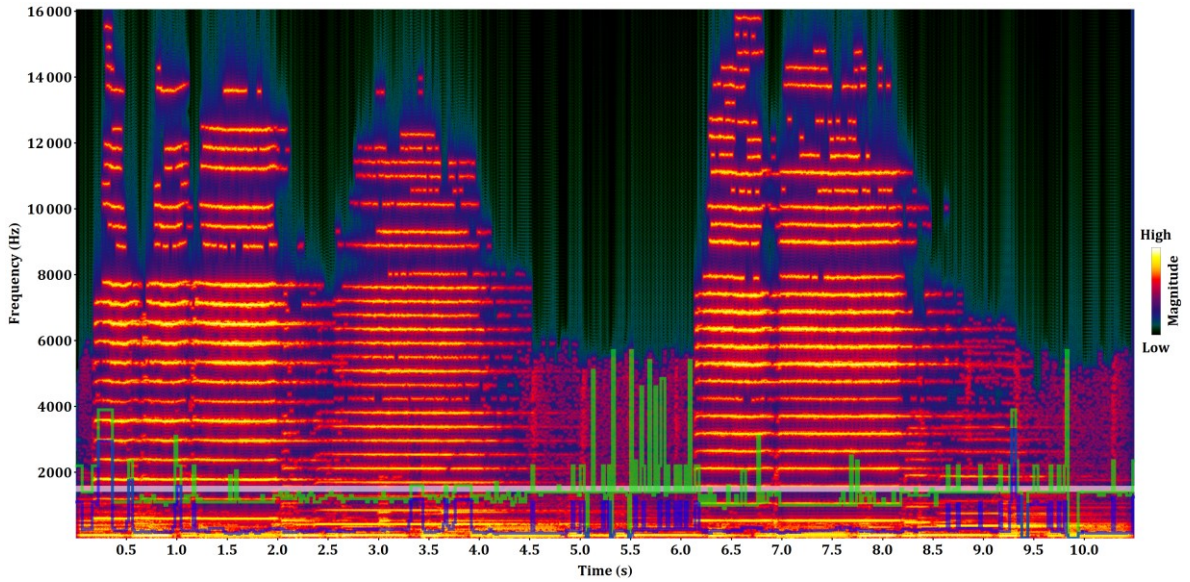

Figure 7.10 Decoded spectrum without the Zero Filling of a trumpet solo [hanco]

In Figure 7.10 the decoded spectrogram without the Zero Filling is shown for a trumpet solo. No pulses are found in this sample. The strongest harmonics are coded across the whole bandwidth, but there is not enough bits to code every harmonic across the whole time sample. At this point the adaptive copy-up of iBPC comes into play. In can be seen in Figure 7.11 that iBPC correctly reconstructs the missing and the discontinued harmonics.

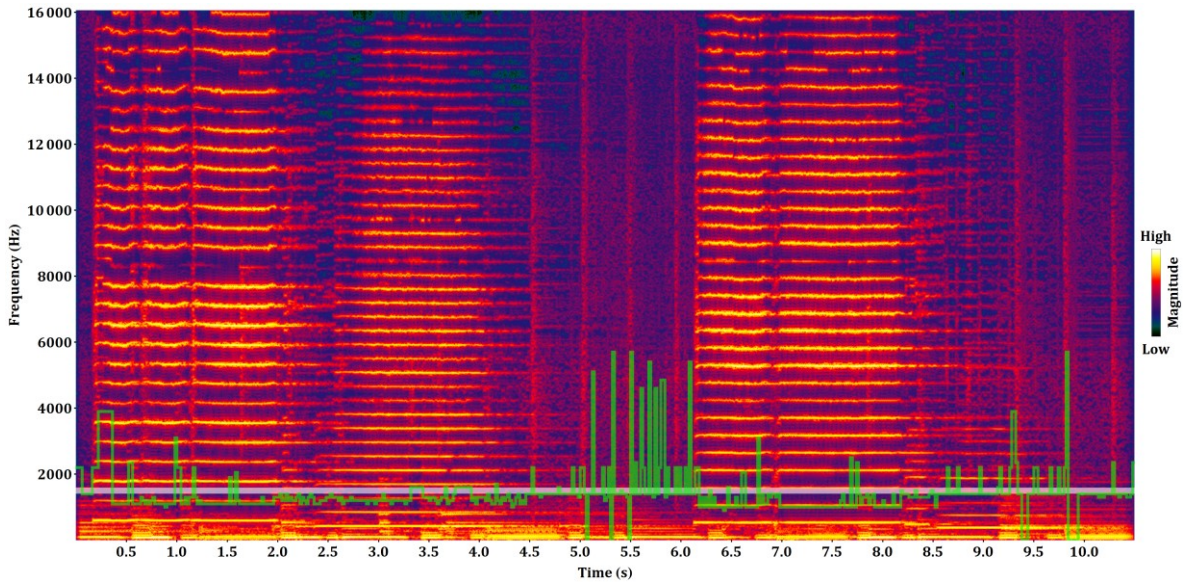

Figure 7.11 Completely decoded signal with the Zero Filling of a trumpet solo

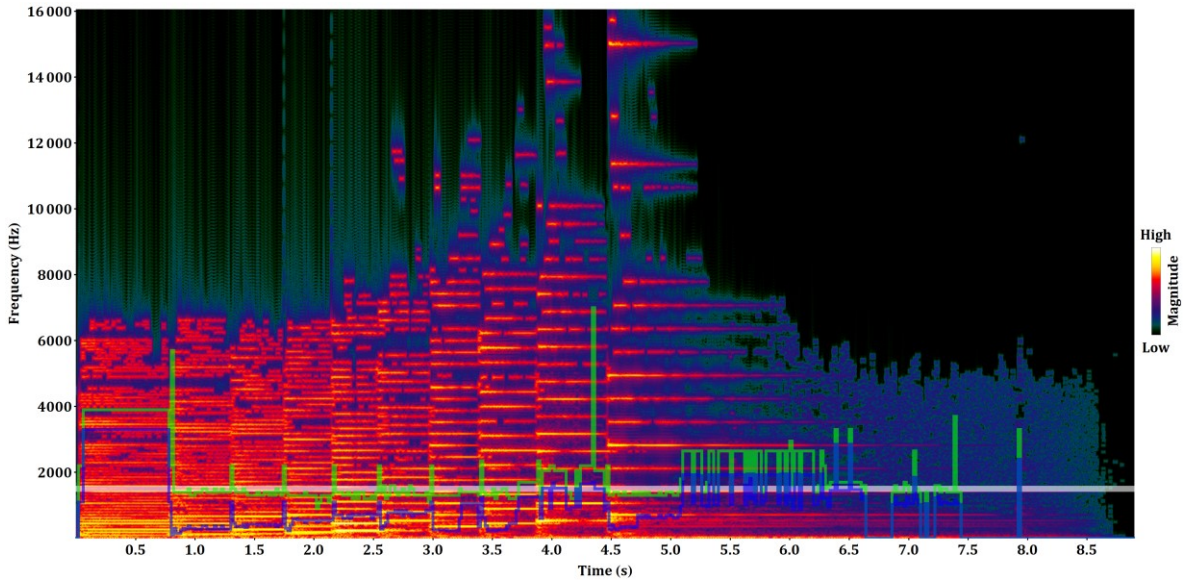

Figure 7.12 Decoded spectrum without the Zero Filling of a harpsichord recording

Another example of the adaptability of iBPC is shown on an example of a harpsichord single note progression recording in Figure 7.12 and Figure 7.13. In this example it can be seen, that as the fundamental frequency and the distance between the harmonics increase, the spectrum becomes sparser, requires less bits for coding and higher harmonics may be coded directly by quantizing the MDCT spectrum. The copy-up distance is chosen so that the copied harmonics appear at the expected position and complement the explicitly coded ones. The pulse coding in this example is only active in few frames and therefore its analysis is not shown.

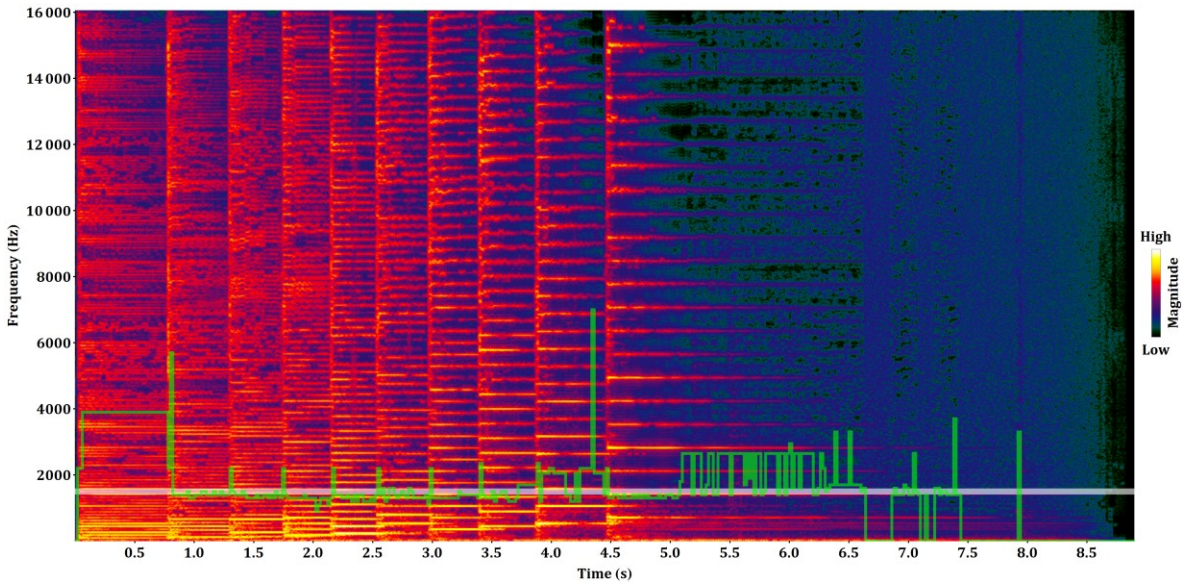

Figure 7.13 Completely decoded signal with the Zero Filling of a harpsichord recording

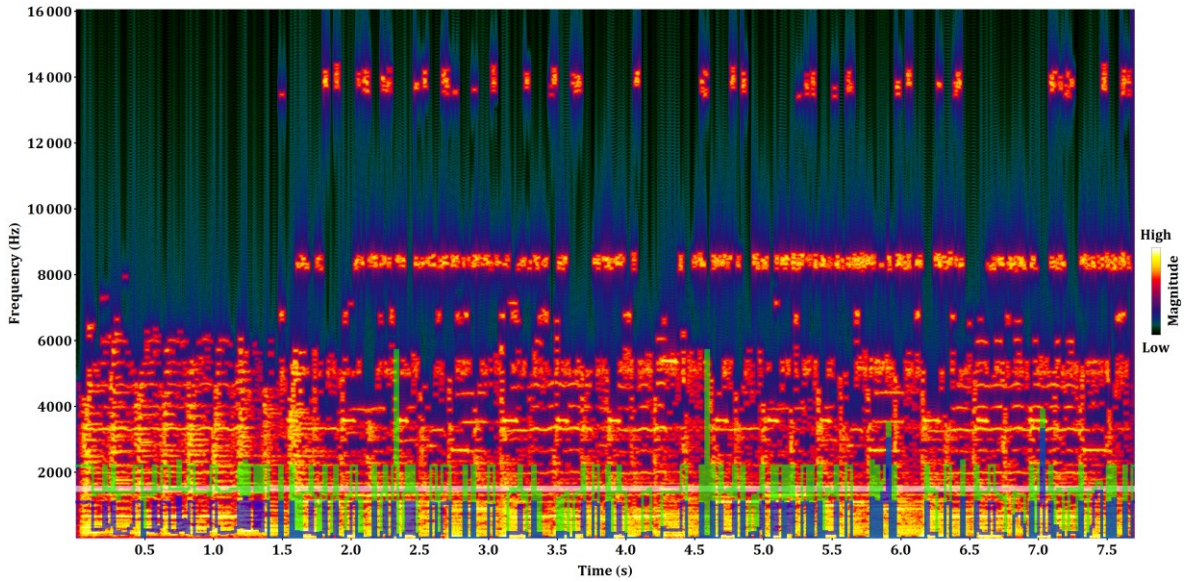

Figure 7.14 Decoded spectrum without the Zero Filling of an excerpt from Ommadawn [omma]

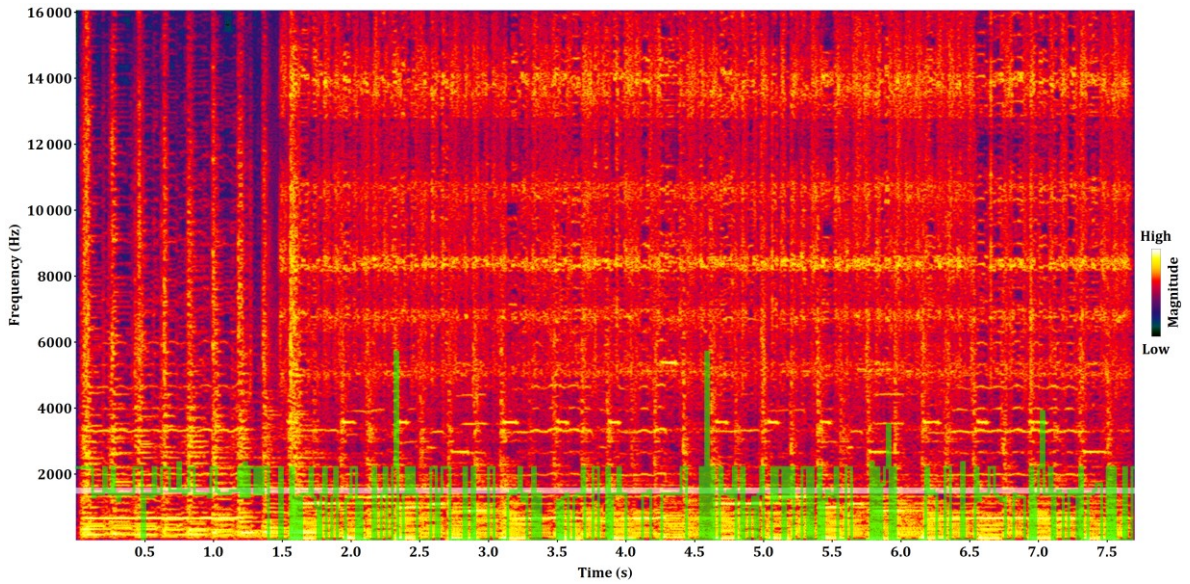

Figure 7.15 Completely decoded signal with the Zero Filling of an excerpt from Ommadawn

It is not just the tonal parts that are waveform preserving coded at high frequencies with iBPC. An example of preserving tambourine sounds, played in the polyphonic recording of Mike Oldfield's Ommadawn, is shown in Figure 7.14. Again there is no significant influence of the pulse coding on the analyzed Ommadawn sample.

7.5 Advantages of the new contributions

iBPC was developed having in mind bitrate constraint for coding a frame of the input audio signal. In IVA this constraint is very strict, requiring constant bitrate.

Different to the previous sophisticated bandwidth extensions, the parameter coding of iBPC is deeply integrated in the waveform preserving spectrum coder (Figure 7.1 and Figure 7.2). The

bitrate is dynamically distributed within the rate-distortion loop between the parametric and the spectrum coding, still preserving the total bitrate constraint. There is no need for changing the loop conditions from EVS. The adaptation of the quantization step also remained as in EVS. The choice which spectrum portions to code parametrically is also integrated into the loop and dependent on the output of the spectrum quantizer and not just on measurements of tonality as in the state of the art. This allows further tuning of the single rate-distortion loop, without need to adapt parameters like the start frequency of the parametric coding.

The coding of iBPC parameters is also unique as:

- there are variable number of parameters
- locations where they are applied is adaptive
- its bitrate is not constant, but below an adaptive limit in each frame

and yet there is no explicit coding of the number of the parameters nor the locations. The number of the parameters, the associated location and even number of used bits is deduced based on the explicitly coded MDCT coefficients.

The spectrum quantization depending decision of the Adaptive band zeroing is different from the tonality based state of the art. Notice that the tonality ϕ_H is used just for making the band zeroing stable over time. The Adaptive band zeroing reduces internal "recoder" iterations in the rate-distortion loop. It also allows that more HF MDCT lines are explicitly coded by setting some of LF bands to zero.

The Zero Filling generates spectrum iteratively, reusing its output. This allows effective handling of two known requirements: the spectrum portion being copied should be big and the copy source should not start at very low frequencies. The explicitly coded spectral coefficients in some cases don't provide big enough continuous source. Spectral holes often appear even at LF of harmonic signals and simple noise filling from the state of the art is not a good way of replacing the missing harmonics. The copy-up starts at very low frequency. At 24.4 kbps it starts already around 1.5 kHz. For an example the IGF start frequency is 6.4 kHz for 24.4 kbps in EVS [37]). Thus, a mixture of the explicitly coded spectrum and the copy-up may be used as a source in iBPC.

Copying from the spectrum of the previous spectrum complements generation of missing tonal portions at LF. The LTP generated spectrum could also be used for the Zero Filling at LF or even at HF, but whenever LTP is active the LF are already well coded and the HF are in most cases not well predicted. Activating LTP just for the Zero Filling would be to big increase in the computational complexity. LTP output was not used for the Zero Filling in IVA.

The coded information for the Zero Filling is minimal, containing just the zero sub-band energies and the single tonality flag. This is achieved by reusing parameters from other tools.

Even though the adaptive choice between the noise filling and the copy-up in the Zero Filling is principally independent of the encoding and the decoding of iBPC, they are conceptually connected. The effectiveness of the Adaptive band zeroing is dependent on premise that the Zero Filling will correctly generate zero bands even for tonal spectrum portions and vice versa.

8 HPF

Post-filtering is a common technique in CELP-type codecs for increasing the perceptual impression [16, 139]. Harmonic Post-Filter (HPF) has recently also gained ground in low-bitrate and low-latency FD coding, for example in EVS [37, 114, 145]. Short window and limited bitrate increase quantization noise between harmonics of a signal coded in the MDCT. HPF acts as a comb-filter, reducing noise between harmonics in the decoded TD signal by filtering the signal across the frame borders. HPF may also increase amplitudes of the harmonics. Sometime the post-filter is accompanied by a pre-filter that reduces amplitudes of harmonics in expectance that an MDCT domain codec would need less bits in coding the pre-filtered signal.

8.1 State of the art

In [162, 163] an adaptive FIR filter $y[n] = a_0 x[n] + \sum_{i=1}^{K} a_i x\left[n - \sum_{j=1}^{i} m_j\right]$ is used for speech enhancement. The parameters m_i are pitch period lengths obtained from glottal movements measurements of an accelerometer. The parameters a_i are fixed and defined by a low-pass windowing function (e.g. Hann or Blackman). The filter taps span several pitch periods, also including changing pitch periods in speech signals. A problem that arises is that when $m_j < n < m_0$ then $x\left[n - \sum_{j=1}^{i} m_j\right]$ doesn't belong to the j-th pitch period. This was named "Overload Problem" and was addressed by setting $a_j = 0$ for $m_j < n$. Voiced/unvoiced detection is also required, so that the first or the last pitch period length from the voiced part is used in the transitions. In the unvoiced portions, the filter is either disabled or continues using the last pitch length.

In [164] a bandwidth expansion and compression/reduction method called Time Domain Harmonic Scaling is used to implement time varying adaptive comb filter, which in fact can be seen as another way of implementing the adaptive FIR filter from [163] with specific window of also adaptive length dependent on pitch.

The filter taps span over at least three pitch periods in both FIR methods, which reduces possibility of modeling rapid pitch changes. The increase of harmonicity that they introduce is fixed and signal independent.

In [113] a pre-/post-filter approach divides the frame into non-overlapping sub-frames of flexible length, where the sub-frame borders are at minimal energy locations. For each sub-frame pitch information is obtained. Post-filter $y[n] = x[n] + \sum_{p=-m}^{m} b_p y[n-d+p]$ is used, where d is pitch estimated in a sub-frame, m is a desired filter order and b_p are prediction coefficients obtained with a closed-loop search. The adaptive filter coefficients b_p allow modeling of the harmonicity level and the fractional pitch lag.

In [113] the pre-filter reduces the harmonic part in the coded signal and consequently limits the quality of coded harmonic components and the post-filter efficiency. All parameters of the post-filter are estimated for each sub-frame and transmitted, significantly increasing the bitrate. Even though the parameters b_p allow implicit modeling of many properties of the filter, the closed-loop search for their values is computationally complex and transmitting all of them implies unneeded burden on the bitrate.

Big problem in all methods from [113, 162–164] is that there is no smoothing at borders where the pitch changes.

In [165] HPF is run on the decoded signal divided in sub-frames of fixed length. A pitch analysis returns a correlation γ and a pitch P_0 per sub-frame. There are no details if the correlation is normalized. The gain g is derived from the correlation γ. The HPF $y[n] = x[n] + g_{-1}y[n-P_{-1}] + g_0 y[n-P_0]$ is run for each sub-frame, with g_0 changing from 0 towards g and g_{-1} changing from the gain in the previous sub-frame towards 0, where P_{-1} is equal to the pitch in the previous sub-frame.

The constant sub-frame length in [165] limits adaptation of the filter to pitch changes and there is no consideration of fractional pitch lags.

In EVS [37, 114] the harmonic filter with the transfer function:

$$H_E(z) = \frac{1 - a_H b_H g_E B(z,0)}{1 - b_H g_E B(z,\{d_E\}) z^{-\lfloor d_E \rceil}}$$

has coefficients derived from the pitch lag d_E and the gain value g_E, which are signal adaptive. The parameters a_H and b_H are constants. A new filter, that will be proposed, has similarities with H_E. The H_E parameters will be analyzed when presenting the new filter.

The post-filter parameters in [37, 114] are constant over a frame, where the frame is defined by a codec. A discontinuity at the frame borders is removed using a cross-fader or a similar method. A disadvantage, similar to most other methods, is its slow adaptation to signal changes because its parameters are bound to the codec's constant framing.

None of the listed state of the art models amplitude modulations of the harmonic components. The post-filter in [165] doesn't model amplitude changes, because g is proportional to the correlation limited between 0 and 1. The post-filter from [37, 114] also does not model well amplitude modulations because $g_E \leq 1$ and because it appears in both numerator (feedforward) and denominator (feedbackward). The reasoning why g and g_E don't model amplitude modulations will be made clear when analyzing the new filter.

8.2 HPF processing

The time domain signal x_D is obtained from $\bar{\bar{X}}_C$ as the output of IMDCT, where IMDCT consists of the inverse MDCT, windowing and adding of the overlapping parts [Figure 4.6]. HPF, that follows the pitch contour, is applied on x_D to reduce the quantization noise between harmonics and outputs x_H [Figure 4.6].

The HPF input for the current frame is $x_D[n], 0 \leq n < H_M$. The past HPF output samples $x_H[n], -\hat{d}_{F_0} \leq n < 0$, are also available. The IMDCT look-ahead samples of length $L_{\hat{O}}$, are also available, including time aliased portion of the inverse MDCT output.

HPF operates either in the constant or the variable-pitch mode. While obtaining the pitch contour, in the encoder, it is also determined if there are no significant changes in the harmonicity nor amplitude modulation and signaled accordingly if HPF should operate in the constant or the variable-pitch mode.

In the constant-pitch mode, pitch parameters still may change between consecutive frames. Thus, a smoothing is needed and is realized in the starting sub-interval of the current frame with the sub-interval length equal to the pitch lag d_H in the current frame. In the remaining of the frame, HPF with constant parameters is used. The pitch lag d_H used in the constant HPF is determined via the pitch search $\mathcal{F}_{F_0}$ [4.3.1] on the signal consisting of the current HPF input $x_D[n]$ for $0 \leq n < H_M$, and the past HPF output samples $x_H[n]$ for $-\hat{d}_{F_0} \leq n < 0$. The borders and locations are schematically presented relative to the MDCT windowing in Figure 8.1. For finding the constant pitch lag, $\mathcal{F}_{F_0}$ is called with $L_H = H_M$, $d_{\check{F}_0} = \dot{d}_{F_0}$, $d_{\check{F}_0} = \dot{d}_{F_0} - 2$, $d_{\hat{F}_0} = \dot{d}_{F_0} + 2$.

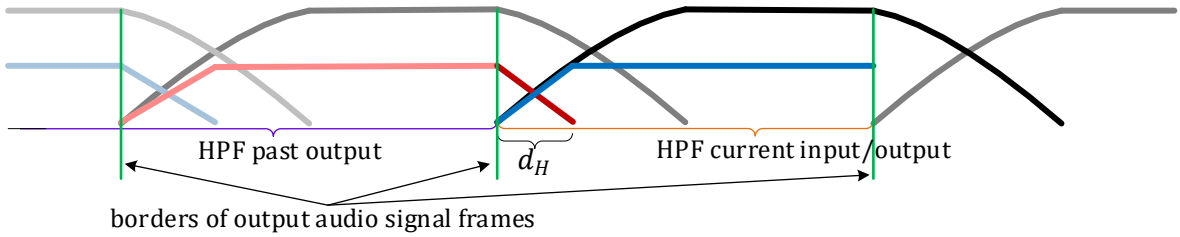

Figure 8.1 HPF in the constant-pitch mode

In the variable-pitch mode, the HPF input is split into N_{F_H} overlapping sub-intervals of length L_{F_H} with the hop size $H_{F_H} = L_{F_H}/2$, similar as in LTP. The smoothing is continuously performed over the overlapping sub-intervals. L_{F_H} is chosen so that no significant pitch change is expected within the sub-intervals. Additionally, it is requested that the frame length H_M is divisible by H_{F_H}. Hence, H_{F_H} is chosen so that it is smaller than $\dot{d}_{F_0}/2$ and that H_M is divisible by H_{F_H} and L_{F_H} is set to $2H_{F_H}$. One could also set H_{F_H} to the half of the minimum expected pitch within the output frame, but this was not used in IVA. In contrast to the LTP sub-interval splitting, HPF must have continuity at the end of the frame borders and there is not enough look-ahead (usually only time aliased output of the inverse MDCT) to do proper pitch search at the end of the current output frame. Not requesting that H_M is divisible by H_{F_H} would significantly complicate the process. The borders and locations in the variable-pitch mode are schematically presented relative to the MDCT windowing in Figure 8.2.

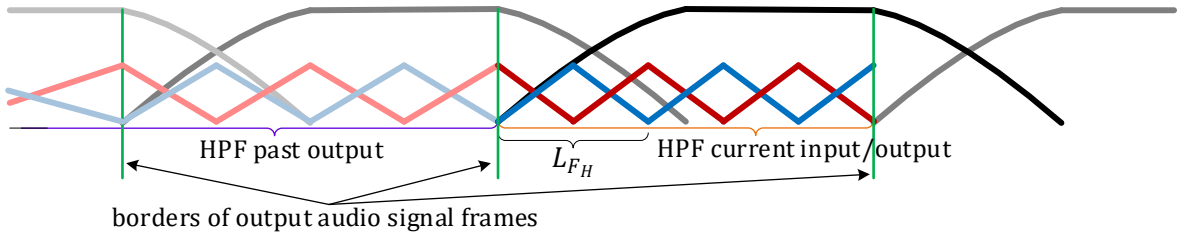

Figure 8.2 HPF in the variable-pitch mode

For smoothing the changes in the filter across the sub-intervals, the method of smoothing with cascaded filters from LC3 is used [79, 80, 144]. In the smoothing method from LC3, the sub-interval is first filtered using parameters from the past sub-interval and the first filter intensity is changed towards no filtering (identity filtering), followed by filtering the output of the first filter by a second filter using parameters from the current sub-interval and the intensity changing from no filtering towards full filtering. This smoothing method has one addition per cycle less than cross-fading of two parallel filters.

8.3 The adaptive filter

A gain adaptive HPF is used for the post-filtering, having the transfer function:

$$H_H(z) = \frac{1 - a_H b_H h_H B(z,0)}{1 - b_H h_H g_H B(z,\{d_H\}) z^{-\lfloor d_H \rfloor}}$$

where $a_H = 0.96$ and $b_H = 0.53$ are constants and h_H, g_H and d_H are adaptive parameters. B is a fractional delay filter with low-pass characteristics, chosen from a list of filters, depending on the fractional part of the pitch lag d_H.

This filter is built upon H_E from EVS [37, 114]. The parameters B, a_H, b_H and d_H have the same function as B, a_E, b_E and d_E in H_E and are just differently tuned. The difference is in replacing g_E with h_H and g_H, how these are calculated and in their function.

The fractional delay filters $B(z,\{d_H\})$ with low-pass characteristics are not necessarily the same as the one used in LTP. Its characteristics can be specifically tuned for HPF. Symmetric 10 tap filters with -6 dB cutoff at 5 kHz is used for $B(z,\{d_H\})$ in HPF at 48 kHz:

$$B(z,0/2) = 0.0000z^{-5} + 0.0164z^{-4} + 0.0646z^{-3} + 0.1297z^{-2} + 0.1856z^{-1} + 0.2075z^{0} + 0.1856z^{1} + 0.1297z^{2} + 0.0646z^{3} + 0.0164z^{4}$$

$$B(z,1/2) = 0.0040z^{-5} + 0.0370z^{-4} + 0.0965z^{-3} + 0.1606z^{-2} + 0.2019z^{-1} + 0.2019z^{0} + 0.1606z^{1} + 0.0965z^{2} + 0.0370z^{3} + 0.0040z^{4}$$

The coefficients of $B(z,0/2)$ and $B(z,1/2)$ are samplings at $t = i$ and $t = i/2$ ($i \in \mathbb{Z}$) of $\frac{\sin\left(\frac{7\pi t}{32}\right)}{\pi t}$ windowed with the Hann window of length 20, with the sum of the coefficients normalized to one. $B(z,0/2)$ and $B(z,1/2)$ have the same magnitude response and differ only in the group delay for $1/2$.

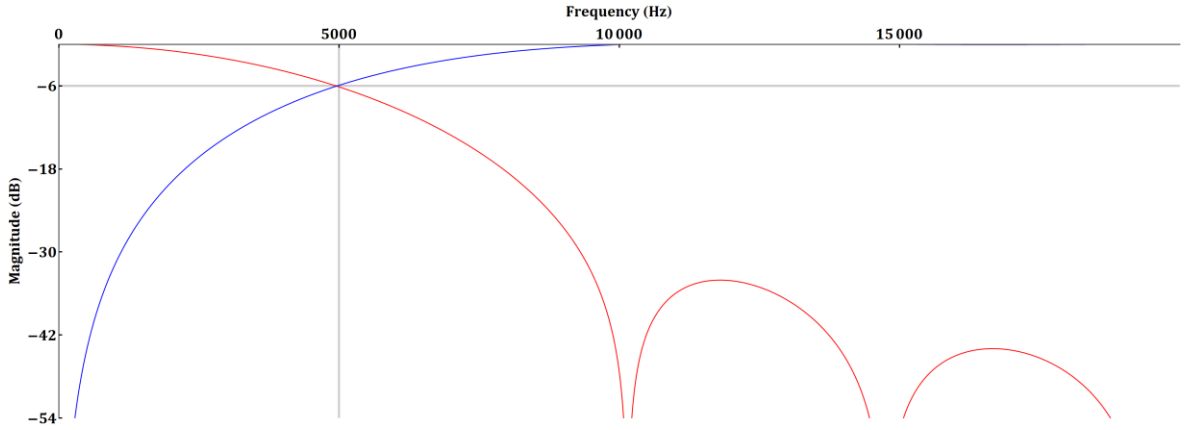

Figure 8.3 Magnitude response of $B(z)$ and $1 - B(z)$

The feedforward part of the filter H_H acts as a high pass filter opposite of B, for $a_H = b_H = h_H = 1$. Both filters' (B and $1 - B$) magnitude responses are shown in Figure 8.3.

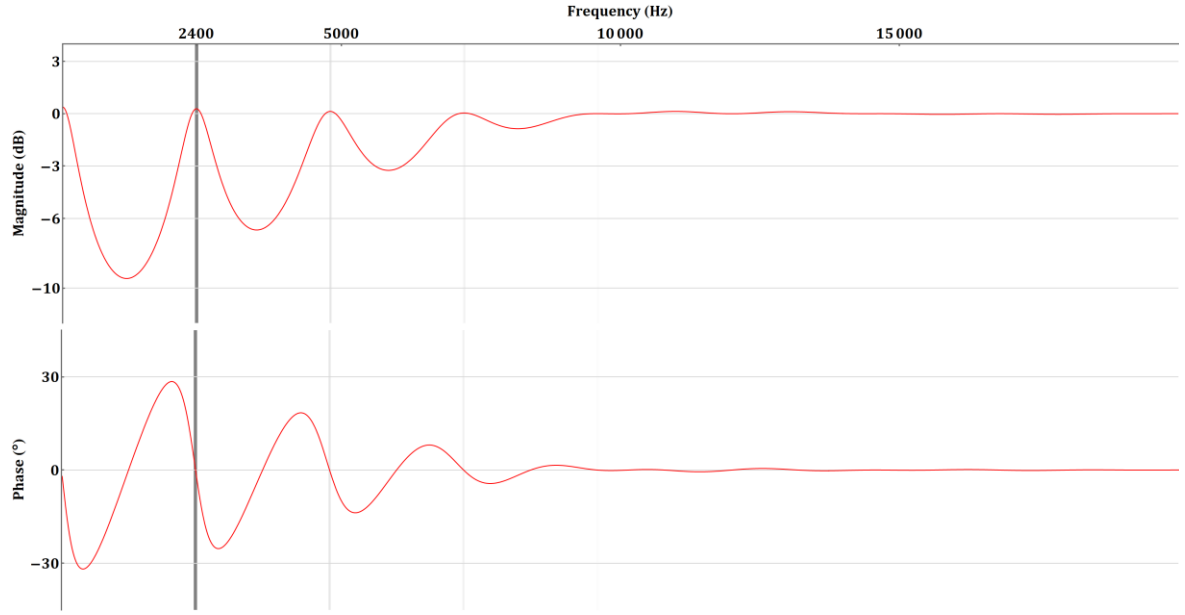

Figure 8.4 Frequency response of $\frac{1-a_H b_H B(z,0)}{1-b_H B(z,0) z^{-20}}$

The frequency response of H_H is plotted in Figure 8.4 for $F_S = 48$ kHz with $d_H = 20$, corresponding to the fundamental frequency of 2.4 kHz. Value 20 of d_H is outside of the used range and is just for presenting the characteristics of H_H; h_H and g_H were set to 1. The filter tap $z^{-\lfloor d_H \rfloor}$ is responsible for harmonicity increase. The role of B in the feedbackward part of H_H is twofold: to account for the fractional delay and to have more reduction between the harmonics at low than at high frequencies.

In most natural signals, speech included, harmonic content is predominant at LF. Natural signals usually consist of noise or inharmonicities at HF. Also low-latency and low-bitrate FD codecs are not able to correctly reproduce HF. Therefore, HPF would attenuate signal at HF without introducing significant increase of harmonicity. Significant advantage of the post-filter from EVS [37, 114] over other methods is that the harmonicity increase is stronger at LF, leaving HF mostly unaffected. The newly proposed filter inherits this characteristic.

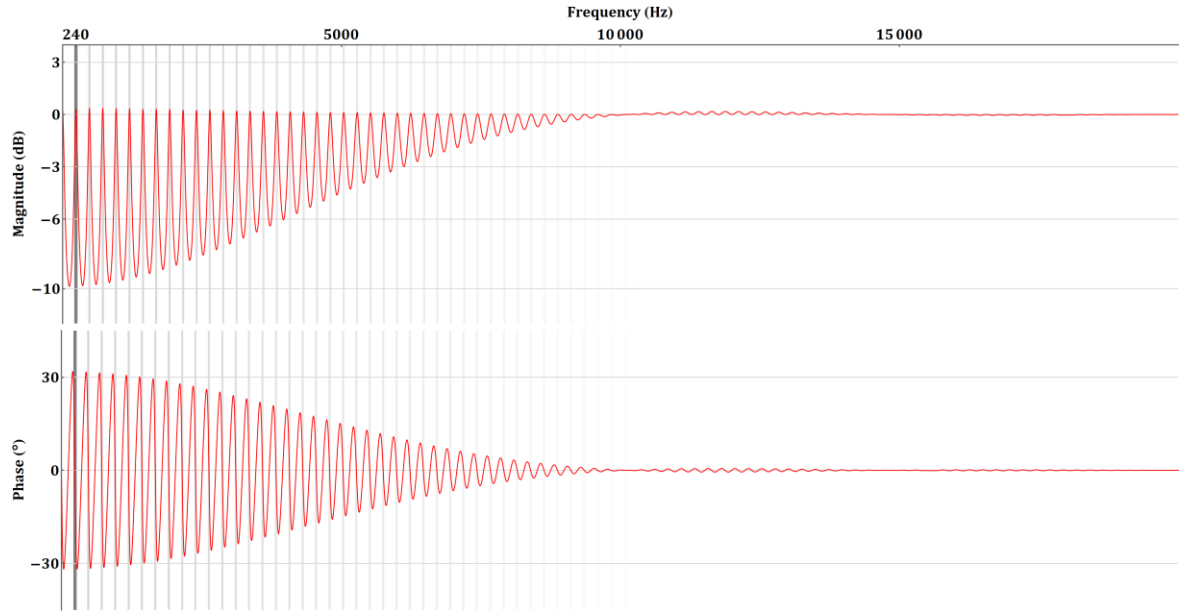

Figure 8.5 Frequency response of $\frac{1-a_H b_H B(z,0)}{1-b_H B(z,0) z^{-200}}$

Increasing d_H to 200, the distance between the notches in the frequency response is reduced as expected for a comb filter and as can be seen in Figure 8.5.

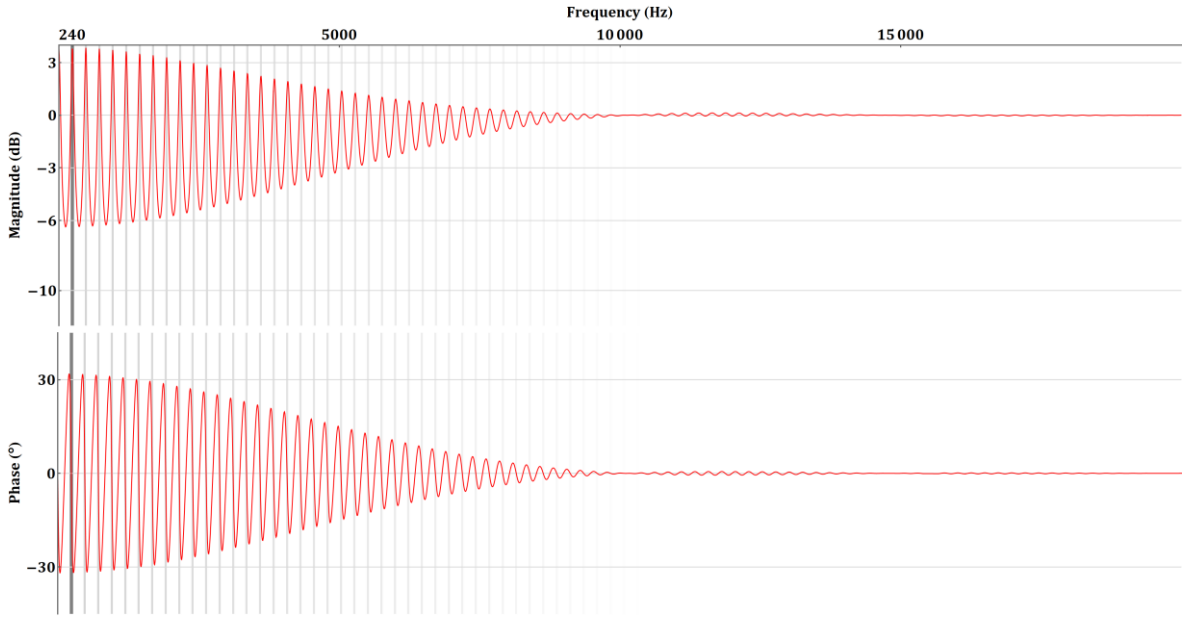

Figure 8.6 Frequency response of $\frac{1-0.5 b_H B(z,0)}{1-b_H B(z,0) z^{-200}}$

Setting a_H to 0.5 in Figure 8.6, it becomes obvious that the role of $a_H b_H g_E B(z,0)$, with a_H close to 1, is to avoid resonance of the harmonics. This is important, as it is expected that the MDCT codec mostly preserves the total energy of the signal. With $a_H = 0.96$, beside avoiding the resonance, the energy between the harmonics is reduced. This energy reduction is less problematic than the resonance as the signal between the harmonics is anyhow masked by the harmonics. Setting $a_H = 1$ would not be optimum, as the harmonics are still spread because of the MDCT's short window; with $a_H = 1$ the harmonic part would be actually a bit

suppressed and the overall energy would be significantly reduced. The role of $B(z,0)$ here also becomes clear, as just using $1-a_H b_H g_E$ for the feedforward part would introduce low-pass filtered output.

The constant parameters a_H and b_H are tuned so that the signal is suppressed between the harmonics defined by d_H and that there is only minimal increase of the harmonics' energy. This leads to an overall increase of the harmonicity of the post-processed signal.

The above analysis of H_H can also be applied on H_E.

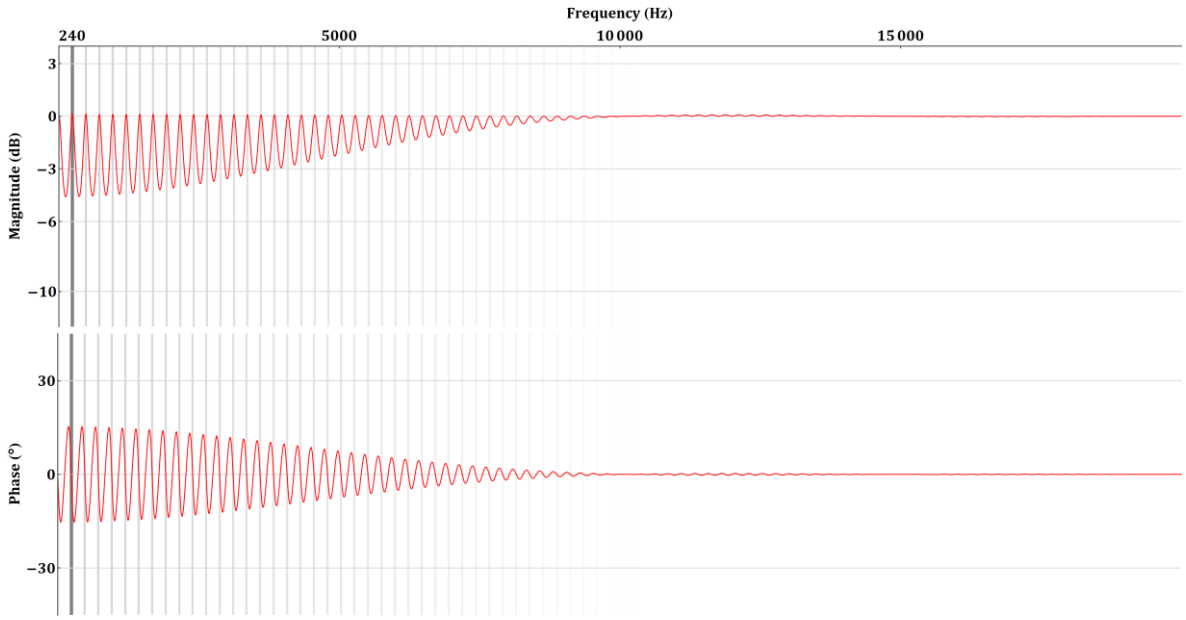

Figure 8.7 Frequency response of $\frac{1-a_H b_H 0.5B(z,0)}{1-b_H 0.5B(z,0)z^{-200}}$

Decreasing h_H to 0.5, the notches of the comb filter are reduced or in other words the filtering effect is lessened [Figure 8.7]. The parameter g_E in H_E and the constants b_H and b_E have the same effect as h_H on regulating the introduced harmonicity. However, the values of the adaptive parameters h_H in H_H and g_E in H_E are calculated in a different way.

In EVS, g_E in H_E is calculated as:

$$g_E = \frac{\sum_{n=0}^{H_M} x[n]y[n]}{\sum_{n=0}^{H_M} y^2[n]}$$

where y is a predicted signal obtained from the ZIR of $H_E(z)$ setting $a_H = 0, b_H = g_E = 1$. Principally, g_E in EVS minimizes root mean square (RMS) error of the prediction. The calculation of g_E occurs on the encoder side and g_E is quantized and limited between 0 and 0.625. The quantized value of g_E is used in the post-filter of the EVS decoder. Also d_E is found for the whole frame of the original signal in EVS and transmitted to the decoder.

All adaptive parameters (h_H, g_H and d_H) of H_H in IVA are calculated on each sub-interval of the decoded signal and not transmitted via a bit-stream nor quantized. When calculating the parameters of H_H, the output of H_H in the previous sub-intervals is made available and used in the calculations.

The pitch lag d_H in each sub-interval is found via the pitch search $\mathcal{F}_{F_0}$ [4.3.1] on the signal consisting of the current HPF input $x_D\left[(i-1)\cdot H_{F_H}+n\right]$ for $0 \le n < L_{F_H}$, and the HPF output samples $x_H\left[(i-1)\cdot H_{F_H}+n\right]$ for $-\hat{d}_{F_0} \le n < 0$, where $0 \le i < N_{F_H}$ is the index of the current sub-interval [8.2]. For finding d_H, $\mathcal{F}_{F_0}$ is called with $L_H = L_{F_H}$, $d_{\tilde{F}_0} = d_V\left[i \cdot H_{F_H}\right]$, $d_{\check{F}_0} = d_{\tilde{F}_0} - 2$ and $d_{\hat{F}_0} = d_{\tilde{F}_0} + 2$. The function $\mathcal{F}_{F_0}$ also returns the normalized correlation ρ_H associated with d_H. The transmitted pitch contour d_V is used to reduce the complexity in the search of d_H.

The parameter h_H is a desired increase in harmonicity, with 1 for the maximum and 0 for no increase:

$$h_H = a_L a_T \rho_H^2$$

h_H is proportional to the square of the normalized correlation ρ_H. Here comes into play, the advantage of only post-filtering and no pre-filtering. Even with the present quantization noise, the signal x_D still contains the harmonic part of the original signal. Short experiments have shown that the normalized correlation in x_D and in x_M [Figure 4.1] are very similar. It was found heuristically that squaring ρ_H already gives good values for h_H and that a modification depending on LTP and depending on a tilt of the spectrum $\bar{\bar{X}}_C$ are beneficial. The modification coming from LTP is:

$$a_L = \begin{cases} 0.5, N_L = 0 \\ 0.7 + \dfrac{0.3 N_L}{\hat{N}_L}, N_L > 0 \end{cases}$$

where N_L is the number of the predicted harmonics in LTP [6.6].

The modification coming from the tilt is:

$$a_T = \frac{\sum_{n=0}^{6}\left(\bar{\bar{X}}_C[n]\right)^2}{\sum_{n=7}^{49}\left(\bar{\bar{X}}_C[n]\right)^2}$$

The constant b_H can also be incorporated into h_H.

Different to g_E, h_H value is not affected by amplitude modulations. g_E could reach its maximum value of 0.625 if $\sum_{n=0}^{H_M} x[n] \gg \sum_{n=0}^{H_M} x[n-d_E]$ even when there is only small correlation of $x[n]$ and $x[n-d_E]$. This could lead to a distortion of the onset of a voiced plosive. Such problem is avoided in EVS using an HPF activation decision based on normalized correlation and transient detection [37, 166]. Another problem arises if $x[n] \sim cx[n-d_E]$ (leading to $g_E \sim c$) for c significantly different from 1. For small values of c, that is at fading out of a harmonic signal, HPF would be deactivated. For large values of c, the signal would be attenuated at LF. The attenuation is happening because by filtering with H_E, the original signal $x[n] \sim cx[n-d_E]$, the inverse of the low-pass filtered $g_E x[n] \sim g_E cx[n-d_E]$ and the low-pass filtered $g_E x[n-d_E]$ are essentially summed up, thus decreasing LF of the original signal with the low-pass filtered $g_E(c-1)x[n-d_E]$.

All of the problems coming from the calculation of g_E are automatically solved by using h_H, without need for any additional decision on disabling HPF.

The value of g_H is calculated in the same way as g_E in EVS, but in each sub-interval and letting it be as high as 5 and setting it to zero if it is higher than 5 or below 0:

$$g_H = \frac{\sum_{n=0}^{L_{F_H}} x[n]y[n]}{\sum_{n=0}^{L_{F_H}} y^2[n]}$$

where $x[n] = x_D\left[(i-1)\cdot H_{F_H} + n\right]$ and y is a predicted signal for the sub-interval i, obtained from the ZIR of $H_H(z)$ setting $a_H = 0, b_H = g_H = h_H = 1$. Basically $y[n]$ is $x_H\left[(i-1)\cdot H_{F_H} - \lfloor d_H \rfloor + n\right]$ low-pass filtered with $B(z,\{d_H\})$. Since $d_H \sim L_{F_H}$, $x[n]$ and $y[n]$ are two consecutive pitch cycles for $0 \leq n < L_{F_H}$ and the role of g_H in H_H is to model amplitude changes of the similar parts of the consecutive pitch cycles in the signal. In other words g_H models amplitude modulations of the harmonic part of the signal.

Even though g_H uses the same formula as g_E, g_H appears only in the feedbackward taps and this is a significant difference.

Figure 8.8 and Figure 8.9 show the effect of increasing g_H to 1.3 and decreasing it to 0.7 respectively. It needs to be taken into account that g_H is changing in every sub-interval and the frequency response analysis is meaningful only for linear time-invariant (LTI) filters. The role of g_H will be explained together with the procedure for its calculation.

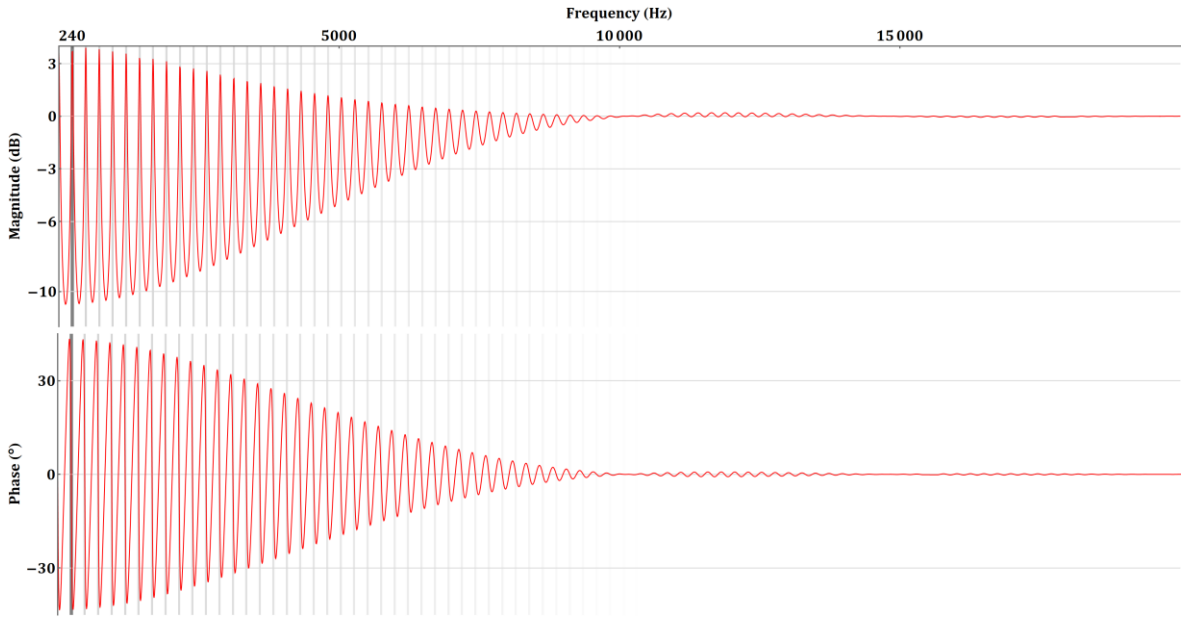

Figure 8.8 Frequency response of $\frac{1-a_H b_H B(z,0)}{1-b_H 1.3 B(z,0) z^{-200}}$

Going back to the plot of the H_H frequency response for $g_H = 1.3$ [Figure 8.8], such response of an LTI filter would be used only for an input signal for which H_H parameters are constant and there is a continuous increase in the amplitude of the input signal. Such signals, with continuously increasing amplitude, are unbounded signals. Real-world signals are bounded and g_H only temporarily goes above 1 and at some point afterwards also goes below 1. Again looking at signals for which H_H parameters are constant, $g_H = 0.7$ would mean that the signal amplitude is continuously decreasing [Figure 8.9]. These kind of signals eventually reach level

below threshold of hearing, the speed of attenuation being proportional to g_H. Such signals are of very limited interest. Real-world signals also exhibit amplitude increase and have g_H only for short time significantly below 1, usually followed by either high g_H or a low h_H. There is very limited usage of the frequency response plots to understand the role of g_H.

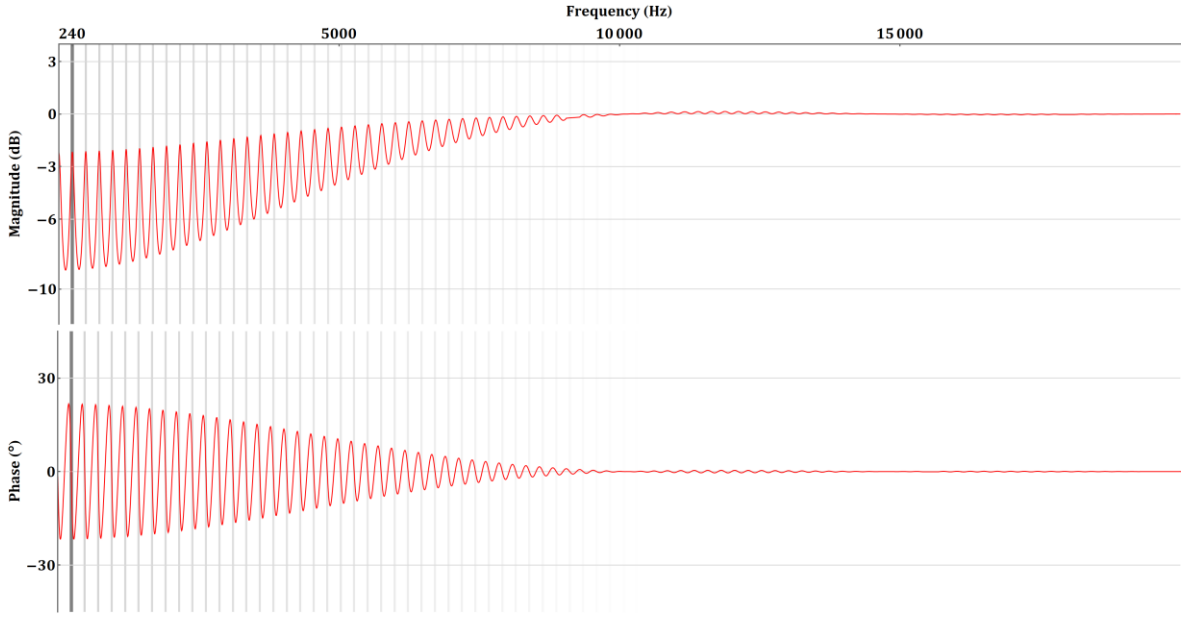

Figure 8.9 Frequency response of $\frac{1-a_H b_H B(z,0)}{1-b_H 0.7 B(z,0) z^{-200}}$

Better understanding of g_H is possible by imaging an amplitude modulated train of unit impulses at the regular interval d_H as the input signal. Such modulated train of unit pulses should not be modified by H_H if $a_H = 1$. Each pitch cycle and also each sub-interval contain only one scaled unit impulse. h_H for such signal is constant and equal to 1, if $a_L = a_T = 1$ (parameters used in calculating h_H). g_H is equal to the ratio of the consecutive impulse amplitudes and the feedbackward part of H_H produces low-pass filtered impulse scaled with b_H. The feedforward part reduces the impulse by a low-pass filtered version of itself scaled with b_H. Adding the feedforward and the feedbackward part then yields the original impulse. Consequently, the modulated train of unit pulses stays unchanged.

8.4 Advantages of the new contributions

The proposed filter H_H together with the splitting of the processed signal into overlapping sub-intervals of adaptive lengths allows within single method:

- continuous and smooth modeling of rapid pitch, amplitude and harmonicity changes
- can be implemented independent of a codec, without any information in the bit-stream

If there is information in the bit-stream needed for an LTP operation, then the proposed approach can use the information to reduce its computational complexity.

9 Objective and subjective performance evaluation

9.1 Experimental results

For evaluating the quality of the proposed methods, objective and subjective measurements were used. Three objective measurements were chosen: Perceptual Objective Listening Quality Analysis (POLQA) [167] and the basic and the advanced versions of Perceptual Evaluation of Audio Quality (PEAQ) [168]. For the subjective measurements, listening tests were organized, based on the ITU-R BS.1534-3 recommendation [169]. The samples for the listening tests are mono, 3 s to 20 s long [A.3]. The speech items include different languages, recorded in various environments. The music samples cover wide range of genres and are expected to produce various types of coding artifacts [170]. For WB tests, the original reference and all test signals are sampled at 16 kHz. For FB tests, all signals are sampled at 48 kHz. The listening tests were conducted on headphones, with the identical signals fed to each earpiece. In most of them, electrostatic earspeakers Stax [171] were employed and listeners were placed in a well isolated environment. Unfortunately, the final test took place during the COVID-19 pandemic and results obtained in a quiet home office with high quality headphones (e.g. Beyerdynamic DT 770 Pro) also had to be accepted. All participants are expert listeners [58], most of them with many years of experience in listening to audio and speech codec artifacts. In rare cases listeners graded a hidden reference with less than 90 points and are post-screened.

The ITU-R BS.1534-3 recommendation [169] dictates the use of the 3.5 kHz low-pass filtered low anchor. It was noticed that listeners can easily find the low anchor, marked as anchor_3k5 in figures, even when the references is band limited to 8 kHz. Thus, the low anchor is not used in the FB tests. The experienced listener have correctly used the whole grading scale, even without the presence of the low anchor. The anchor exclusion allows participants to concentrate on differences between the codecs under test.

In the plots of the listening tests' results, mean values with 95 % confidence intervals based on the Student's t-distribution are shown. The last column shows the averages and the confidence intervals over all items and all listeners for each codec under test. In the absolute score plots, grading categories from excellent to bad are displayed. The absolute grade 100

means that there is no distinction to the reference, being the original signal. In the differential score plots, one condition is chosen as the reference and the other conditions are displayed relative to this reference with 0 meaning no difference, positive score meaning that the condition is better than the reference and negative score that it is worse than the reference.

IVA was always tested at 24.4 kbps constant bitrate.

9.1.1 WB listening tests

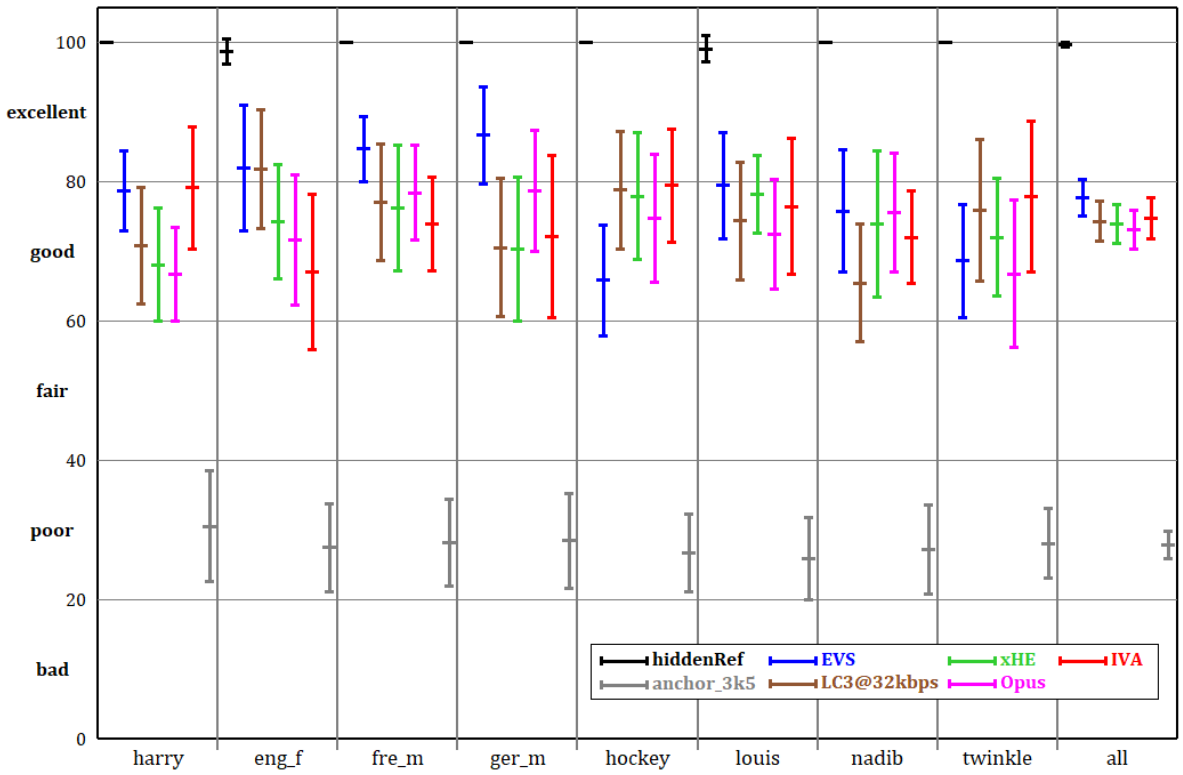

Figure 9.1 Listening test results at WB for speech items with IVA v73, 12 listeners

The results in Figure 9.1, Figure 9.2, Figure 9.3 and Figure 9.4 show absolute scores from the WB tests with older versions of IVA, conducted at 16 kHz. These IVA versions use the LP filtering for spectral shaping [3.1]. They employ a block switching, as in EVS, between 20 ms, 10 ms and 5 ms frames. The block switching is disabled in the final IVA version. The old IVA versions include an initial implementation of the pulse coding, operating in the LP filtered domain, and have tunings different from the final version.

In the first two listening tests (Figure 9.1 and Figure 9.2) the IVA version marked as v73 was used. In v73, LTP with constant pitch over the frame is used (resembling LTP in [108]). The HPF implementation in v73 is very similar as in EVS. IVA v73 was compared with:

- EVS: 3GPP EVS v13.1.0 at 24.4 kbps constant bitrate [120]
- xHE: Fraunhofer xHE-AAC v2.00 at 25 kbps average bitrate [33, 172]
- Opus: libopus 1.2 and opus-tools 0.1.10 at 24.4 kbps constant bitrate [38, 173]
- LC3@32kbps: LC3 v1.0.0 at 32 kbps constant bitrate [80, 174]

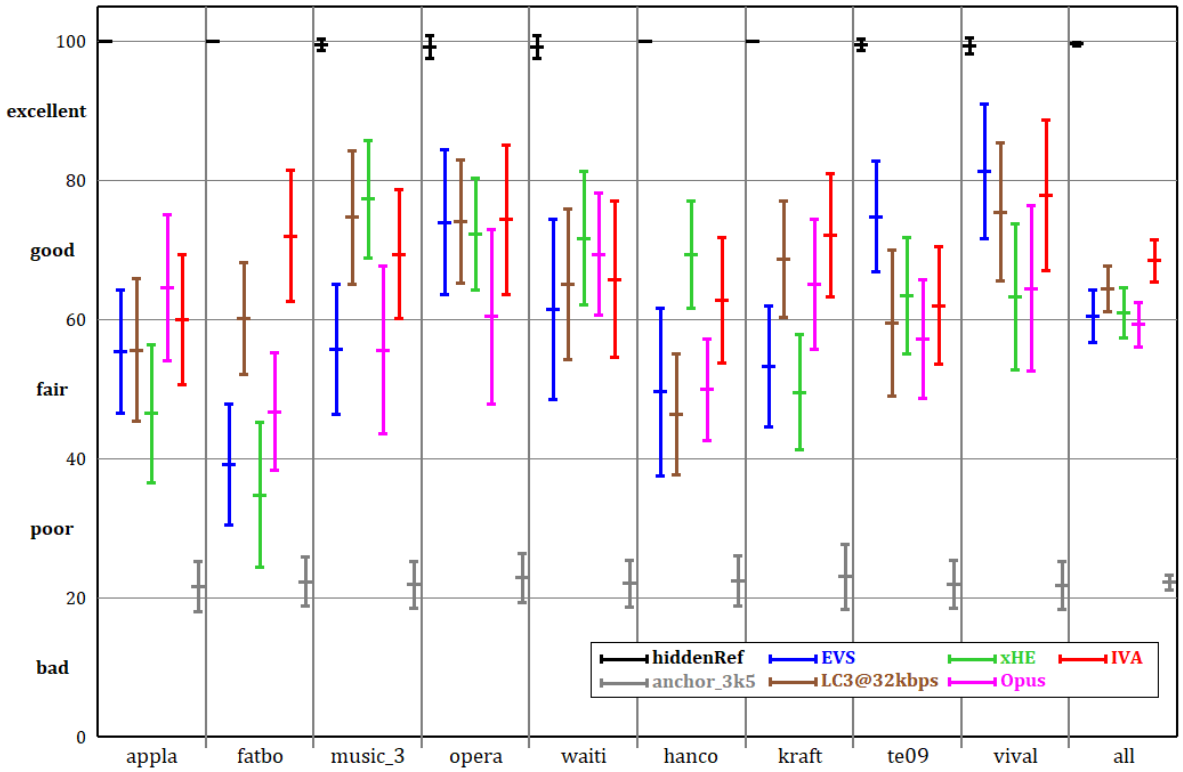

Figure 9.2 Listening test results at WB for music items with IVA v73, 13 listeners

It should be pointed out that the codecs in the test have different algorithmic delays. EVS has 32 ms, LC3 12.5 ms, Opus 26.5 ms and IVA v73 34.5 ms delay. xHE-AAC has much longer delay of about 200 ms and its bitrate may vary between the frames within the super frame, but it has constant super frame size and produces 25 kbps bit-streams [34]. Because LC3 has much lower delay and computational complexity than other codecs in the test, it was run at 32 kbps.

The first test includes clean speech, speech with background sounds and speech mixed with music. The second test includes various music genres [A.3].

After conducting the first listening tests, IVA was further developed. New HPF [8.2,8.3] was implemented, low-pass with an adaptive cutoff was introduced in LTP, some tunings were introduced in the pulse coding (e.g. disabling of pulse extraction and coding for high pitched signals) and some smaller bugs were fixed. The new version was marked as v93 and compared in listening tests to v73.

The results of the listening tests for v93 are shown in Figure 9.3 and Figure 9.4. EVS is included in the speech listening test. EVS and xHE-AAC are included in the music listening test. The inclusion of EVS and xHE-AAC allows comparison to the results of the first listening tests with v73.

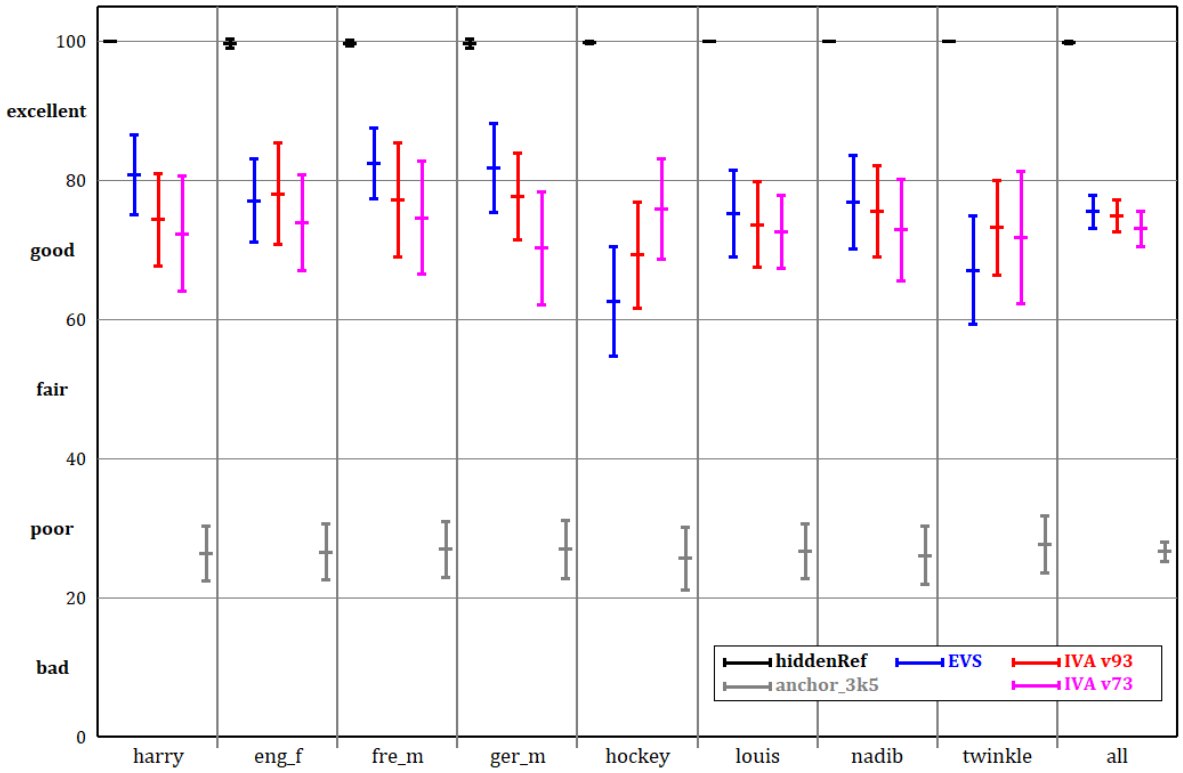

Figure 9.3 Listening test results at WB for speech items with v93, 17 listeners

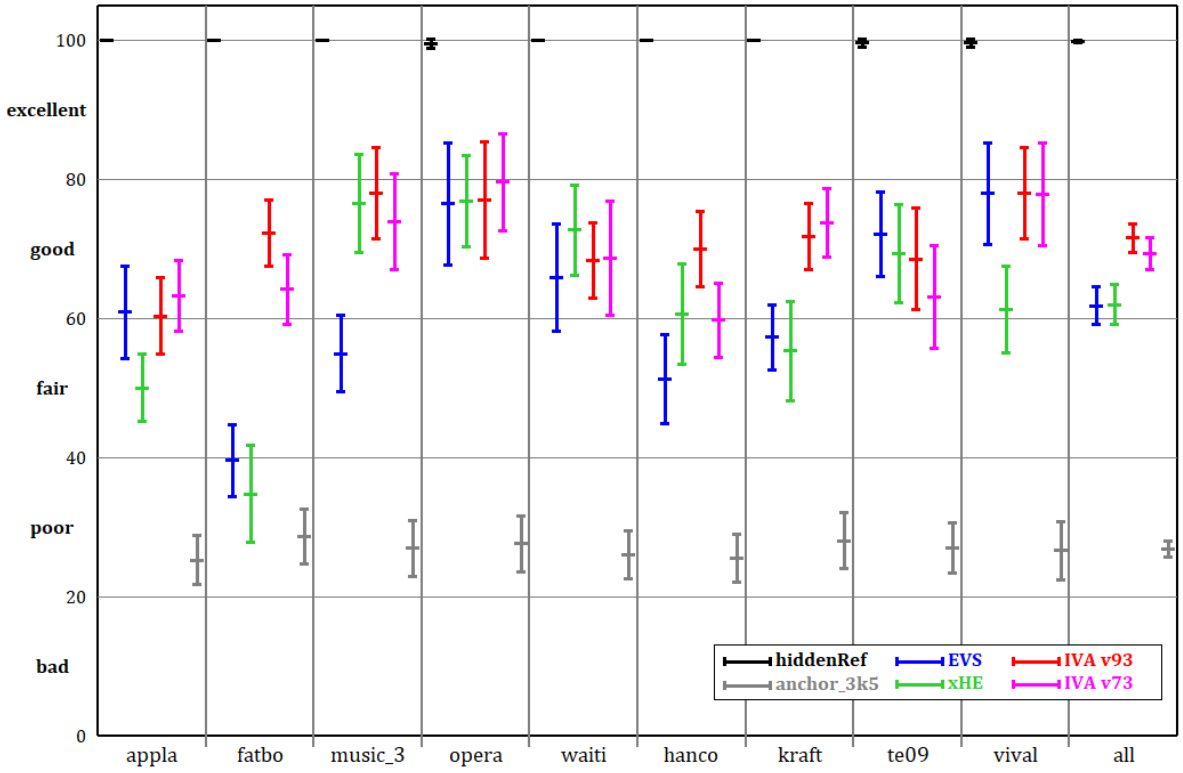

Figure 9.4 Listening test results at WB for music items with v93, 18 listeners

9.1.2 FB listening tests

After the experiments in WB, IVA was reorganized and extended to FB. In the initial FB implementations, the pulse coding is disabled and only simple noise filling, similar to the noise filling in EVS, is used (there was no IGF nor iBPC). The FB IVA version uses SNS, as described in [4.5]. The low anchor is not used in the FB listening tests.

New pitch contour LTP [6.2-6.6] was implemented and evaluated in the following experiment. HPF is disabled in this experiment. The new LTP was compared against frequency-domain prediction (FDP) [109] and a simple TD LTP (similar to [107, 110, 140, 175]). The simple TD LTP finds a portion in the past decoded signal, most similar to the signal in the current MDCT window. The found portion is filtered with fractional delay filters B, the same filters as used in the new LTP, and hence fractional delays are accounted for. The portion is then transformed via the MDCT to produce the predicted MDCT. The rest of the method in the simple TD LTP is the same as in the new LTP.

To choose items for the listening tests, objective measurements and informal listening were used. Averages from the objective measurements on 92 various items are presented in Table 9.1. The samples with the highest expected impact of LTP were chosen, also trying to cover various types of signals. The averages from the objective measurements on the chosen 17 items are presented in Table 9.2.

	no LTP	Simple TD LTP	FDP	Pitch contour LTP
PEAQ Basic	-3.10	-3.10	-3.10	-3.07
PEAQ Advanced	-3.24	-3.23	-3.24	-3.19
POLQA	4.05	4.04	4.06	4.15

Table 9.1 Objective measurements for LTP on 92 items

	no LTP	Simple TD LTP	FDP	Pitch contour LTP
PEAQ Basic	-3.13	-3.11	-3.13	-3.05
PEAQ Advanced	-3.34	-3.29	-3.31	-3.17
POLQA	3.67	3.66	3.70	3.93

Table 9.2 Objective measurements for the 17 LTP test items

Figure 9.5 and Figure 9.7 show absolute scores for the LTP listening tests for speech and music items, respectively. Figure 9.6 and Figure 9.8 show differential scores relative to the new LTP for the same listening tests. In the differential plots, the new LTP is positioned at 0 and the differential scores may be positive and negative. The only difference between the tested variants is in LTP; all other codec parts are the same.

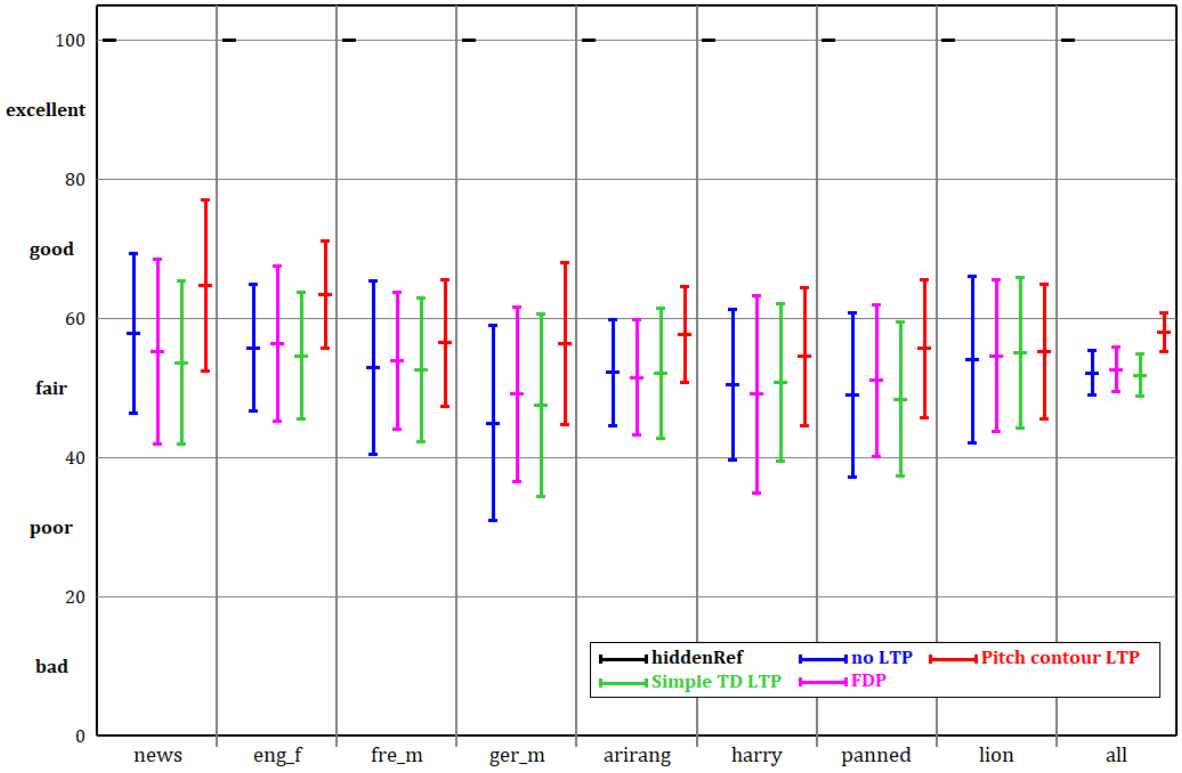

Figure 9.5 The LTP listening test results at FB for speech items, 7 listeners

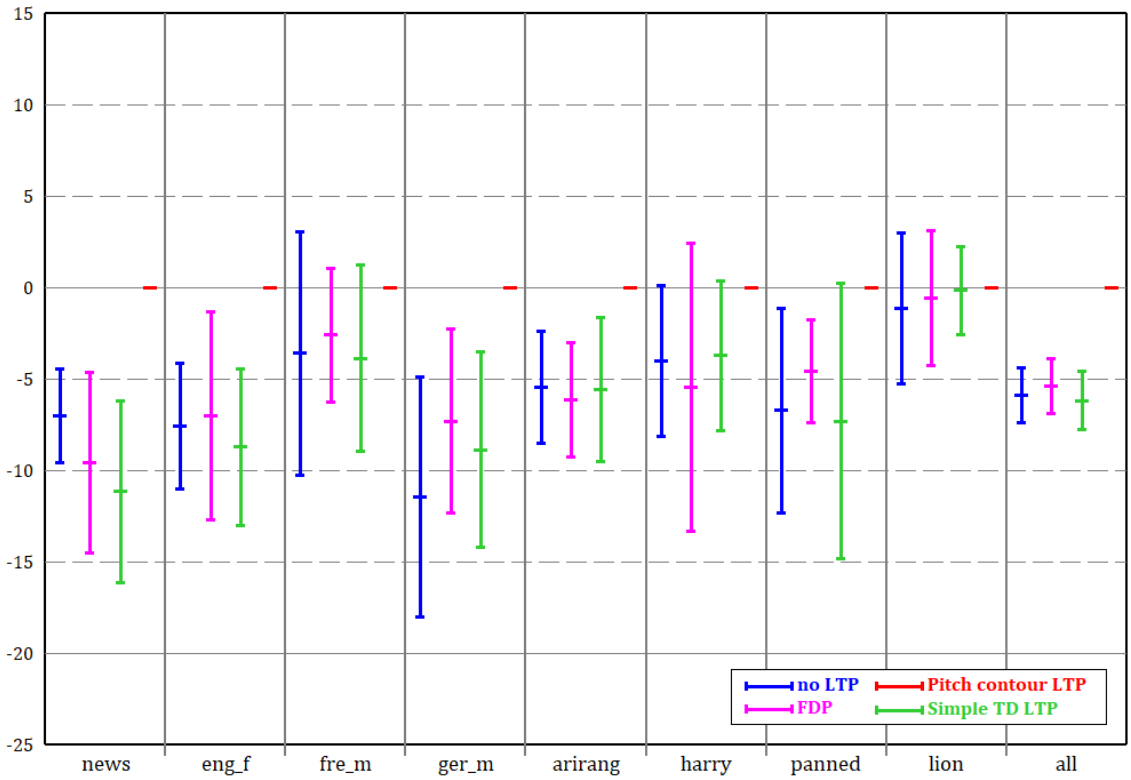

Figure 9.6 Differential scores for the LTP listening test results for speech items

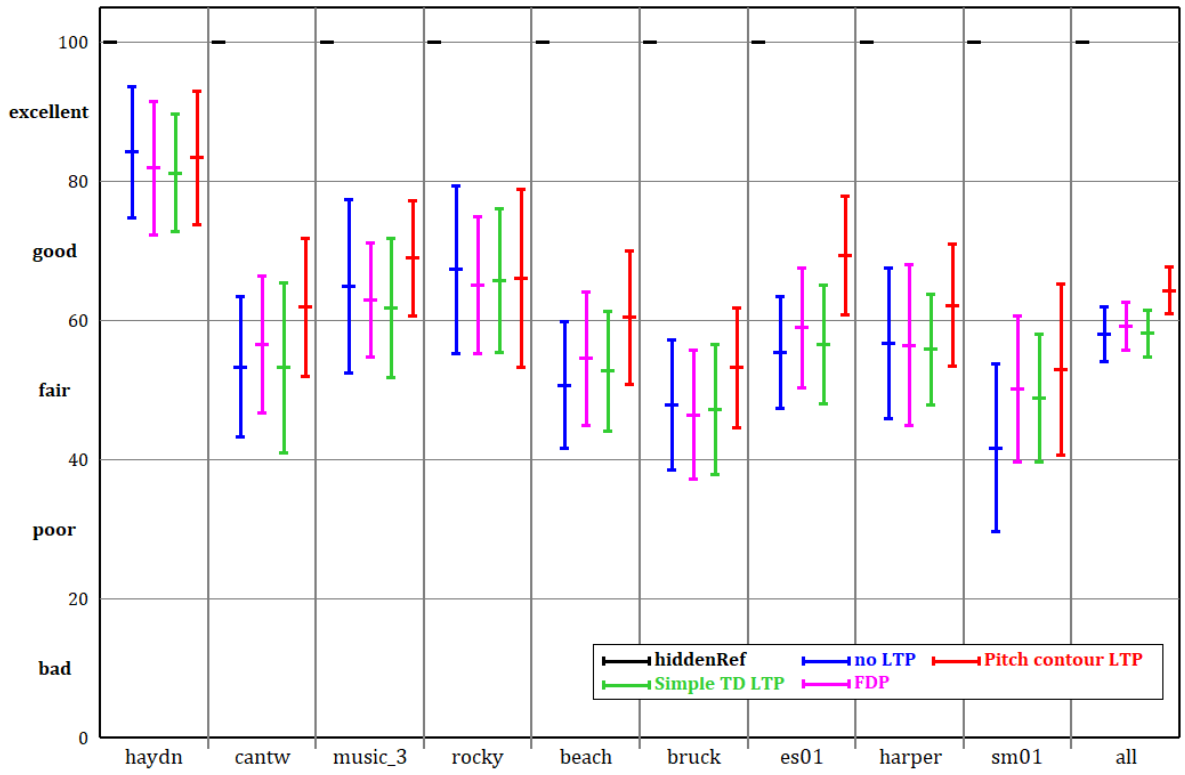

Figure 9.7 The LTP listening test results at FB for music items, 8 listeners

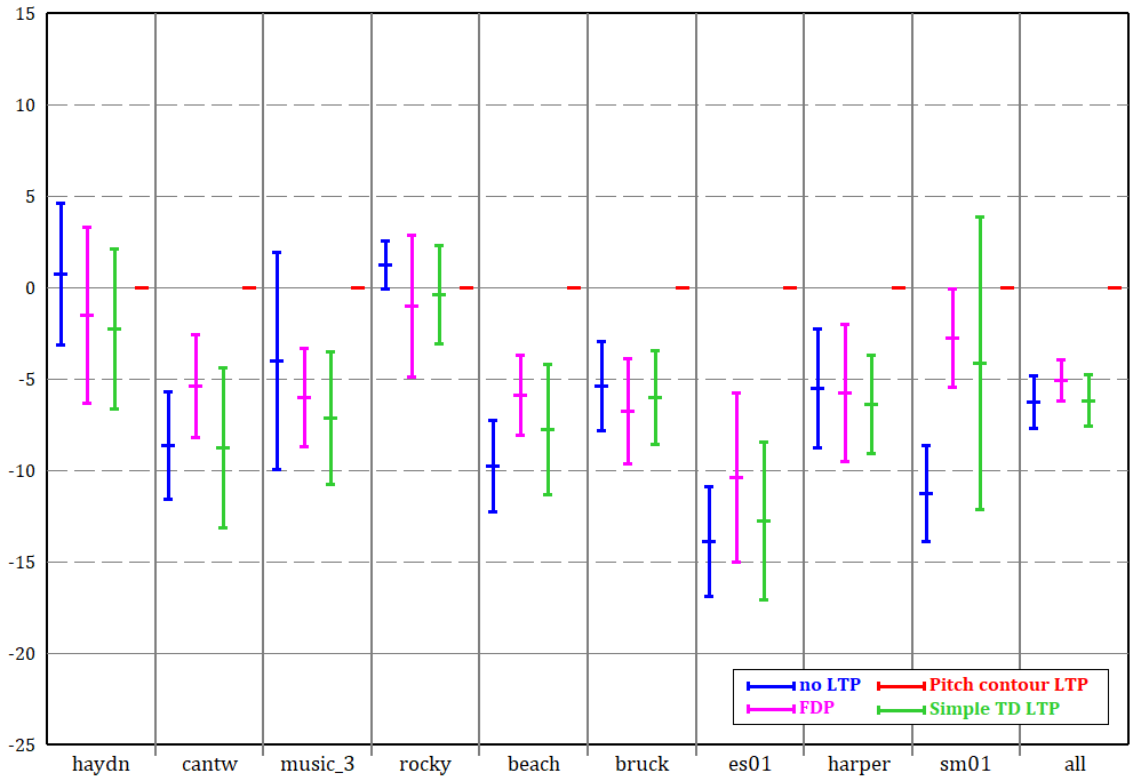

Figure 9.8 Differential scores for the LTP listening test results for music items

In the following experiments, the effect of LTP and HPF was compared. To distinguish them from the previous tests, they are called the HPF listening tests. As in the LTP tests, the pulse

coding is disabled and only simple noise filling is used. The items from the LTP tests were reused, adding one more item to the speech test and two items to the music test [A.3].

The arithmetic coder, used to code the spectral coefficients in the MDCT, has two configurations: one for low bitrates and one for high bitrates. The configurations are the same as in EVS. Up to the HPF listening tests, the high bitrate configuration was used for the arithmetic coder at 24.4 kbps. Between the LTP and HPF tests, it was found that using the low bitrate configuration improves quality and was onward used per default in IVA. Because of this change, the results for the variants (e.g. with HPF disabled) that appear in both experiments, i.e. in the HPF and the LTP tests, cannot be compared. Yet, in an informal listening, the perceptual impact of the change in the arithmetic coder configuration is estimated to be significantly smaller than the effect of LTP or HPF.

Five variants were tested in the HPF listening tests:

- no LTP, no HPF
- with the pitch contour LTP [6.2-6.6], no HPF
- no LTP, with HPF using constant pitch over frame (EVS HPF)
- with the pitch contour LTP, with the constant pitch HPF (EVS HPF)
- with the pitch contour LTP, with the new HPF following pitch contour (IVA HPF)

The EVS HPF is basically the same as HPF in EVS [37, 120]. The difference is that the same smoothing method as in IVA or LC3 [144], the same constants (a_H and b_H) and the same fractional delay filters (B) as in IVA are used. The only difference between the tested variants is in LTP and HPF; all other codec parts are the same.

The averages from the objective measurements on 95 various items are presented in Table 9.3. New samples were collected during the development of IVA and this is the reason for increasing the number of items in the objective measurements between the LTP and the HPF test. The averages from the objective measurements on the chosen 20 items are presented in Table 9.4.

	no LTP no HPF	with LTP no HPF	no LTP EVS HPF	with LTP EVS HPF	with LTP IVA HPF
PEAQ Basic	-2.99	-3.02	-3.23	-3.23	-3.26
PEAQ Advanced	-3.10	-3.11	-3.18	-3.18	-3.20
POLQA	4.16	4.21	4.25	4.26	4.26

Table 9.3 Objective measurements for HPF on 95 items

	no LTP no HPF	with LTP no HPF	no LTP EVS HPF	with LTP EVS HPF	with LTP IVA HPF
PEAQ Basic	-3.03	-3.04	-3.25	-3.23	-3.28
PEAQ Advanced	-3.22	-3.17	-3.23	-3.19	-3.22
POLQA	3.89	4.05	4.15	4.19	4.19

Table 9.4 Objective measurements for the 20 HPF test items

Figure 9.9 and Figure 9.11 show absolute scores for the HPF listening tests for speech and music items, respectively. Figure 9.10 and Figure 9.12 show differential scores for the same listening tests. The differential scores are relative to the variant with the new LTP and the new HPF.

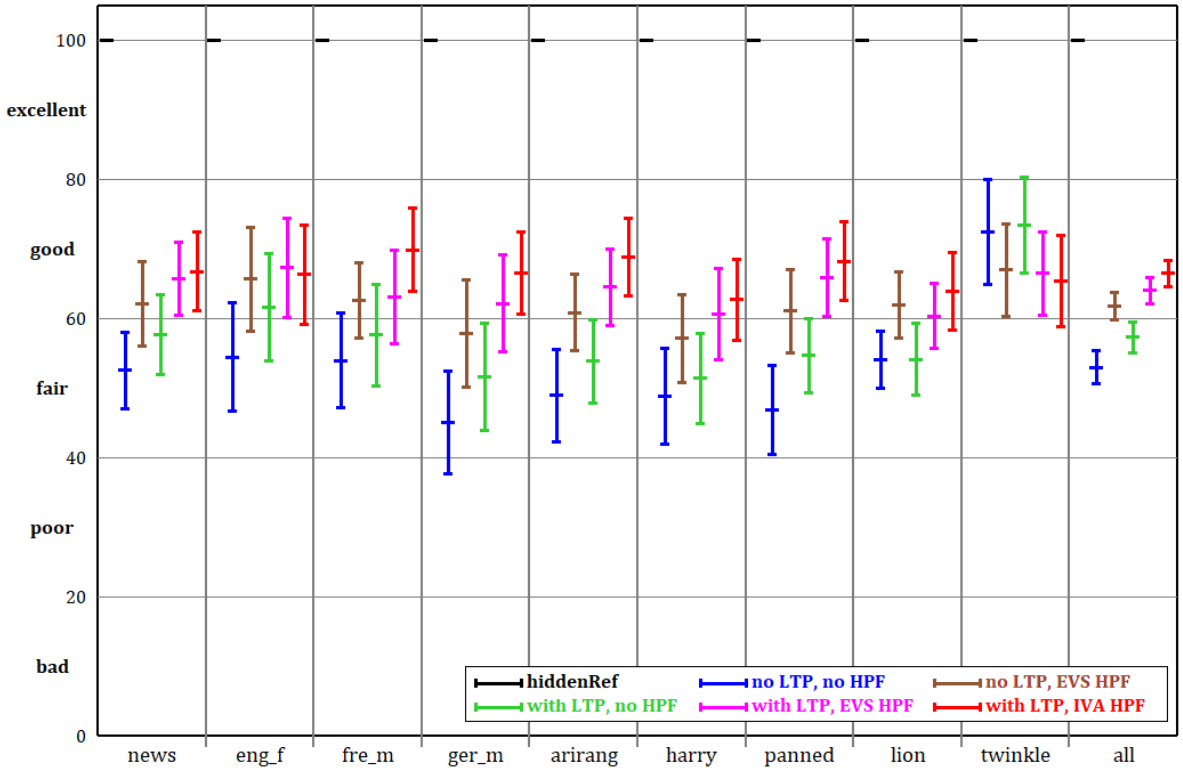

Figure 9.9 The HPF listening test results at FB for speech items, 22 listeners

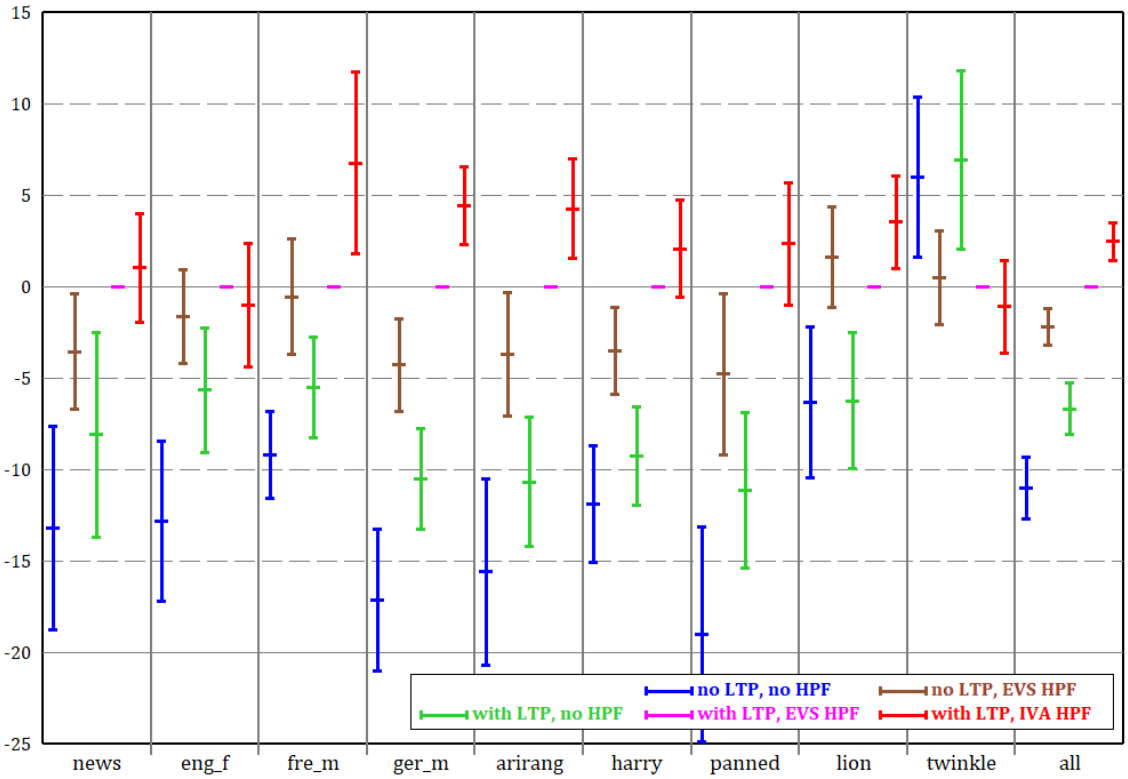

Figure 9.10 Differential scores for the HPF listening test results for speech items

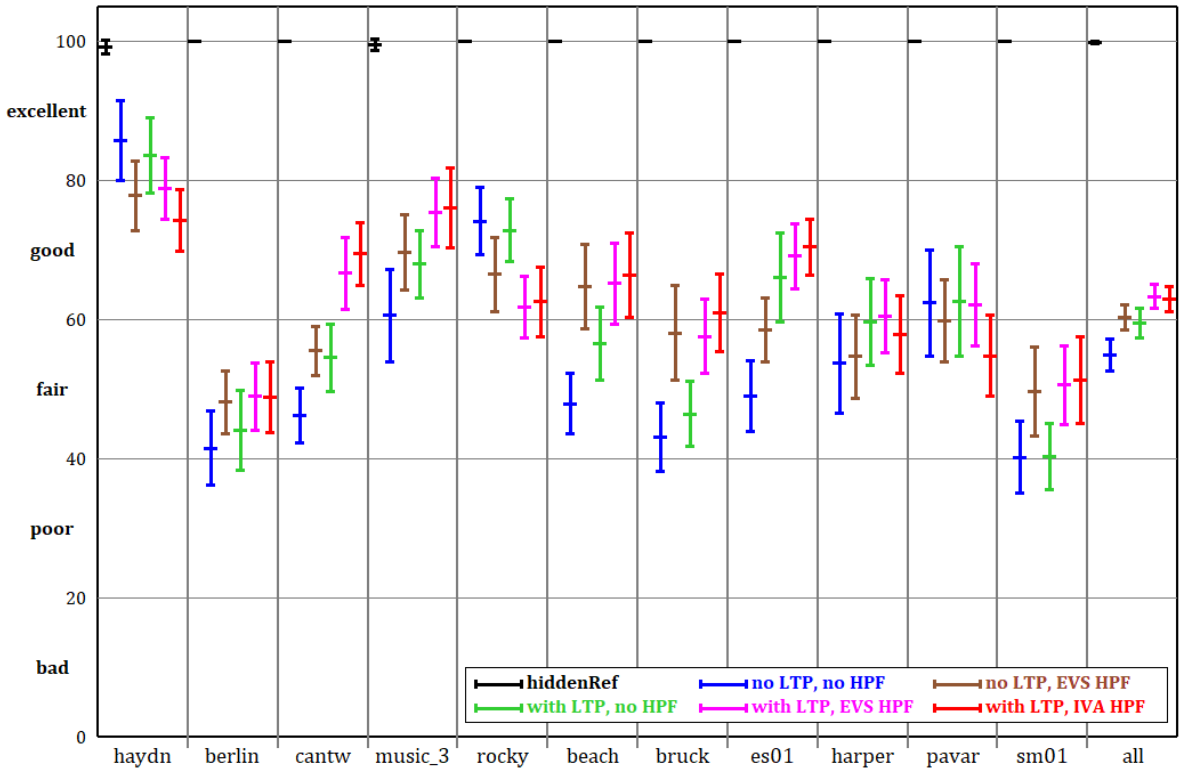

Figure 9.11 The HPF listening test results at FB for music items, 26 listeners

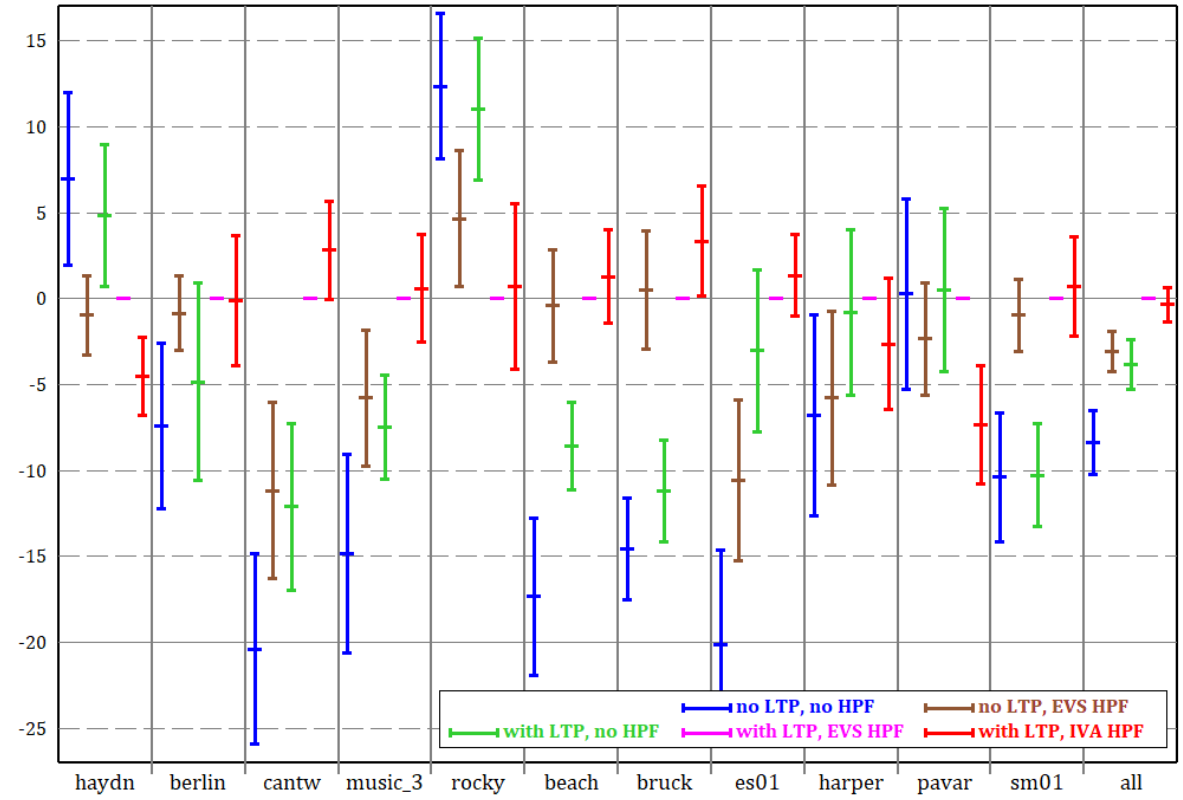

Figure 9.12 Differential scores for the HPF listening test results for music items

In the final listening tests, IVA as described in chapters 4-8 was used. IVA was compared with:

- EVS: 3GPP EVS v13.1.0 at 24.4 kbps constant bitrate [120]
- AAC: Fraunhofer HE-AAC from Winamp 5.6.6.3516 (enc_fhgaac.dll v1.0.8.0) at 24 kbps constant bitrate [176]
- Exhale: open source implementation of the xHE-AAC v1.0.8, preset 0 [177]
- Opus: libopus 1.3 and opus-tools 0.2 at 24.4 kbps constant bitrate [38, 173]

The tested codecs have different algorithmic delays. EVS has 32 ms, AAC 129 ms [178], Exhale approximately 82 ms, Opus 26.5 ms and IVA 35.875 ms delay.

Exhale v1.0.8 uses only AAC-based coding from xHE-AAC and has no SBR [34]. It uses variable bitrate (VBR), the bitrate ranging from 18 kbps to 44 kbps with the mean of 27 kbps on a set of 123 samples. The mean is 28 kbps on the items used in the listening tests, ranging from 23 kbps to 33 kbps, with the exception of 37 kbps for fatbo sample; mostly clean speech samples have above average bitrate.

Half of the samples, in each of the final listening tests, were not available during the development of IVA.

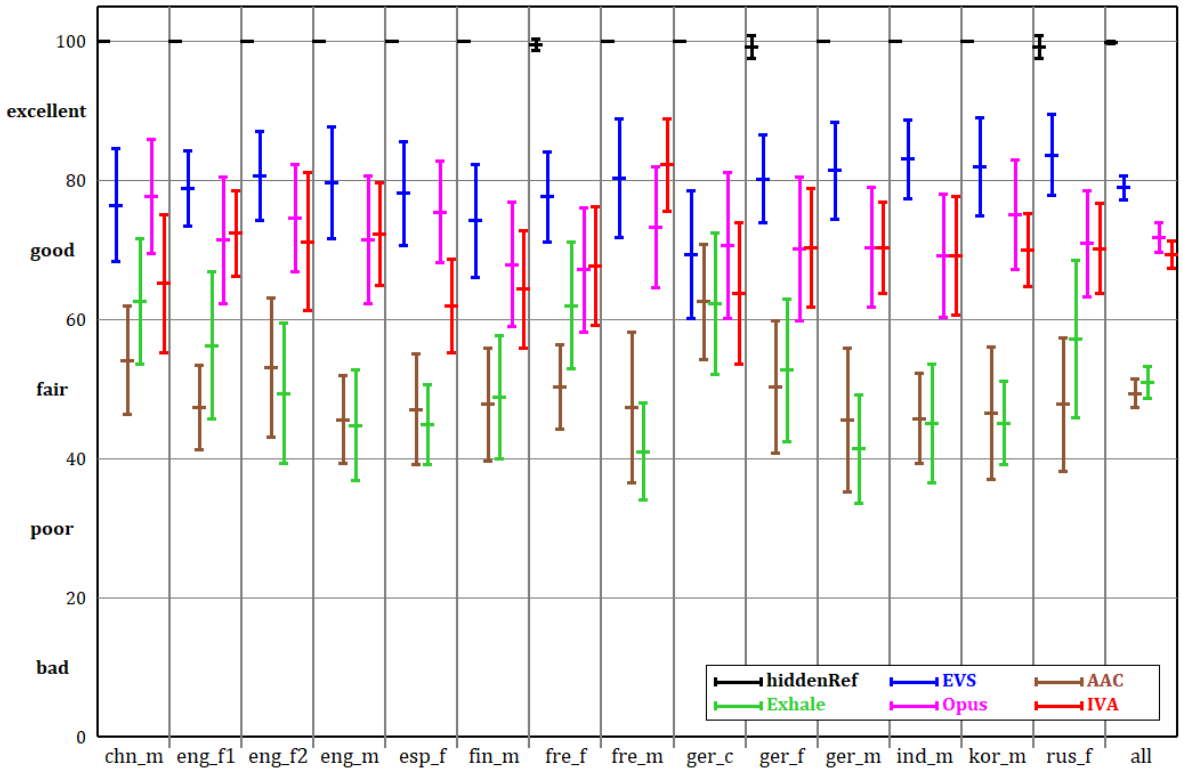

Figure 9.13 The final listening test results at FB for clean speech items, 13 listeners

	EVS	AAC	Exhale	Opus	IVA
PEAQ Basic	-2.05	-3.32	-2.55	-2.78	-3.21
PEAQ Advanced	-3.07	-3.70	-3.37	-2.93	-3.52
POLQA	4.63	4.30	4.49	4.57	4.41

Table 9.5 Objective measurements for the final listening test clean speech items

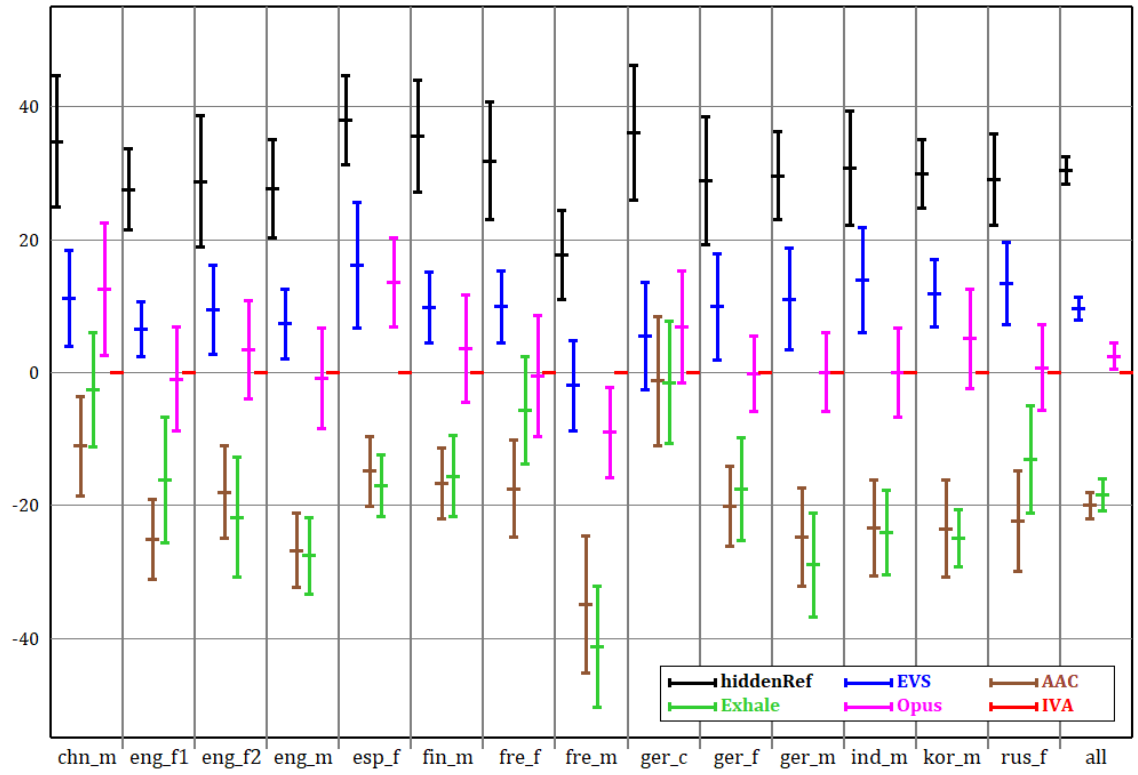

Figure 9.14 Differential scores for the final listening test results for clean speech items

The first among the final tests includes only clean speech. The absolute scores are shown in Figure 9.13. The differential scores, relative to IVA, are displayed in Figure 9.14. Mean values from the objective measurements of the tested clean speech items are listed in Table 9.5.

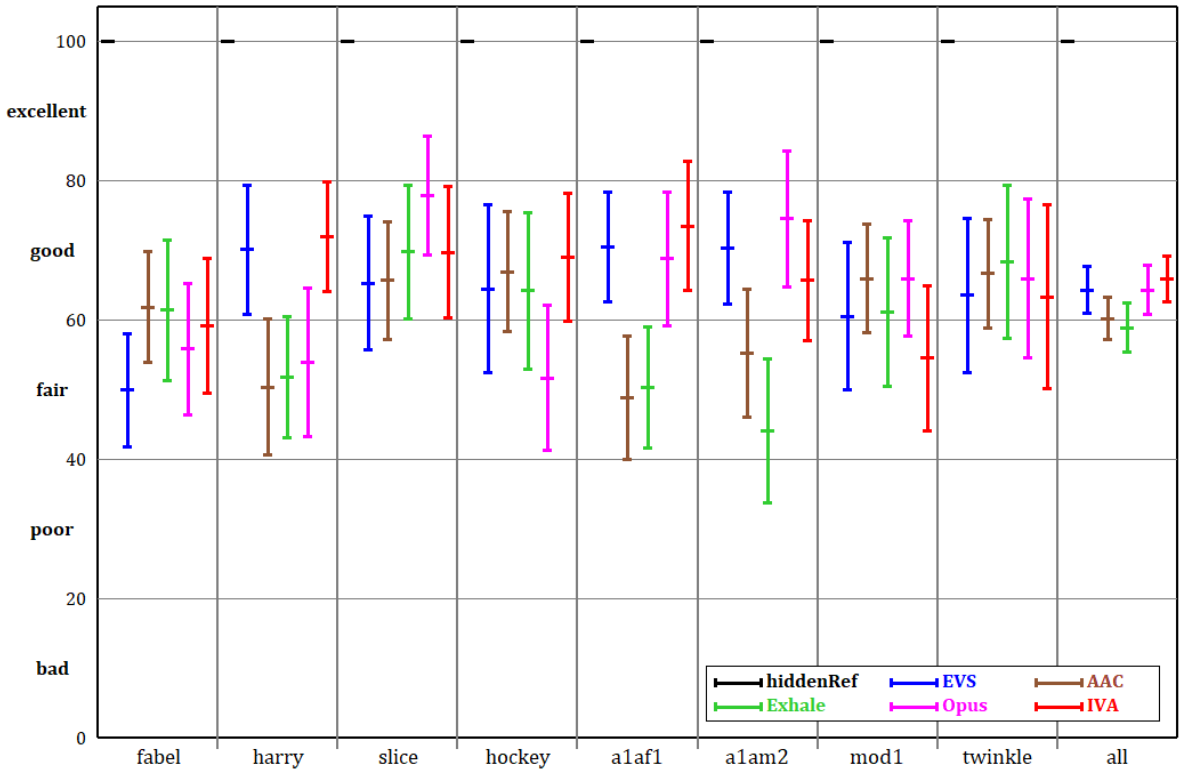

Figure 9.15 The final listening test results at FB for noisy and mixed items, 15 listeners

	EVS	AAC	Exhale	Opus	IVA
PEAQ Basic	-2.44	-3.05	-2.36	-2.67	-3.12
PEAQ Advanced	-3.30	-3.22	-2.78	-3.05	-3.17
POLQA	4.62	4.67	4.75	4.55	4.68

Table 9.6 Objective measurements for the final listening test noisy speech and mixed items

The second among the final tests includes speech with background noise and speech mixed with music. The absolute scores are shown in Figure 9.15. The differential scores, relative to IVA, are displayed in Figure 9.16. Mean values from the objective measurements of the tested noisy speech and mixed items are listed in Table 9.6. The noise and mixed items were tested separately to the clean speech, so that separate comparison of codecs is achieved in each of these 2 categories.

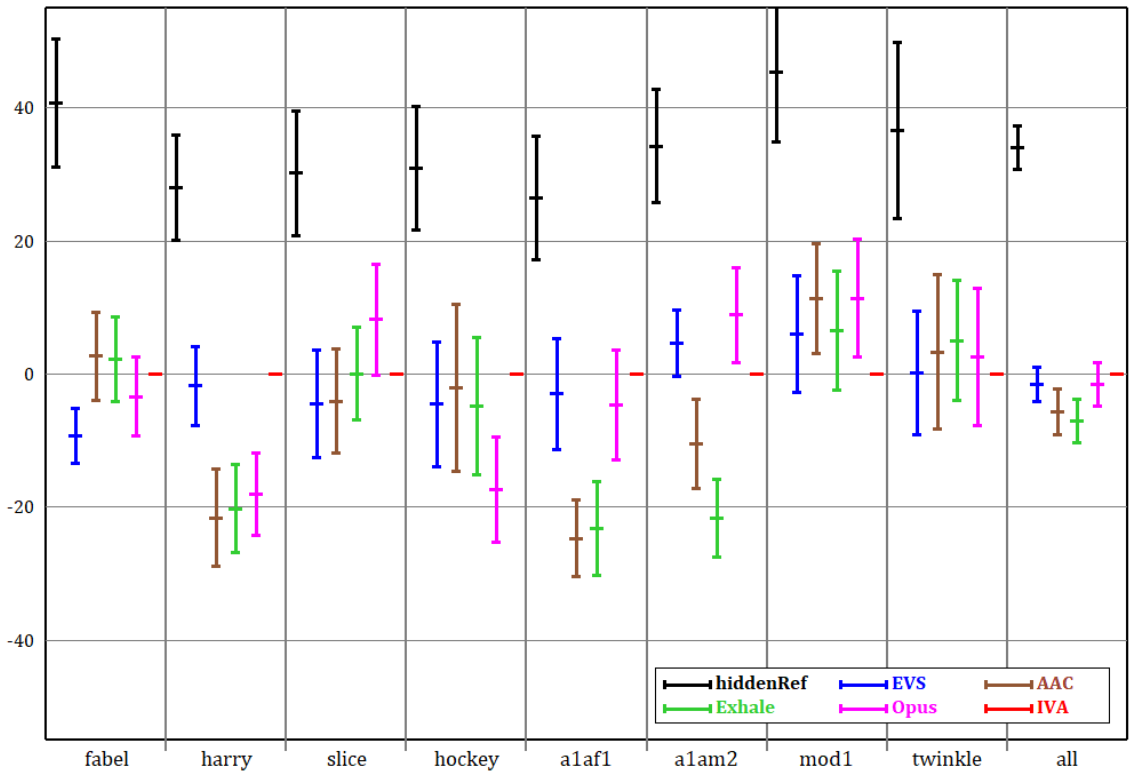

Figure 9.16 Differential scores for the final listening test results for noisy and mixed items

The third among the final tests includes music. The absolute scores are shown in Figure 9.17. The differential scores, relative to IVA, are displayed in Figure 9.18. Mean values from the objective measurements of the tested music items are listed in Table 9.7.

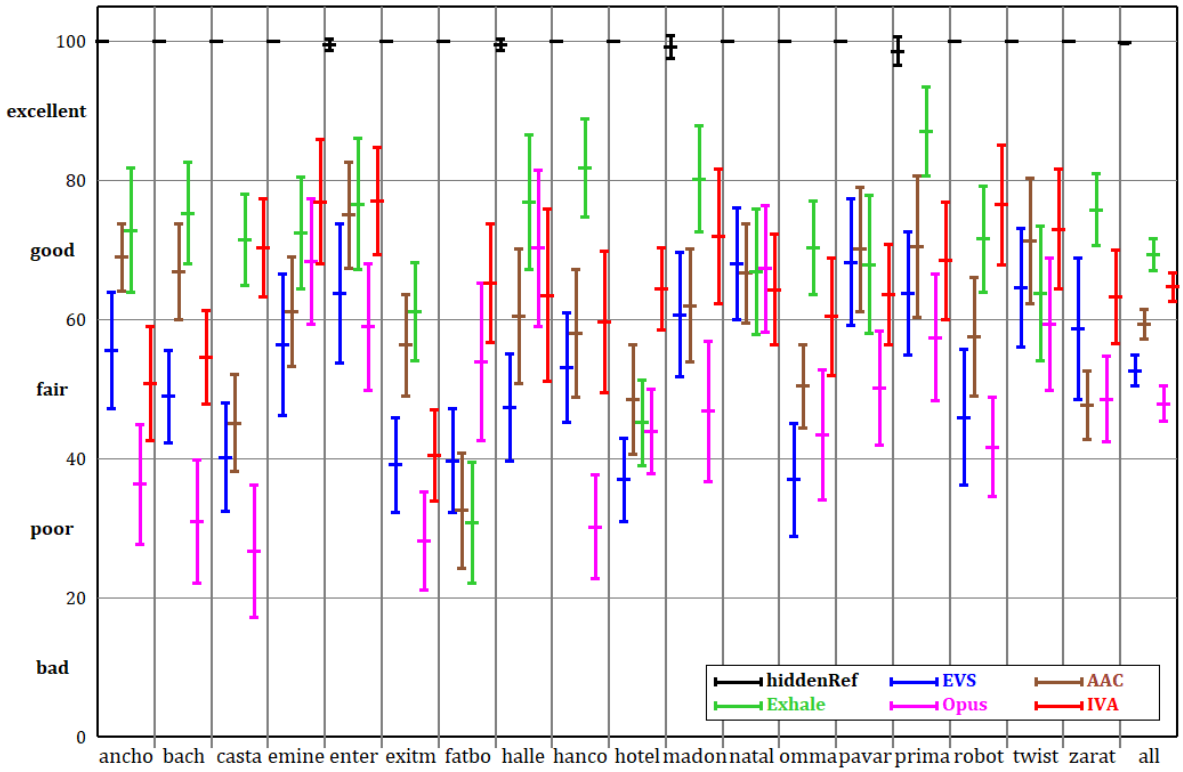

Figure 9.17 The final listening test results at FB for music items, 13 listeners

	EVS	AAC	Exhale	Opus	IVA
PEAQ Basic	-2.19	-3.07	-2.56	-2.92	-3.00
PEAQ Advanced	-3.39	-3.37	-3.31	-3.17	-3.34
POLQA	4.62	4.56	4.66	4.56	4.61

Table 9.7 Objective measurements for the final listening test music items

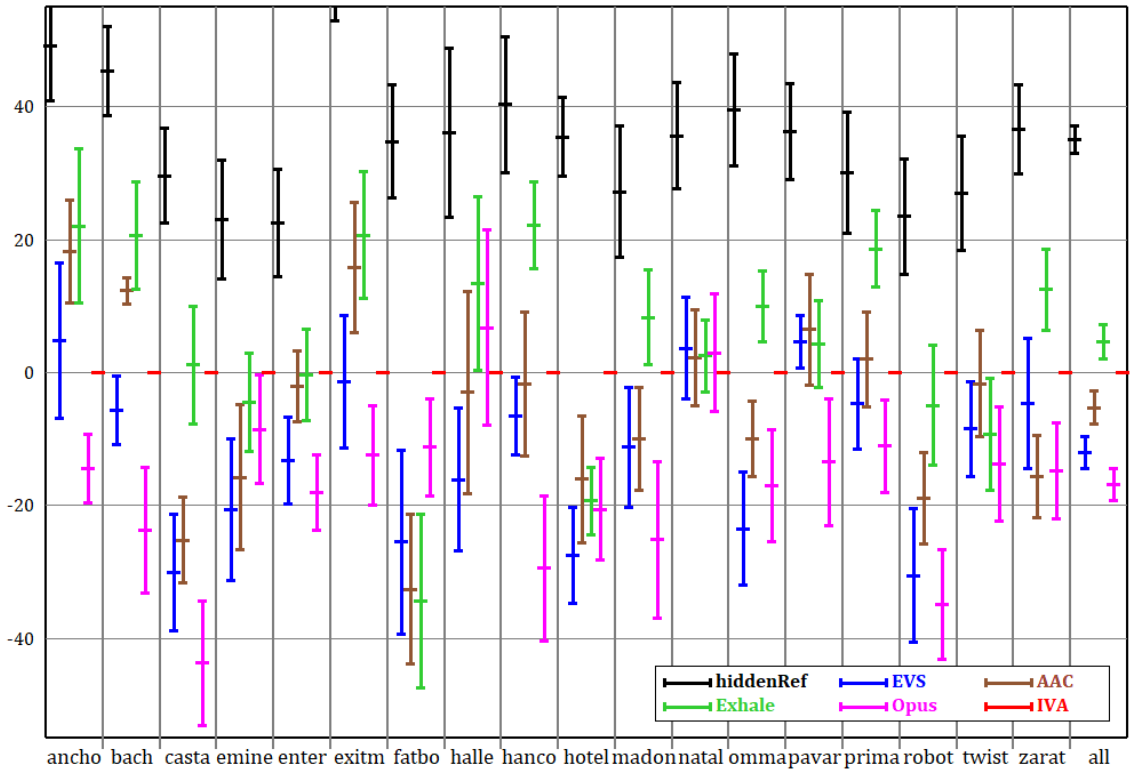

Figure 9.18 Differential scores for the final listening test results for music items

Figure 9.19 shows the averages from each of the three final tests and also the average across all items from all three tests. Differential scores, relative to EVS, are also displayed for average across all items from all three tests.

Table 9.8 lists mean objective scores among all items from the three final tests. Table 9.9 lists mean objective scores for 123 various speech and music items, also including the items used in the final tests.

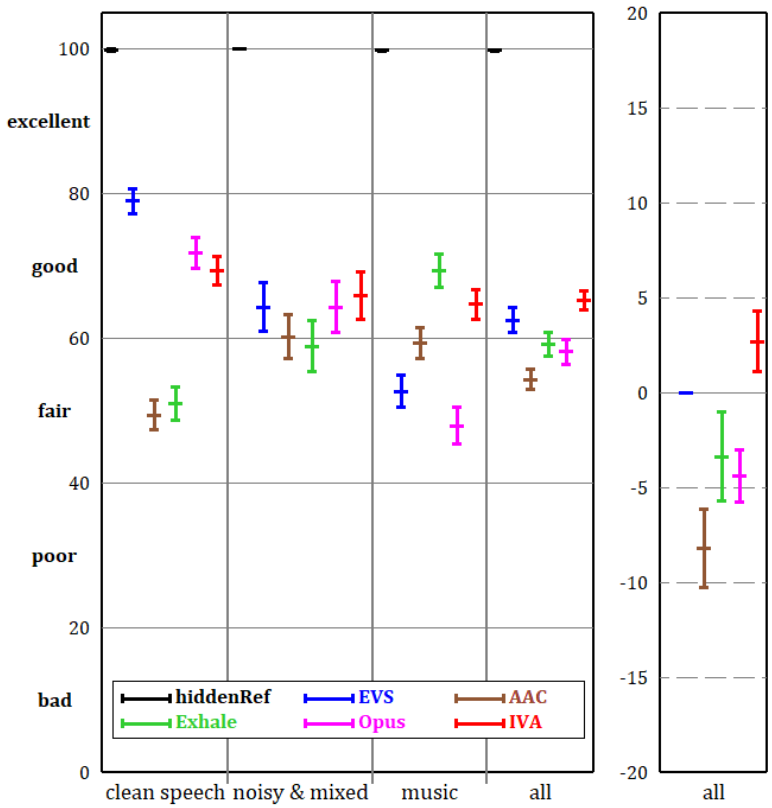

Figure 9.19 The final listening test results for all items

	EVS	AAC	Exhale	Opus	IVA
PEAQ Basic	-2.25	-3.15	-2.47	-2.76	-3.13
PEAQ Advanced	-3.24	-3.42	-3.09	-3.03	-3.33
POLQA	4.62	4.52	4.64	4.56	4.57

Table 9.8 Objective measurements for all final listening tests' items

	EVS	AAC	Exhale	Opus	IVA
PEAQ Basic	-2.38	-3.05	-2.40	-2.72	-3.08
PEAQ Advanced	-3.28	-3.17	-2.81	-3.03	-3.22
POLQA	4.50	4.56	4.62	4.42	4.57

Table 9.9 Objective measurements for 123 items

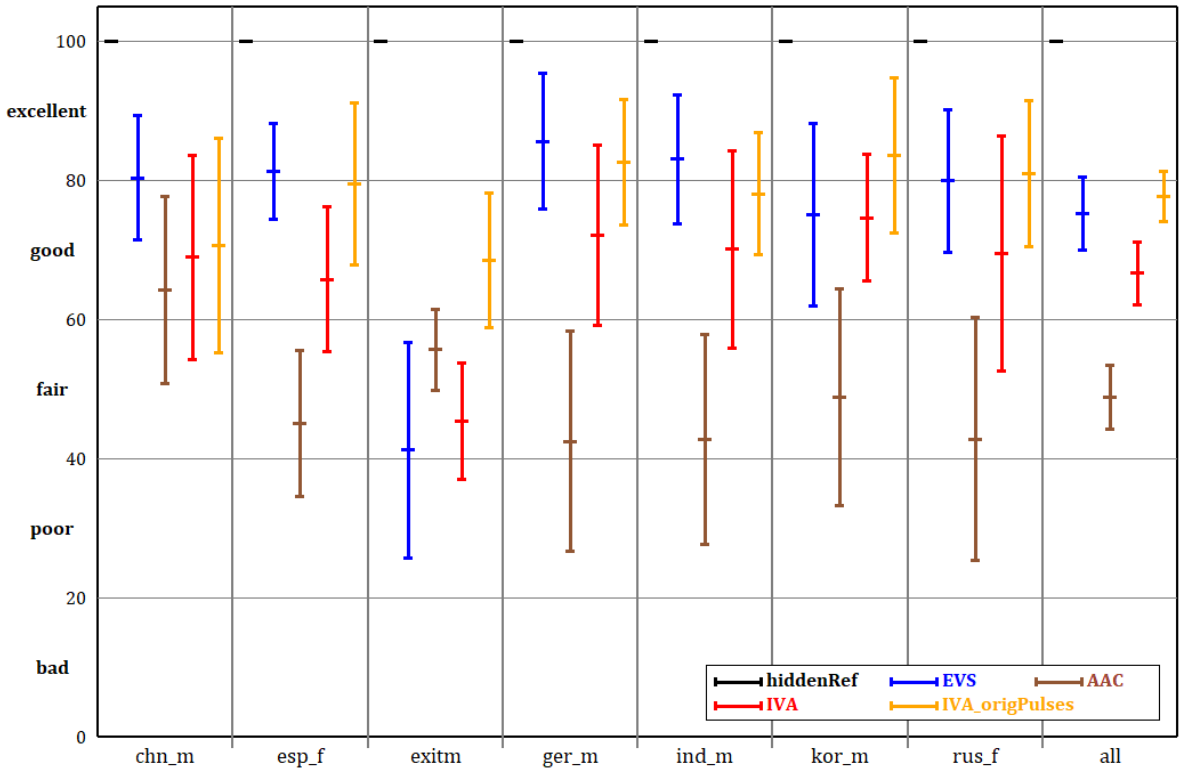

Figure 9.20 The listening test results of checking pulse coding, 7 listeners

IVA has significantly worse quality than EVS for most clean speech items [Figure 9.14] and poor performance for one music item [Figure 9.17]. It was checked, in an additional listening test, if the pulse coding is the reason for this. The worst performing music item exitm and 6 clean speech items, with biggest difference to EVS, were tested. The condition IVA_origPulses is IVA with original pulses added back, instead of the coded pulses; the rest of the codec remains unchanged, including the same pulse extraction. Listeners from the final clean speech test [Figure 9.14], that were most critical towards IVA, participated in this test. The results are shown in Figure 9.20.

9.2 Discussion

It should be stressed out again that most of the listeners have years of experience in listening to audio and speech coding artifacts and their experience provides valid results with smaller number of test participants [179].

The averaging over all items in the last columns of the result figures should be regarded only as an indicator or an overall tendency. It is highly dependent on the items chosen for the test and may be misleading [180]. This has to be especially considered for Figure 9.19.

Assuming symmetrically distributed data, t-test can be implicitly carried out by looking at the confidence intervals [180]. It is under question whether our sample of data is representative of the population and for sure the sample size is very small. Unfortunately, because of the cost of the experiments, it is not possible to obtain more data. It is also open question, what is the population in this case: is it the general public or is it just the expert listeners [58]. To not overcomplicate the problem, the test data will be simply assumed symmetrically distributed.

The purpose of the initial listening tests at WB is to show that good quality for various types of signals, including speech, can be achieved with the proposed system [Figure 3.1], the system being based on the MDCT codec and including the pulse coding, LTP and HPF. For the WB speech items, IVA produces results (Figure 9.1, Figure 9.3) that are on par with the contemporary codecs, including currently the best available speech codec EVS. IVA is the only codec having at least good quality for all music items at 24.4 kbps WB (Figure 9.2, Figure 9.4). Every other codec, except IVA, shows significant weakness for some of the tested music signals. It should be noted that 24.4 kbps WB mono is not target operating point for xHE-AAC and that it might not be well tuned for this operating point. Nonetheless, xHE-AAC has much bigger delay and VBR allocation among the frames in the super frame, in that manner allowing the results to be considered valid.

The difference of the quality between the IVA versions v73 and v93, prompted for detailed analysis of the importance of LTP and HPF. The new LTP and the new HPF were compared against the baselines (with no LTP and no HPF) and against existing methods.

The advantage of the proposed LTP [6.2-6.6], that adapts to pitch contour independent of the codec framing, is visible in Figure 9.6 for speech items and in Figure 9.8 for music items. The listeners have noticed advantage of the new LTP, over FDP and the simple TD LTP, for half of the speech items and for 6 out of 9 music items. There is also no significant degradation of the new LTP over the other LTP methods. The objective scores on the 17 tested items (Table 9.2) are in concordance with the averages of the listening tests. The objective scores on a bigger set of items show the same tendency (Table 9.1). It is also visible in Figure 9.10 and Figure 9.12 that "with LTP, EVS HPF" is significantly better than "no LTP, EVS HPF" for 9 out of 20 items. This means that LTP is advantageous even when HPF is active. The average objective scores also confirm this tendency (Table 9.3 and Table 9.4). The only noticed potential problem is the degradation over the version without LTP for [rocky] item. From all these results it can be concluded that the new LTP provides improvements over the baseline and over the existing methods. Therefore, it is activated per default in the IVA versions used for other listening tests.

A tendency that "no LTP, EVS HPF" performs better than "with LTP, no HPF" is visible from Figure 9.9 for speech items. The picture is mixed for music items [Figure 9.11]. Additionally, the advantage of "with LTP, EVS HPF" over "no LTP, EVS HPF" is for most items not very big [Figure 9.10, Figure 9.12]. This is important for low computational complexity systems. The complexity of LTP, that includes internal decoder and an additional MDCT, is significantly higher than the complexity of the simple HPF from EVS. If increase in computational complexity doesn't pose a problem, including both LTP and HPF will give significant improvements over having only LTP or HPF [Figure 9.10, Figure 9.12].

Comparing the newly proposed HPF in "with LTP, IVA HPF" to the EVS HPF in "with LTP, EVS HPF", significant improvements on 4 speech and 1 music items are observed. However, there are 2 significant degradations for music items. For one of the degradations the quality is close to the excellent range and could be sacrificed for improving perceptual quality of speech coding. Significant degradation for item [pavar] should be investigated. The new LTP includes fallback to constant pitch over the 20 ms framing; the decision to use constant pitch could be tuned, so that the degradations in quality are avoided.

The scores in Table 9.4 show that PEAQ is incapable of predicting the effect of HPF. POLQA shows the same ranking as the listeners for the tested items. From the POLQA results over 95 items in Table 9.3, it is expected that the new LTP and the new HPF don't introduce overall degradation.

In the final clean speech test [Figure 9.13, Figure 9.14], IVA performs close to Opus, having 2 significantly worse scored items and 1 significantly better. For most of the clean speech items, IVA has significant and big advantage over the state of the art FD codecs (AAC and Exhale). EVS is significantly better than IVA for most of the clean speech items, the average score of EVS being about 10 points higher than the IVA's score. The objective measurements in Table 9.5 fail to predict even the ranking of the codecs for the items used in the listening test.

The final test results with noisy speech and mixed samples [Figure 9.15, Figure 9.16] show item dependent listeners' preference for a codec. The samples [a1af1] and [a1am2] have low level of background noise and their quality ranking resembles the results for the clean speech. The items with significant amount of mixed music (fable, slice, twinkle) show good performance even for the FD codecs. The item [mod1] has multiple speakers, high level of background noises and seems to be a critical item for IVA. Overall IVA shows most balanced quality and highest mean value. Again, all of the objective scores listed in Table 9.6 are unreliable quality predictors even for the mean or the ranking of the tested codecs.

Exhale shows its advantage, with long window and VBR, over the low latency codecs (EVS, Opus, IVA) in the final music listening test in Figure 9.17 and Figure 9.18. Improvements of the new generation of FD codecs and the advantage of VBR are evident when comparing Exhale and AAC. IVA has significantly better quality than Opus in 16 out of 18 music items and significantly better quality than EVS in 12 out of 18 music items. This shows that quality of music coding can be significantly improved without significantly increasing latency. EVS is significantly better than IVA in only 1 music item - [pavar], which already showed problems with the new HPF. IVA shows just one outlier with bad quality – [exitm]. The objective measurements' mean values in Table 9.7, once more show that they are not reliable.

Overall, the quality of IVA seems more balanced than with other codecs, when looking at the grades for all items in the three final tests. The averages in the three final tests in Figure 9.19 also show this tendency of balanced quality for IVA.

The average objective scores for all items in the three final tests [Table 9.8] confirm previous findings in [56, 181] that the objective perceptual measurements are not precise. Nevertheless, checking objective scores on a large set of items is needed to confirm that the selected items are no special cases. Mean values of the objective measurements, across a set of 123 items are shown in Table 9.9. Based on the objective measurement, total discrepancy to the presented listening tests' results shouldn't be expected on other set of test items.

The listening test results in Figure 9.20 demonstrate that there is still potential in improving the pulse coding. IVA with uncoded (i.e. original) pulses has at least fair performance for every item and provides clean speech quality on par to EVS.

It is envisioned for a future work to test the importance of each tool (namely LTP, HPF, iBPC and the pulse coding) in the final version of IVA.

9.3 Computational complexity analysis

The computation complexity of IVA was estimated in the terms of weighted million operations per second (WMOPS) as defined in the ITU-T Software Tool Library 2009 [182]. The estimated WMOPS values are listed in Table 9.10 and Table 9.11. For reference, LC3 requires 19 WMOPS [129] and EVS 88 WMOPS [91], both operating in the SWB mode.

	Worst case encoder complexity (WMOPS)	Worst case decoder complexity (WMOPS)	Worst/worst combined complexity (WMOPS)
SWB	102	34	136
FB	176	66	242

Table 9.10 Worst case complexity for IVA

	Worst case pulse encoder complexity (WMOPS)	Worst case pulse decoder complexity (WMOPS)	Worst/worst combined complexity (WMOPS)
SWB	60	21	81
FB	125	48	173

Table 9.11 Worst case complexity for the pulse coding

Even though IVA was not well tested in SWB mode, it produces reasonable output at 32 kHz. It is unlikely that a tuning of the parameters for the SWB mode would significantly change the complexity.

The fast Fourier transform (FFT) operates on the length of 192 samples in the IVA pulse coding in the FB mode. Its implementation is not optimized for this length and requires 85 WMOPS in total (for both the encoder and the decoder). In the SWB mode, the FFT operates on 128 samples and requires 25 WMOPS. Even in the SWB mode, the FFT is yet to be optimized for real sequence inputs and outputs.

Significant complexity contributors are also the transformations of the spectrum in the pulse coding, namely between the Cartesian and the polar coordinates and between the linear and the logarithmic scale. The cosine, sine, logarithm and exponential functions require 16 WMOPS in the pulse extraction on the encoder side in the SWB mode. The complexity could be reduced by implementing the cosine and the sine functions with less precision. It would also be worth to investigate using non-logarithm scales in the pulse extraction.

Another source of the high complexity is the cross-correlation between the pulses [5.3.2], requiring 15 WMOPS on the encoder side. It could be optimized by adaptively reducing the shift range over which the cross-correlation is calculated, for example using the available pitch information.

The initial implementation of IVA has reasonable complexity in the SWB mode and there is a possibility to optimize the newly proposed methods.

10 Conclusion

10.1 Summary

In this thesis, a new coding scheme for coding any type of audio signal is proposed. First, existing solutions are presented. Shortcomings of recently standardized solutions are identified: specialized and separate approaches for speech and music coding, need for signal classification with contradicting requirements, problematic switching between specialized approaches, separation of parametric bandwidth extension from core coding. Afterwards, the coding scheme, named Implicit Voice or Anything (IVA), is described. It uses only one paradigm for coding both music and speech, making switching obsolete. IVA extends MDCT-based coding with: TD coding of pulse-like portions of input signals, Long-Term Prediction (LTP) and Harmonic Post-Filtering (HPF). Further, parametric coding of spectral portions, named integral Band-wise Parametric Coding (iBPC), is tightly integrated in IVA's MDCT coding. Differences of the proposed pulse coding, LTP, HPF and iBPC to similar existing methods are presented and their advantage is argued. The proposed MDCT-based coding scheme is the first one that includes all three common speech coding tools (TD pulse coding, LTP, HPF). It provides operation at low bitrate and low latency (< 36 ms), outputting signal at full bandwidth.

Finally, performance of IVA is compared to state-of-the-art codecs at around 24 kbps. For clean speech signals, its performance is close to the best existing LP speech codecs, far exceeding quality of other transform codecs. For music signals, its performance is for most items on par with the best high-latency transform codecs, far exceeding quality of other low-latency codecs. Overall, IVA provides the most consistent quality over different signal types. The tests show that it is possible to keep music performance of high-latency transform codecs, even at constant low bitrate and low latency. At the same time performance for speech signals is significantly improved.

Extending low-latency MDCT-based coding, with tools known from speech coding, is a promising approach at low bitrates. Improvements are evident for both speech and music signals. Unified and universal coding scheme can deliver perceptual quality on par to switched systems, even at low latency and at bitrates as low as 24 kbps.

10.2 Considerations for future research

Based on the results in [Figure 9.20], it seems that there is significant potential in improving quantization and coding of pulses. Perceptual irrelevancy in pulse quantization remains an open subject. Additionally, techniques for pulse coding from ACELP or a more parametric approach may be investigated.

Optimal distribution of bits between pulse and spectrum coding is not investigated. This gives another opportunity for future research.

Degradations are observed for some music samples [Figure 9.12] with the activation of LTP and HPF. Since the activation is signal adaptive, these degradations should be avoidable and it remains to be investigated how to devise a signal adaptive decision.

Another noticed problem is low quality of polyphonic music coded with low-latency codecs [Figure 9.17]. Spectral leakage of short MDCT window reduces discrimination of tonal signal components. LTP and HPF significantly improve coding of monophonic signals, but may even bring degradations in polyphonic signals. Potential direction is a research on specialized prediction and post-filtering of polyphonic signals. The challenge is that there are multiple signals that need to be independently predicted with small amount of side information.

The computational complexity of the initial implementations, especially of the pulse extraction and coding, is high. Optimizations of the proposed methods or an investigation of less complex approaches is needed.

Coding of LP residual was not investigated in FB because focus was put on FDNS. Future work may be done on solving problems with TDNS [3.2,3.3] and introducing new tools in TDNS-based coding. Comparison of such approach with FDNS-based IVA would be very interesting.

Data-driven approaches have recently shown significant improvements in many areas, including speech and audio coding. Deep Neural Networks (DNNs) may be employed to improve post-filtering [183] or for context modeling in the arithmetic coder [184].

A.1 Design details

A.1.1 Choice of the source code base

The MDCT is used in most general audio codecs and provides decorrelation even for polyphonic music signals. Even though the coding gain of the MDCT is close to the KLT, other transformations should not be a priori discarded. In [185] different frequency domain approaches are investigated including the DFT, the DCT, the KLT and Vandermonde decomposition [186]. The listening test results show that the MDCT has advantage over other transforms also for speech signals. Considering all this, the MDCT is used in this thesis, either for transforming the LP residual or the input signal, followed by the frequency adaptive quantization.

Looking at the listening test results in [185], MDCT-based TCX from EVS is a good starting point for development of an universal FD-based codec. The C source code of MDCT-based TCX is available at [120] and the version 13.1 is used as a starting point for developing IVA. Only the 48 kbps MDCT-based TCX configuration from EVS is used. There is no switching between ACELP and MDCT-based TCX at 48 kbps in EVS and the transform coder is tuned at this configuration to also deliver good performance for speech signals. The source code is stripped of the remaining algorithms, including algorithms needed exclusively for the ACELP and the HQ MDCT modes. Also the code for packet-loss concealment (PLC), discontinuous transmission (DTX), comfort noise generator (CNG), AMR-WB interoperable mode and most of the pre- and post-processing is removed [36]. PLC, DTX and CNG are important tools used in speech transmission, but investigating them is beyond the scope of this thesis. AMR-WB interoperable mode is not of interest as being only a backward compatibility necessity. Only processing needed for MDCT-based TCX remained, namely: high-pass/DC rejection filter, time domain transient detector and HPF. The other pre- and post-processing tools are not needed. Adaptive low-frequency emphasis (ALFE), the harmonic model from the arithmetic coder and the residual TCX coding are also removed. From the two arithmetic coding methods, the envelope-based arithmetic coding is removed and the context-based arithmetic coding is kept, as only the latter is used in EVS at the bitrates of interest. Such stripped codec basically resembles the LC3 codec [79, 80] which was developed at the same time period as IVA. The new methods for pitch analysis [4.3] and HPF [8.2,8.3] replace the methods from EVS.

A.1.2 MDCT Windowing

In the MPEG-D USAC standard [32], a so-called low overlap window is used [187]. The low overlap window is a combination of the rectangular and sine window and equal to the square root of the Tukey window. In the MP3 and AAC terminology it can be regarded as the stop and start window. The overlap length of the low overlap window can be instantaneously adapted for handling transients [128].

As a reminder, the input signal is divided into overlapping blocks, element-wise multiplied with an analysis window w_A, transformed into FD, coded and decoded, transformed back from FD to TD and element-wise multiplied with a synthesis window w_S. The overlapping decoded and windowed signal blocks are added. The perfect reconstruction is required, meaning that in the absence of the quantization and rounding errors, the reconstructed signal obtained from the added overlapping blocks is exactly the same as the input signal. The Princen-Bradley conditions assure window characteristics for the perfect reconstruction: $w_A[i+H]w_S[i+H] + w_A[i]w_S[i] = 1$, where $w_A[i]$ $(0 \leq i < 2H)$ is the analysis window, $w_S[i]$ is the synthesis window and H is the block hop size or half of the window length (window including the trailing zeros).

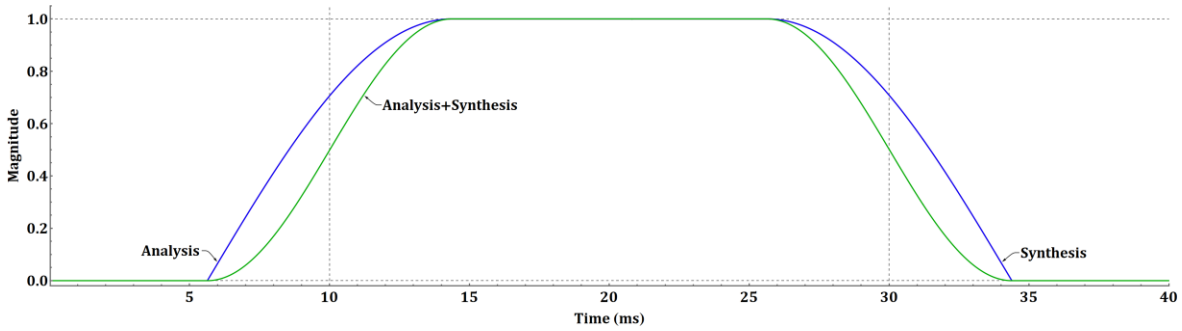

Figure A.1 Low overlap window

As can be seen in Figure A.1, the low overlap window is symmetric and thus the analysis window, used in the encoder before the MDCT, and the synthesis window, used in the decoder after the inverse MDCT, are the same. A block of the input signal is temporally shaped after encoding and decoding (assuming no quantization) as presented with the Analysis+Synthesis in Figure A.1, which is the square of the low overlap window, being equal to the Tukey window. Folding points of the MDCT are presented with the vertical grid lines.

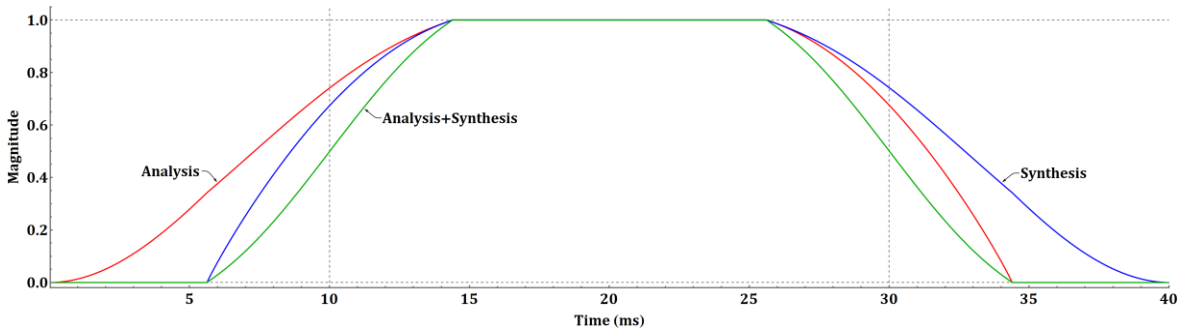

Figure A.2 ALDO window

The ALDO window from EVS [37] is asymmetric, but the joint windowing of the analysis and the synthesis is still symmetric, as presented in Figure A.2. The synthesis ALDO window is the time reversal of the analysis window.

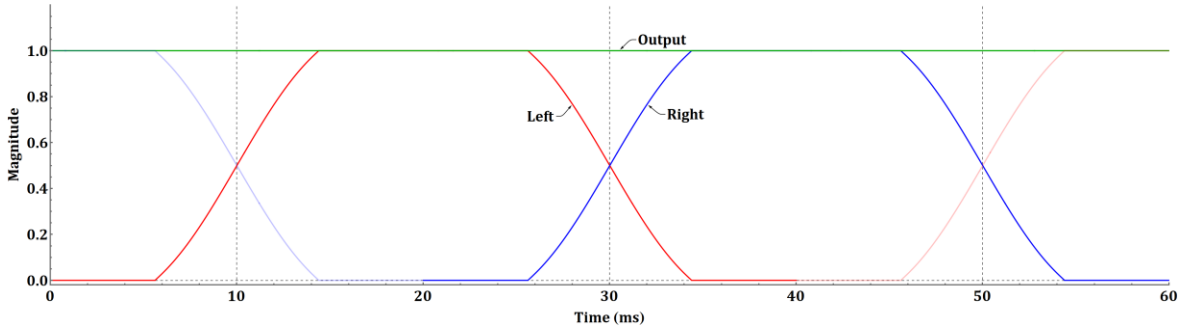

Figure A.3 Perfect reconstruction with the ALDO window

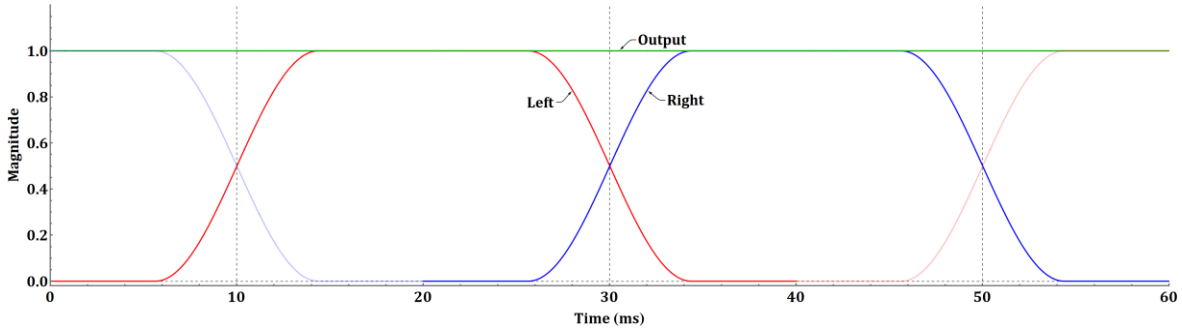

Figure A.4 Perfect reconstruction with the low overlap window

Adding the consecutive overlapping signal blocks, windowed by the analysis and the synthesis window, there is no change in the magnitude of the signal, as shown in Figure A.3 for the ALDO and in Figure A.4 for the low overlap window. This is expected from the perfect reconstruction requirement for the windows and the Princen-Bradley conditions [64].

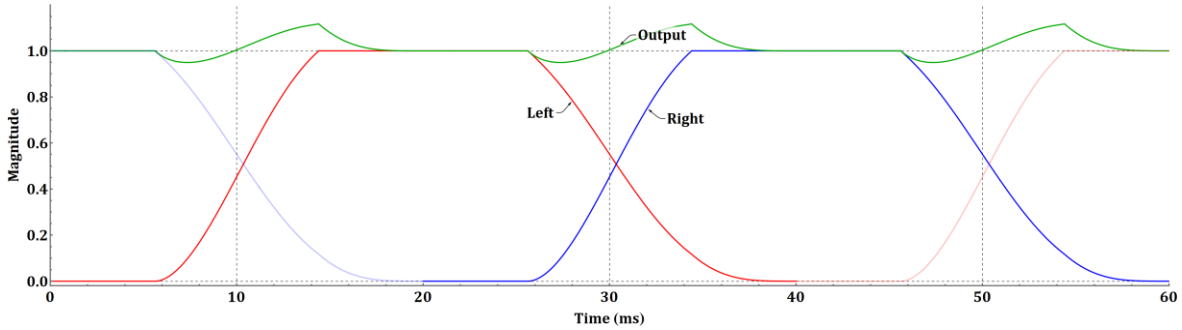

Figure A.5 Temporal shaping of the quantization noise with the ALDO window

Assuming that the quantization noise has additive white noise characteristics and that it is uncorrelated in the consecutive blocks, its temporal envelope is affected only by the synthesis window [87]. The error of a codec output consequently has a temporal envelope shaped by $(w_S[i+H])^2 + (w_S[i])^2$, where $w_S[i]$ is the synthesis window. For symmetric windows, w_S is equal to w_A and from the Princen-Bradley conditions it follows that $(w_S[i+H])^2 + (w_S[i])^2 = 1$, making the error signal's temporal envelope flat for the symmetric windows (the same temporal envelope as "Output" in Figure A.4). The temporal envelope of the error signal

exhibits amplitude modulation for asymmetric windows, with the example for the ALDO window as shown in Figure A.5.

The ALDO window improves spectral characteristics of the transform at the expense of the temporal characteristics. It is expected that the advantage for coding tonal signal will be smaller if LTP and HPF are present. Because of ALDO's temporal modulation and because it is easier to maintain and adapt low overlap window, the ALDO window is replaced with the low overlap window. Tuning of other aspects of a codec are sometimes affected by the window choice, yet it is possible to later change the window and use the most appropriate one for the desired application.

A.1.3 Psychoacoustic model

At low bitrate and low delay audio coding, coding artifacts become increasingly complex [188] and cannot be modeled by a white noise additive model [189]. Using elaborate psychoacoustic models as in [76, 77] or even optimized fastenc model [190] introduces computational complexity burden. Even the most sophisticated objective perceptual measurements are not able to evaluate quantization errors coming from parametric tools [191]. For coders with parametric tools, the quantization error is above masking threshold and simply satisfying requirements of low noise-to-masking ratio (NMR) [192] is of limited usefulness. For low bitrate coding, masking models need to be replaced with annoyance models. An annoyance model would give answer how annoying is an artifact coming from the quantization. Such models would be very complex, probably require analysis by synthesis approach and lack experimental background. It is questionable whether additional complexity, even by the optimized fastenc, is justifiable for the codec targeted in this thesis. Instead, simple and implicit psychoacoustic models were used, as in [37, 79, 80]. When tuning specific parameters of the codec, informal listening or listening tests with small number of participants were employed to assess the annoyance of the artifacts.

A.2 Spectrogram

The spectrogram is a heat map of the short-time Fourier transform (STFT) magnitudes. For calculating the STFT, signal is divided using a sliding window into overlapping blocks and each block is transformed using the discrete Fourier transform (DFT). Of course the fast Fourier transform (FFT) is used for calculating the DFT. For the spectrogram figures in this thesis, unless a spectrogram is obtained directly from the codec, the consecutive blocks have at least 75 % overlap and sine window is used. In coding, the large overlap would be counterproductive. For the signal presentation, the larger overlap improves visualization of the signal continuity and makes visible changes in pitch.

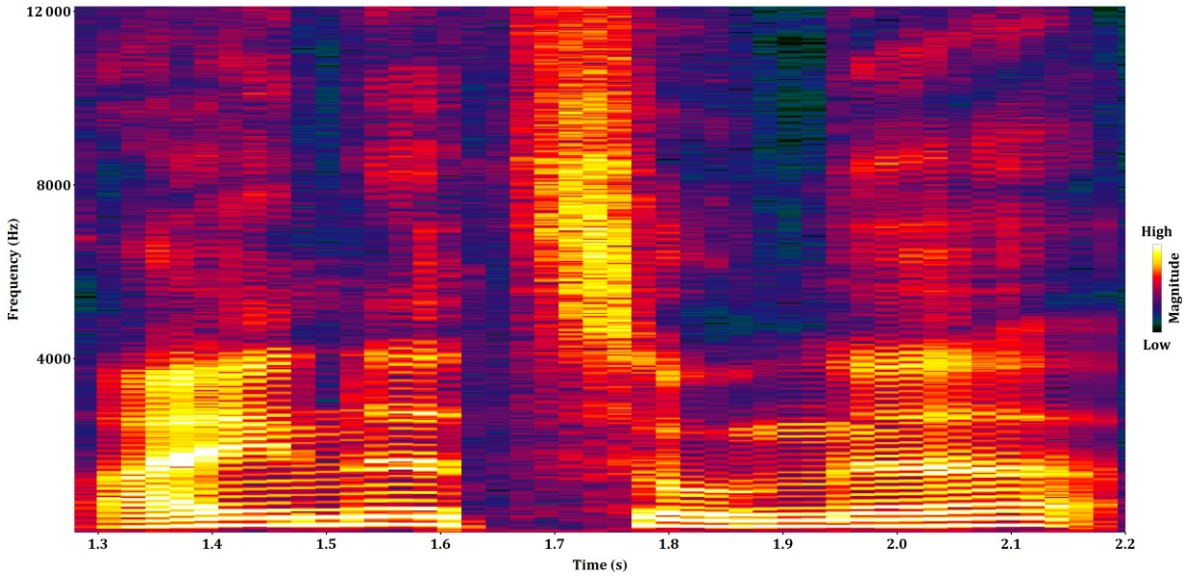

Figure A.6 Spectrogram of a speech with 42.7 ms window of 50 % overlap

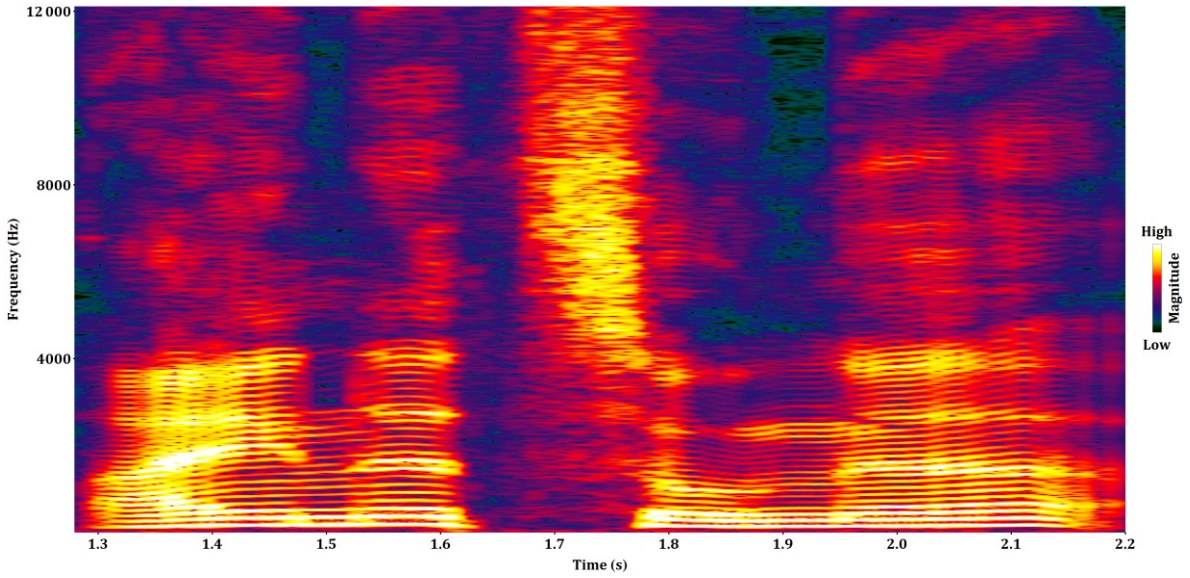

Figure A.7 Spectrogram of a speech with 42.7 ms window of 97 % overlap

With the long window, high spectral resolution is achieved in Figure A.6 and Figure A.7. The advantage of the longer overlap in the visualization is obvious, as the continuous pitch change and distinction of harmonic and noisy parts of the signal become visible.

With the short window in Figure A.8 and Figure A.9, temporal resolution is increased at the expense of reduced frequency resolution. The same speech signal is presented as in Figure A.6 and Figure A.7, but only from 1.65 s to 2.2 s. For such high temporal resolution, the longer overlap helps in noticing the regular pulse structure in the voiced phones.

It could be argued that the higher overlap compensates for the unavailability of a phase presentation in a spectrogram.

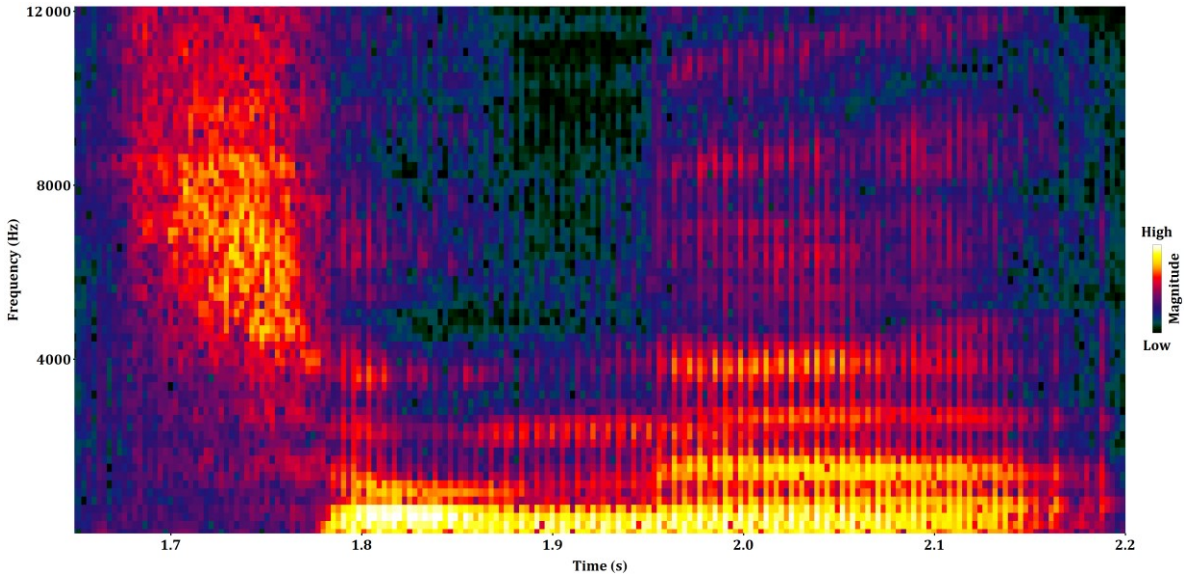

Figure A.8 Spectrogram of a speech with 5.3 ms window of 50 % overlap

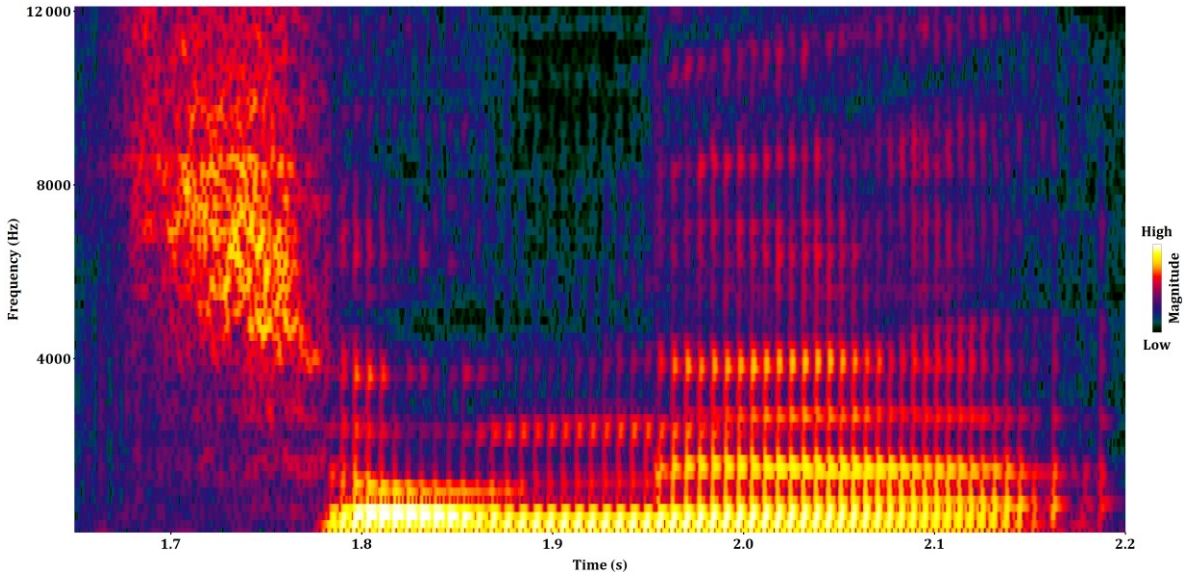

Figure A.9 Spectrogram of a speech with 5.3 ms windows of 94 % overlap

For presenting magnitudes, both visual and auditory perception needs to be considered. A detailed analysis of mapping audio spectrogram magnitudes to colors, for optimal presentation of subjective loudness sensation, is beyond the scope of this thesis. Instead, it was determined heuristically how to present as wide dynamic range as possible and to put emphasis on the most relevant magnitude differences. To handle broad dynamic range of human hearing, the magnitudes are presented on a log scale. The log magnitudes are linearly mapped to a specific range, with a signal specific choice so that most details are visible, and then raised to the power of 1.3. This approach for presenting log magnitudes may be connected to the gamma correction, which utilizes non-linear perception of color and light. Another important aspect is the choice of the color gradient. The basis is the sunset colors gradient going from white over yellow, red and violet towards black. The sunset gradient is extended with shades of dark green at low levels. Because of the signal specific mapping of magnitudes into color, the exact scale of magnitudes is in most cases omitted.

Other presentations could be used for what humans hear, for example excitation patterns [193, 194] or specific loudness [195]. These presentations use non-linear frequency and intensity scale in accordance with human perception. In contrast to these psychoacoustically motivated presentations, the linear frequency spectrograms show more details, that are irrelevant for the perception. The abundance of details may also make it hard to notice which differences are important when comparing two versions of a signal. However, the linear frequency spectrograms can be obtained fast and using them, a possible errors or an incompleteness in a psychoacoustic model is avoided. The linear frequency spectrograms also show which redundancy in a signal could be exploited, which may not be directly visible in psychoacoustically motivated presentations.

In general, the presented spectrograms are for visualization purposes. They can give insight into sources of some ideas for proposed methods, but cannot be used to check validity of them. More details on obtaining and configuring a spectrogram can be found in [196].

A.3 Samples used in the listening tests

V73 and v93 WB tests

Item	Description
	Speech
harry	English male narrator; audio book; starts with music background, that fades out; HarryPotter from USAC test items [24, 197–200]
eng_f	English female speaker; excerpt from the EBU SQAM [201]
fre_m	French male speaker; excerpt from EBU SQAM
ger_m	German male speaker; excerpt from EBU SQAM
hockey	English female commentator of a sport match with crowd cheering; SpeechOverMusic_1 from USAC test items
louis	French male speakers without crosstalk; with street noise; louis_raquin_15 from USAC test items
nadib	Japanese female and male speaking simultaneously; nadib41 from USAC test items
twinkle	French female speaker with jazz style “Twinkle twinkle little star” in background; twinkle_ff51 from USAC test items
	Music
appla	Applause with distinct and dense clapping sounds; from EBU evaluation [202]
fatbo	Starting 8s of “Kalifornia” by Fatboyslim from “You've Come a Long Way Baby”; speech heavily processed with vocoder effects
music_3	Starting 16s of “I Feel Fine” by The Beatles; Music_3 from USAC test items
opera	"Der Hölle Rache kocht in meinem Herzen" by Mozart from „Die Zauberflöte K. 620, Act 2“; soprano
waiti	Starting 14s of “Waiting” by Green Day from “Warning”; male singing accompanied by electric guitars
hanco	Jazz trumpet solo at 2:21-2:31 from “All blues live” by Herbie Hancock Quintet from “A Tribute to Miles”
kraft	0:21-0:29 second of “Robots” by Kraftwerk from “The Man-Machine”; electronic music
te09	0:34- 0:43 “Mountains O’ Things” by Tracy Chapman from “Tracy Chapman”
vival	“La primavera” by Vivaldi from “Le quattro stagioni”

LTP and HPF tests

Item	Description
	Speech
news	English male reading news; starts with very short music background, that fades out
eng_f	English female speaker; excerpt from EBU SQAM
fre_m	French male speaker; excerpt from EBU SQAM
ger_m	German male speaker; excerpt from EBU SQAM
arirang	Korean male speaker; Arirang_speech from USAC test items
harry	English male narrator; audio book; starts with music background, that fades out; HarryPotter from USAC test items
panned	Finish male speaker; PannedSpeechEkonomiekot from USAC test items
lion	English male speaker alternating with animal sound effects; from USAC test items
	Music
haydn	"Trumpet Concerto in E-Flat Major Hob. VIIe:1: III. Finale. Allegro" by Haydn; excerpt from EBU SQAM
cantw	"Can´t Wait Until Tonight" by Max Mutzke; English male singing accompanied by percussions
music_3	Starting 16s of "I Feel Fine" by The Beatles; Music_3 from USAC test items
rocky	Starting 11s of "Rock You Gently" by Jennifer Warnes from "The Hunter"; starting with deep bass and percussions, guitar and other instruments join at the end
beach	"Sloop John B" by Beach Boys from "The Pet Sounds Sessions"; vocals only
bruck	Sustained trumpets from "Symphony No.4 in E Flat Major Romantic WAB 104" by Bruckner
es01	0:24-0:34 of "Tom's Diner" by Suzanne Vega from "Solitude Standing"; a cappella English female single voice singing
harper	2:32-2:45 of "Steal My Kisses" by Ben Harper from "Burn to Shine"; multiple male English voices; a cappella at beginning, accompanied with pop rock instruments at the end
sm01	Bagpipes from USAC test items

Additional samples in HPF test

Item	Description
	Speech
twinkle	French female speaker with Jazz style "Twinkle twinkle little star" in background; twinkle_ff51 from USAC test items
	Music
berlin	Starting 12s of "Drug" by Berlin from "Voyeur"; electronic music
pavar	2:31-2:38 of "'O Sole Mio" by Luciano Pavarotti from "Pavarotti & Friends 2"; opera singer

Final tests

Item	Description
	Clean speech
chn_m	Chinese male speaker; 452374__st888__chinese-poem_noise_reduced_short.wav from freesound.org
eng_f1	English female speaker; excerpt from EBU SQAM
eng_f2	English female speaker; Fraunhofer IIS recording
eng_m	English male speaker; Fraunhofer IIS recording
esp_f	Spanish female speaker; Fraunhofer IIS recording
fin_m	Finish male speaker; PannedSpeechEkonomiekot from USAC test items
fre_f	French female speaker; Fraunhofer IIS recording
fre_m	French male speaker; excerpt from EBU SQAM
ger_c	German female and male speakers with partial crosstalk; Fraunhofer IIS recording
ger_f	German female speaker; Fraunhofer IIS recording
ger_m	German male speaker; excerpt from EBU SQAM
ind_m	Indian male speaker; Fraunhofer IIS recording
kor_m	Korean male speaker; Arirang_speech from USAC test items
rus_f	Russian female speaker; Fraunhofer IIS recording
	Noisy speech and mixed
fable	German male narrator with music background; from the audio book “Die fabelhafte Welt der Amélie”
harry	English male narrator; audio book; starts with music background, that fades out; HarryPotter from USAC test items
slice	German female radio commentator with background music; excerpt from the Slices DVD issue 4/06 including Digitalism
hockey	English female commentator of a sport match with crowd cheering; SpeechOverMusic_1 from USAC test items
a1af1	German female speaker with office noise in background. Fraunhofer IIS recording
a1am2	German male speaker with car noise in background. Fraunhofer IIS recording
mod1	Sub-saharan language speakers with background noises and drums
twinkle	French female speaker with jazz style “Twinkle twinkle little star” in background; twinkle_ff51 from USAC test items
	Music
ancho	Starting 8 seconds of “The Anchor Song” by Björk from “Debut”; saxophone, polyphonic
bach	“Allemande” by Bach from “English Suite No. 5” played on a harpsichord
casta	Flamenco guitar and castanets
emine	1:13-1:20 “White America” by Eminem from “The Eminem Show”; male vocal hip hop with rock music
enter	0:22-0:36 of “Enter Sandman” by Metallica from “Metallica (Black album)”; heavy metal/hard rock drums with electric guitars and bass

exitm	0:31-0:38 of “Exit Music (For A Film)” by Radiohead from “OK Computer”; male vocal with acoustic guitar
fatbo	Starting 8s of “Kalifornia” by Fatboyslim from “You've Come a Long Way Baby”; speech heavily processed with vocoder effects
halle	Chorus “Hallelujah” by Händel from Messiah
hanco	Jazz trumpet solo at 2:21-2:31 from “All blues live” by Herbie Hancock Quintet from “A Tribute to Miles”
hotel	6.5 s applause and crowd cheering at the end of “Hotel California” by Eagles from “Live Unplugged on MTV, Hell Freezes Over”, followed by 3.0 s of applause recording from EBU evaluation [202]
madon	0:57-1:09 of “Music” by Madona from “Music”; female vocal with disco/funk music
natal	0:50-1:10 of “Avalon” by Natalie Cole from “Unforgettable: With Love”; jazz with female vocal
omma	7:35-7:44 of “Ommadawn Pt.1” by Mike Oldfield from “Ommadawn”; flute, guitar, tambourine/rattles
pavar	2:31-2:38 of “'O Sole Mio” by Luciano Pavarotti from “Pavarotti & Friends 2”; opera singer
prima	“La primavera” by Vivaldi from “Le quattro stagioni”; different recording, with more bandwidth than the sample vival used in V73 and v93 tests
robot	0:21-0:29 of “Robots” by Kraftwerk from “The Man-Machine”; electronic music
twist	1:39-1:45 of “Twist and Shout” by The Beatles from “Please Please Me”; male vocals with pop/rock music
zarat	End of “Einleitung, oder Sonnenaufgang” by Richard Strauss from “Also sprach Zarathustra”; symphony orchestra

In many cases, the samples are shortened, compared to the original test samples, by cutting out a characteristic excerpt and removing repetitions. For samples where the duration is stated, it is the duration as used in the listening tests.

Many of the music samples are originating from public listening tests organized at Hydrogenaudio [203–205], where they are found to be critical for audio codecs.

List of Symbols

$\lfloor x \rfloor$ – Rounding down of x

$\{x\}$ – Fractional part of x

$\mathbb{i}$ – Imaginary unit

a, b, c – Scaling factor

d – Pitch lag or distance for maximum correlation

$\check{d}_{F_0}$ – Minimum pitch lag

$\hat{d}_{F_0}$ – Maximum pitch lag

$d_{\check{F}_0}$ – Pitch initial candidate in a call to $\mathcal{F}_{F_0}$

$d_{\check{F}_0}$ – Pitch search range start in a call to $\mathcal{F}_{F_0}$

$d_{\hat{F}_0}$ – Pitch search range end in a call to $\mathcal{F}_{F_0}$

$\dot{d}_{F_0}$ – Pitch in the middle of the current MDCT window

$\acute{d}_{F_0}$ – Pitch at the end of the current MDCT window

$\grave{d}_{F_0}$ – Pitch at the beginning of the current MDCT window

$\dot{d}_{F_C}$ – Pitch in the middle after the half pitch lag correction

$\acute{d}_{F_C}$ – Pitch at the end after the half pitch lag correction

d_V – Pitch contour

$\bar{d}_{F_0}$ – Average pitch lag in the current frame

$\bar{d}_{F_C}$ – Average pitch lag of the half pitch lag correction

d_L – Pitch in an LTP sub-interval

d_H – Pitch in an HPF sub-interval

d_{P_i,P_j} – Distance between the i^{th} and the j^{th} pulse waveform

d_{P_i} – Pitch lag at the position of the i^{th} pulse

$\dot{d}_C$ – Optimal distance for the iBPC copy-up

$\dot{d}_{\check{C}}$ – Minimum optimal distance for the IGF copy-up

$\dot{d}_{\hat{C}}$ – Maximum optimal distance for the iBPC copy-up

$\check{d}_C$ – Minimum distance for the iBPC copy-up

$\grave{d}_C$ – Optimal distance for the iBPC copy-up in the previous frame

e – Envelope

e_T – Temporal envelope of the high-pass filtered input signal, mean removed

f – Frequency or corresponding frequency line index

f_{F_0} – Fractional index of the MDCT bin corresponding to the fundamental frequency

f_{P_i} – Pulse starting frequency

f_N – Starting frequency for iBPC

h – Harmonicity

h_H – Desired harmonicity increase in an HPF sub-interval

g – Gain, quantization step size

g_H – Amplitude modulation in an HPF sub-interval

$g_{P_{P_i}}$ – Pulse prediction gain

$g_{I_{P_i}}$ – Pulse innovation gain

g_Q – The global gain, quantization step size

i, j, k, l, m, n, s – Index

i_{F_0} – The sub-interval center sample index

$i_{P_{P_i}}$ – Index of the pulse prediction source

s_C – iBPC copy-up start

$\check{s}_C$ – Minimum iBPC copy-up start

p – Probability or percentage

p_{P_i,P_j} – Probability that the i^{th} and the j^{th} pulse belong to a train of pulses

p_{P_i,P_j} – Probability that the i^{th} and the j^{th} pulse belong to a train of pulses

p_{E_L,P_i} – Percentage of the local energy in the pulse

p_{E_F,P_i} – Percentage of the frame energy in the pulse

p_{N_E,P_i} – Percentage of bands with the pulse energy above half of the local energy

p_{P_i} – Probability that the i^{th} extracted pulse is true pulse

t – Position in TD

$\dot{t}_{P_i}$ – Exact pulse position

t_{P_i} – Pulse center position

w - Window

x, y, z – TD signal or vector

x_I – The original input TD signal

x_{P_O} – The TD signal consisting of the original non-coded pulse waveforms

x_M – The input to the MDCT that is the residual of the pulse removal

x_D – The output of the IMDCT

x_H – The output of the HPF

x_{P_c} – The decoded TD signal consisting of pulse waveforms

x_O – The final decoded output TD signal

x_{P_i} – Pulse waveform

y_{P_i} – Spectrally flattened pulse waveform

z_{P_i} – Quantized flattened pulse waveform consisting of the prediction and the innovation

$\tilde{z}_{P_i}$ – Pulse waveform prediction without the gain

$\dot{z}_{P_i}$ – Pulse waveform innovation without the gain

A – LP filter

$\hat{A}$ – Decoded LP filter

$\hat{A}_P$ – Decoded LP filter from the previous frame

$\hat{A}_C$ – Decoded LP filter from the current frame

$\hat{A}_I$ – LP filter from the arithmetic mean of LSFs of $\hat{A}_P$ and $\hat{A}_C$

$A_{M,j}$ – LP filters from small windows

B – Sub-band; low-pass filter transfer function

B_i – Sub-band

D – Distance between pulses

$D_{\rho_{e_T}}$ – Exact delay for the maximum autocorrelation of the temporal envelope

$\widetilde{D}_P$ – Expected average distance between pulses in the current frame

$\bar{D}_P$ – Average distance between pulses in the current frame

E – Energy or noise filling level

E_{N_i} – Zero filling level for the i^{th} sub-band

$E_{\hat{A}_P}$ – Energy of $X_{\hat{A}_P}$

$E_{\hat{A}_C}$ – Energy of $X_{\hat{A}_C}$

F – Sampling rate

F_S – Sampling rate of the input signal

H – Hop size of a time to frequency transform; filter transfer function

H_M – Frame length, 20 ms

H_{F_0} – Hop size of the LTP sub-intervals

H_L – LTP filter transfer function

H_P – Hop size of the STFT for the pulse extraction and the pulse reconstruction

H_{F_H} – Hop size of the HPF sub-intervals

H_H – HPF filter transfer function

L – Length of a vector or window

$L_{\hat{O}}$ – Maximum MDCT window overlap, MDCT look-ahead length

L_H – Length of the pitch search interval in a call to $\mathcal{F}_{F_0}$

L_{F_0} – Length of the LTP sub-intervals

L_{F_H} – Length of the HPF sub-intervals

L_{W_P} – Length of the pulse waveform

L_{B_i} – Sub-band length

N – Integer number of items

N_T – Number of tonal MDCT bins

N_{F_0} – Number of the LTP sub-intervals

N_L – Number of predicted harmonics

$\widehat{N}_L$ – Maximum number of predicted harmonics

N_{F_H} – Number of the HPF sub-intervals

N_{P_X} – Number of extracted pulses in the current frame

N_{P_C} – Number of coded pulses in the current frame

N_{P_P} – Number of past pulses before the current frame, kept in memory

N_Q – Number of spectral lines that are explicitly coded

P_i – i^{th} pulse in a frame for $0 \leq i \leq N_{P_X}$. It is a pulse preceding the frame for $0 < i$

Q – Quantization operator

R – Relative differences

$R_{\hat{A}_P,\hat{A}_C}$ – Relative differences in magnitude responses of $\hat{A}_P$ and $\hat{A}_C$

$R_{\hat{A}_P,A_{M,j}}$ – Relative differences in magnitude responses of $\hat{A}_P$ and $A_{M,j}$

$R_{\hat{A}_C,A_{M,j}}$ – Relative differences in magnitude responses of $\hat{A}_c$ and $A_{M,j}$

$R_{\hat{A}_I,A_{M,j}}$ – Relative differences in magnitude responses of $\hat{A}_I$ and $A_{M,j}$

X, Y, Z – FD signal or vector

X_M – The original MDCT spectrum

$\bar{X}_M$ – The MDCT spectrum with perceptually flattened spectral envelope

$\bar{\bar{X}}_M$ – The MDCT spectrum with perceptually flattened spectral envelope and flattened temporal envelope

$\bar{X}_{F_0}$ – The predicted MDCT spectrum

X_{F_0} – The predicted and perceptually flattened MDCT spectrum

$\dot{X}_M$ – The MDCT spectrum that is to be scalar quantized

X_Q – The quantized MDCT spectrum

X_{Q_g} – The quantized MDCT spectrum scaled with the global gain g_Q

X_D – The dequantized MDCT spectrum with the added prediction

X_C – The dequantized MDCT spectrum after zero filling

$\bar{X}_C$ – The dequantized MDCT spectrum after restoring its temporal envelope

$\bar{\bar{X}}_C$ – The dequantized MDCT spectrum after restoring its spectral envelope

Z_C – Estimated magnitude spectrum used in the copy-up distance search

X_N – Random noise MDCT spectrum used in iBPC

$X_{S_{B_i}}$ – The iBPC source spectrum

$X_{G_{B_i}}$ – The scaled iBPC source spectrum

$X_{\hat{A}_P}$ – Magnitude response of $\hat{A}_P$

$X_{\hat{A}_C}$ – Magnitude response of $\hat{A}_C$

$X_{\hat{A}_I}$ – Magnitude response of $\hat{A}_I$

$X_{A_{M,j}}$ – Magnitude response of $A_{M,j}$

Δ – Offset/delta/shift

$\Delta_{\rho_{P_i,P_j}}$ – Offset for the maximum correlation between the i^{th} and the j^{th} pulse waveform

$\Delta_{P_{P_i}}$ – Pulse prediction offset

Δ_{P_i} – Pulse position deltas

Δ_C – Shift for the copy-up in iBPC

γ – Exponential weighting factor for LPC bandwidth expansion

ϵ – Error

ϵ_{P_i,P_j} – Error between the pitch and the pulse distance

ρ – Correlation

ρ_H – Normalized autocorrelation in the pitch search

ρ_{H_F} – Normalized autocorrelation at the high frequencies

$\hat{\rho}_{e_T}$ – Maximum autocorrelation of the temporal envelope

ρ_{P_i,P_j} – Correlation between the i^{th} and the j^{th} pulse waveform

τ – Threshold

τ_{B_i} – Threshold for the Adaptive Band Zeroing in the i^{th} sub-band

ϕ – Flag, single bit, 1/0, True/False

ϕ_H – Flag indicating if the high frequencies are tonal

$\phi_{N_{B_i}}$ – Flag indicating if the i^{th} sub-band is noise like

$\grave{\phi}_{N_{B_i}}$ – Flag indicating if the i^{th} sub-band was noise like in the previous frame

$\grave{\phi}_{T_C}$ – Flag indicating if there was a change of tonality in the previous frame affecting copy-up

$\mathcal{F}$ – Function

$\mathcal{F}_{F_0}$ – The pitch search

$\mathcal{F}_C$ – Function for obtaining the optimal iBPC copy-up distance

min( ) – Minimum

max( ) – Maximum

List of Abbreviations

ACELP	algebraic code-excited linear prediction
AES	the Audio Engineering Society
ALFE	adaptive low-frequency emphasis
AMR-WB+	Extended Adaptive Multi-Rate Wide Band
CD	compact disc
CDDA	Compact Disc Digital Audio
CELP	code-excited linear prediction
CELT	Constrained Energy Lapped Transform
CNG	comfort noise generator
DC	direct current
DCT	the discrete cosine transform
DFT	the discrete Fourier transform
DTX	discontinuous transmission
EBU	the European Broadcasting Union
ETSI	European Telecommunications Standards Institute
EVS	Enhanced Voice Services
FB	fullband
FD	frequency domain
FDNS	frequency domain noise shaping
FDP	frequency domain prediction
FFT	the fast Fourier transform
FIR	finite impulse response
HF	high frequencies
HPF	harmonic post-filter
HREP	High Resolution Envelope Processing
iBPC	integral Band-wise Parametric Coder
ICASSP	the International Conference on Acoustics, Speech, and Signal Processing
IEEE	the Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGF	Intelligent Gap Filling
ISO	the International Organization for Standardization
ITU	the International Telecommunication Union
IVA	Implicit Voice or Anything
JND	just noticeable difference
LC3	Low Complexity Communication Codec
LF	low frequencies
LP	linear prediction or linear predictor
LPC	linear prediction coefficients
LSF	line spectral frequencies
LTI	linear time-invariant
LTP	long-term predictor
MCLT	modulated complex lapped transform

MDCT	the modified discrete cosine transform
MDST	modified discrete sine transform
MP3	MPEG-1 Layer III
MPEG	the Moving Picture Experts Group
MTPC	Multimode Transform Predictive Coder
NMR	noise-to-masking ratio
PAC	Perceptual Audio Coder
PCM	pulse code modulation
PEAQ	Perceptual Evaluation of Audio Quality
PLC	packet-loss concealment
PNS	Perceptual Noise Substitution
POLQA	Perceptual Objective Listening Quality Analysis
QMF	quadrature mirror filterbank
RMS	root mean square
SBR	Spectral Band Replication
SNR	signal-to-noise ratio
SNS	spectral noise shaping
SQAM	Sound Quality Assessment Material
STFT	short-time Fourier transform
SWB	superwideband
TCX	transform-coded excitation
TD	time domain
TDAC	time domain aliasing cancelation
TDNS	time domain noise shaping
TNS	temporal noise shaping
USAC	Unified Speech and Audio Coding
VBR	variable bitrate
WB	wideband
WMOPS	weighted million operations per second
xHE-AAC	Extended High-Efficiency Advanced Audio Coding
ZFL	the Zero Filling Level
ZIR	zero input response

Bibliography

[1] R. N. E. Haughton, "Alexander Graham Bell 1847-1922 Inventor of the Bell System," *The Telecommunications Mosaic: An Introduction to the Information Age*, Canadian Telecommunications

[2] N. S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video," Prentice Hall, 1984.

[3] J. G. Proakis and D. K. Manolakis, "Digital Signal Processing (4th Edition): Principles. Algorithms and Applications," Prentice-Hall, 2006.

[4] L. R. Rabiner and R. W. Schafer, "Digital Processing of Speech Signals," Dorling Kindersley, 2009.

[5] C. Montgomery, "24/192 Music Downloads...and why they make no sense," retrieved on 27. April 2021., https://web.archive.org/web/20130707161555/http://xiph.org/~xiphmont/demo/neil-young.html, 2012.

[6] J. R. Stuart, "Coding Methods for High Resolution Recording Systems," *103rd AES Convention, New York*, 1997.

[7] J. Issing and N. Färber, "Conversational quality as a function of delay and interactivity," *20th SoftCOM, Split*, 2012.

[8] "Sustainability of Digital Formats: Planning for Library of Congress Collections," retrieved on 29. April 2021, https://www.loc.gov/preservation/digital/formats/fdd/fdd000011.shtml, 2008.

[9] H. Pihkola, M. Hongisto, O. Apilo, and M. Lasanen, "Evaluating the Energy Consumption of Mobile Data Transfer—From Technology Development to Consumer Behaviour and Life Cycle Thinking," *Sustainability*, vol. 10, 2018.

[10] M. Yan, C. A. Chan, A. F. Gygax, J. Yan, L. Campbell, A. Nirmalathas, and C. Leckie, "Modeling the Total Energy Consumption of Mobile Network Services and Applications," *Energies*, vol. 12, 2019.

[11] G. Kamiya, "Factcheck: What is the carbon footprint of streaming video on Netflix," retrieved on 29. April 2021., https://www.carbonbrief.org/factcheck-what-is-the-carbon-footprint-of-streaming-video-on-netflix, 2020.

[12] L. U. Marks, J. Clark, J. Livingston, D. Oleksijczuk, and L. Hilderbrand, "Streaming Media's Environmental Impact," *Media+ Environment*, vol. 2, 2020.

[13] "Connecting for Inclusion: Broadband Access for All," retrieved on 29. April 2021, https://www.worldbank.org/en/topic/digitaldevelopment/brief/connecting-for-inclusion-broadband-access-for-all

[14] I. Abdulqadir and S. Asongu, "The asymmetric effect of internet access on economic growth in sub-Saharan Africa: Insight from a dynamic panel threshold regression," African Governance and Development Institute, 2021.

[15] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–515, 2000.

[16] A. Spanias, "Speech coding: a tutorial review," *Proceedings of the IEEE*, vol. 82, no. 10, pp. 1541–1582, 1994.

[17] B. S. Atal, "The history of linear prediction," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 154–161, 2006.

[18] R. M. Gray, "A History of Realtime Digital Speech on Packet Networks: Part II of Linear Predictive Coding and the Internet Protocol," *Foundations and Trends in Signal Processing*, vol. 3, no. 4, pp. 203–303, Apr. 2010.

[19] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; Transcoding functions," 3GPP TS 26.190 v16, 2020.

[20] B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, "The adaptive multirate wideband speech codec (AMR-WB)," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 8, pp. 620–636, 2002.

[21] J. Mäkinen, B. Bessette, S. Bruhn, P. Ojala, R. Salami, and A. Taleb, "AMR-WB+: a new audio coding standard for 3rd generation mobile audio services," *IEEE ICASSP 2005, Philadelphia*, 2005.

[22] M. Neuendorf, P. Gournay, M. Multrus, J. Lecomte, B. Bessette, R. Geiger, S. Bayer, G. Fuchs, J. Hilpert, N. Rettelbach, R. Salami, G. Schuller, R. Lefebvre, and B. Grill, "Unified speech and audio coding scheme for high quality at low bitrates," *IEEE ICASSP 2009, Taipei*, 2009.

[23] K. Brandenburg and H. Popp, "An introduction to MPEG Layer-3," EBU technical review No.283, 2000.

[24] S. Quackenbush and R. Lefevbre, "Performance of MPEG Unified Speech and Audio Coding," *131st AES Convention, New York*, 2011.

[25] P. Combescure, J. Schnitzler, K. Fischer, R. Kircherr, C. Lamblin, A. Le Guyader, D. Massalaux, C. Quinquis, J. Stegmann, and P. Vary, "A 16, 24, 32 kbit/s wideband speech codec based on ATCELP," *IEEE ICASSP 99, Phoenix*, 1999.

[26] M. Jelinek, T. Vaillancourt, and J. Gibbs, "G.718: A new embedded speech and audio coding standard with high resilience to error-prone transmission channels," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 117–123, 2009.

[27] R. Lefebvre, R. Salami, C. Laflamme, and J.-P. Adoul, "High quality coding of wideband audio signals using transform coded excitation (TCX)," *IEEE ICASSP 94, Adelaide*, 1994.

[28] K. Makino and J. Matsumoto, "Hybrid audio coding for speech and audio below medium bit rate," *IEEE 2000 Digest of Technical Papers. 19th International Conference on Consumer Electronics, Los Angeles*, 2000.

[29] S. A. Ramprashad, "A multimode transform predictive coder (MTPC) for speech and audio," *1999 IEEE Workshop on Speech Coding, Porvoo*, 1999.

[30] I. Varga, S. Proust, and H. Taddei, "ITU-T G.729.1 scalable codec for new wideband services," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 131–137, 2009.

[31] M. Neuendorf, P. Gournay, M. Multrus, J. Lecomte, B. Bessette, R. Geiger, S. Bayer, G. Fuchs, J. Hilpert, N. Rettelbach, F. Nagel, J. Robilliard, R. Salami, G. Schuller, R. Lefebvre, and B. Grill, "A Novel Scheme for Low Bitrate Unified Speech and Audio Coding – MPEG RM0," *126th AES Convention, Munich*, 2009.

[32] M. Neuendorf, M. Multrus, N. Rettelbach, G. Fuchs, J. Robilliard, J. Lecomte, S. Wilde, S. Bayer, S. Disch, C. Helmrich, R. Lefebvre, P. Gournay, B. Bessette, J. Lapierre, K. Kjörling, H. Purnhagen, L. Villemoes, W. Oomen, E. Schuijers, K. Kikuiri, T. Chinen, T. Norimatsu, K. S. Chong, E. Oh, M. Kim, S. Quackenbush, and B. Grill, "The ISO/MPEG Unified Speech and Audio Coding Standard—Consistent High Quality for All Content Types and at All Bit Rates," *Journal of the Audio Engineering Society*, vol. 61, no. 12, pp. 956–977, 2013.

[33] M. Neuendorf, M. Multrus, N. Rettelbach, G. Fuchs, J. Robilliard, J. Lecomte, S. Wilde, S. Bayer, S. Disch, C. Helmrich, R. Lefebvre, P. Gournay, B. Bessette, J. Lapierre, K. Kjörling, H. Purnhagen, L. Villemoes, W. Oomen, E. Schuijers, K. Kikuiri, T. Chinen, T. Norimatsu, C. K. Seng, E. Oh, M. Kim, S. Quackenbush, and B. Grill, "MPEG Unified Speech and Audio Coding - The ISO/MPEG Standard for High-Efficiency Audio Coding of All Content Types," *132nd AES Convention, Budapest*, 2012.

[34] "Digital Radio Mondiale (DRM); System Specification," ETSI ES 201 980 v4.1.1, 2013.

[35] S. Bruhn, H. Pobloth, M. Schnell, B. Grill, J. Gibbs, L. Miao, K. Järvinen, L. Laaksonen, N. Harada, N. Naka, S. Ragot, S. Proust, T. Sanda, I. Varga, C. Greer, M. Jelínek, M. Xie, and P. Usai, "Standardization of the new 3GPP EVS codec," *IEEE ICASSP 2015, South Brisbane*, 2015.

[36] M. Dietz, M. Multrus, V. Eksler, V. Malenovsky, E. Norvell, H. Pobloth, L. Miao, Z. Wang, L. Laaksonen, A. Vasilache, Y. Kamamoto, K. Kikuiri, S. Ragot, J. Faure, H. Ehara, V. Rajendran, V. Atti, H. Sung, E. Oh, H. Yuan, and C. Zhu, "Overview of the EVS codec architecture," *IEEE ICASSP 2015, South Brisbane*, 2015.

[37] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Codec for Enhanced Voice Services (EVS); Detailed algorithmic description," 3GPP TS 26.445 v16, 2019.
[38] J.-M. Valin, K. Vos, and T. B. Terriberry, "Definition of the Opus Audio Codec," IETF RFC 6716, 2012.
[39] K. oe. Vos, K. V. Sørensen, S. S. Jensen, and J.-M. Valin, "Voice Coding With Opus," *Journal of the Audio Engineering Society*, Oct. 2013.
[40] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-Quality, Low-Delay Music Coding in the Opus Codec," *135th AES Convention, New York*, 2013.
[41] "Opus description at HydrogenAudio," retrieved on 03. May 2021, https://wiki.hydrogenaud.io/index.php?title=Opus
[42] A. Rämö and H. Toukomaa, "Subjective quality evaluation of the 3GPP EVS codec," *IEEE ICASSP 2015, South Brisbane*, 2015.
[43] J. G. Beerends, N. M. P. Neumann, E. L. van den Broek, A. Llagostera Casanovas, J. T. Menendez, C. Schmidmer, and J. Berger, "Subjective and Objective Assessment of Full Bandwidth Speech Quality," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 440–449, 2020.
[44] "WikiJournal of Medicine," https://en.wikiversity.org/wiki/WikiJournal_of_Medicine/Medical_gallery_of_Blausen_Medical_2014, 2014.
[45] B. Atal, "Predictive Coding of Speech at Low Bit Rates," *IEEE Transactions on Communications*, vol. 30, no. 4, pp. 600–614, Apr. 1982.
[46] B. Atal and J. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *IEEE ICASSP 82, Paris*, 1982.
[47] T. Bäckström, "Comparison of windowing in speech and audio coding," *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz*, 2013.
[48] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
[49] J. O. Smith, "Comb Filters in Physical Audio Signal Processing," retrieved on 7. May 2021, https://ccrma.stanford.edu/~jos/pasp/Comb_Filters.html, 2010.
[50] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *The Bell System Technical Journal*, vol. 49, no. 8, pp. 1973–1986, 1970.
[51] B. Atal and M. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 3, pp. 247–254, 1979.
[52] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," *IEEE ICASSP 85, Tampa*, 1985.
[53] J.-P. Adoul, P. Mabilleau, M. Delprat, and S. Morissette, "Fast CELP coding based on algebraic codes," *IEEE ICASSP 87, Dallas*, 1987.
[54] H. Fastl and E. Zwicker, "Psychoacoustics: Facts and Models," Springer, 2006.
[55] B. C. J. Moore, "An Introduction to the Psychology of Hearing," Emerald, 2012.
[56] H. Pulakka, V. Myllylä, A. Rämö, and P. Alku, "Speech quality evaluation of artificial bandwidth extension: Comparing subjective judgments and instrumental predictions," *INTERSPEECH 2015, Dresden*, 2015.
[57] "Compact disc hits 25th birthday," retrieved on 07. May 2021, http://news.bbc.co.uk/2/hi/technology/6950845.stm, 2007.
[58] "Methods for the subjective assesment of small impairments in audio systems," Recommendation ITU-R BS.1116-3, 2015.
[59] D. Purves, G. Augustine, D. Fitzpatrick, and et al., "Neuroscience. 2nd edition.," https://www.ncbi.nlm.nih.gov/books/NBK10946/, 2001.
[60] R. F. Lyon, "Human and Machine Hearing - Extracting Meaning from Sound," Cambridge University Press, 2018.

[61] R. Zelinski and P. Noll, "Adaptive transform coding of speech signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 299–309, 1977.
[62] K. Brandenburg, C. Faller, J. Herre, J. D. Johnston, and W. B. Kleijn, "Perceptual Coding of High-Quality Digital Audio," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 1905–1919, 2013.
[63] V. Britanak, "A survey of efficient MDCT implementations in MP3 audio coding standard: Retrospective and state-of-the-art," *Signal Processing*, vol. 91, no. 4, pp. 624–672, 2011.
[64] J. Princen and A. Bradley, "Analysis/Synthesis filter bank design based on time domain aliasing cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1153–1161, 1986.
[65] B. Edler, "Äquivalenz von Transformation und Teilbandzeriegung in der Quellencodierung," Universität Hannover, 1995.
[66] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, pp. 969–978, 1990.
[67] M. Temerinac and B. Edler, "LINC: A common theory of transform and subband coding," *IEEE Transactions on Communications*, vol. 41, no. 2, pp. 266–274, 1993.
[68] T. Bäckström, "Speech Coding: with Code-Excited Linear Prediction," Springer, 2017.
[69] K. Sayood, "Introduction to Data Compression," 3. ed., Morgan Kaufmann Publishers, 2006.
[70] B. R. U. Bhaskar, "Adaptive Prediction With Transform Domain Quantization For Low-rate Audio Coding," *1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz*, 1991.
[71] J.-H. Chen and D. Wang, "Transform predictive coding of wideband speech signals," *IEEE ICASSP 96, Atlanta*, 1996.
[72] T. Moriya and M. Honda, "Transform coding of speech with weighted vector quantization," *IEEE ICASSP 87, Dallas*, 1987.
[73] V. E. Sanchez and J.-P. Adoul, "Low-delay wideband speech coding using a new frequency domain approach," *IEEE ICASSP 93, Minneapolis*, 1993.
[74] K. Brandenburg, "OCF–A new coding algorithm for high quality sound signals," *IEEE ICASSP 87, Dallas*, 1987.
[75] "Model showing the distribution of frequencies along the basilar membrane of the cochlea," retrieved on 11. May 2021, https://www.britannica.com/science/inner-ear#/media/1/288499/18100
[76] "Information technology — Coding of audio-visual objects — Part 3: Audio," International Standard ISO/IEC 14496-3, 2001.
[77] "Information technology — Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s — Part 3: Audio," International Standard ISO/IEC 11172-3, 1993.
[78] M. Schnell, E. Ravelli, J. Büthe, M. Schlegel, A. Tomasek, A. Tschekalinskij, J. Svedberg, and M. Sehlstedt, "LC3 and LC3plus: The new audio transmission standards for wireless communication," *150th AES Convention*, 2021.
[79] "Low Complexity Communication Codec," Bluetooth Specification v1.0, 2020.
[80] "Digital Enhanced Cordless Telecommunications (DECT); Low Complexity Communication Codec plus (LC3plus)," ETSI TS 103 634 v1.1.1, 2019.
[81] B. Edler and G. Schuller, "Audio coding using a psychoacoustic pre-and post-filter," *IEEE ICASSP 2000, Istanbul*, 2000.
[82] A. Härmä, M. Vaalgamaa, and U. K. Laine, "A warped linear predictive stereo codec using temporal noise shaping," *Third IEEE Nordic Signal Processing Symposium*, 1998.
[83] B. Edler, "Codierung von Audiosignalen mit überlappender Transformation und adaptiven Fensterfunktionen," *Frequenz*, vol. 43, pp. 252–256, 1989.

[84] J. Herre and J. D. Johnston, “Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS),” *101st AES Convention, Los Angeles*, 1996.
[85] B. Edler, C. Faller, and G. Schuller, “Perceptual audio coding using a time-varying linear pre-and post-filter,” *109th AES Convention, Los Angeles*, 2000.
[86] V. Atti, V. Krishnan, D. Dewasurendra, V. Chebiyyam, S. Subasingha, D. J. Sinder, V. Rajendran, I. Varga, J. Gibbs, L. Miao, V. Grancharov, and H. Pobloth, “Super-wideband bandwidth extension for speech in the 3GPP EVS codec,” *IEEE ICASSP 2015, South Brisbane*, 2015.
[87] C. R. Helmrich, G. Marković, and B. Edler, “Improved low-delay MDCT-based coding of both stationary and transient audio signals,” *IEEE ICASSP 2014, Florence*, 2014.
[88] “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Audio codec processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec; Transcoding functions,” 3GPP TS 26.290 v16, 2020.
[89] J. Herre and D. Schultz, “Extending the MPEG-4 AAC Codec by Perceptual Noise Substitution,” *104th AES Convention, Amsterdam*, 1998.
[90] S. Disch, A. Niedermeier, C. R. Helmrich, C. Neukam, K. Schmidt, R. Geiger, J. Lecomte, F. Ghido, F. Nagel, and B. Edler, “Intelligent Gap Filling in Perceptual Transform Coding of Audio,” *141st AES Convention, Los Angeles*, 2016.
[91] “Universal Mobile Telecommunications System (UMTS); LTE; Codec for Enhanced Voice Services (EVS); Performance characterization (3GPP TR 26.952 version 12.4.0 Release 12),” ETSI TR 126 952 v12.4.0, 2016.
[92] R. L. Bleidt, D. Sen, A. Niedermeier, B. Czelhan, S. Füg, S. Disch, J. Herre, J. Hilpert, M. Neuendorf, H. Fuchs, and others, “Development of the mpeg-h tv audio system for atsc 3.0,” *IEEE Transactions on broadcasting*, vol. 63, no. 1, pp. 202–236, 2017.
[93] “Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 3: 3D audio,” International Standard ISO/IEC 23008-3, 2015.
[94] K. Kjörling, J. Rödén, M. Wolters, J. Riedmiller, A. Biswas, P. Ekstrand, A. Gröschel, P. Hedelin, T. Hirvonen, H. Hörich, J. Klejsa, J. Koppens, K. Krauss, H.-M. Lehtonen, K. Linzmeier, H. Muesch, H. Mundt, S. Norcross, J. Popp, H. Purnhagen, J. Samuelsson, M. Schug, L. Sehlström, R. Thesing, L. Villemoes, and M. Vinton, “AC-4 – The Next Generation Audio Codec,” *Journal of the Audio Engineering Society*, 2016.
[95] L. Villemoes, J. Klejsa, and P. Hedelin, “Speech coding with transform domain prediction,” *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz*, 2017.
[96] “Digital Audio Compression (AC-4) Standard,” ETSI TS 103 190 v1.1.1, 2014.
[97] B. Bessette, “Forward Time-Domain Aliasing Cancellation Using Linear-Predictive Filtering,” Patent Application Number: PCT/CA2011/000040, 2011.
[98] J. Lecomte, P. Gournay, R. Geiger, B. Bessette, and M. Neuendorf, “Efficient cross-fade windows for transitions between LPC-based and non-LPC based audio coding,” *126th AES Convention, Munich*, 2009.
[99] E. Ravelli, C. R. Helmrich, G. Fuchs, and M. Multrus, “Low-Complexity And Robust Coding Mode Decision In The EVS Coder,” *IEEE ICASSP 2015, South Brisbane*, 2015.
[100] “Information technology - MPEG audio technologies - Part 3: Unified speech and audio coding 2nd edition,” International Standard ISO/IEC 23003-3, 2020.
[101] G. Fuchs, “A robust speech/music discriminator for switched audio coding,” *IEEE 23rd European Signal Processing Conference (EUSIPCO), Nice*, 2015.
[102] V. Malenovsky, T. Vaillancourt, W. Zhe, K. Choo, and V. Atti, “Two-stage speech/music classifier with decision smoothing and sharpening in the EVS codec,” *IEEE ICASSP 2015, South Brisbane*, 2015.
[103] J.-M. Valin, “Opus 1.3 Released,” https://jmvalin.ca/opus/opus-1.3/, 2018.
[104] J. Makinen, A. Lakaniemi, and P. Ojala, “Low complex audio encoding for mobile, multimedia,” *IEEE 63rd Vehicular Technology Conference*, 2006.
[105] H. W. Strube, “Linear prediction on a warped frequency scale,” *The Journal of the Acoustical Society of America*, vol. 68, no. 4, pp. 1071–1076, 1980.

[106] D. Sinha, J. D. Johnston, S. Dorward, and S. Quackenbush, "The perceptual audio coder (PAC)," *The digital signal processing handbook*, pp. 42–1, 1998.
[107] J.-M. Valin, T. B. Terriberry, C. Montgomery, and G. Maxwell, "A high-quality speech and audio codec with less than 10-ms delay," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 1, pp. 58–67, 2009.
[108] G. Cohen, Y. Cohen, D. Hoffman, H. Krupnik, and A. Satt, "Digital audio signal coding," Patent Application Number: US09/034,516, 1998.
[109] B. Edler, C. Helmrich, M. Neuendorf, and B. Schubert, "Audio Encoder, Audio Decoder, Method For Encoding An Audio Signal And Method For Decoding An Encoded Audio Signal," Patent Application Number: PCT/EP2016/054831, 2016.
[110] J. Ojanperä, M. Väänänen, and L. Yin, "Long term predictor for transform domain perceptual audio coding," *107th AES Convention, San Jose*, 1999.
[111] J.-H. Chen and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 59–71, 1995.
[112] J.-H. Chen, "Pitch Prefiltering and Postfiltering Techniques for Improving the Audio Quality of the IETF Opus Mode 1 Codec (the Original CELT Codec)," https://studylib.net/doc/7730106/pitch-prefiltering-and-postfiltering-techniques-for-impro
[113] J. Song, C.-H. Lee, H.-O. Oh, and H.-G. Kang, "Harmonic Enhancement in Low Bitrate Audio Coding Using an Efficient Long-Term Predictor," *EURASIP Journal on Advances in Signal Processing Volume 2010*, 2010.
[114] E. Ravelli, C. Helmrich, G. Marković, M. Neusinger, S. Disch, M. Jander, and M. Dietz, "Apparatus and Method for Processing an Audio Signal Using a Harmonic Post-Filter," Patent Application Number: PCT/EP2015/066998, 2015.
[115] S. Disch, C. Helmrich, E. Ravelli, and M. Neuendorf, "3DA Phase 2 Core Experiment on frequency-domain prediction and time- domain post-filtering," ISO/IEC JTC1/SC29/WG11, MPEG2015/M36534, Jun. 2015.
[116] P. Prokein, "A vocoding-based precoding for transform-based audio coders," M.Sc. Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2008.
[117] "4G/5G Network Experience Evaluation Guideline," GSMA, https://www.gsma.com/futurenetworks/resources/4g-5g-network-experience-evaluation-guideline/, 2020.
[118] T. Braud, T. Kämäräinen, M. Siekkinen, and P. Hui, "Multi-Carrier Measurement Study of Mobile Network Latency: The Tale of Hong Kong and Helsinki," *15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2019.
[119] M. Donegan, "5G is 450% faster than other mobile services, finds Ookla," retrieved on 20. May 2021, https://5g.co.uk/news/5g-is-450-faster/5102/, 2019.
[120] "Codec for Enhanced Voice Services (EVS); ANSI C code (floating-point)," 3GPP TS 26.443 v13.1, https://www.3gpp.org/DynaReport/26443.htm
[121] G. Fuchs, M. Multrus, M. Neuendorf, and R. Geiger, "MDCT-based coder for highly adaptive speech and audio coding," *IEEE 17th European Signal Processing Conference (EUSIPCO), Glasgow*, 2009.
[122] P. Kabal, "Ill-Conditioning and Bandwidth Expansion in Linear Prediction of Speech," Department of Electrical & Computer Engineering McGill University, 2003.
[123] T. Islam, "Interpolation of Linear Prediction Coefficients for Speech Coding," M.Sc. Thesis, McGill University, 2000.
[124] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations," *4th European Conference on Speech Communication and Technology*, 1995.
[125] S. Vittappan, "Comparison of spectral envelope quantization and coding methods in MDCT based codec at low bitrate with low complexity constraints," M.Sc. Thesis, Technische Universität Ilmenau, 2019.

[126] A. Hysneli, “Coding of Psychoacoustic Pre-/Post-Filter Coefficients,” M.Sc. Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2017.
[127] E. Ravelli, M. Schnell, C. Benndorf, M. Lutzky, M. Dietz, and S. Korse, “Apparatus And Method For Encoding And Decoding An Audio Signal Using Downsampling Or Interpolation Of Scale Parameters,” Patent Application Number: PCT/EP2018/080137, 2018.
[128] C. Helmrich, J. Lecomte, G. Marković, M. Schnell, B. Edler, and S. Reuschl, “Apparatus And Method For Encoding Or Decoding An Audio Signal Using A Transient-Location Dependent Overlap,” Patent Application Number: PCT/EP2014/053293, 2014.
[129] “Digital Enhanced Cordless Telecommunications (DECT); Study of Super Wideband Codec in DECT for narrowband, wideband and super-wideband audio communication including options of low delay audio connections (≤ 10 ms framing),” ETSI TR 103 590 v1.1.1, 2018.
[130] H. Malvar, “A modulated complex lapped transform and its applications to audio processing,” *IEEE ICASSP 99, Phoenix*, 1999.
[131] L. R. Rabiner and R. W. Schafer, “Introduction to digital speech processing,” *Foundations and Trends in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.
[132] J. Herre, R. Geiger, S. Bayer, G. Fuchs, U. Krämer, N. Rettelbach, and B. Grill, “Audio Encoder For Encoding An Audio Signal Having An Impulse-Like Portion And Stationary Portion, Encoding Methods, Decoder, Decoding Method; And Encoded Audio Signal,” Patent Application Number: PCT/EP2008/004496, 2008.
[133] O. Niemeyer and B. Edler, “Detection and Extraction of Transients for Audio Coding,” *120th AES Convention, Paris*, 2006.
[134] F. Ghido, S. Disch, J. Herre, F. Reutelhuber, and A. Adami, “Coding Of Fine Granular Audio Signals Using High Resolution Envelope Processing (HREP),” *IEEE ICASSP 2017, New Orleans*, 2017.
[135] A. Adami, A. Herzog, S. Disch, and J. Herre, “Transient-to-noise ratio restoration of coded applause-like signals,” *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz*, 2017.
[136] R. Füg, A. Niedermeier, J. Driedger, S. Disch, and M. Müller, “Harmonic-percussive-residual sound separation using the structure tensor on spectrograms,” *IEEE ICASSP 2016, Shanghai*, 2016.
[137] D. Griffin and J. Lae, “Signal estimation from modified short-time Fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, pp. 236–243, 1984.
[138] “Mathematica,” Wolfram Research, https://www.wolfram.com/mathematica/
[139] W. C. Chu, “Speech Coding Algorithms: Foundation and Evolution of Standardized Coders,” 1st ed.., John Wiley & Sons, 2003.
[140] J. Ojanperä, “Method for improving the coding efficiency of an audio signal,” Patent Application Number: PCT/FI2000/000619, 2000.
[141] L. Yin, M. Suonio, and M. Väänänen, “Proposal for a Core Experiment on Long Term Prediction in MPEG4 Audio,” ISO/IEC JTC1/SC29/WG11, 1830, Apr. 1997.
[142] N. Guo and B. Edler, “Frequency Domain Long-Term Prediction for Low Delay General Audio Coding,” *IEEE Signal Processing Letters*, 2021.
[143] N. Guo and B. Edler, “Encoder, Decoder, Encoding Method And Decoding Method For Frequency Domain Long-Term Prediction Of Tonal Signals For Audio Coding,” Patent Application Number: PCT/EP2019/082802, 2019.
[144] G. Marković, E. Ravelli, M. Dietz, and B. Grill, “Signal Filtering,” Patent Application Number: PCT/EP2018/080837, 2018.
[145] G. Fuchs, C. R. Helmrich, G. Marković, M. Neusinger, E. Ravelli, and T. Moriya, “Low delay LPC and MDCT-based audio coding in the EVS codec,” *IEEE ICASSP 2015, South Brisbane*, 2015.
[146] M. Dietz, L. Liljeryd, K. Kjorling, and O. Kunz, “Spectral Band Replication, a novel approach in audio coding,” *112nd AES Convention, Munich*, 2002.

[147] S. Disch, M. Gayer, C. Helmrich, G. Marković, and M. Luis Valero, "Noise Filling Concept," Patent Application Number: PCT/EP2014/051630, 2014.
[148] N. Rettelbach, B. Grill, G. Fuchs, S. Geyrsberger, M. Multrus, H. Popp, J. Herre, S. Wabnik, G. Schuller, and J. Hirschfeld, "Audio Encoder, Audio Decoder, Methods For Encoding And Decoding An Audio Signal, Audio Stream And Computer Program," Patent Application Number: PCT/EP2009/004602, 2009.
[149] F. Nagel, S. Disch, and S. Wilde, "A continuous modulated single sideband bandwidth extension," *IEEE ICASSP 2010, Dallas*, 2010.
[150] C. Neukam, F. Nagel, G. Schuller, and M. Schnabel, "A MDCT based harmonic spectral bandwidth extension method," *IEEE ICASSP 2013, Vancouver*, 2013.
[151] S. Disch, R. Geiger, C. Helmrich, F. Nagel, C. Neukam, K. Schmidt, and M. Fischer, "Apparatus, Method And Computer Program For Decoding An Encoded Audio Signal," Patent Application Number: PCT/EP2014/065118, 2014.
[152] S. Disch, F. Nagel, R. Geiger, B. N. Thoshkahna, K. Schmidt, S. Bayer, C. Neukam, B. Edler, and C. Helmrich, "Apparatus And Method For Encoding Or Decoding An Audio Signal With Intelligent Gap Filling In The Spectral Domain," Patent Application Number: PCT/EP2014/065109, 2014.
[153] S. Disch, F. Nagel, R. Geiger, B. N. Thoshkahna, K. Schmidt, S. Bayer, C. Neukam, B. Edler, and C. Helmrich, "Apparatus And Method For Encoding And Decoding An Encoded Audio Signal Using Temporal Noise/Patch Shaping," Patent Application Number: PCT/EP2014/065123, 2014.
[154] S. Disch, S. van de Par, A. Niedermeier, E. Burdiel Pérez, A. Berasategui Ceberio, and B. Edler, "Improved Psychoacoustic Model for Efficient Perceptual Audio Codecs," *145th AES Convention, New York*, 2018.
[155] C. R. Helmrich, A. Niedermeier, S. Disch, and F. Ghido, "Spectral envelope reconstruction via IGF for audio transform coding," *IEEE ICASSP 2015, South Brisbane*, 2015.
[156] C. Neukam, S. Disch, F. Nagel, A. Niedermeier, K. Schmidt, and B. N. Thoshkahna, "Apparatus And Method For Decoding And Encoding An Audio Signal Using Adaptive Spectral Tile Selection," Patent Application Number: PCT/EP2014/065116, 2013.
[157] A. Niedermeier, C. Ertel, R. Geiger, F. Ghido, and C. Helmrich, "Apparatus And Method For Decoding Or Encoding An Audio Signal Using Energy Information Values For A Reconstruction Band," Patent Application Number: PCT/EP2014/065110, 2013.
[158] M. Dietz, G. Fuchs, C. Helmrich, and G. Marković, "Low-Complexity Tonality-Adaptive Audio Signal Quantization," Patent Application Number: PCT/EP2014/051624, 2014.
[159] S. Disch, B. Schubert, R. Geiger, and M. Dietz, "Apparatus And Method For Audio Encoding And Decoding Employing Sinusoidal Substitution," Patent Application Number: PCT/EP2012/076746, 2012.
[160] S. Disch, B. Schubert, R. Geiger, B. Edler, and M. Dietz, "Apparatus And Method For Efficient Synthesis Of Sinusoids And Sweeps By Employing Spectral Patterns," Patent Application Number: PCT/EP2013/069592, 2013.
[161] M. Oger, S. Ragot, and M. Antonini, "Model-based deadzone optimization for stack-run audio coding with uniform scalar quantization," *IEEE ICASSP 2008, Las Vegas*, 2008.
[162] R. Frazier, S. Samsam, L. Braida, and A. Oppenheim, "Enhancement of speech by adaptive filtering," *IEEE ICASSP 76, Philadelphia*, 1976.
[163] R. H. Frazier, "An adaptive filtering approach toward speech enhancement.," Ph.D. Thesis, Massachusetts Institute of Technology, 1975.
[164] D. Malah and R. Cox, "A generalized comb filtering technique for speech enhancement," *IEEE ICASSP 82, Paris*, 1982.
[165] T. Morii, "Post Filter And Filtering Method," Patent Application Number: PCT/JP2007/074044, 2007.
[166] G. Marković, C. Helmrich, E. Ravelli, M. Jander, and S. Döhla, "Harmonicity-Dependent Controlling Of A Harmonic Filter Tool," Patent Application Number: PCT/EP2015/067160, 2015.

[167] "Perceptual objective listening quality prediction," Recommendation ITU-T P.863, 2018.
[168] "Method for objective measurements of perceived audio quality," Recommendation ITU-R BS.1387-1, 2001.
[169] "Method for the subjective assessment of intermediate quality level of audio systems," Recommendation ITU-R BS.1534-3, 2015.
[170] M. Erne, "Perceptual audio coders "what to listen for," *Journal of the Audio Engineering Society*, Nov. 2001.
[171] Stax Ltd from Wikipedia, https://en.wikipedia.org/wiki/Stax_Ltd
[172] "Extended HE-AAC - Bridging the gap between speech and audio coding," Fraunhofer IIS, 2019.
[173] Opus downloads, https://opus-codec.org/downloads/
[174] LC3Plus Software from ETSI, https://www.etsi.org/deliver/etsi_ts/103600_103699/103634/01.02.01_60/ts_103634v010201p0.zip
[175] J. Ojanpera, "Method, apparatus and computer program to provide predictor adaptation for advanced audio coding (AAC) system," Patent Application Number: PCT/IB2005/002341, 2005.
[176] "Winamp 5.666 Released (Build 3516)," http://forums.winamp.com/showthread.php?t=373755, 2013.
[177] R. C. Helmrich, "ecodis extended high-efficiency and low-complexity encoder - an open-source ISO/IEC 23003-3 (USAC, Extended HE-AAC) encoder," https://gitlab.com/ecodis/exhale
[178] M. Lutzky, M. Gayer, G. Schuller, U. Krämer, and S. Wabnik, "A guideline to audio codec delay," *116th AES Convention, Berlin*, 2004.
[179] N. Schinkel-Bielefeld, N. Lotze, and F. Nagel, "Audio quality evaluation by experienced and inexperienced listeners," *21st ICA Meetings on Acoustics, Montreal*, 2013.
[180] F. Nagel, T. Sporer, and P. Sedlmeier, "Toward a statistically well-grounded evaluation of listening tests - Avoiding pitfalls, misuse, and misconceptions," *128th AES Convention, London*, 2010.
[181] P. Počta and J. G. Beerends, "Subjective and Objective Assessment of Perceived Audio Quality of Current Digital Audio Broadcasting Systems and Web-Casting Applications," *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 407–415, 2015.
[182] "ITU-T Software Tool Library 2009 User's Manual," Recommendation G.191 STL-2009, 2009.
[183] S. Korse, K. Gupta, and G. Fuchs, "Enhancement of coded speech using a mask-based post-filter," *IEEE ICASSP 2020, Barcelona*, 2020.
[184] M. Li, K. Ma, J. You, D. Zhang, and W. Zuo, "Efficient and effective context-based convolutional entropy modeling for image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 5900–5911, 2020.
[185] P. P. Zarazaga, "Frequency Domain Methods for Coding the Linear Predictive Residual of Speech Signals," M.Sc. Thesis, Aalto University School of Electrical Engineering, 2017.
[186] T. Bäckström and C. R. Helmrich, "Decorrelated innovative codebooks for ACELP using factorization of autocorrelation matrix," *15th Annual Conference of the International Speech Communication Association*, 2014.
[187] E. Allamanche, R. Geiger, J. Herre, and T. Sporer, "MPEG-4 low delay audio coding based on the AAC codec," *106th AES Convention, Munich*, 1999.
[188] C.-M. Liu, H.-W. Hsu, and W.-C. Lee, "Compression artifacts in perceptual audio coding," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 4, pp. 681–695, 2008.
[189] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of quantization," *IEEE Transactions on instrumentation and measurement*, vol. 45, no. 2, pp. 353–361, 1996.

[190] J. Herre and S. Dick, “Psychoacoustic models for perceptual audio coding - A tutorial review,” *Applied Sciences*, vol. 9, no. 14, p. 2854, 2019.
[191] M. Torcoli and S. Dick, “Comparing the effect of audio coding artifacts on objective quality measures and on subjective ratings,” *144th AES Convention, Milan*, 2018.
[192] J. Herre, “Temporal noise shaping, qualtization and coding methods in perceptual audio coding: A tutorial introduction,” *AES 17th International Conference on High Quality Audio Coding, Florence*, 1999.
[193] G. Marković, “Analysis Of Methods For Objective Evaluation Of Quality Of Audio Signals And Application In Implementation Of An Encoder On A Class Of Digital Signal Processors,” M.Sc. Thesis, Faculty of Technical Sciences at University of Novi Sad, 2006.
[194] T. Thiede, “Perceptual Audio Quality Assessment using a Non-Linear Filter Bank,” Ph.D. Thesis, Technical University of Berlin Berlin, Germany, 1999.
[195] F. Baumgarte, “Ein psychophysiologisches Gehörmodell zur Nachbildung von Wahrnehmungsschwellen für die Audiocodierung,” Ph.D. Thesis, Universität Hannover, 2000.
[196] M. Müller and F. Zalkow, “FMP Notebooks: Educational Material for Teaching and Learning Fundamentals of Music Processing,” *International Conference on Music Information Retrieval (ISMIR), Delft*, 2019.
[197] “Bitstreams cross-check on improved stereo coding in USAC,” ISO/IEC JTC1/SC29/WG11, MPEG2010/m17957, Jul. 2010.
[198] “Workplan on Speech and Audio Material Selection,” ISO/IEC JTC1/SC29/WG11, MPEG2008/N9639, Jan. 2008.
[199] “Framework for Exploration of Speech and Audio Coding,” ISO/IEC JTC1/SC29/WG11, MPEG2007/N8855, Jan. 2007.
[200] “MPEG Audio Content Assets,” ISO/IEC JTC1/SC29/WG11, MPEG2007/N8856, Jan. 2007.
[201] “SQAM CD - Sound Quality Assessment Material recordings for subjective tests,” EBU, https://tech.ebu.ch/publications/sqamcd, 2008.
[202] “Evaluations of Multichannel Audio Codecs,” EBU technical report 3324, http://tech.ebu.ch/docs/tech/tech3324.pdf, 2007.
[203] “Listening tests at HydrogenAudio,” retrieved on 12. November 2021, https://hydrogenaud.io/index.php?board=40.0
[204] “Hydrogenaudio Listening Tests,” retrieved on 12. November 2021, https://wiki.hydrogenaud.io/index.php?title=Hydrogenaudio_Listening_Tests
[205] “Codec listening test,” retrieved on 12. November 2021, https://en.wikipedia.org/wiki/Codec_listening_test