Low Delay Transform for High Quality Low Delay Audio Coding

David Virette

N° d'ordre : 2012REN1E014 **ANNÉE 2012**

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du signal et télécommunications

Ecole doctorale MATISSE

présentée par

David Virette

préparée à l'unité de recherche (6074 IRISA UMR)
(Institut de Recherche en Informatique et Systèmes Aléatoires)

Étude de transformées temps-fréquence pour le codage audio faible retard en haute qualité (Low Delay Transform for High Quality Low Delay Audio Coding)

Thèse soutenue à Rennes le 10 décembre 2012

devant le jury composé de :

Bernd EDLER
Professor Dr.-Ing.
International Audio Laboratories Erlangen / *rapporteur*

Dominique MASSALOUX
Directrice scientifique adjointe
Télécom Bretagne / *rapporteur*

Sylvain MARCHAND
Professeur des universités
Université Bretagne Occidentale / *examinateur*

Laurent GIRIN
Professeur des universités
Grenoble INP / *examinateur*

Frédéric BIMBOT
Directeur de Recherche CNRS
IRISA / *examinateur*

Hervé TADDEI
Examinateur OEB / *examinateur*

Pascal SCALART
Professeur des universités
ENSSAT / *directeur de thèse*

Pierrick PHILIPPE
Ingénieur R&D Orange Labs / *co-directeur de thèse*

Acknowledgments

I would like to express my gratitude to my thesis supervisors Dr. Pierrick Philippe and Prof. Pascal Scalart, for their patience, guidance, enthusiastic encouragement and useful critiques of this research work. I would especially like to thank Pierrick for his unshakeable support and confidence. I have particularly appreciated the long technical and non-technical discussions that we had during the course of this research. With him, performing research work can sometimes lead to have the giggles…
Many thanks to Balázs Kövesi, who inspired me part of this work and gave me a different point of view. I would also like to thank Duncan Menzies for his huge help in extensively testing the seamless reconstruction during his M.Eng degree. I am also extremely grateful to Jean-Pierre Petit for encouraging me to pursue this PhD work.
I would like to thank the reviewers of this thesis, Prof. Bernd Edler and Dr Dominique Massaloux, for their valuable and constructive comments on the manuscript of the thesis. I am also grateful to the members of the Doctorate committee, Prof. Sylvain Marchand, Prof. Laurent Girin, Dr. Frédéric Bimbot, Dr. Hervé Taddéi, for devoting their time in reading the manuscript and showing interest to this work.
During these years I have had the pleasure to work with great colleagues at France Telecom: Alex, Greg, Rozenn, Jérôme, Marc, Claude, Catherine, Arnault, Martine, Adrien, Ludo, Charles, Florent, Manuel, Alain, Bruno, Stéphane and Stéphane. I have also appreciated the discussions on the "front montant" and "front descendant" of the windows with Hervé and Anisse.
Last but not least, I wish to express my warmest thanks to Katell for encouraging me and to Elliott, Zélie and Timothé for bringing joy during all these years.

Contents

Abstract

In recent years there has been a phenomenal increase in the number of products and applications which make use of audio coding formats. Among the most successful audio coding schemes, the MPEG-1 Layer III (mp3), the MPEG-2 Advanced Audio Coding (AAC) or its evolution MPEG-4 High Efficiency-Advanced Audio Coding (HE-AAC) can be cited.

More recently, perceptual audio coding has been adapted to achieve coding at low-delay such to become suitable for conversational applications. Traditionally, the use of filter bank such as the Modified Discrete Cosine Transform (MDCT) is a central component of perceptual audio coding and its adaptation to low delay audio coding has become an important research topic. Low delay transforms have been developed in order to retain the performance of standard audio coding while reducing dramatically the associated algorithmic delay.

This work presents some elements allowing to better accommodate the delay reduction constraint. Among the contributions, a low delay block switching tool which allows the direct transition between long transform and short transform without the insertion of transition window. The same principle has been extended to define new perfect reconstruction conditions for the MDCT with relaxed constraints compared to the original definition. As a consequence, a seamless reconstruction method has been derived to increase the flexibility of transform coding schemes with the possibility to select a transform for a frame independently from its neighbouring frames. Finally, based on this new approach, a new low delay window design procedure has been derived to obtain an analytic definition for a new family of transforms, permitting high quality with a substantial coding delay reduction.

The performance of the proposed transforms has been thoroughly evaluated, an evaluation framework involving an objective measurement of the optimal transform sequence is proposed. It confirms the relevance of the proposed transforms used for audio coding. In addition, the new approaches have been successfully applied to the recent standardisation work items, such as the low delay audio coding developed at MPEG (LD-AAC and ELD-AAC) and they have been evaluated with numerous subjective testing, showing a significant improvement of the quality for transient signals. The

new low delay window design has been adopted in G.718, a scalable speech and audio codec standardized in ITU-T and has demonstrated its benefit in terms of delay reduction while maintaining the audio quality of a traditional MDCT.

Keywords: Low delay audio coding – Transform coding – Block switching – MDCT – Seamless reconstruction – Low delay window design

List of Figures

List of Tables

Frequently Used Terms, Abbreviation and Notations

AAC	Advanced Audio Coding
AC	Alias Component
BCC	Binaural Cue Coding
CELP	Code Excited Linear Prediction
CCR	Comparison Category Rating
DFT	Discrete Fourier Transform
ELD-AAC	Enhanced Low Delay - Advanced Audio Coding
ERB	Equivalent Rectangular Bandwidth
FFT	Fast Fourier Transform
ICC	Inter-Channel Coherence
ICLD	Inter-Channel Level Difference
ICPD	Inter-Channel Phase Difference
ICTD	Inter-Channel Time Difference
ITU	International Telecommunication Union, –T (Telecommunication Sector) and –R (Radiocommunication Sector)
kbit/s	Kilo-bit per second
KLT	Karhunen-Loève Transform
LD-AAC	Low Delay - Advanced Audio Coding
LOT	Lapped Orthogonal Transform
LPC	Linear Predictive Coding
MDCT	Modified Discrete Cosine Transform
MLT	Modulated Lapped Transform
MUSHRA	MUlti Stimuli with Hidden Reference and Anchors
MPEG	Moving Picture Experts Group
NMR	Noise to Mask Ratio
PAT	Perceptual Audio Transposition
PCA	Principal Component Analysis
PQMF	Pseudo-Quadrature Mirror Filter
PR	Perfect Reconstruction
PS	Parametric Stereo
SBR	Spectral Band Replication
SMR	Signal to Mask Ratio

SNR	Signal to Noise Ratio
TDAC	Time Domain Aliasing Cancellation
USAC	Unified Speech and Audio Coding

Chapter 1

Introduction

In recent years there has been a phenomenal increase in the number of products and applications making use of audio coding formats. Among the most successful audio coding schemes, MPEG-1 Layer III [ISO 92][Brandenburg 99], the MPEG-2 Advanced Audio Coding (AAC) [ISO 09][Grill 99] or its evolution MPEG-4 High Efficiency-Advanced Audio Coding (HE-AAC and HE-AACv2) [ISO 09][Dietz 02] can be listed. These codecs are based on the perceptual audio coding paradigm. Usually, the perceptual audio codecs find their applications in broadcasting services, streaming or storage. Indeed, historically few delay constraints were imposed to those audio coding standards and they are consequently not suitable for conversational applications. As opposed to the broadcast applications, communication services are usually based on speech coding format such as Algebraic Code Excited Linear Prediction (ACELP). The ACELP coding scheme [Schroeder 85] [Adoul 87] is used in the most widely deployed communication codecs such as AMR [3GPP 99], 3GPP AMR-WB [3GPP 02] or ITU-T G.729 [ITU-T G.729 96]. This coding algorithm is based on the source-filter model of the speech production and it provides good quality for speech signals with a limited delay which makes it compatible with conversational applications.
Perceptual audio coding has been adapted to achieve low delay audio coding and to become suitable for conversational applications. The wideband codec ITU-T G.722.1 [ITU-T G.722.1 99] and its superwideband extension ITU-T G.722.1 annex C [ITU-T G.722.1C 05], the MPEG-4 AAC-Low Delay [Allamanche 99] or the scalable extension of speech codec such as ITU-T G.729.1 [ITU-T G.729.1 06] [Ragot 07] can be cited. These communication codecs target not only toll quality for speech signals but also address any audio contents. Consequently the speech production paradigm can not be solely used and transform perceptual coding has been adapted in this application domain.

However, as discussed in this thesis, due to the delay constraint, some of the tools are usually not used in low delay transform codecs leading to quality limitation compared to larger delay perceptual audio codecs. Moreover, as most of the delay comes from the transform itself, care must be particularly taken for the window design in order to reduce the algorithmic delay. Some advanced filter bank design has been proposed to reduce the delay associated with the transform [Schuller 00], but they have the drawback to extend the window or prototype size in order to achieve this delay reduction. A longer prototype leads to a longer temporal noise spreading which increases the risk of perceived noise.

1.1 Thesis motivation

The purpose of this work is to develop coding tools and transform coding schemes that are adapted to low delay audio coding. The quality limitation introduced in perceptual audio coding by the low delay constraint is mainly due to the lack of flexibility in the time-frequency resolution. Indeed, the transform size is fixed with a frame size which is usually between 10 and 20 ms. The overall delay of the transform coding operation lies between 20 and 40 ms. For most of the low bit rate conversational codecs, the 20 ms frame size is used with a reduced sampling frequency (8, 16, 24 or 32 kHz). From this frame size constraint, one can deduce the maximum window size which can be selected. A smaller delay can be obtained, given the transform size, for larger sampling rate.

It is known from the successful broadcast perceptual audio codec that the ability to change the time-frequency resolution provides an improved quality for non-stationary sounds and more specifically for transient signals. This thesis presents a time-frequency resolution adaptation scheme, called low delay block switching, which is fully compatible with nowadays transform coding and offering this additional transform flexibility for low delay audio codecs.

A second goal of the thesis was to develop transforms for embedded/scalable speech and audio codecs. In this particular context, the core layer is based on ACELP coding with a fixed 20 ms frame size and the transform coding is then used to encode the residual signal. For this specific application, the low delay filter banks introduced in [Schuller 00] are not always efficient as the temporal support is longer than the MDCT and leads to a longer temporal spreading of the quantization noise. Note that the underlying framework is flexible and would allow also shorter temporal support, even if it has not been used in practice. This work presents a low delay window design solution for MDCT. It is based on the relaxation of the perfect reconstruction conditions which have been introduced in [Princen 86]. A general flexible perfect reconstruction system based on the

MDCT is presented and the derivation of new window prototypes is explained.

1.2 Standardization context

This work was closely related to the development of low delay audio coding standards in ISO/MPEG and ITU-T. The technologies which have been proposed during this work have been developed keeping in mind the constraints which were imposed by the development of MPEG-4 AAC-Enhanced Low Delay (AAC-ELD) and ITU-T Embedded Variable Bit rate (EV-VBR) that led to the G.718 standard and its Superwideband extension G.718 Annex B. The close connection to standardization activity has led to the development of competitive MPEG-4 AAC-LD and AAC-ELD encoder implementations in order to get the possibility to demonstrate the new tools performance and to adapt the encoder accordingly. This development has also required a large amount of work to perform the fine tuning to achieve a state-of-the-art quality which was used to assess the benefit of the proposed technologies.

1.3 Contributions and thesis overview

Chapter 2 introduces the basis of perceptual audio coding which is needed to understand the place and role of the coding transform.
Chapter 3 presents a detailed description of the Modified Discrete Cosine Transform (MDCT) which is, by far, the most common transform in perceptual audio codecs. It is used as central component for this work. The MDCT and its perfect reconstruction conditions are first defined. Then the associated tools, such as block switching, used for quality improvement with transient signals are presented. It should be noted that most of those tools are exclusively used in broadcast applications in state-of-the-art codecs due to the additional delay required to ensure a perfect behaviour.
The contributions of this work are then presented in Chapter 4. The first tool is the low delay block switching tool for the rapid adaptation of the time-frequency resolution between long transform and short transform without the need of transition windows. This low delay block switching tool has been adapted to the MDCT and low delay filter bank which are used in the low delay MPEG audio codec. The impact of the quantization noise in that context is also discussed.
In section 4.3, the seamless reconstruction method is introduced. This new method allows to develop audio coding schemes without any constraint on window selection and window transition. An audio coding experiment is described to demonstrate the benefit of the method in the context of audio coding with an extended transform windows set. This technique has been adapted to the design of new window prototype for MDCT. In the last part

of the Chapter, a new low delay window design method is introduced. While ensuring the perfect reconstruction, this new window definition provides at the same time a better objective performance compared to traditional low delay window (Low Overlap window) used by MDCT.
Chapter 5 provides the results of the extensive subjective listening assessment which were performed in the context of the standardization processes in order to assess the benefit of the proposed method in real world low delay audio codecs.
Finally Chapter 6 gives the conclusion with an overview of the thesis document, the contributions of this work and particularly the achievements and the perspective offered by this thesis.

Chapter 2

Filter banks and Transforms in audio coding

In this chapter, the usage of filter banks and transforms for audio coding is introduced and especially in perceptual audio coding where they are used in many state of the art coding schemes and standards. First, a short introduction of filter banks and transforms is given. Then, an overview of the principles of perceptual audio coding is presented. As the thesis focuses on conversational applications, care must be taken to obtain good performance for speech content with low delay and low bit rates. Hence, a review of the recently developed parametric tools, which are nowadays widely used in low delay and low bit rate audio coding standards, is provided.

2.1 Filter banks and transforms – an introduction

2.1.1 Characteristics of audio signals

The main characteristic of the audio signal is its wide diversity. All the acoustic signals with a frequency range lying between 20 Hz and 20 kHz can be assimilated to audio signals. However, clear differences between transient and harmonic contents can be perceived. This distinction leads to the definition of two important aspects of audio signals as far as coding is concerned: temporal and frequency aspects.

The main temporal feature of audio signals that has to be taken into account to design a time-frequency transform is its non stationary property. Sounds can be assumed to be pseudo-stationary processes, which means that for short segments (few milliseconds), the short-term stationarity assumption can be used. As such the audio signal can be framed into short segments, each processed independently.

An audio signal excerpt extracted from a short segment of *Pitchpipe* sound is given in Figure 1. It shows the stationarity and periodicity of this highly harmonic signal over a segment of 25 milliseconds (1200 samples at 48 kHz). This gives a perfect example of the stationarity of the sound over short periods. Moreover, this kind of signal has the advantage of being particularly adapted for frequency analysis as the frequency domain signal is mostly represented with a limited number of frequency coefficients. The periodic aspect of this signal facilitates the detection and exploitation of the time redundant components.

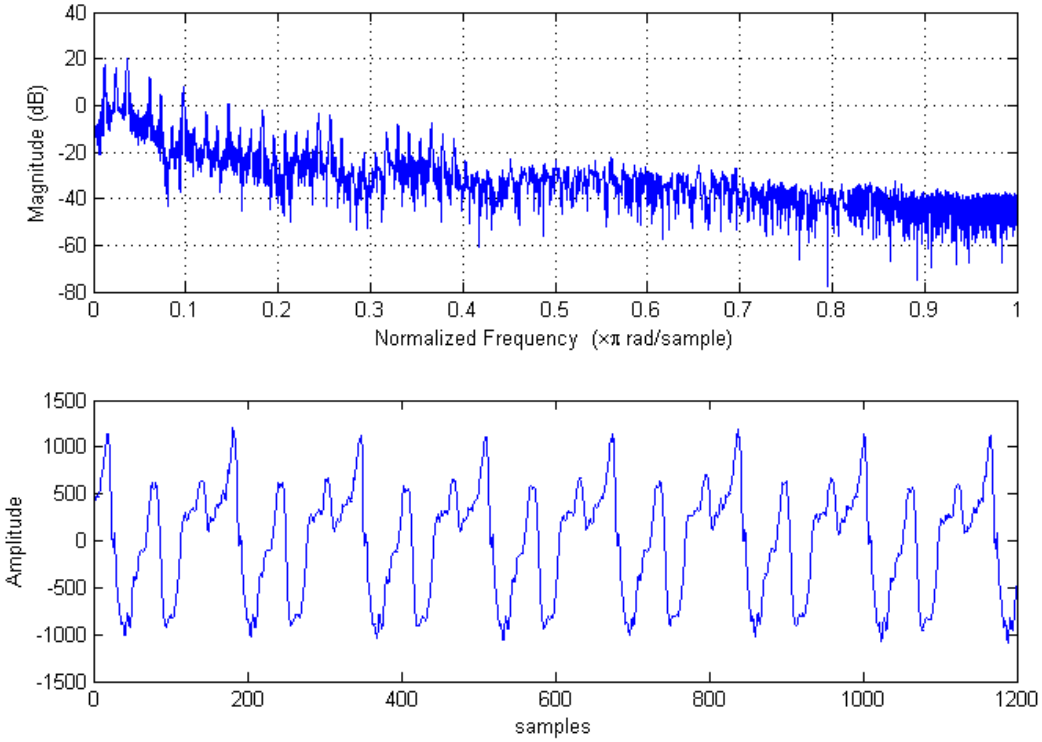


Figure 1 – Frequency and temporal representation of a short segment extracted from *Pitchpipe* sequence at 48 kHz sampling rate

On the contrary, the lack of stationarity in some audio signals is really a problem for the design of a coding scheme, redundancies are inevitably hard to exploit. Indeed, the coding system must be able to adapt automatically to quick variations of the signal properties. This second temporal property of audio signals can be defined as potential high dynamic energy variations. In a few milliseconds (ms) time interval, the signal energy may change very quickly. This can be illustrated with a percussive sound that appears just after a period of relative silence. Figure 2 shows a short segment of the *castanets* audio sequence. As explained in Chapter 3, this kind of audio signal, which is usually referred as transients or attacks, is usually not optimally represented in the frequency domain and special processing must be provided.

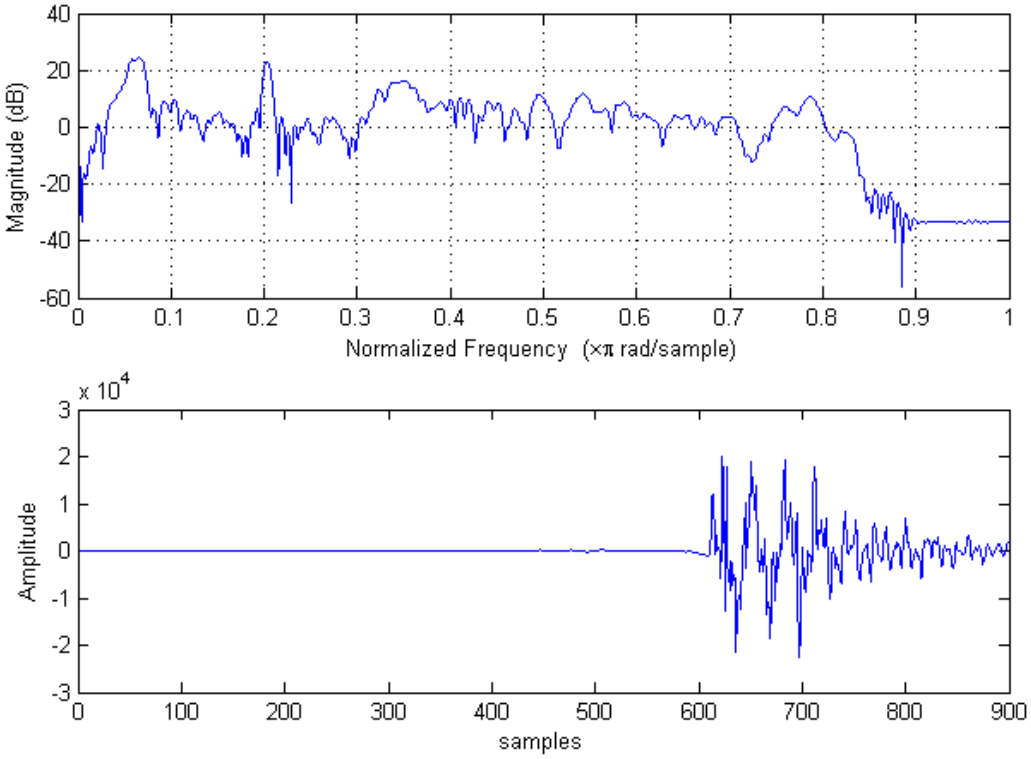


Figure 2 – Short segment extracted from *Castanets* sequence at 48 kHz sampling rate

General audio signals can at any time switch between steady states and transition phases. As such the choice of a time-frequency analysis method always involves a fundamental trade-off between time and frequency resolution requirements. A filter bank or a transform is used for mapping audio signals from the time domain into the frequency domain. It is often used as a basic component of audio signal processing. In that context, the main feature of a time-frequency transformation is its ability to provide a compact representation of any audio signal by subdividing its content into a compact representation. For this purpose, the time-frequency transformation must maximally reduce the redundancy. However, several aspects of the human audio perception need also to be considered in filter bank audio processing, especially in order to respect both temporal and spectral properties of the audio signal during filter bank processing. Indeed, the perceptual relevance of the time-frequency component of the audio signal and of possible coding artefacts must be considered.

2.1.2 General structure of filter banks

Filter banks and transforms play an important role in audio signal processing and perceptual audio coding. The input time domain audio signal $X(z)$ is split into several band-limited signals $Y_k(z)$ with $0 \leq k \leq M - 1$ (spectral or sub-band coefficients) obtained through the application of a set of analy-

sis band-pass filters H_k(z). The reconstructed signal $\hat{X}(z)$ is the sum of the recombined sub-band signals filtered via the synthesis filters F_k(z).
H_k(z) and F_k(z) are given by their transfer functions [Bellanger 76]:

$$H_k(z) = \sum_{n=-\infty}^{+\infty} h_k(n) z^{-n}$$
$$F_k(z) = \sum_{n=-\infty}^{+\infty} f_k(n) z^{-n} \quad (2.1)$$

Figure 3 illustrates an M sub-bands analysis and synthesis filter bank.

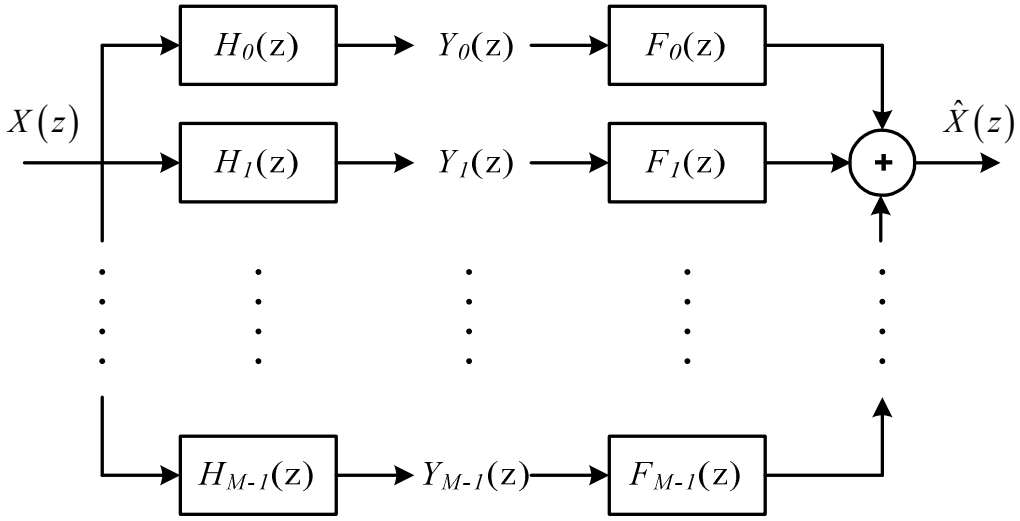


Figure 3 – Structure of filter bank with M sub-bands

A large number of filter banks and transforms can be represented by this basic representation. However, the filter bank displayed on Figure 3 is not suitable for a compact representation of the audio signal: albeit split in relative independent signals, there are more samples in the sub-band domain than in the full band initial time domain. A decimation operator needs to be introduced to reduce the amount of samples transmitted leading to the introduction of the multi-resolution filter bank theory.
The filter banks which are typically used in audio processing and coding can generally be defined by the maximally decimated filter banks theory introduced in [Vaidyanathan 93, Malvar 92b]. Indeed, in that case the number of samples in sub-band domain is equivalent to the number of samples in time domain. For instance, the pseudo-quadrature mirror filter banks (PQMF), the modified discrete cosine transform (MDCT), the modulated lapped transform (MLT) and the lapped orthogonal transforms (LOT) [Bosi 99, Malvar 92b, Shlien 97] can be cited. The MDCT and MLT, which are basically defining the same transform, will be presented in details in Chapter 3.

2.1.3 Maximally decimated filter banks

2.1.3.1 General structure

Following the presentation of the analysis and synthesis filter bank structure, the critically sampled uniform filter banks are introduced now. As every analysis filter output represents only a part of the audio signal bandwidth, this signal can be downsampled according to the associated bandwidth to which it corresponds [Shannon 49]. According to the Nyquist theorem the sampling frequency shall be twice the bandwidth. Figure 4 describes the M sub-bands maximally decimated filter bank processing scheme.

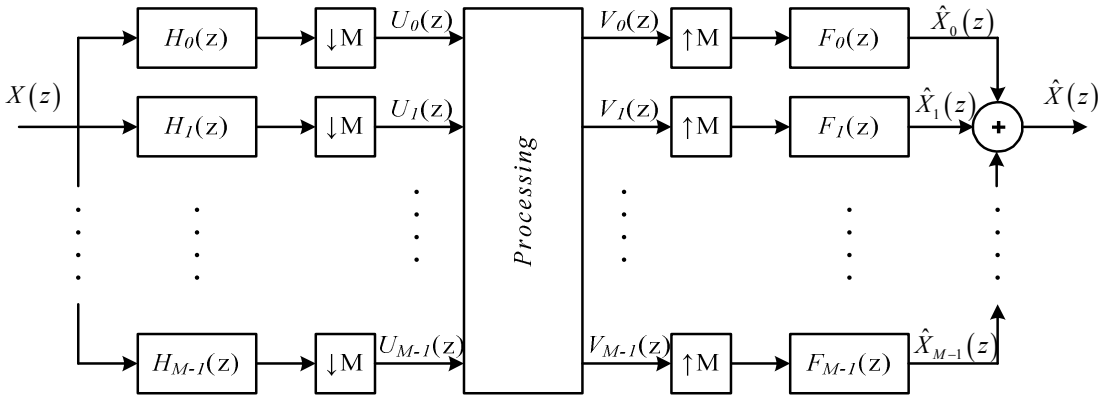


Figure 4 – Maximally decimated uniform filter bank with M sub-bands

The critically sampled uniform filter bank is defined by the set of analysis filters H_k(z) and the associated synthesis filters F_k(z) with $0 \le k \le M-1$. The output of the k-th analysis filter is obtained by the operation of filtering followed by the decimation by a factor of M:

$$\mathrm{U}_k(z) = \frac{1}{M}\sum_{l=0}^{M-1} \mathrm{H}_k\left(z^{\frac{1}{M}}\,\mathrm{W}^l\right)\mathrm{X}\left(z^{\frac{1}{M}}\,\mathrm{W}^l\right), 0 \le k \le M-1 \qquad (2.2)$$

with $\mathrm{W} = e^{j(2\pi/M)}$. In this equation, only the decimated component given for $l = 0$ corresponds to the useful signal, while the other components ($l \neq 0$) represent the frequency shifted version of the input signal spectrum filtered by $H_k(z)$. Those components come from the decimation and are named aliasing components.

Without any processing, the sub-band signal $V_k(z)$ (= $U_k(z)$) is first interpolated and then filtered through the synthesis filters to obtain the sub-band outputs:

$$\hat{X}_k(z) = V_k\left(z^M\right)F_k(z) = \frac{1}{M}\sum_{l=0}^{M-1} H_k\left(z\,\mathrm{W}^l\right)X\left(z\,\mathrm{W}^l\right)F_k(z), 0 \le k \le M-1 \qquad (2.3)$$

After summation, the reconstructed signal is given by:

$$\hat{X}(z) = \sum_{k=0}^{M-1} \hat{X}_k(z) = \frac{1}{M} \sum_{l=0}^{M-1} \left(\sum_{k=0}^{M-1} H_k\left(z\,\mathrm{W}^l\right) F_k(z) \right) X\left(z\,\mathrm{W}^l\right) \tag{2.4}$$

This can be rewritten as:

$$\hat{X}(z) = \frac{1}{M} \sum_{l=0}^{M-1} X\left(z\,\mathrm{W}^l\right) \left(\sum_{k=0}^{M-1} H_k\left(z\,\mathrm{W}^l\right) F_k(z) \right) = \sum_{l=0}^{M-1} X\left(z\,\mathrm{W}^l\right) A_l(z) \tag{2.5}$$

where $A_0(z)$ is the amplitude distortion affecting the input signal $X(z)$ and $A_l(z)$ for $l \neq 0$ are the gains of the l^{th} aliasing terms that are unwanted components.
The reconstructed spectrum is then a linear combination of the input signal $X(z)$ and its $M - 1$ uniformly frequency shifted versions $X(zW^l)$. The aliasing components are then cancelled for $1 \leq l \leq M - 1$ if:

$$\sum_{k=0}^{M-1} \mathrm{H}_k\left(z\,\mathrm{W}^l\right) \mathrm{F}_k(z) = 0 \tag{2.6}$$

The distortion function (amplitude and phase distortion) or transfer function is defined by:

$$A_0(z) = \mathrm{T}(z) = \frac{1}{M} \sum_{k=0}^{M-1} \mathrm{H}_k(z) \mathrm{F}_k(z) \tag{2.7}$$

Thus, the filter bank becomes a linear and time invariant system when aliasing is cancelled. The complete system is said to be a Perfect Reconstruction (PR) system if the transfer function corresponds to a pure delay. This condition, which is called paraunitary, is then expressed by:

$$\mathrm{T}(z) = cz^{-d}. \tag{2.8}$$

2.1.3.2 Alias component matrix

The aliasing terms can be written as a matrix:

$$M \begin{bmatrix} A_0(z) \\ A_1(z) \\ \vdots \\ A_{M-1}(z) \end{bmatrix} = M \cdot \mathbf{A}(z) = \mathbf{H}(z) \cdot \mathbf{f}(z) \tag{2.9}$$

where $\mathbf{f}(z)$ corresponds to the synthesis filter vector $\begin{bmatrix} F_0(z) & F_1(z) & \cdots & F_{M-1}(z) \end{bmatrix}^T$ and $\mathbf{H}(z)$ is a $M \times M$ matrix called the Alias Component (AC) matrix and is of the form:

$$\mathbf{H}(z) = \begin{bmatrix} H_0(z) & H_1(z) & \cdots & H_{M-1}(z) \\ H_0(zW) & H_1(zW) & \cdots & H_{M-1}(zW) \\ \vdots & \vdots & \ddots & \vdots \\ H_0\left(zW^{M-1}\right) & H_1\left(zW^{M-1}\right) & \cdots & H_{M-1}\left(zW^{M-1}\right) \end{bmatrix} \tag{2.10}$$

In order to cancel the aliasing terms, $A_l(z)$ for $1 \le l \le M-1$ has to be forced to zero and aliasing cancellation condition can be rewritten as:

$$\mathbf{H}(z) \cdot \mathbf{f}(z) = \mathbf{t}(z) = \begin{bmatrix} M \cdot A_0(z) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} M \cdot \mathrm{T}(z) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.11}$$

with:

$$\mathbf{f}(z) = \mathbf{F}(z)\mathbf{v}(z) = \begin{bmatrix} F_0(z) & 0 & \cdots & 0 \\ 0 & F_1(z) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & F_{M-1}(z) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{2.12}$$

The matrix of aliasing gain rewrites as:

$$\mathbf{t}(z) = \mathbf{H}(z) \cdot \mathbf{F}(z) \cdot \mathbf{v}(z) \tag{2.13}$$

And $\mathbf{U}(z) = \mathbf{H}(z).\mathbf{F}(z)$ is defined as the composite alias component matrix:

$$\mathbf{U}(z) = \begin{bmatrix} F_0(z)H_0(z) & F_1(z)H_1(z) & \cdots & F_{M-1}(z)H_{M-1}(z) \\ F_0(z)H_0(zW) & F_1(z)H_1(zW) & \cdots & F_{M-1}(z)H_{M-1}(zW) \\ \vdots & \vdots & \ddots & \vdots \\ F_0(z)H_0\left(zW^{M-1}\right) & F_1(z)H_1\left(zW^{M-1}\right) & \cdots & F_{M-1}(z)H_{M-1}\left(zW^{M-1}\right) \end{bmatrix} \tag{2.14}$$

As described in equation (2.8), we can go a step further and obtain the perfect reconstruction by requiring the additional constraint on $A_0(z) = T(z)$. The Alias Component matrix notation is useful for the representation of the aliasing terms and is used by some optimization algorithm for filter bank

design when a specific constraint has to be put on the aliasing terms as presented in section 2.3.3.

2.1.3.3 Polyphase representation

Polyphase representations were introduced by Bellanger [Bellanger 76] to facilitate the design of filter banks and their implementations through fast algorithms [Vaidyanathan 93, Malvar 92b]. This theory provides an alternate view on the reconstruction process.

The polyphase decomposition of the analysis filter bank, $H_k(z)$, (Type 1 polyphase) and of the synthesis filter bank, $F_k(z)$, (Type 2 polyphase) is used to decompose the analysis and synthesis filters given by their transfer functions, as defined in equation (2.1), into a sum of M terms expressed in the form:

$$H_k(z) = \sum_{l=0}^{M-1} z^{-l} \, \mathrm{E}_{kl}\left(z^M\right) \tag{2.15}$$

and

$$F_k(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} \, \mathrm{R}_{lk}\left(z^M\right) \tag{2.16}$$

Using the matrix notation, the previous equation can be written as:

$$\mathbf{h}(z) = \mathbf{E}\left(z^M\right)\mathbf{e}(z) \tag{2.17}$$

$$\mathbf{f}^{\mathbf{T}}(z) = z^{-(M-1)}\tilde{\mathbf{e}}(z)\mathbf{R}\left(z^M\right) \tag{2.18}$$

where

$$\mathbf{h}(z) = \left[H_0(z) \;\; H_1(z) \;\; H_2(z) \;\; \cdots \;\; H_{M\text{-}1}(z)\right]^{\mathbf{T}} \tag{2.19}$$

and

$$\mathbf{f}(z) = \left[F_0(z) \;\; F_1(z) \;\; F_2(z) \;\; \cdots \;\; F_{M\text{-}1}(z)\right] \tag{2.20}$$

are the analysis and synthesis vectors respectively and the vector $\mathbf{e}(z)$ represents the delay chain vector which is expressed in the form:

$$\mathbf{e}(z) = \left[1 \;\; z^{-1} \;\; \cdots \;\; z^{-(M-1)}\right]^{\mathbf{T}} \tag{2.21}$$

The notation $\tilde{\mathbf{e}}(z)$, for the matrix $\mathbf{e}(z)$, corresponds to $\mathbf{e}_*^{\mathrm{T}}\left(z^{-1}\right)$ where * indicates that the components of the matrix are the conjugates.

The matrix $\mathbf{E}(z)$ is the $M \times M$ polyphase component matrix (also called polyphase matrix) which is given by:

$$\mathbf{E}(z)=\begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) & \cdots & E_{0,M\text{-}1}(z) \\ E_{1,0}(z) & E_{1,1}(z) & \cdots & E_{1,M\text{-}1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M\text{-}1,0}(z) & E_{M\text{-}1,1}(z) & \cdots & E_{M\text{-}1,M\text{-}1}(z) \end{bmatrix}^{\mathbf{T}} \tag{2.22}$$

where each row represents one analysis filter. Similarly, the synthesis filter bank can be represented by a polyphase matrix $\mathbf{R}(z)=\left[R_{lk}(z)\right]$.

Based on equations (2.17) and (2.18), by migrating the decimation and up-sampling operations before and after the polyphase matrix operations, the polyphase implementation of the analysis and synthesis filter bank can be illustrated by the Figure 5.

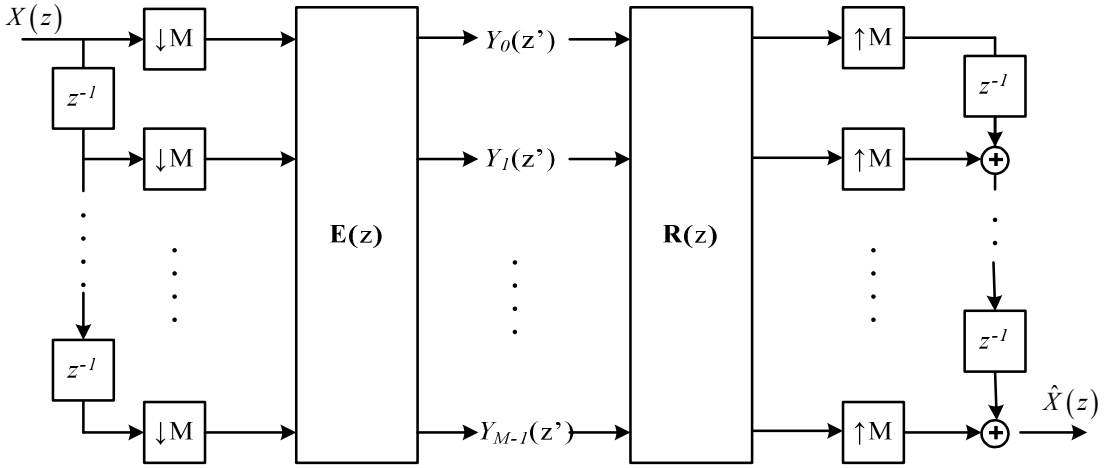


Figure 5 – Polyphase representation of critically sampled analysis and synthesis filter banks

In that context of polyphase representation, [Vaidyanathan 93] has shown that the analysis and synthesis filter banks verify the perfect reconstruction condition if and only if the product $\mathbf{R}(z)\mathbf{E}(z)$ can be written as follows:

$$\mathbf{R}(z)\mathbf{E}(z)=cz^{-\lambda}\begin{bmatrix} \mathbf{0} & \mathbf{I_{M\text{-}r}} \\ z^{-1}\mathbf{I_r} & \mathbf{0} \end{bmatrix} \tag{2.23}$$

where λ is a positive integer and for some integer r with $0 \le r \le M-1$. Hence, in order to satisfy the perfect reconstruction constraint, it has been demonstrated by Vaidyanathan that a sufficient condition is r = 0 and the synthesis polyphase matrix has to be the inverse of the analysis polyphase matrix:

$$\begin{aligned} &\mathbf{R}(z)\mathbf{E}(z)=cz^{-\lambda}\mathbf{I_M} \\ &\mathbf{R}(z)=z^{-K}\tilde{\mathbf{E}}(z) \end{aligned} \tag{2.24}$$

K is a positive integer which is selected in order to ensure the causality of the matrix $\mathbf{R}(z)$ and then of the synthesis filter bank. The polyphase representation is used for some filter banks or transforms definition and prototype design (as illustrated in paragraph 3.1.3).

2.1.3.4 Cosine modulated filter banks

The cosine modulated filter banks are obtained by the modulation of a low-pass prototype filter $h(n)$ with a cosine function. The main advantage is the possibility to derive low complexity implementations which are based on a simple filtering operation followed by a modulation. The pseudo-QMF (PQMF) filter bank has been introduced by Rothweiler in [Rothweiler 83]. Even if the PQMF does not achieve perfect reconstruction, aliasing components due to adjacent bands and the phase distortion are cancelled. The performance in terms of amplitude distortion depends on the optimization procedure which is used to design the filter bank prototype. Vaidyanathan has proposed several optimization methods to limit the amplitude distortion and shown that using the same definition the perfect reconstruction can be achieved [Vaidyanathan 93].

The impulse responses of the cosine modulated analysis and synthesis filter banks are expressed by:

$$\begin{aligned} h_k(n) &= h(n)\cos\left(\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)+\theta_k\right), 0\le k\le M-1 \\ f_k(n) &= h(n)\cos\left(\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)-\theta_k\right), 0\le k\le M-1 \end{aligned} \tag{2.25}$$

where $h(n)$ is the impulse response of the prototype filter of length L, $0\le n\le L-1$ and $\theta_k = \frac{(-1)^k\pi}{4}$

The perfect reconstruction is obtained only if the following conditions are met:

1) $L = 2mM$, where m is an integer
2) The synthesis filters are given by $f_k(n) = h_k(L-1-n)$
3) The prototype filter is a linear phase filter $h(n) = h(L-1-n)$
4) The polyphase components of order $2M$, noted $G_k(z)$ for $0\le k\le 2M-1$, of the prototype filter $H(z)$ must verify the following condition

$$\tilde{G}_k(z)G_k(z)+\tilde{G}_{M+k}(z)G_{M+k}(z) = \text{constant},\ 0\le k\le M-1 \tag{2.26}$$

Using the order $K = 2m-1$ polyphase matrix $\mathbf{E}(z)$ of equation (2.23), the condition 4) simply expresses the paraunitary property of this matrix. A cosine modulated filter bank is then a paraunitary filter bank with the synthesis polyphase matrix obtained by:

$$\mathbf{R}(z) = z^{-(2m-1)}\tilde{\mathbf{E}}(z) \tag{2.27}$$

Based on this ability to achieve perfect reconstruction, the cosine modulated filter banks are widely used in audio signal coding and more specifically for perceptual audio coding.

2.2 Filter banks for perceptual audio coding

The principle of perceptual audio coding consists in reducing the bit rate to represent the audio signal by taking into account the property of the human auditory system. The less relevant components of an audio signal can be quantized with less precision or even completely discarded. Perceptual relevancy is evaluated; this operation relies on the masking phenomena that are described in section 2.2.1. Perceptual audio coding mainly exploits the simultaneous masking which is described in the frequency domain. Hence, these coding schemes are based on filter banks or time-frequency transforms leading to a frequency domain representation of the audio signal. This representation holds for both perceptual property interpretation and application of the derived rules. Figure 6 gives a basic block diagram of a perceptual audio coding system.

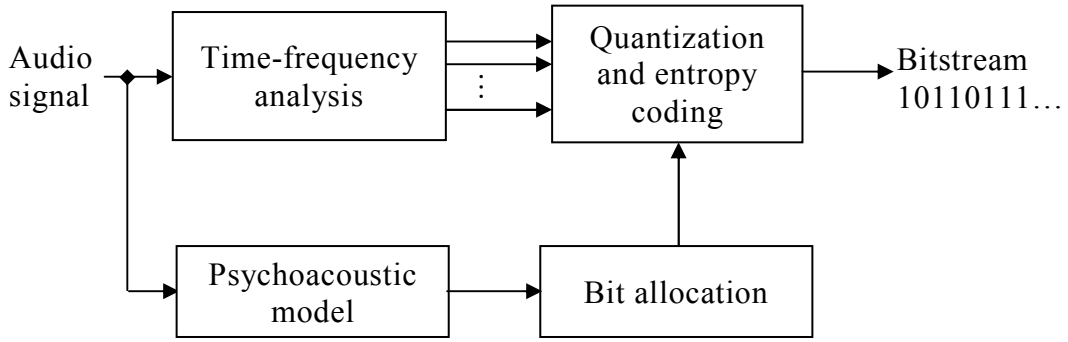


Figure 6 – General structure of a perceptual audio encoder

The blocks of Figure 6 are described as follows:

Time-frequency analysis: a frequency domain representation requires the use of a filter bank or a transform to decompose the time domain input audio signal into spectral components. This decomposition allows to better isolate the frequency content over each sub-band, this is called energy compaction or energy concentration. Moreover, this time-frequency transform decorrelates the sub-band signals and then eliminates some statistical redundancy. The main goal of this filter bank or transform is to divide the signal spectrum into frequency sub-bands or spectral coefficients which can then be conveniently exploited to shape the coding distortion according to

the auditory model. The main characteristics of a filter bank, which must be carefully evaluated during its design, are listed below:
- Good energy concentration: the basic idea in a data reduction scheme is to filter the signal into various sub-band signals. Instead of quantizing the samples of the signal with desired number of bits per sample, the quantization can be performed on the transform coefficients with a different number of bits for each. The total number of bits should be the same, but the mean square error is lower in the transform coding case compared to quantizing the samples in the time domain as the system can allocate the bits according to the spectrum characteristics.
- Signal adaptive time-frequency tiling: the audio signal is composed of stationary and transient parts. An audio coding scheme must adapt to the time- frequency content of the signal.
- Perfect or near-perfect reconstruction: this characteristic is important in order to recover the original audio signal at the decoding stage, or at least to approach its reconstruction, this is important, especially for higher rates (any reconstruction quality can be achieved, and transparent coding quality obtained).
- Critically sampled: each band-pass filter is sub-sampled by a factor corresponding to the number of sub-bands. No increase in terms of components to be encoded is required for efficient compression applications. In these systems, the overall rate at the output of the analysis stage equals the overall rate at the input of the analysis stage.
- Limited blocking artefacts: the audio signal is processed by blocks or frames. The reconstruction must ensure a smoothed transition from one block to the other.
- Good sub-band separation and stop-band attenuation: This characteristic corresponds to the minimum attenuation which can be obtained with a filter. It ensures a good separation between the sub-band to reduce the redundancy.

Psychoacoustic model: it aims at representing the behaviour of the human auditory system. Based on the input signal, a perceptual model, also known as psychoacoustic model, is used to estimate the perceptual masking threshold based on the simultaneous masking and in quiet masking phenomena [Zwicker 07]. Two slightly different effects are usually considered depending on the tonality characteristic of the masker. This model provides the necessary indication on how to inject quantization noise during the coding stage. This perceptual model relies on the time-frequency analysis of the audio signal if the frequency resolution is sufficient. Otherwise, a dedicated Discrete Fourier Transform (DFT) is directly applied to the time domain input signal in order to obtain an adequate frequency representation.

Quantization and entropy coding: the quantization stage associated with an efficient coding scheme aims at reducing the bit rate required to represent the audio signal. The sub-band signal is approximated on a reduced number of levels (quantization levels) and an indication of the selected level is further compacted using an entropy-coding scheme. This quantization step is applied in accordance with bit allocation based on the psychoacoustic masking properties. The spectral coefficients are then quantized with the objective of hiding the quantization noise just below the masking threshold, according to the decision performed during bit allocation.

Bit allocation: this procedure distributes the available bit budget, driven by the coding bit rate, in all the frequency sub-bands. It determines the amount of bits that each sub-band must receive in order to shape the quantization noise according to the perceptual model. The bit allocation is carried out based on the masking curve obtained from the perceptual model.

The basic components of an audio coding scheme are further described hereunder.

2.2.1 Perceptual model

In order to have a bit rate reduction guided by perception, transform coding is based on a perceptual model indicating how the spectrum of the quantization noise can be shaped. The main goal of the perceptual model is to describe as precisely as possible how the sound and the quantization noise might be perceived by the human auditory system.

2.2.1.1 Absolute threshold of hearing

Based on listening tests and experiments [Zwicker 07], it is considered that the human auditory system is able to perceive sound in the frequency range of 20 Hz – 20 kHz. However, the ear is not equally sensitive at all the frequencies. The hearing area is the region in the Sound Pressure Level (*SPL*)/frequency plane in which the sound is audible. It is defined as the region in which listeners can perceive a stimulus made of a pure tone or a narrow band noise in a noiseless environment, and the lower energy level for which this stimulus can be heard defines the absolute threshold of hearing. The threshold of hearing, also called the threshold in quiet, can be approximated by:

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \quad (\text{dB } SPL) \quad (2.28)$$

where f is the frequency in Hz and the absolute threshold of hearing in dB *SPL*. It is represented on Figure 8. It can be seen that the human auditory system has a maximum sensibility between 2 and 5 kHz. However, for

higher frequencies the curve increases exponentially and only a sound with a significant level of energy can be perceived. It is particularly difficult to hear sounds above 16 kHz where only sounds with energy levels higher than 60 dB *SPL* are perceived.

2.2.1.2 Critical bands

The concept of critical bands, introduced by Fletcher [Fletcher 40], is particularly important in psychoacoustic. The loudness perceived in a critical band corresponds to the integration of the loudness for all the frequency components of that band. When measuring the hearing threshold of a narrow band noise as a function of its bandwidth while keeping its overall pressure level constant, this threshold remains constant as long as the bandwidth does not exceed the critical bandwidth. When exceeding the critical bandwidth, the hearing threshold of the noise increases.
Critical bands play a role in sound intensity perception and are also related to masking phenomena. If a sound is presented simultaneously with maskers, the maximum masking contribution is obtained only when the masking components fall within the same critical band. The masking property is then constant over critical bandwidths and drop rapidly outside this band. Moreover, the perceived loudness of a sound is independent of its bandwidth as long as it is smaller than the corresponding critical bandwidth.
Zwicker and Fastl [Zwicker 07] have proposed an analytic expression of critical bandwidths as a function of their centre frequencies:

$$\Delta f = 25 + 75\left[1 + 1.4\left(0.001 f_c\right)^2\right]^{0.69} \tag{2.29}$$

Table 1 lists the bands used to model the human ear in the form of a filter bank with 24 abutting critical bands. It is typically used for audio coding applications. A new frequency scale has been introduced, the Bark scale, which follows the critical band rate and has a unit “Bark”. According to this scale, 1 Bark defines the lower and upper limits of each critical band. The Bark scale is then a mapping of frequencies onto a relation Herz/Bark. This relation is almost linear up to 500 Hz and then becomes quasi logarithmic. The critical band rate $z(f)$ (expressed in Bark) can be approximated using the following expression from [Zwicker 07]:

$$z\left(f\right) = 13\arctan\left(0.00076 f\right) + 3.5\arctan\left[\left(\frac{f}{7500}\right)^2\right] \tag{2.30}$$

Several alternative methods have been used to simulate the hearing frequency scale leading to alternative definitions. Moore and Glasberg [Moore 03] have proposed to replace the Bark scale by what they call the *Equiva-*

lent Rectangular Bandwidth (ERB) scale. The ERB *scale* is defined as the number of ERBs below each frequency:

$$ERBS(f) = 21.4\left(4.37\left(\frac{f}{1000}\right)+1\right) \tag{2.31}$$

f is expressed in Hertz. And the critical bandwidth/ERB is given by:

$$ERB(f) = 0.108f + 24.7 \tag{2.32}$$

Critical band index	Centred frequency (Hz)	Critical bandwidths (Hz)	Lower frequency limit (Hz)	Upper frequency limit (Hz)
1	50	100	0	100
2	150	100	100	200
3	250	100	200	300
4	350	100	300	400
5	450	110	400	510
6	570	120	510	630
7	700	140	630	770
8	840	150	770	920
9	1000	160	920	1080
10	1170	190	1080	1270
11	1370	210	1270	1480
12	1600	240	1480	1720
13	1850	280	1720	2000
14	2150	320	2000	2320
15	2500	380	2320	2700
16	2900	450	2700	3150
17	3400	550	3150	3700
18	4000	700	3700	4400
19	4800	900	4400	5300
20	5800	1100	5300	6400
21	7000	1300	6400	7700
22	8500	1800	7700	9500
23	10500	2500	9500	12000
24	13500	3500	12000	15500

Table 1 – Critical bands

2.2.1.3 Temporal and frequency masking

Several perception criteria are used to reduce the quantity of data to be stored or transmitted without any degradation of the subjective audio quality. Two kinds of masking phenomena can be experienced and are presented in this section: frequency masking (also known as simultaneous masking) and temporal masking.

Frequency masking is commonly used in audio coding schemes. The masking curves are used to distribute the quantization noise below the perceptual masking threshold. Temporal masking offers a more limited interest for the perceptual audio coding as it is more difficult to exploit. Masking models are usually not exploiting the temporal masking. However, the perceptual audio coding models must ensure a sufficient temporal concentra-

tion of the quantization noise and the filter banks are usually designed to provide this time-frequency resolution trade-off as it will be explained in section 3.2.

Frequency masking

The first phenomenon is defined as frequency masking or simultaneous masking. It occurs when a so-called masker masks simultaneous signals presented at the auditory system and more specifically when stimuli are present at nearby frequencies. The curve, represented in Figure 7 as masking threshold, describes the modified audibility threshold in presence of some masking signal. This curve is based on the absolute threshold of hearing, on top of which, with the additional presence of the masker, the masking threshold is modified. Two kinds of maskers are commonly used to derive the masking curve: maskers made of pure tones and of band limited noise. Both are studied to measure their capability at masking (quantization) noise.

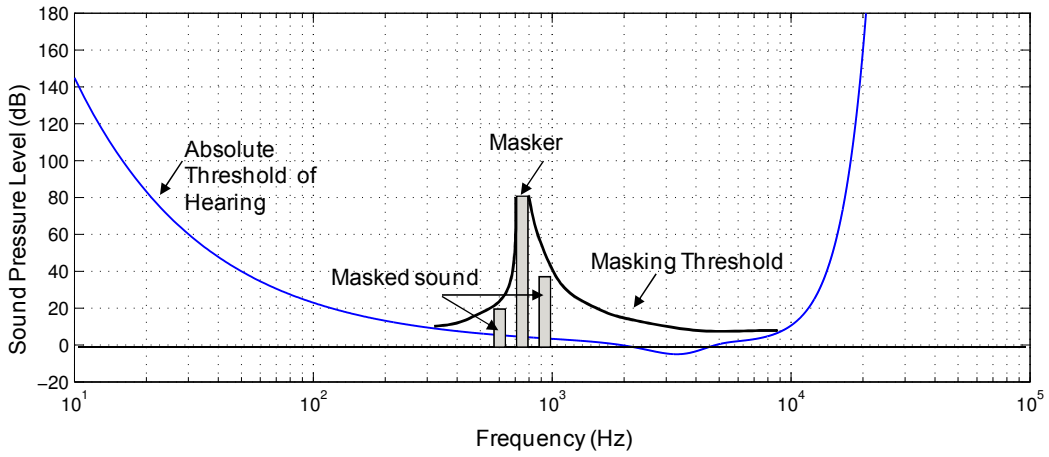


Figure 7 – Frequency masking phenomena

The first simultaneous masking phenomenon is defined as tone-masking-noise. In this first category of frequency masking, a pure tone in a critical band masks a band limited noise as long as this noise belongs to the same critical band. The minimum signal-to-mask ratio (SMR), which represents the smallest difference between the intensity of the masking signal and the intensity of the masked signal, tends to lie between 21 and 28 dB. In the example provided in Figure 8, a noise of one Bark bandwidth centred on the central frequency of the critical band is masked by a pure tone of 80 dB at the same central frequency.

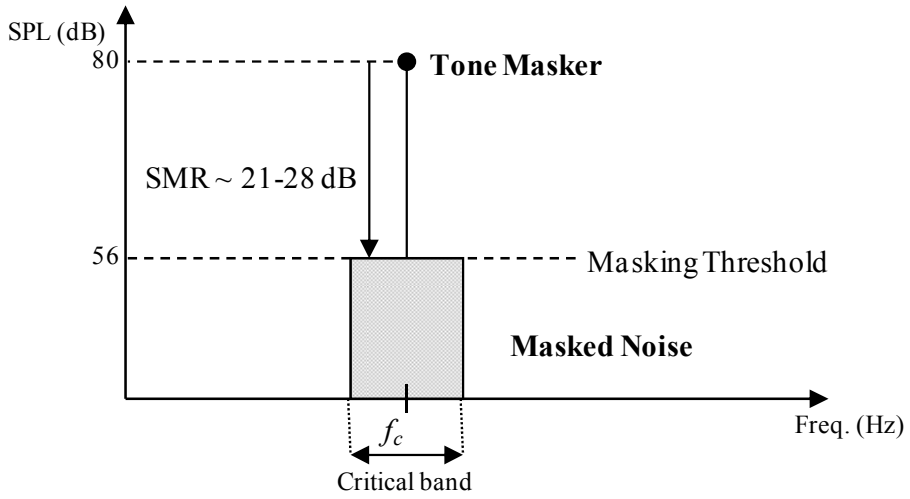


Figure 8 – Tone masking noise

The second important masking phenomenon is called noise-masking-tone. In this scenario, a band limited noise masks a pure tone within the same critical band. This masking property occurs if the intensity of the tone is below a certain threshold which depends directly on the intensity. In that case, the range in which the SMR tends to lie is between -5 and +5 dB. Figure 9 illustrates this phenomenon for a pure tone with an intensity which is smaller than the intensity of the noise.

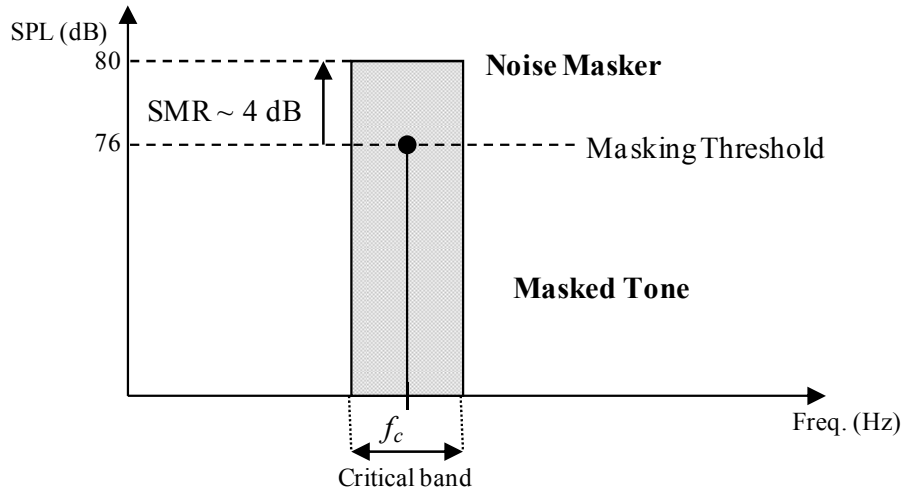


Figure 9 – Noise masking tone

It should be noted that the noise-masking-tone and tone-masking-noise are asymmetric in masking power between the noise masker and the tone masker.

Finally, a third phenomenon can be defined as a band limited noise masking another band limited noise. This noise-masking-noise scenario is usually more difficult to characterize, but an intensity detection threshold is usually of 26 dB. The simultaneous masking effect is not limited within a single critical band. The masking effect of a pure tone signal positioned at

the boundaries of a critical band will spread over other critical bands. This effect is known as the spread of masking and is often modeled with a slope of +25 dB per Bark for frequency lower that the masker and with a slope of -10 dB per Bark for the higher frequencies as schematically illustrated on Figure 7.

Temporal masking

When the human auditory system has been excited by a pure tone, a loss of sensibility occurs around this frequency during few hundreds of milliseconds. This phenomenon is known as post-masking and is the most commonly used. The temporal masking is actually composed of two categories: pre-masking and post-masking. Pre-masking is effective a few milliseconds before the onset of the masker, it is usually considered to last only 1-2 ms. As stated above, post-masking is a stronger and longer phenomenon (lasting generally between 50 and 100 ms, but no longer than 150 ms) occurring after the masker offset and depending on the masker level, duration and relative frequency of masker and probe signals. Figure 10 illustrates the temporal masking phenomena.

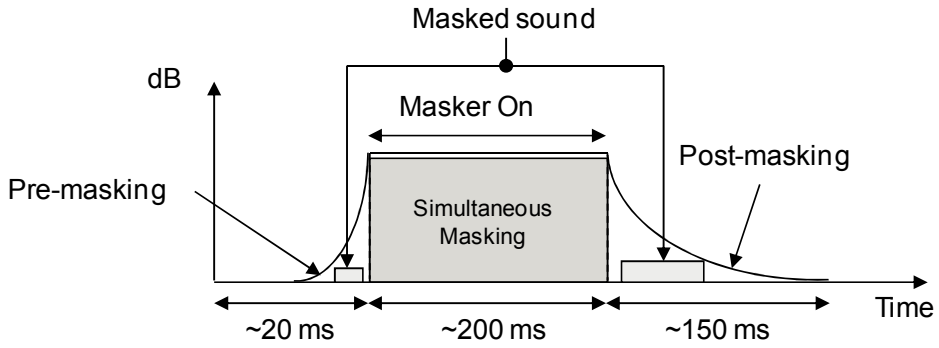


Figure 10 – Temporal masking phenomena

2.2.2 Quantization and entropy coding

In this paragraph, some quantization methods that are used to reduce the necessary bit rate to represent and transmit the audio signal are briefly introduced. The quantization consists in limiting the number of states associated to a signal, i.e. a signal taking its value in $\mathbb{R}$ is directly associated to a finite set of values (C). The quantization operation can be defined as:

$$\hat{x} = Q(x) = c_i \tag{2.33}$$

where $x \in \mathbb{R}$ and $\hat{x}$ is the quantized value of x taken its value c_i in the codebook C. The quantization error is usually measured by the distortion measure which is commonly defined as the mean squared error (MSE):

$$\sigma_d^2 = \mathrm{E}\left[\left(x - c_i\right)^2\right] \tag{2.34}$$

where σ_d^2 is the variance of the distortion expressed as $d = x - c_i$. The MSE distortion is simple to use but it does not directly match the subjective perception for audio signal. The signal to noise ratio (SNR) is also defined as a quality measure for a quantizer:

$$SNR = \frac{\sigma_x^2}{\sigma_d^2} \tag{2.35}$$

In order to measure the bit rate associated with the quantizer, the association of a unique codeword is used with each value of the quantizer. If the quantizer is composed of L codewords, in order to code each codeword with a fixed number of bits, the necessary number of bits R is defined as:

$$R \geq \log_2 L \tag{2.36}$$

To measure the minimum necessary quantity of information to transmit L words, the first order entropy of the quantization indices is frequently used [Shannon 48] and defined by:

$$E = -\sum p_{c_i} \log_2\left(p_{c_i}\right) \tag{2.37}$$

where p_{c_i} is the probability to have the codeword c_i ($\hat{x} = c_i$). E is the entropy given the theoretical limit of redundancy reduction for a memory-less discrete source and is expressed in bits/codeword. Hence, it is generally used as an estimation of the rate per sample.

Quantization can be classified as scalar or vector quantization. In the next sections, the two main classes of quantizers and the principle of entropy coding are introduced.

2.2.2.1 Scalar quantization

In this section, we describe shortly the main scalar quantization schemes: uniform, non-uniform and differential scalar quantizers.

Uniform scalar quantization

A uniform scalar quantizer is a memory-less process that has a "cell" decision c_i of regular size. The cell defines the interval in which the input signal will be associated with one of the quantized value. The quantized value is then obtained by rounding each sample to one of a set of discrete values

(*C*). The difference between the adjacent quantization levels is defined by the step size *q*.

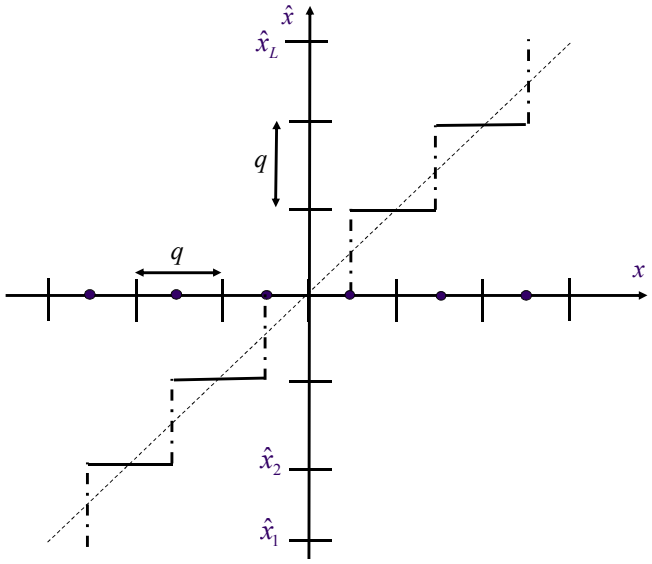


Figure 11 – Uniform scalar quantizer

Assuming equiprobability of the signal in the quantization intervals, the variance of the quantization noise or average distortion of the uniform scalar quantizer in case of uniform distribution of quantization error (or in case of high rate quantization) can be shown to be:

$$\sigma_d^2 = \frac{q^2}{12} . \tag{2.38}$$

For a uniform scalar quantization scheme followed by an entropy coder as described in section 2.2.2.3, the distortion can be expressed [Moreau 1995] as a function of the number of bits in the form:

$$\sigma_d^2 = \frac{1}{12} 2^{2e} 2^{-2b} \tag{2.39}$$

where *e* represents the differential entropy of the input signal *x*, and *b* denotes the number of bits per symbol:

$$e = -\int_{-\infty}^{+\infty} p_x(x) \log_2 \left(p_x(x) \right) dx \tag{2.40}$$

$p_x(x)$ is the probability density function of the input signal *x*.

Non-uniform scalar quantization

As opposed to uniform scalar quantizer, the non-uniform scalar quantizer uses non-uniform step sizes. The definition of the step sizes can be opti-

mally derived from the probability density function of the input signal. This class of quantizer is more efficient in terms of MSE, if the step sizes and corresponding centroid are adapted to the signal statistics aiming to minimize the mean square quantization error.
The centroid represents the quantized value associated to any value lying within an interval.
Another commonly used non-uniform quantizer relies on the use of compression and expansion functions respectively before and after a uniform scalar quantizer. This non-linear mapping function maps the non-uniform step sizes to the uniform step sizes. This permits the use of the simple uniform scalar quantization with a limited complexity. Some logarithmic companding functions have been widely used in speech coding such as the A-law and the μ-law.

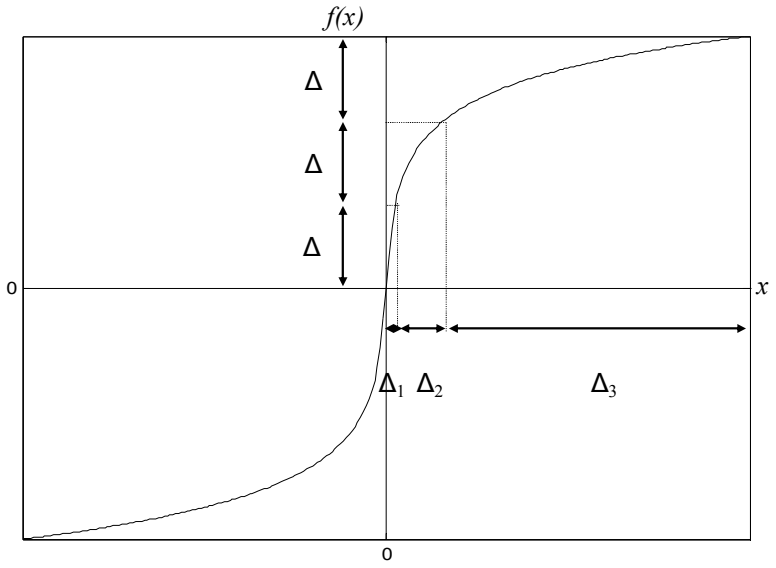


Figure 12 – Example of companding function

Alternatively, the Lloyd algorithm can be used to design the quantizer associated with a signal depending on its statistics [Lloyd 57].

Differential scalar quantization
The last class of scalar quantizer is based on the exploitation of the correlation between consecutive values to be quantized. The differential scalar quantizer removes the temporal redundancy of the input signal based on a short-term prediction. Figure 13 provides the general scheme of a predictive differential scalar quantizer.

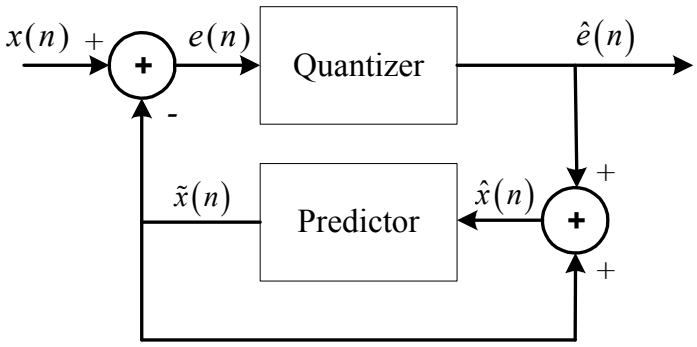


Figure 13 – General scheme of predictive scalar quantizer

The principle of differential quantizer is as follows. The estimation of the input signal $\tilde{x}(n)$ is obtained from the past samples $x(n-1),\ldots,x(n-L)$. The prediction error $e(n)$ is quantized and transmitted to the decoder. The quantized signal $\hat{x}(n)$ is then obtained when the quantized prediction error $\hat{e}(n)$ is added to the predicted signal $\tilde{x}(n)$ which is computed similarly at the encoder and decoder.

2.2.2.2 Vector quantization

As opposed to scalar quantization, vector quantization aims at jointly quantizing a group of samples. The vector $\mathbf{x}$ is built from the consecutive samples $x(n)$:

$$\mathbf{x} = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(M-1) \end{bmatrix} \tag{2.41}$$

where M represents the vector dimension. The vector quantization maps the input vector $\mathbf{x}$ to a vector $\hat{\mathbf{x}}$ from a codebook of L code vectors. The quantized vector $\hat{\mathbf{x}}$ is usually obtained based on the nearest neighbour algorithm. The distortion measure is commonly defined with squared error as the L_2 norm:

$$d = \left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2 = \sum_{n=0}^{M-1} \left(x(n) - \hat{x}(n)\right)^2 \tag{2.42}$$

Based on this distortion measure, the distance between the input vector and each code vector of the codebook is computed and the code vector minimizing the distortion measure is selected.

In order to build the codebook, several algorithms have been developed. The Linde-Buzo-Gray (LBG) algorithm [Linde 80], which is an iterative codebook design algorithm defined as an extension of the Lloyd algorithm for scalar quantization, is one of the most commonly used codebook design algorithm. Figure 14 illustrates the partitioning of a 2D space (two dimensions vector quantizer). The grey points represent the signal input, the black points give the graphical representation of the code vectors (codebook with 24 code vectors) and the areas delimited by the black lines gives the corresponding Voronoi cells for the vector quantizer.

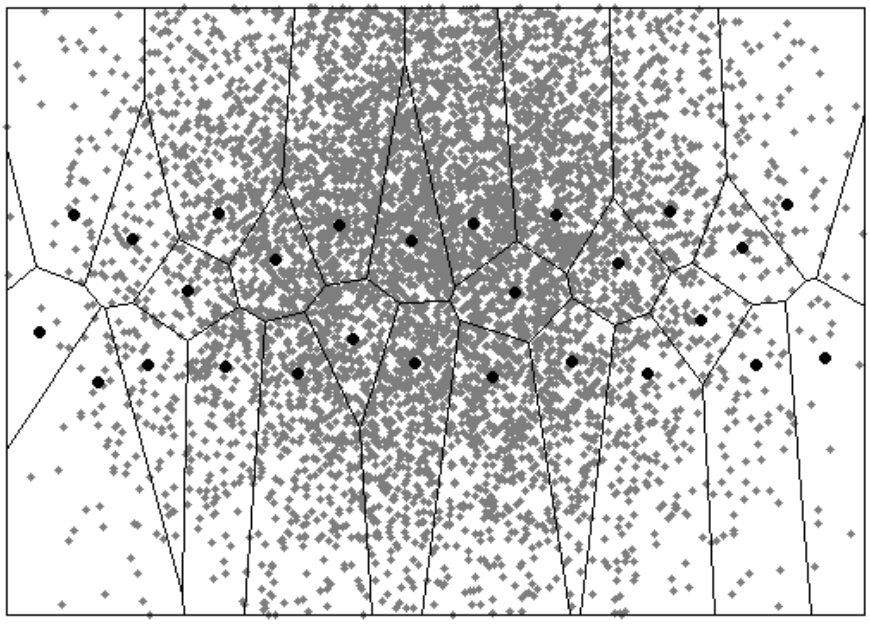


Figure 14 – Partitioning based on vector quantization with 24 code vectors (black dots)

The codebook can be structured in order to limit the required complexity by the search algorithm for the quantization. For instance, the k-nearest neighbour algorithm can be used to classify the components of a quantizer. The *Tree Structured Vector Quantization* [Gersho 92], which is usually organized as a binary tree structure where the code vectors lie on branches of a tree, allows to dramatically reduce the complexity if the codebook is well balanced. Indeed, the search is performed in stages. In each stage, a substantial subset of candidate vectors is eliminated from consideration. In a binary tree search, the input code vector is compared with two candidate vectors at each node of the tree. The nearest candidate vector determines which of the two paths through the tree must be selected in order to reach the next stage of the search. At each stage, the number of candidate vectors is reduced to roughly half the previous set of candidates.

The structured vector quantizations are also organized in several families of vector quantizer which are not detailed in this document. For instance, the following vector quantizers have been used in standardized speech and audio codecs: the *Lattice Vector Quantization* [Conway 93], where the codebook is a subset of regular lattice, the *Multistage Vector Quantization* [Gersho 92] which uses successive stages of vector quantization. It should be noted that several vector quantizations: Splits Vector Quantization [Paliwal 91], Conjugate Structure Vector Quantization [Kataoka 93] or the Gain-Shape Vector Quantization [Sabin 84] have also been widely published and used in speech and audio coding schemes, such as for the Line Spectral Pairs (LSP) representing the Linear Prediction Coding (LPC) coefficients in the Code Excited Linear Prediction Coding (CELP) [So 07] or for the quantization of the transform coefficients [Iwakami 96].
Vector quantization has led to a large number of methods with different efficiencies; the performance depends on the signal statistics and correlation. All these vector quantization methods have been used in different speech or audio coding schemes with various quality/complexity trade-offs.

2.2.2.3 Entropy coding

In order to reduce the bit rate, two solutions can be used. The quantization can be based on a limited size codebook which limits the associated bit rate, but at the cost of an increased distortion. The alternative and complementary solution efficiently encodes the quantization index taking into account the statistics of the quantized signal.
Entropy coding is a lossless coding scheme which is used on top of a lossy quantization stage. The following paragraphs present Huffman Coding and Arithmetic Coding which are the most commonly used entropy coding schemes which are part of the large number of variable-size code variants: Huffman coding, Rice coding, Golomb coding and Arithmetic coding [Salomon 00].

Huffman coding
Huffman coding [Huffman 52] defines a method to build variable-size codes aiming to approach the theoretical entropy limit. The principle of Huffman coding is to assign shorter code words to the quantization indices with highest probability and inversely longer code words to the indices with lowest probability. An example of quantization with 5 levels with the associated probability and corresponding Huffman code is given in Table 2.

Index	Probability	Huffman Code
0	0.30	00
1	0.25	01
2	0.25	10
3	0.10	110
4	0.10	111
R	3	2.20

Table 2 – Huffman code example

The entropy *E* of this source is equal to 2.19 and the Huffman coding which is provided in the third column obtains an average bit rate of 2.20 bits per sample compared to the 3 bits which would be obtained without entropy coding. In the most common usage of Huffman coding, the entropy coding table is built from a training database which is used to learn the statistics of the source. This table is then stored in memory on both encoder and decoder sides.

Arithmetic coding

In general, entropy coding based on Huffman coding does not remove all statistic redundancy. In order to exploit this remaining redundancy, several symbols can be combined during the coding stage. Arithmetic coding [Witten 87] is an entropy coding method which overcomes the limitation of integer lengths of code words and then further improves the performance of the coding. This coding scheme is based on the coding of a sequence of input symbols as a large fractional number. In each stage of an arithmetic encoding scheme, the current interval of values is divided into sub-intervals, each sub-interval representing a fraction of the current interval which is proportional to the symbol probability. The sub-interval, which represents the symbol to be coded, becomes the new interval for the next stage of the encoding process as illustrated on Figure 15.

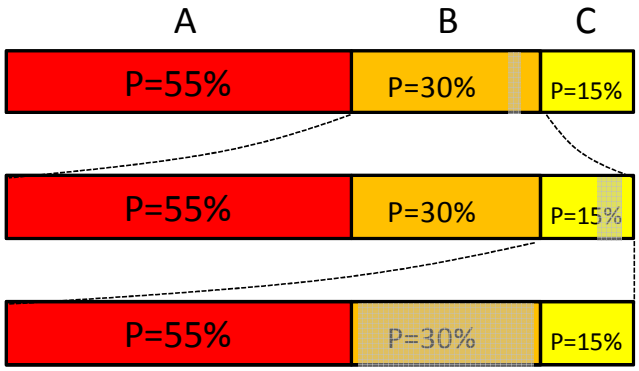


Figure 15 – Arithmetic coding (B-C-B are emitted)

When all symbols have been encoded, the resulting interval, represented in grey on Figure 15, unambiguously identifies the sequence of symbols. It is not necessary to transmit the corresponding interval but only a binary code which represents this interval.

2.2.3 Bit allocation

As explained previously, transform coding consists in quantizing the spectral components so as to minimize the quantization error according to a certain criteria. The bit allocation plays an important role in the coding scheme as this module aims at distributing the bit budget among the spectral components. As the spectrum evolves with time, the bit allocation is a dynamic process which is applied for each input frame.
For transform coding, when the number of spectral component is relatively high, those spectral components are grouped per frequency band following a perceptual scale (critical bands, Bark scale, or equivalent rectangular bandwidth - ERB). In that case, the bit allocation is performed for each group of frequency bins.
The first criterion which can be used for bit allocation is a Least Mean Square Error (LMSE). The resulting quantization noise is constant for all frequency components (i.e. white noise) as defined in [Jayant 84]. Perceptual audio coding allocates the bits per band in order to mask the quantization noise when the bit rate is sufficient, or at least to minimize its audibility. The perceptual criterion takes into account the masking effect, calculated on the original signal, in order to minimize the audible quantization noise. The bit allocation spectrally shapes the noise to maintain it below the masking curve. First the Noise to Mask Ratio (*NMR*) is computed for each band *k*:

$$NMR(k) = 10\log_{10}\frac{\sigma_q^2\left(k, b(k)\right)}{\tau(k)} \tag{2.43}$$

where $\sigma_q^2\left(k, b(k)\right)$ is the power of the quantization noise with $b(k)$ bits and $\tau(k)$ is the masking threshold for the band *k*. The quantization noise is completely masked as long as the *NMR* (in dB) is negative for all the bands. However, for limited bit budgets, the masking of quantization noise cannot be completely achieved and the *NMR* is positive for some of the bands indicating that some degradation should be perceived.
In order to solve the problem of the minimization of this distortion, the perceptual bit allocation is expressed as a constrained optimization problem. Thus, the problem consists in minimizing the total distortion which is e.g. defined as the sum of *NMR* given the bit budget *B* and the quantizers *Q*. Hence, the bit allocation algorithm can simply be described as the com-

putation of the number of allocated bits per band ($b(k)$ with $0 \le k \le QuantizerBand - 1$) which minimizes the total distortion:

$$\sum_{k=0}^{QuantizerBand-1} NMR(k) \tag{2.44}$$

under the bit budget constraint:

$$\sum_{k=0}^{QuantizerBand-1} b(k) = B, \text{ and } b(k) \in Q \tag{2.45}$$

where B is the total number of available bits (bit budget), Q represents the available quantizers and *QuantizerBand* is the number of bands.
Here, high-resolution scalar quantization followed by an entropy coding is assumed. Under this hypothesis, the quantization noise power in the band k quantized with $b(k)$ bits per sample is given by:

$$\sigma_q^2\left(k, b(k)\right) = \varepsilon^2 2^{-2b(k)} \sigma_X^2(k) \tag{2.46}$$

where $\sigma_X^2(k)$ is the sub-band signal power in band k and ε^2 is a constant representing the performance of the quantizer (entropy coder pair). It should be noted that under the hypothesis of using an entropy coder, equation (2.39) can replace (2.46). Using the Lagrange multiplier, the optimal solution is written:

$$b_{opt}(k) = \max\left(\frac{1}{2}\log_2\left[\frac{\sigma_X^2(k)}{\tau(k)}\right] + C, 0\right) = \max\left(\frac{SMR(k)_{dB}}{6.02} + C, 0\right) \tag{2.47}$$

where $SMR(k)$ is the Signal to Mask Ratio in decibels and the constant C is defined by:

$$C = \frac{B}{M} - \frac{1}{2M}\sum_{k=0}^{M-1} \log_2\left[\frac{\sigma_X^2(k)}{\tau(k)}\right] \tag{2.48}$$

From the equation (2.47) and (2.48), the power of the quantization noise is obtained using the following equation:

$$\sigma_q^2\left(k, b_{opt}(k)\right) = \tau(k)\varepsilon^2 2^{-2C} \tag{2.49}$$

In that bit allocation scenario, the Noise to Mask Ratio is consequently constant over all the frequency bands. The quantization noise is then shaped such as to be parallel to the masking curve. Based on these results, most audio encoders proceed in an iterative fashion: the most demanding

sub-band in terms of *NMR* is given additional bits and the procedure is repeated until the bit budget is used.

The main components of perceptual audio coding have been introduced. It has been shown that the transform coding relies on a frequency representation allowing a better energy concentration which is exploited to better allocate the bits where it is perceptually more important. The audio quality obtained with a perceptual audio coder is limited at low bit rate, due to the fact that the bit budget is not sufficient to correctly encode the complete spectrum. Hence, to overcome this problem, the audio bandwidth is usually limited to concentrate the bit rate in the low frequency range.

2.3 Filter banks for parametric audio coding tools

In this section, some "parametric tools" are presented. They have been introduced to overcome the quality degradation which is usually experienced with perceptual audio codecs at low bit rates. These tools do not aim at achieving faithful reconstruction of the audio signal but they provide a perceptually relevant synthesis of the audio content. Two types of parametric tools are presented.
In section 2.3.1, we introduce the bandwidth extension tools. Classical audio coding reduces the audio bandwidth to maintain a reasonable perceived quality and to limit degradations (at lower bit rates the quantization noise grows rapidly and some bands are not allocated). Hence, the input signal is low pass filtered to limit the audio bandwidth to be encoded. Bandwidth extension has been introduced to overcome this problem [Dietz 02]. It is based on the similarity of the frequency content between low and high frequency parts of the audio spectrum. The method exploits signal redundancy in the spectral domain and uses the lower band components to synthesize the higher band components.
A second type of parametric tool has been developed to transmit stereo or multichannel audio signals at low bit rates. It is presented in section 2.3.2. Classical speech or audio coding schemes require almost doubling the bit rate to encode a stereo signal. To achieve good performance with multichannel audio signals, the same rule applies and classical audio coders rely on an almost linear increase of the necessary bit rate with the number of channels. Indeed, each channel requires the same bit rate to be independently encoded. To improve the compression efficiency, some tools have been proposed and used in the stereo and multichannel audio coding schemes [Herre 92][Johnston 92]. A well established stereo tool is the intensity stereo method that merges the spectrum of the two channels in high frequency and transmits a small amount of side information on intensity information in the corresponding sub-bands to pan the stereo image [Herre

94]. However, those tools are really efficient for intermediate bit rates and offer only a limited efficiency improvement at low bit rates.
In the last decade, parametric representations of stereo and multichannel audio signals have been investigated and efficient models have been developed. Those models rely on the spatial perception of the human auditory system and are based on the coding of spatial cues which are transmitted at a very limited bit rate together with a down-mix representation of the stereo or multichannel audio signals [Faller 02, Breebaart 04]. Parametric stereo coding can be seen as an extension of the Intensity Stereo method by taking into account the spatial perception more precisely.

2.3.1 Bandwidth extension

Several bandwidth extension solutions have been developed to address the problem of low bit rate speech and audio coding. By using those methods, one can reduce the frequency bandwidth which is encoded by a traditional audio encoder and then have a better quality/efficiency. Moreover, the high frequency components are encoded with a very small amount of information leading to a larger bandwidth compared to traditional audio coder. In this paragraph, we present two solutions which have competed for the standardization of a bandwidth extension tool in MPEG-4. The PAT (Perceptual Audio Transposition) and SBR (Spectral Band Replication) are based on the same modules: extension of the spectral fine structure and coding of the high frequency spectral envelope. Both components represent the main contribution to the perceived audio quality improvement. However, some small differences between the bandwidth extension methods provide some further improvements with critical audio signals, such as non-harmonic high frequency tonal signals.

2.3.1.1 Perceptual Audio Transposition

The PAT codec has been associated with Code Excited Linear Prediction (CELP) speech codec and transform audio coding. It has been extensively tested with the ITU-T G.729 speech codec and the MPEG Advanced Audio Coding (AAC) as low band core coder [Philippe 01] in the MPEG standardization competition. The block diagram of the PAT encoder and decoder are described on Figure 16.

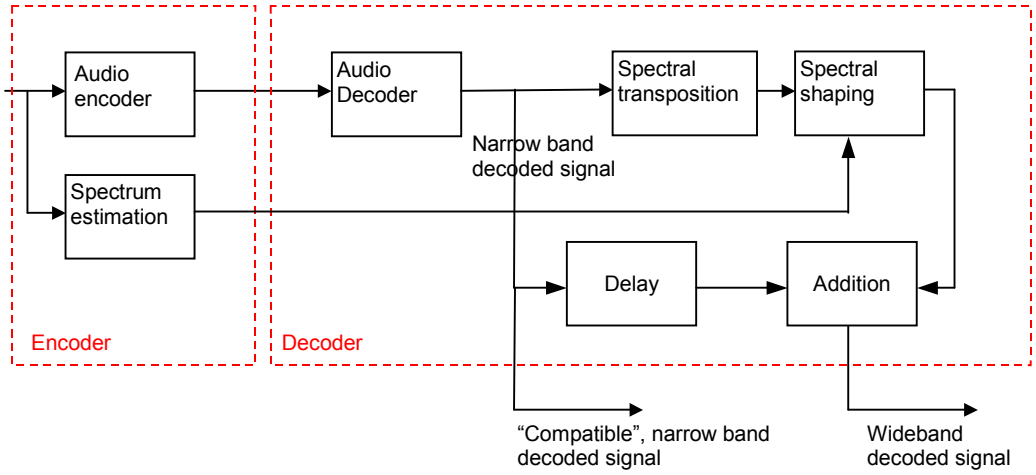


Figure 16 – Principle of the PAT codec

At the encoder side, the audio signal is separated in the low frequency part, which is fed into the audio core encoder (ITU-T G.729 or MPEG AAC), and the high frequency part which is encoded by the bandwidth extension encoder. The PAT encoder uses the complete spectrum to efficiently detect the redundancy between the low and high frequency parts. As described in [Collen 02], a transient detection module is used to adapt and more precisely reduce the frequency analysis window length in case an attack is present in the current frame. A second detection module is used to determine if the input signal is harmonic. The harmonic detection is based on a spectral analysis which consists in finding the peaks in the Discrete Fourier Transform domain of the high frequency signal. It uses a similar method to the one used in the MPEG-1 psychoacoustic model 1 [ISO 92]. Depending on the detected input signal, the encoder estimates the necessary spectral transposition which has to be done and encodes the high frequency spectral envelope.

The two main components of the bandwidth extension decoder are the spectral fine structure synthesis and the spectral envelope adjustment. The fine structure is generated by translation of the low frequency spectrum obtained from the core decoder. The translation uses a frequency-reversed version of the low frequency spectrum in adjacent bands in order to ensure the continuity in the high frequency spectrum. The DFT of the decoded core signal is first computed and the spectral fine structure module operates in the DFT domain by the translation of DFT coefficients. For harmonic signals, it has been noticed that breaking the harmonicity at the translated band boundary generates some perceived dissonances. In order to avoid those artefacts, an attenuation of the transition frequency band (the first 200 Hz) of the translated band in case of harmonic signal allows to attenu-

ate the tonal components which generate the perceived buzzy noise. This buzzy noise is created by the introduction of one harmonic in the translated band near the last harmonic of the previous band. This effect breaks the harmonicity and is perceived as a very annoying artefact. In case of harmonic signals, a whitening filter is also applied to the high frequency bands in order to control the tonal/noise ratio (which is higher in the lower frequency range than in the higher frequencies). For instance, with speech signals, the whitening filter attenuates the harmonic components in high frequencies to avoid over-voicing effect. Finally, the spectral fine structure is adjusted using the decoded spectral envelope. Two methods have been investigated during the PAT development. The first one uses a Linear Predictive Coding (LPC) filter to represent the spectral envelope. This method is efficient for speech signals, but generates some artefacts for music signals, and more specifically for highly harmonic signals. The second method, which has been selected in the final version of the PAT, describes the spectral envelope with scale factors which are computed at the encoder and applied at the decoder on the DFT spectrum. The PAT has been successfully tested with G.729 and AAC codec with a bit rate around 2 kbit/s for the bandwidth extension module [Philippe 01].
The PAT has demonstrated the benefit of an adaptive time-frequency representation. Indeed, the PAT is based on a DFT with variable length depending on the audio input characteristic. Shorter DFT size is selected when a transient is detected.

2.3.1.2 Spectral Band Replication

The Spectral Band Replication (SBR) [Dietz 02] has been selected as bandwidth extension tool in the MPEG-4 audio standard [ISO 03]. The SBR is jointly used with a conventional audio codec. In MPEG-4, the association with the Advanced Audio Coding (AAC) has been extensively tested and finally standardized as High Efficiency AAC (HE-AAC). The block diagram of the initial SBR decoder is illustrated on Figure 17. The SBR usually operates at a bit rate of around 2 kbit/s.
The SBR technique is based on a 64 sub-bands complex-exponential modulated filter bank. The high frequency spectral fine structure is obtained in the HF Generator by the translation of low frequency sub-bands obtained by the core decoder. The low frequency sub-bands are eventually whitened using filters with a fixed order (2). The spectral envelope is transmitted in the form of scale factors, which are first calculated based on group of sub-bands, and then quantized and Huffman coded. Multiple spectral envelopes can be encoded for shorter sub-frames within each frame in order to increase the temporal resolution. At the decoder side, the generated high frequency fine structure is then scaled in the envelope adjuster. In addition, the harmonic extension module allows to transmit and synthesize additional

tonal components in the filter bank. This feature is particularly useful when no tonal component is present in the low frequency spectrum of the core decoder (bell sounds for example exhibit a high tonal behaviour in the higher frequencies with no reference in the lower spectrum). A noise generator can correct the noise level in the high frequency sub-bands.

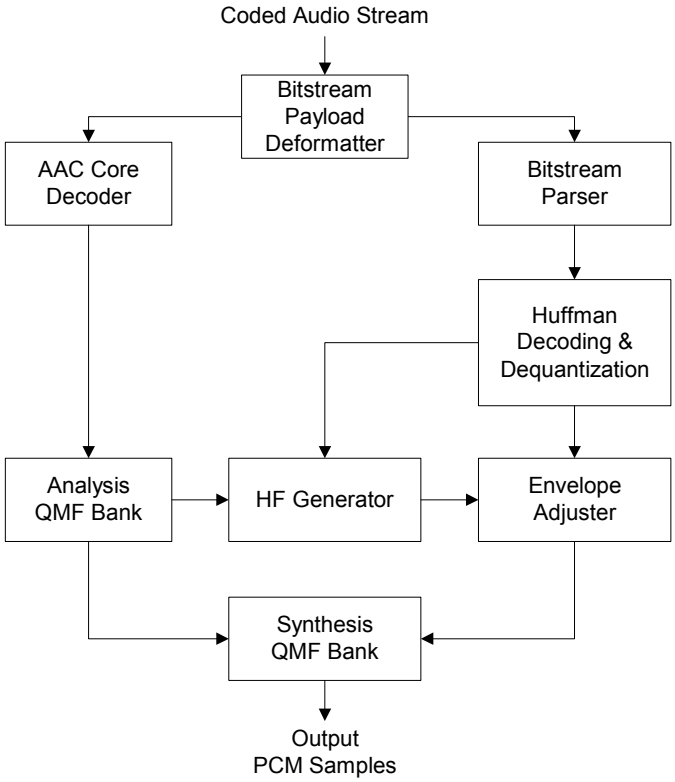


Figure 17 – HE-AAC decoder block diagram

An enhanced version of the SBR (eSBR) has been more recently developed in the context of the Unified Speech and Audio Coding (USAC) standardization [Neuendorf 09]. Compared to the initial SBR, this enhanced version allows to have a more flexible cross-over frequency between the core and the extended high frequency bands. The new bandwidth extension module can operate with frequency ratio 1/4 or 3/8 in addition to the traditional ratio 1/2 and 1/1 which were initially the only possibilities. Those ratios represent the sampling frequency ratio between the core encoder/decoder and the complete system with SBR. There is also the possibility to transmit more spectral envelopes per frame for a better temporal resolution. A predictive vector coding scheme of the spectral envelope [Chinen 11] has also been adopted for very low bit rates. The spectral envelope quantization uses a prediction of the high frequency spectral envelope using the low frequency band transmitted with the core codec. It has been shown that this module improves the coding efficiency of the SBR for very low bit rate and has demonstrated a quality improvement for speech signals. Finally, an im-

proved harmonic frequency reconstruction module has been integrated based on the harmonic transposer described in [Nagel 09].

The two bandwidth extension tools, which have been introduced in section 2.3.1, are based on two different frequency representations. The PAT uses an adaptive DFT depending on the transient characteristic of the input signal, while the SBR is based on a complex modulated filter bank using a small number of frequency components. These two approaches offer different trade-offs in terms of filter bank adaptability and selectivity.

2.3.2 Parametric stereo

The binaural cue coding (BCC) has been introduced in [Faller 02, Faller 04] to efficiently encode the stereo signals with a reduced bit rate. It is based on the representation of the stereo signals as monaural signal plus auxiliary spatial parameters which describe the stereo image, as shown on Figure 18. This method uses the spatial perception to efficiently reconstruct a stereo signal using intensity and time panning effect. Indeed, the inter-channel level differences (ICLD), the inter-channel phase differences (ICPD) or inter-channel time differences (ICTD) and the inter-channel coherence (ICC) describe the perceptually relevant spatial cues [Blauert 97].

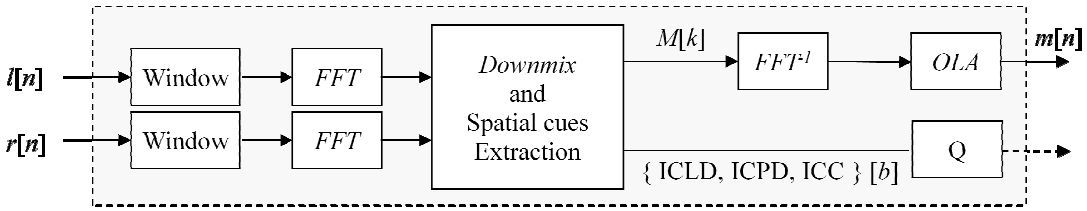


Figure 18 – BCC encoder block diagram

An alternative implementation, named Parametric Stereo (PS) [Breebaart 04, Breebaart 05a], has been developed in the context of MPEG standardization and has been associated with the SBR in HE-AACv2. The PS parameters extraction and synthesis modules operate in the complex-exponential modulated filter bank domain (the same as SBR). In order to achieve a finer frequency resolution in the low frequency sub-bands, an additional stage of band-pass filtering is used to approximate an ERB filter bank.
A similar parametric multichannel coding tool has been developed by MPEG targeting mainly 5.1 format compression, and has been extended to support stereo, 7.1 and alternative format with eventually more channels [Breebaart 05b, Herre 08]. Alternative stereo coding schemes, based on Principal Component Analysis (PCA) or Karhunen-Loève transform (KLT) have also been proposed and it has been shown that those approaches can offer similar performance to the binaural cue coding [Briand 06a, 06b].

2.3.3 Complex filter bank for parametric audio coding tools

As explained in the previous paragraphs, the parametric audio coding tools rely on complex transforms (DFT) or complex filter banks. Those tools can be seen as equalizers which control the spectral envelope. For this purpose, filter banks are the most relevant signal processing tool as fast algorithms are available for practical implementation. However, this specific equalization application requires a limitation of the aliasing effect between sub-bands and at the same time, there is no need to keep the critical sampling as the sub-band coefficients are not directly quantized and transmitted, but just locally processed, at the receiver.

In this paragraph, the main advantages of the complex exponential modulated filter banks are highlighted.
The cosine modulated filter banks were introduced in section 2.1.3.4. The corresponding analysis filter bank produces real-value sub-band samples. The sub-band samples are then decimated by a factor equal to the number of sub-bands in order to obtain a critically sampled system. With traditional real valued filter bank, each filter has two pass-bands, one in the positive frequency range and the corresponding sub-band in the negative frequency range. With a maximally decimated filter bank, it can be shown that the main alias terms are generated by the overlap of the negative and positive frequency bands with their frequency modulated versions due to the decimation.

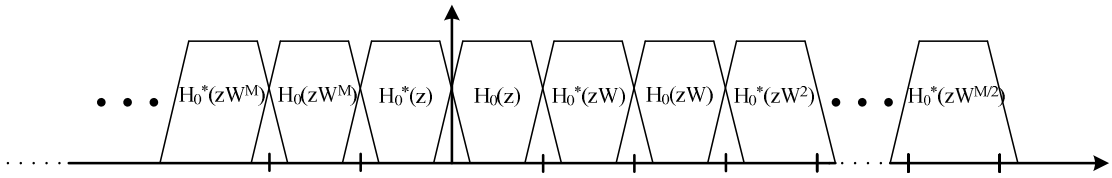


Figure 19 – Illustration of aliasing terms generated by negative and positive frequency bands in real valued filter bank

Figure 19 illustrates the aliasing between the positive and negative frequency band corresponding to the first band-pass filter and their corresponding frequency shifted versions. Indeed, at the boundary of each band, there is an overlapping region with the neighbouring bands (being the frequency shifted bands).
It has been shown that Perfect Reconstruction (PR) can be obtained at the cost of some constraints on the modulation function which ensure that the alias terms are cancelled. Even if the cosine modulated filter banks offer very effective implementation with fast algorithms, they are not completely suitable for the parametric audio coding tools due to the strong aliasing components in case of simple equalization processing. As shown in the

previous paragraphs, the parametric tools usually rely on spectral envelope adjuster which behaves like an equalizer. In this section, the complex-exponential modulated filter bank is introduced. Then, the main differences to the cosine modulated filter bank are illustrated and the behaviour of the complex filter bank for equalization-like sub-band processing is presented.

2.3.3.1 Complex-exponential modulated filter bank

The complex-exponential modulated filter banks are defined by extending the cosine modulation to complex-exponential modulation. The analysis and synthesis filters can be expressed as:

$$\begin{aligned} h_k(n) &= h(n)\exp\left(j\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)+\theta_k\right), 0\le k\le M-1 \\ f_k(n) &= h(n)\exp\left(j\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)-\theta_k\right), 0\le k\le M-1 \end{aligned} \tag{2.50}$$

where $h(n)$ is the impulse response of a prototype filter of length L and $0\le n\le L-1$. In order to keep the PR property, the same constraints on θk can be applied separately on the cosine and sine modulated parts.
As opposed to the cosine modulated filter bank, the complex-exponential modulated filter bank has only one pass-band in the positive frequency range as illustrated in Figure 20.

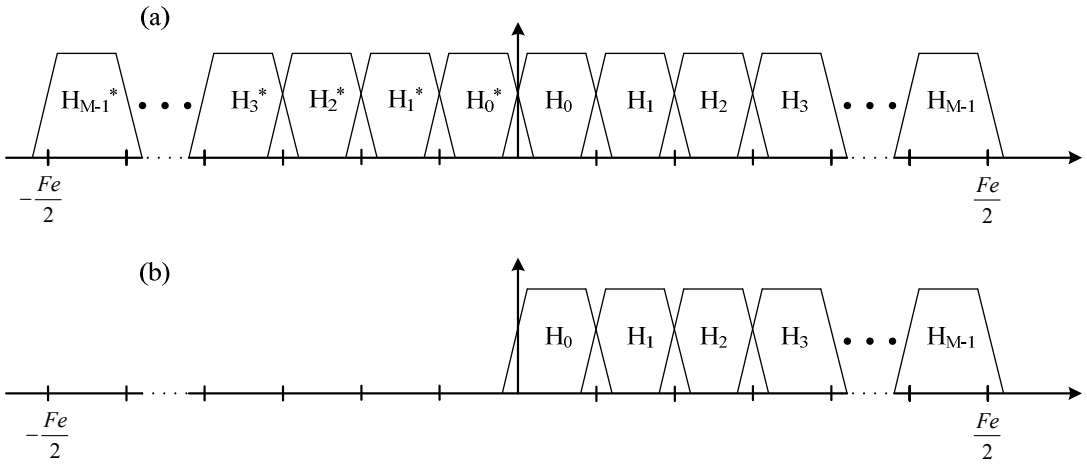


Figure 20 – Illustration of frequency bands in cosine modulated filter bank (a), and complex modulated filter bank (b)

Based on this feature, it can be shown that the complex-exponential modulated filter banks are free of the main alias terms coming from the overlap of negative and positive bands with their shifted versions. Hence, the aliasing cancellation constraint described in [Vaidyanathan 93] is obsolete. Thus, the analysis and synthesis filters can be simplified and written as:

$$h_k(n) = h(n)\exp\left(j\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)\right), 0 \le k \le M-1$$
$$f_k(n) = h(n)\exp\left(j\frac{\pi}{M}\left(n-\frac{L-1}{2}\right)\left(k+\frac{1}{2}\right)\right), 0 \le k \le M-1 \quad (2.51)$$

This gives an additional degree of freedom in the selection of the modulation in the context of complex filter banks. Of course, different modulations can be selected (different θ_k) without affecting the performance of the filter bank for its application in parametric tools as described in the previous sections.

2.3.3.2 Characteristics of complex filter bank

In order to illustrate the limitation of the alias components, a similar example as the one shown in [Ekstrand 02] is presented. This example is based on the MPEG-1 Layer II filter bank, which is a 32-band cosine modulated filter bank. An input signal which is a harmonic series with a fundamental frequency of 1200 Hz is fed into the analysis filter banks (real cosine and complex-exponential modulated filter banks), the sub-band signals are then equalized using the equalizing curves (dash line) illustrated in Figures 21 and 22. After equalization, the sub-band samples are fed into the synthesis filter banks to reconstruct the equalized signal. The equalized signal obtained from the cosine modulated filter bank is shown on Figure 21, and the one from the complex-exponential modulated filter bank is illustrated on Figure 22.

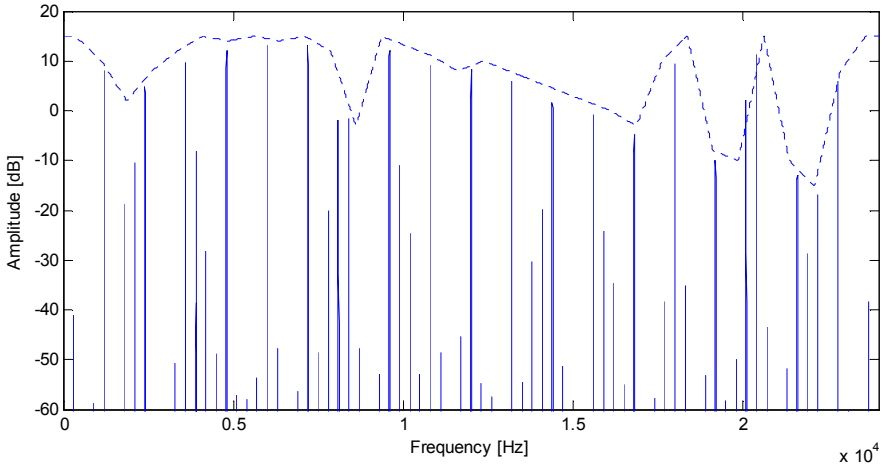


Figure 21 – Equalization using cosine modulated filter bank

It can be seen that the aliasing components are seriously corrupting the spectrum of the equalized signal in case of cosine modulated filter bank. For instance, around 8.4 kHz, the generated aliasing component has the

same energy as the desired signal which will lead to audible artefacts. The position and energy of those aliasing components obviously depend on the fundamental frequency, the equalizing curve and the characteristics of the prototype filter (stop-band attenuation).

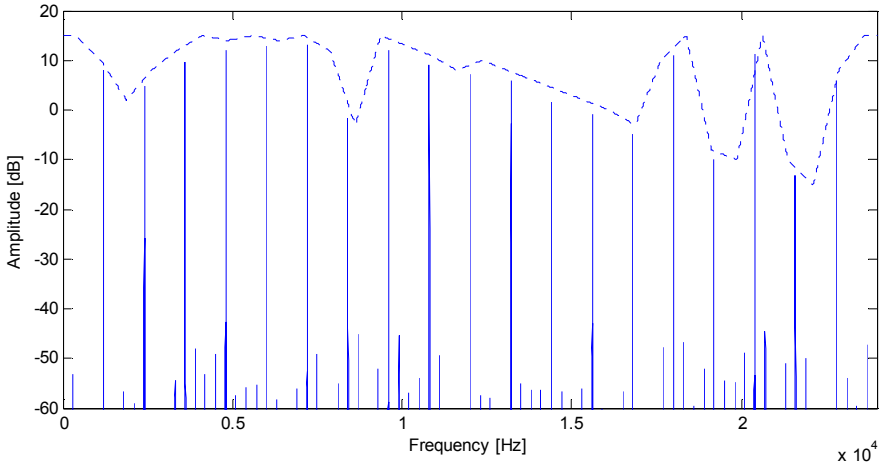


Figure 22 – Equalization using complex-exponential modulated filter bank

In the context of complex-exponential modulated filter bank, using exactly the same prototype filter and the same equalizing curve, the level of aliasing components become very low and does not harm the quality of the processed signal.
To better illustrate graphically the effect of the alias terms, the composite alias component matrix ($\mathbf{U}(z)$) as defined in equation (2.14) in section 2.1.3.2 is represented.

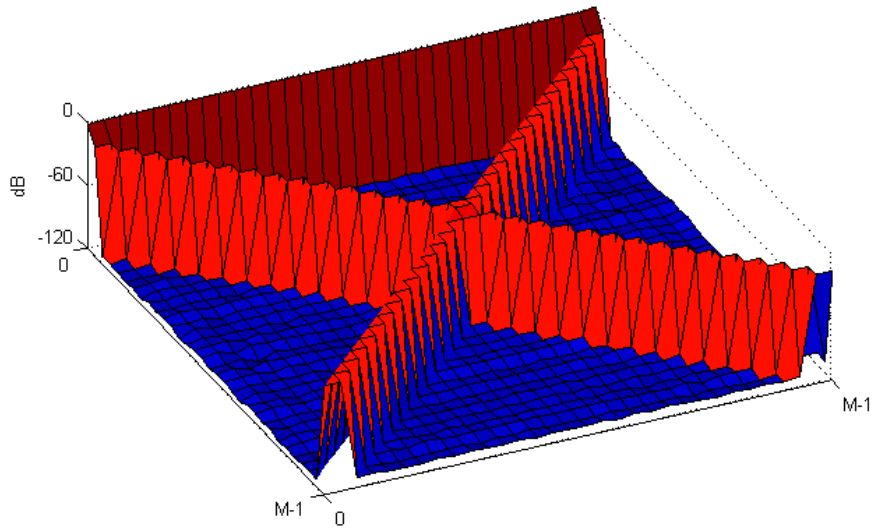


Figure 23 – Magnitude of composite alias component matrix for cosine modulated filter bank

Figure 23 represents the matrix $\mathbf{U}(z)$ for the cosine modulated filter bank. The dominant terms are given by the first row, which represents the trans-

fer function of the filter bank. The diagonals, which consist in the main alias terms representing the overlap of the filters with their closest frequency shifted versions. Indeed, according to the matrix $\mathbf{U}(z)$, the synthesis filtering operation $F_k(z)$ introduces some aliasing due to the aliasing between the current band of $F_k(z)$ and the frequency shifted band of the analysis filter $H_k(zW^l)$. As illustrated in Figure 19, those main alias terms are generated from the overlap between a positive sub-band with the frequency modulated version of the corresponding negative sub-band, and reciprocally. It can be seen on Figure 19 that applying a synthesis filter, which would have the same frequency response as $H_0(z)$ to the decimated signal, would introduce some frequency aliasing coming from $H_0^*(z)$ and $H_0^*(zW)$. This can be extended to all the bands k leading to the two diagonals of Figure 23.

As opposed to the real filter bank, using a complex-exponential modulated filter bank, the dominant terms of the composite alias component matrix lie only on the first row. The main alias terms are completely absent in that case as shown on Figure 24.

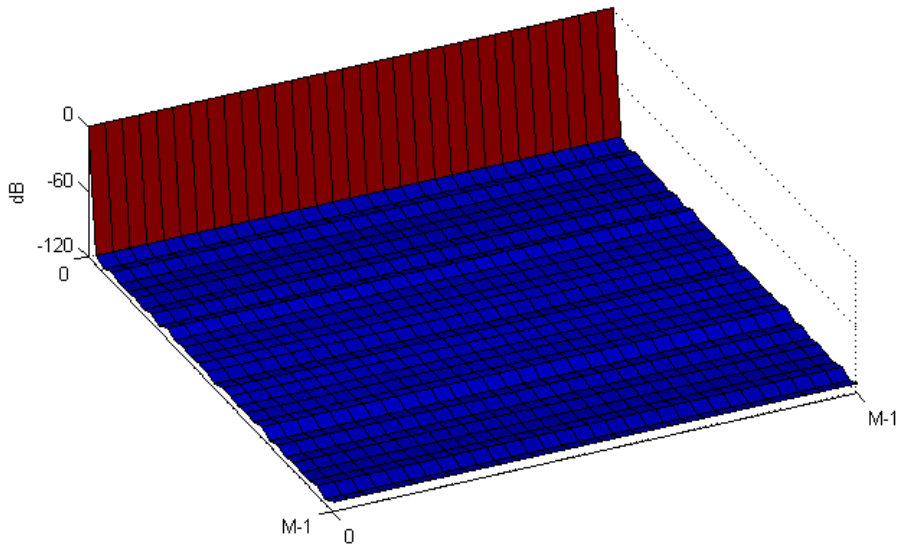


Figure 24 – Magnitude of composite alias component matrix for complex-exponential modulated filter bank

This aliasing reduction can also be quantified using the estimation of the energy of the total aliasing e_a which is defined as:

$$e_a = \frac{1}{8\pi M^2}\int_{-\pi}^{\pi}\left|A_{M/2}\left(e^{j\omega}\right)+A^*_{M/2}\left(e^{-j\omega}\right)\right|^2 d\omega + \frac{1}{8\pi M^2}\sum_{l=1}^{M/2-1}\int_{-\pi}^{\pi}\left(\left|A_l\left(e^{j\omega}\right)+A^*_{M-l}\left(e^{-j\omega}\right)\right|^2+\left|A_{M-l}\left(e^{j\omega}\right)+A^*_l\left(e^{-j\omega}\right)\right|^2\right)d\omega \quad (2.52)$$

Based on this measure, the aliasing rejection of the filter banks illustrated at the beginning of this paragraph is estimated to 12.4 dB for the cosine modulated filter bank and to 97.4 dB for the complex modulated filter bank.

In order to specifically design prototype adapted for the complex-exponential modulated filter bank, it has been proposed in [Ekstrand 01] to define a prototype design algorithm based on the alias term minimization (ATM). This algorithm is adapted to design prototype for equalizer-like applications. During the optimization process, a random equalization curve is applied to the sub-bands. The energy of the total aliasing is then calculated based on equation (2.51) using the filters multiplied by the corresponding gain of the equalization curve. The error energy of the transfer function is also calculated and the optimization is performed minimizing a composite objective function which is defined as a weighted sum of the error energy of the transfer function and the energy of the total aliasing. The author proposed to use the Downhill Simplex Method (also known as Nelder–Mead method). This standard non linear optimization algorithm demonstrated an improvement of the rejection of total aliasing by about 20 dB in a 32 bands complex-exponential filter bank as the one used in the example provided in Figures 22 and 24.

2.4 Conclusion

In this Chapter, the perceptual audio coding schemes were introduced with all the components which need to be taken into account in the development of such codec. The first goal of this thesis is to investigate potential optimization of filter banks for low delay audio coding applications. It has been shown that the filter banks or transforms play an important role in the audio coding. The cosine and complex exponential modulated filter banks have been reviewed. The first one has been introduced for perceptual audio coding, first with PQMF and later with the perfect reconstruction which is a key element of a filter bank. The complex version has been studied for the parametric coding tool such as bandwidth extension or parametric stereo. It does not necessarily achieve the perfect reconstruction and it is not maximally decimated, but it offers the property of reducing the impact of aliasing for sub-band audio processing.

Thus, the development of modern low delay perceptual audio coding schemes must carefully consider the transform design. This work focuses

on MDCT based transform for low delay applications, but it can be easily extended to complex transform to be used in low delay parametric audio coding tools.

Chapter 3

Transform for audio coding

In Chapter 2, the bases of perceptual audio coding have been introduced. The filter bank is one of the main components of this audio coding model and it has been widely studied. The main purpose of this thesis is to contribute to the definition of transforms for low delay perceptual audio coding. This Chapter is composed of two parts: on one hand the introduction of block transforms for perceptual audio coding with the existing solutions for low delay audio coding and on the other hand the newly developed transforms for low delay audio coding.
In this Chapter, the block transforms which have demonstrated their efficiency for perceptual audio coding are presented. They have led to the alternative name: transform audio coding. This Chapter is organized as follows. First the Modified Discrete Cosine Transform (MDCT) is presented. It is probably the most widespread transform in nowadays audio codecs. Then, its generalization called Extended Lapped Transform (ELT) is presented. Subsequently, the low delay transform allowing to reduce the delay associated with the transform itself is presented. Finally, block switching, or window switching, is presented as it is an important component of transform coding for the adaptation of the time/frequency resolution to the input signal characteristics.

3.1 MDCT

The block transforms are widely used for digital audio signal processing. The Discrete Fourier Transform (DFT) is commonly used in signal processing as efficient implementations have been developed, i.e. Fast Fourier Transform (FFT), but is usually not convenient for coding applications. For an input block of N samples, the DFT generates N complex spectral values in the frequency range $[0,2\pi[$. For an input signal with real values, the spectral values follow the Hermitian symmetry allowing to reconstruct the

input signal with only half the spectral components. However, this transform is critically sampled only when it is used without overlapping part of the input samples for consecutive transform blocks, leading to block artefacts during the decoding due to the discontinuities between blocks.
Block transforms with overlapping have been used in the early ages of audio coding, but suffered form the absence of critical sampling: there were more samples in the transformed domain than in the time domain.
In order to overcome this drawback, several overlapping block transforms have been defined. Among them, the MDCT is certainly the most used in perceptual audio coding.

3.1.1 MDCT definition

The MDCT is a cosine modulated perfect reconstruction filter bank based on time domain aliasing cancellation (TDAC). Credit for this filter bank is often given to Princen and Bradley [Princen 86], where it is referred to as Time Domain Aliasing Cancellation (TDAC) filter banks. The oddly-stacked TDAC was later recognized as a specific case of the more general class of Modulated Lapped Transforms (MLT) [Malvar 92b] and connected to the Perfect Reconstruction Modulated Filter bank theory (PRMF). The key argument for the MDCT is that it is a critically sampled and overlapping transform. As introduced in 2.1.3, a transform, which is critically sampled, has an equivalent number of spectral coefficients and time-domain samples, which is crucial for coding efficiency. Moreover, fast algorithms have been developed and have proved to offer very efficient implementation of such transform [Malvar 90, Duhamel 91]. Those fast algorithms have also been one of the key factors of the success of the MDCT in audio coding.

3.1.1.1 Definition

The MDCT maps a discrete signal segment (or frame) $x_{t,n}= x(n+tM)$, $0 \le n \le 2M\text{-}1$ into M frequency components $X_{t,k}$ at frame t using the following equation:

$$X_{t,k} = \sum_{n=0}^{2M-1} x_{t,n} p_{k,n} \tag{3.1}$$

for $0 \le k \le M\text{-}1$, with:

$$\begin{aligned} p_{k,n} &= w(n) c_{k,n}, \\ c_{k,n} &= \sqrt{\frac{2}{M}} \cos\left(\frac{\pi}{M} \left(n + \frac{M+1}{2} \right) \left(k + \frac{1}{2} \right) \right), \end{aligned} \tag{3.2}$$

where $p_{k,n}$ are the basis functions for the direct and inverse transforms, and $w(n)$ denotes a weighting function acting as the analysis and synthesis win-

dow. This window is equivalent to the low pass prototype filter in the filter bank terminology.
The MDCT considers overlapping frames composed of M past samples and M new incoming samples resulting in M frequency components; hence it corresponds to a maximally decimated filter bank.
In order to recover the original sequence x, an inverse transform is applied according to:

$$\tilde{x}_{t,n} = \sum_{k=0}^{M-1} X_{t,k}\, p_{k,n} \tag{3.3}$$

for $0 \le n \le 2M$-1. The $2M$ terms of $\tilde{x}_{t,n}$ are recovered using only M frequency components of $X_{t,k}$, and hence cannot exactly represent the original $x_{t,n}$ signal. The samples $\tilde{x}_{t,n}$ contain time aliasing terms, consisting particularly in unwanted components reversed in time. These aliasing terms are cancelled in the time domain using a combination of two consecutives frames (overlapped and added) such that:

$$\hat{x}_{t,n} = \tilde{x}_{t-1,n+M} + \tilde{x}_{t,n} \tag{3.4}$$

for $0 \le n \le M$-1. This gives the origin of the name Time Domain Aliasing Cancellation (TDAC).
To guarantee the perfect reconstruction (PR) property, as defined in [Princen 86], the window $w(n)$, which must be used for the forward and inverse transform, needs to satisfy the following conditions such that the time aliasing is cancelled:

$$\begin{cases} w(n) = w(2M-1-n) \\ w^2(n) + w^2(M+n) = 1 \end{cases}, \quad 0 \le n \le M-1 \tag{3.5}$$

Several windows have been defined in the literature; three common ones are given below. First, the sine window based on a sine arch is considered. This window satisfies the PR conditions and offers good frequency domain behaviour with good pass-band selectivity, characterized by a narrow main lobe for better discrimination of tonal components close to each other. The sine window is defined by:

$$w(n) = \sin\left[\left(n + \frac{1}{2}\right)\frac{\pi}{2M}\right], \quad 0 \le n \le 2M-1 \tag{3.6}$$

The second widely used window for MDCT based audio coding is the Kaiser-Bessel derived (KBD) window [Fielder 96]. It is constructed using the following formula:

$$w(n)=\begin{cases}\sqrt{\dfrac{\sum_{l=0}^{n} w_K(l)}{\sum_{l=0}^{M} w_K(l)}} & , 0 \le n \le M-1 \\ \sqrt{\dfrac{\sum_{l=n-M+1}^{M} w_K(l)}{\sum_{l=0}^{M} w_K(l)}} & , M \le n \le 2M-1\end{cases} \tag{3.7}$$

with $w_K(n)$ being the M+1 points Kaiser-Bessel window which is computed as:

$$w_K(n)=\frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{M}-1\right)^2}\right)}{I_0(\pi\alpha)} \tag{3.8}$$

where $I_0(x)$ is the 0th order modified Bessel function of the first kind given by:

$$I_0(x)=\sum_{k=0}^{\infty}\frac{\left(\frac{x^2}{4}\right)^k}{(k!)^2} \tag{3.9}$$

α, in equation (3.8), is a control parameter that determines the shape of the window and thus influences the associated time-frequency behaviour. A larger value of α leads to a larger main lobe and a better stop-band attenuation.

A third window, used in Low Delay - Advanced Audio Coding (LD-AAC) [Allamanche 99] is now presented. This window is characterized by two parts at the beginning and at the end that are set to zero. Due to the PR conditions defined in equation (3.5), the parts equal to zero impose that the central part of the window is equal to one. The transition parts are defined by a shorter sine window (first half for the transition between 0 and 1, and second part for the transition between 1 and 0). This transition part can also be defined with a KBD window. In LD-AAC, the size of the shorter window used for the definition corresponds to the size of a long window di-

vided by 8. The exact definition of the low overlap window is given in equation (3.10). It should be noted that the name “low overlap’ comes from the fact that the overlap region between two consecutive transforms is limited to a length M_s as opposed to the normal overlap size which is M.

$$w_{LOV}(n)=\begin{cases}\sin\left(\dfrac{\left(n-\dfrac{M-M_s}{2}+\dfrac{1}{2}\right)\pi}{2M_s}\right) & ,\dfrac{M-M_s}{2}\le n\le\dfrac{M+M_s}{2}-1\\ 1 & ,\dfrac{M+M_s}{2}\le n\le\dfrac{3M-M_s}{2}-1\\ \sin\left(\dfrac{\left(n-\dfrac{3M-3M_s}{2}+\dfrac{1}{2}\right)\pi}{2M_s}\right) & ,\dfrac{3M-M_s}{2}\le n\le\dfrac{3M+M_s}{2}-1\\ 0 & ,\text{otherwise in } 0\le n\le 2M-1\end{cases} \quad (3.10)$$

with e.g. $M_s = M/8$.
In order to compare the time and frequency characteristics of each window, Figure 25 represents the impulse and magnitude responses. On the left side, the impulse responses show that the sine window offers rather low time selectivity, but the right side which represents the magnitude response shows good pass-band selectivity. On the contrary, at the bottom, the low overlap window is defined with a reduced transition part which gives the highest temporal localization, but with very poor frequency selectivity and stop-band attenuation. The KBD window, which is shown in the middle of Figure 25, presents a good trade-off with good time selectivity as the overlap is rather low and very good stop-band attenuation. However, the frequency selectivity is worse than the sine window as the main and second lobes are rather large. The trade-off between temporal and frequency performance can be adjusted with parameter α. The low overlap window, which is the last window on Figure 25, provides a better time selectivity, but the frequency response offers poor performance in terms of selectivity and stop-band attenuation. The use of the low overlap window is then restricted to applications which require a delay reduction of the transform.

The delay associated with the MDCT can be defined as the number of samples between the first sample of the synthesis window and the last sample of the analysis window minus one. For symmetric window, the delay is simply defined by the number of non-zero coefficients minus one. For the window illustrated on Figure 25, the sine and KBD windows have an associated delay of $2M$-1 samples, whereas the delay of the low overlap window is only M+M_s.

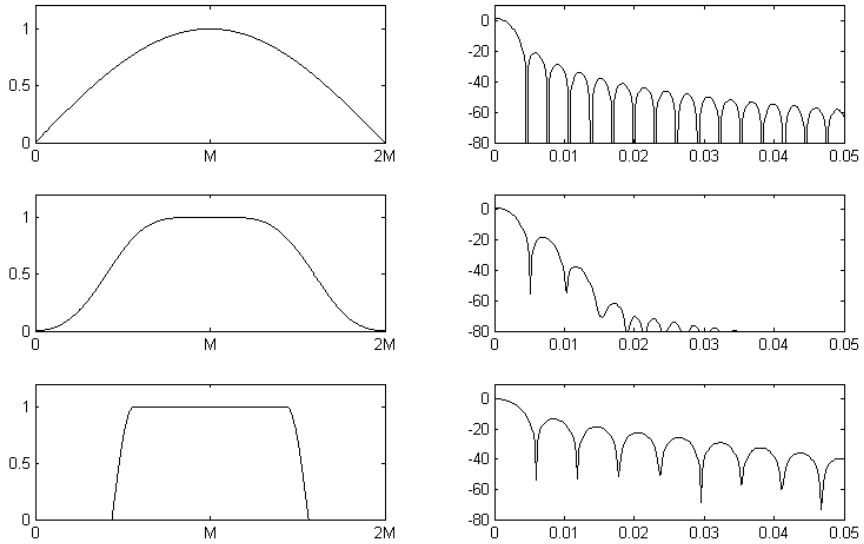


Figure 25 – Impulse and magnitude response for 1) sine window, 2) Kaiser-Bessel derived window and 3) low overlap window

3.1.1.2 Matrix notation

In this paragraph, the direct and inverse transforms of two consecutive frames are introduced using a matrix notation. This notation allows to easily represent the time aliasing introduced in the direct transform and spread in time with the inverse transform. The transform operation with a long window at time *t*-1 and *t* is presented to illustrate the reconstruction. Boldface letters indicate vectors and matrix with elements defined in 3.1.1.1. The symbols $\mathbf{I}_M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$ and $\mathbf{J}_M = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & \iddots & 0 \\ 0 & 1 & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$ denote the $M \times M$ identity and counter-identity (or anti-identity) matrixes. **diag** is an $M \times M$ diagonal matrix, and the operator ^{T} denotes the transpose operation.

At instant *t*-1, similar to equation (3.1) the direct transform is expressed by:

$$\mathbf{X}_{t-1}^{M} = \mathbf{P}\,\mathbf{x}_{t-1}^{2M} \tag{3.11}$$

where:

$$\mathbf{x}_t^{2M} = \left[x(tM), x(tM+1), ..., x(tM+2M-1)\right]^T \tag{3.12}$$

is the input buffer of length $2M$, $\mathbf{X}_{t-1}^{M}$ is the vector of length M with transform coefficients and the matrix $\mathbf{P}$ is the $2M \times M$ transform matrix given by equation (3.13).

$$\mathbf{P} = \mathbf{C}\ \mathbf{diag}(\mathbf{w}) = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,2M-1} \\ p_{1,0} & p_{1,1} & & \vdots \\ \vdots & & \ddots & \vdots \\ p_{M-1,0} & \cdots & \cdots & p_{M-1,2M-1} \end{bmatrix} \tag{3.13}$$

with $\boldsymbol{w} = [w(0),\ w(1),\ \ldots,\ w(2M-1)]^{\mathrm{T}}$ being the window vector of length $2M$ and $\mathbf{C}$ is the modulation matrix defined by:

$$\mathbf{C} = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,2M-1} \\ c_{1,0} & c_{1,1} & & \vdots \\ \vdots & & \ddots & \vdots \\ c_{M-1,0} & \cdots & \cdots & c_{M-1,2M-1} \end{bmatrix} \tag{3.14}$$

where the components $c_{k,n}$ of the modulation matrix $\mathbf{C}$ are given in equation (3.2). The windowing matrix **diag(w)** is a $2M \times 2M$ matrix containing zeros but on the main diagonal carrying the vector $\mathbf{w}$:

$$\mathbf{diag}(\mathbf{w}) = \begin{bmatrix} w(0) & & & 0 \\ & w(1) & & \\ & & \ddots & \\ 0 & & & w(2M-1) \end{bmatrix} \tag{3.15}$$

The corresponding inverse transform is defined by:

$$\tilde{\mathbf{x}}_{t-1}^{2M} = \mathbf{P}^{\mathrm{T}}\ \mathbf{X}_{t-1}^{M} \tag{3.16}$$

The PR condition can be derived by cascading the direct and inverse transforms:

$$\tilde{\mathbf{x}}_{t-1}^{2M} = \mathbf{P}^{\mathrm{T}}\mathbf{P}\ \mathbf{x}_{t-1}^{2M} = \mathbf{diag}(\mathbf{w})\ \mathbf{C}^{\mathrm{T}}\mathbf{C}\ \mathbf{diag}(\mathbf{w})\ \mathbf{x}_{t-1}^{2M} \tag{3.17}$$

Using the cosine orthogonal properties, one notices that:

$$\mathbf{C}^{\mathrm{T}}\mathbf{C} = \begin{bmatrix} \mathbf{I}_M - \mathbf{J}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M + \mathbf{J}_M \end{bmatrix} \tag{3.18}$$

Hence, using this notation for the combination of the modulation functions, the time aliasing in the direct and inverse transforms are clearly identified through the $\mathbf{J}_M$ matrixes.

$$\begin{aligned}\mathbf{P}^{\mathrm{T}}\mathbf{P} &= \mathbf{diag}(\mathbf{w})\begin{bmatrix}\mathbf{I}_M - \mathbf{J}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M + \mathbf{J}_M\end{bmatrix}\mathbf{diag}(\mathbf{w}) \\ \mathbf{P}^{\mathrm{T}}\mathbf{P} &= \begin{bmatrix}\mathbf{U_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U_1}\end{bmatrix}\end{aligned} \tag{3.19}$$

with:

$$\mathbf{U_0} = \mathbf{diag}(\mathbf{w}_0)\left[\mathbf{I}_M - \mathbf{J}_M\right]\mathbf{diag}(\mathbf{w}_0) \tag{3.20}$$

and:

$$\mathbf{U_1} = \mathbf{diag}(\mathbf{w}_M)\left[\mathbf{I}_M + \mathbf{J}_M\right]\mathbf{diag}(\mathbf{w}_M) \tag{3.21}$$

where $\mathbf{w}_0 = [w(0), w(1), \ldots, w(M - 1)]^{\mathrm{T}}$ and $\mathbf{w}_M = [w(M), \ldots, w(2M - 1)]^{\mathrm{T}}$ are two portions of length M of the window w. It is reminded that the analysis and synthesis windows are identical and symmetric according to the Princen and Bradley definition [Princen 86] in equations (3.5).
The reconstruction of the input signal is obtained through the identity part of the equation (3.19) and the time aliasing terms of MDCT are determined by the anti-identity part of the equation. Indeed, the anti-identity corresponds to a time-reversed version of the input samples.

For illustrative purpose, a distinct notation for the analysis (w_a) and synthesis (w_s) windows is now introduced. The following example presents the form of the matrix $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ for M=4 which can be expressed by:

$$\mathbf{P}^{\mathrm{T}}\mathbf{P} = \begin{bmatrix} w_a(0)w_s(0) & 0 & 0 & -w_a(3)w_s(0) & 0 & 0 & 0 & 0 \\ 0 & w_a(1)w_s(1) & -w_a(2)w_s(1) & 0 & 0 & 0 & 0 & 0 \\ 0 & -w_a(1)w_s(2) & w_a(2)w_s(2) & 0 & 0 & 0 & 0 & 0 \\ -w_a(0)w_s(3) & 0 & 0 & w_a(3)w_s(3) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_a(4)w_s(4) & 0 & 0 & w_a(7)w_s(4) \\ 0 & 0 & 0 & 0 & 0 & w_a(5)w_s(5) & w_a(6)w_s(5) & 0 \\ 0 & 0 & 0 & 0 & 0 & w_a(5)w_s(6) & w_a(6)w_s(6) & 0 \\ 0 & 0 & 0 & 0 & w_a(4)w_s(7) & 0 & 0 & w_a(7)w_s(7) \end{bmatrix} \tag{3.22}$$

On next page, the matrix representation of the application of the block transform MDCT to an input signal **x** representing the time signal over several frames (m frames) is provided. This matrix notation expresses the use of multiple overlapping window/MDCT of size $2M$ as given in [Wang 03]. The matrix **F** represents the transform operation which is used to derive $m \times M$ spectral coefficients from the $(m+1) \times M$ input samples. m is the number of consecutive frames of the input signal **x**. $(m+1)$ input frames are transformed to the frequency domain to generate m sets of spectral coefficients, which are then transformed back to the time domain to finally obtain $(m-1)$ alias free time samples. This general transform matrix is composed of several transform sub-matrixes **P** which are applied with an overlap of M samples. The corresponding inverse transform matrix is defined by $\mathbf{F}^{\mathbf{T}}$, and its size is $(m+1) \times mM$.

$$\mathbf{F} = \begin{bmatrix} \mathbf{P} & & & \\ & \mathbf{P} & \mathbf{0} & \\ & & \ddots & \\ & \mathbf{0} & & \mathbf{P} \\ & & & \mathbf{P} \end{bmatrix}_{\mathrm{mM}\times(\mathrm{m}+1)\mathrm{M}} \tag{3.23}$$

$$\mathbf{F} = \begin{bmatrix}
p_{0,0} & \cdots & p_{0,M-1} & p_{0,M} & \cdots & p_{0,2M-1} & & & & & & & & & & & & \\
p_{1,0} & \ddots & & & & \vdots & & & & & & & & & & & & \\
\vdots & & & & \ddots & \vdots & & \mathbf{0} & & \cdots & \cdots & \cdots & \cdots & \cdots & & & \mathbf{0} & \\
p_{M-1,0} & \cdots & p_{M-1,M-1} & p_{M-1,M} & \cdots & p_{M-1,2M-1} & & & & & & & & & & & & \\
 & & & p_{0,0} & \cdots & p_{0,M-1} & p_{0,M} & \cdots & p_{0,2M-1} & & & & & & & & \vdots & \\
 & & & p_{1,0} & \ddots & & & & \vdots & & & & & & & & \vdots & \\
 & \mathbf{0} & & \vdots & & & & \ddots & \vdots & & & & & & & & \vdots & \\
 & & & p_{M-1,0} & \cdots & p_{M-1,M-1} & p_{M-1,M} & \cdots & p_{M-1,2M-1} & & & & & & & & & \\
 & \vdots & & & & & & & & \ddots & & & & & & & \mathbf{0} & \\
 & \vdots & & & & & & & & & \ddots & & & & & & & \\
 & \vdots & & & & & & & & & & p_{0,0} & \cdots & p_{0,M-1} & p_{0,M} & \cdots & p_{0,2M-1} & \\
 & & & & & & & & & & & p_{1,0} & \ddots & & & & \vdots & \\
 & \mathbf{0} & & \cdots & \cdots & \cdots & \cdots & \cdots & & \mathbf{0} & & \vdots & & & & \ddots & \vdots & \\
 & & & & & & & & & & & p_{M-1,0} & \cdots & p_{M-1,M-1} & p_{M-1,M} & \cdots & p_{M-1,2M-1} &
\end{bmatrix}_{\mathrm{mM}\times(\mathrm{m}+1)\mathrm{M}} \tag{3.24}$$

The consecutive transformed input frames are then expressed as:

$$\mathbf{X}^{mM} = \mathbf{F}.\mathbf{x}^{(m+1)M} \tag{3.25}$$

where $\mathbf{x}^{(m+1)M}$ is the vector of input signal for m+1 frames. The consecutive application of the direct transform **P** is illustrated in Figure 26. Consecutive overlapping frames are windowed and then multiplied by the modulation matrix.

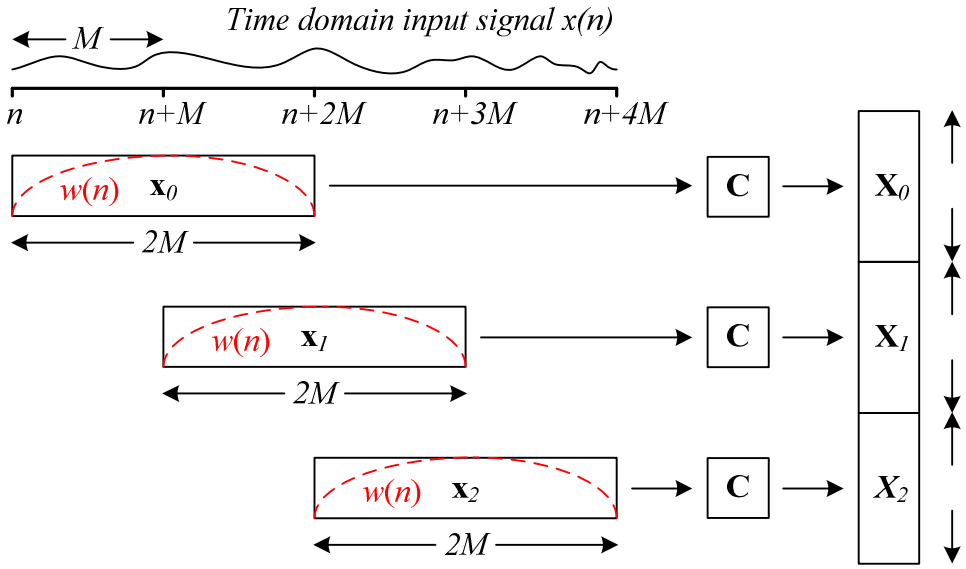


Figure 26 – Direct MDCT of multiple consecutive frames of input signal $x(n)$

The reconstructed signal, which is obtained after the direct and inverse transform operations, is then defined by:

$$\hat{\mathbf{x}}^{(m+1)M} = \mathbf{F}^{\mathrm{T}}.\mathbf{X}^{mM} = \mathbf{F}^{\mathrm{T}}.\mathbf{F}.\mathbf{x}^{(m+1)M} \tag{3.26}$$

Figure 27 visualizes the inverse MDCT operation of multiple consecutive frames. The MDCT coefficients are first multiplied by the modulation matrix $\mathbf{C}^{\mathrm{T}}$, leading to the inverse transformed time domain which contains the time aliasing terms and the signal is then reconstructed after the application of the synthesis window and overlap-add of two consecutive blocks.

The system satisfies the Perfect Reconstruction property if the matrix product $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ equals the identity matrix **I**. As illustrated in Figures 26 and 27, the matrixing $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ can be decomposed in windowing operations, direct and inverse modulations $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and finally overlap-add of consecutive blocks. The PR condition can be reduced to the reconstruction of a frame $\mathbf{x}_t$. Hence, the PR problem can be decomposed on a frame-by-frame basis and reduced to a simple constraint on the two parts of the $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ matrix, noted as $\mathbf{U_0}$ and $\mathbf{U_1}$ hereunder in equation (3.27).

By adding two consecutive frames, the reconstructed signal of length M is obtained by a simple overlap-add operation, the PR condition can then be reduced to the following equation:

$$\mathbf{U_0}+\mathbf{U_1}=\mathbf{I}_M \tag{3.27}$$

By replacing $\mathbf{U_0}$ and $\mathbf{U_1}$ defined in equation (3.20) and (3.21) into (3.27), it comes that:

$$\mathbf{diag}(\mathbf{w}_M)\mathbf{J}_M\mathbf{diag}(\mathbf{w}_M)-\mathbf{diag}(\mathbf{w}_0)\mathbf{J}_M\mathbf{diag}(\mathbf{w}_0)=\mathbf{0} \tag{3.28}$$

and

$$\mathbf{diag}(\mathbf{w}_0)\mathbf{diag}(\mathbf{w}_0)+\mathbf{diag}(\mathbf{w}_M)\mathbf{diag}(\mathbf{w}_M)=\mathbf{I}_M \tag{3.29}$$

Equations (3.5) can then be derived from (3.28) and (3.29) for symmetric windows.

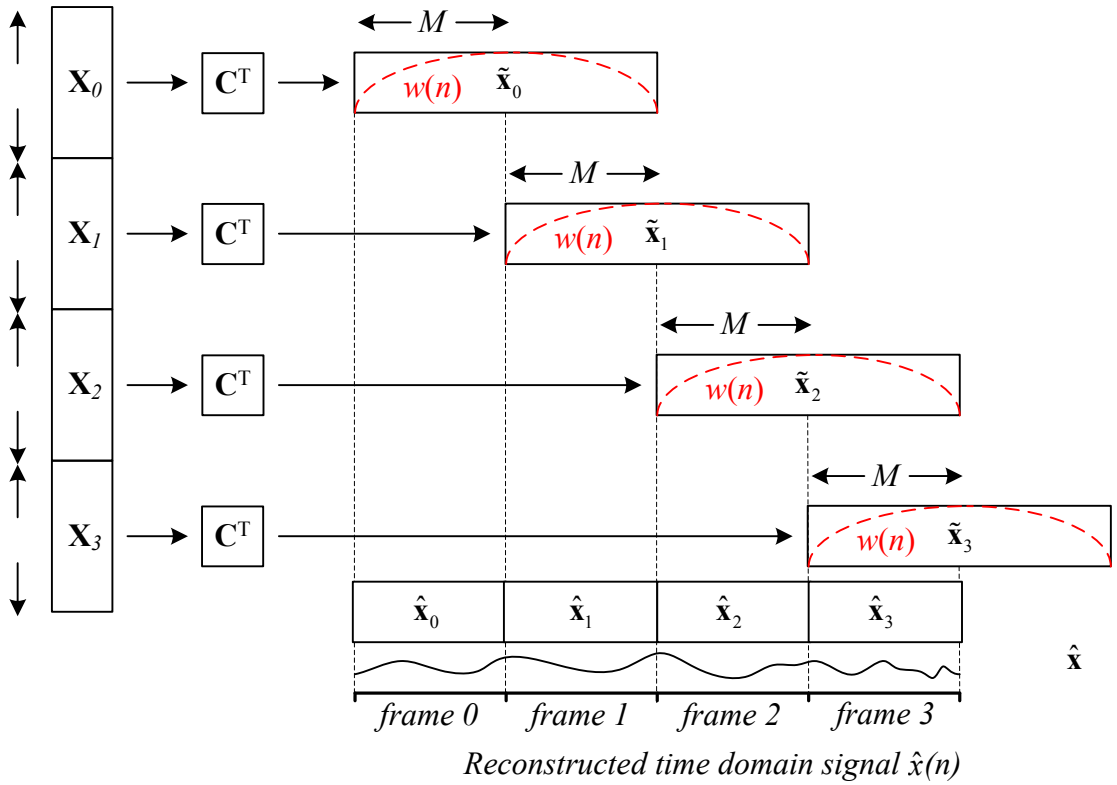


Figure 27 – Inverse MDCT of multiple frames for reconstruction of $\hat{x}(n)$

In order to develop fast algorithms for the implementation of direct and inverse MDCT, it has been shown that the MDCT can be decomposed by considering its polyphase representation [Malvar 92b]. As the MDCT is a cosine modulated filter bank, the polyphase decomposition as described in section 2.1.3.3 can be applied (See Annex A). Taking into account the definition of the Discrete Cosine Transform of type IV (DCT_{IV}) as:

$$X_k = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} x_n \cos\left(\frac{\pi}{4M} (2n+1)(2k+1) \right), \tag{3.30}$$

for $0 \le k \le M$-1. And the inverse $\mathrm{DCT_{IV}}$ is defined identically by:

$$x_n = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X_k \cos\left(\frac{\pi}{4M} (2n+1)(2k+1) \right), \tag{3.31}$$

the MDCT can be decomposed in pre-processing time domain aliasing, followed by a $\mathrm{DCT_{IV}}$ of length M. Based on the symmetry of the cosine function and using the condition (3.5) which impose the symmetry of the window, the MDCT can then be defined as:

$$\mathrm{MDCT} = \mathrm{DCT_{IV}} \cdot \left[-\mathbf{J_{2M}} \right] \begin{bmatrix} \mathbf{J_{M/2}} & \mathbf{I_{M/2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I_{M/2}} & -\mathbf{J_{M/2}} \end{bmatrix} \mathbf{diag}(\mathbf{w}) \tag{3.32}$$

With the components of the matrix $\mathrm{DCT_{IV}}$ defined by $\mathrm{DCT_{IV}}(n,k) = \sqrt{\frac{2}{M}} \cos\left(\frac{\pi}{4M} (2n+1)(2k+1) \right)$. By combining the windowing and time domain aliasing operations and integrating the delay z^{-1} of one block of M samples, equation (3.32) results in:

$$\begin{aligned} \mathrm{MDCT} = {} & \mathrm{DCT_{IV}} \cdot \left[-\mathbf{J_{2M}} \right] \\ & \cdot \begin{bmatrix} \mathbf{J_{M/2}}\mathbf{diag}\left(w_0 \ldots w_{M/2\text{-}1} \right) z^{-1} & \mathbf{diag}\left(w_{M/2} \ldots w_{M-1} \right) z^{-1} \\ \mathbf{diag}\left(w_{M-1} \ldots w_{M/2} \right) & -\mathbf{J_{M/2}}\mathbf{diag}\left(w_{M/2-1} \ldots w_0 \right) \end{bmatrix} \end{aligned} \tag{3.33}$$

In this polyphase decomposition of the MDCT, it appears clearly that consecutive frames can be first processed by the windowing and time domain aliasing operations before applying the $\mathrm{DCT_{IV}}$. The inverse MDCT can be defined similarly by using the operations in the reversed order.

3.1.2 Extended Lapped Transform (ELT)

The extended version of the modified discrete cosine transform (MDCT) is now reviewed. It gives more flexibility on transform prototype or window design at the cost of additional constraints for the perfect reconstruction. The Extended Lapped Transform (ELT) has been introduced and extensively studied in [Malvar 91, Malvar 92a, Malvar 92b]. This increased flexibility on the design of filter prototype, allowing for better filtering performance with longer windows and larger overlap region. In order to obtain better performance, the size of the prototype and basis functions of the transform is increased to a length $L > 2M$. Actually, the ELT uses a larger

block overlap of length $L = KM$, with the overlapping factor $m = K/2$ and $m \in \mathbb{N}$. This transform is a particular instance of the general definition of the perfect reconstruction cosine modulated filter banks defined in [Koilpillai 92]. The basis functions of the ELT are just a simple extension of the MDCT cosine modulation function, but with a longer temporal support, made identical to the window prototype length:

$$p_{k,n} = w(n)c_{k,n},$$
$$c_{k,n} = \sqrt{\frac{2}{M}}\cos\left(\frac{\pi}{M}\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\right), \tag{3.34}$$

for $0 \le k \le M$-1, and $0 \le n \le L$-1. In the ELT definition given in [Malvar 91], the analysis and synthesis windows $w(n)$ are identical and symmetric. Hence, the condition, which must be fulfilled to achieve perfect reconstruction, is given by:

$$\sum_{l=0}^{2m-2s-l} w(n+lM)\,w(n+lM+2sM) = \delta(s) \tag{3.35}$$

for $s = 0,\ldots, m-1$ and $n = 0,.., M/2 - 1$. It should be noted that when $m = 1$, the ELT is reduced to a simple MDCT and the conditions on the window (3.5) can be deducted from equation (3.35). However, no simple analytic formulation of the ELT analysis and synthesis windows exists. In [Malvar 92b], the author has provided several optimization techniques in order to design the appropriate windows.

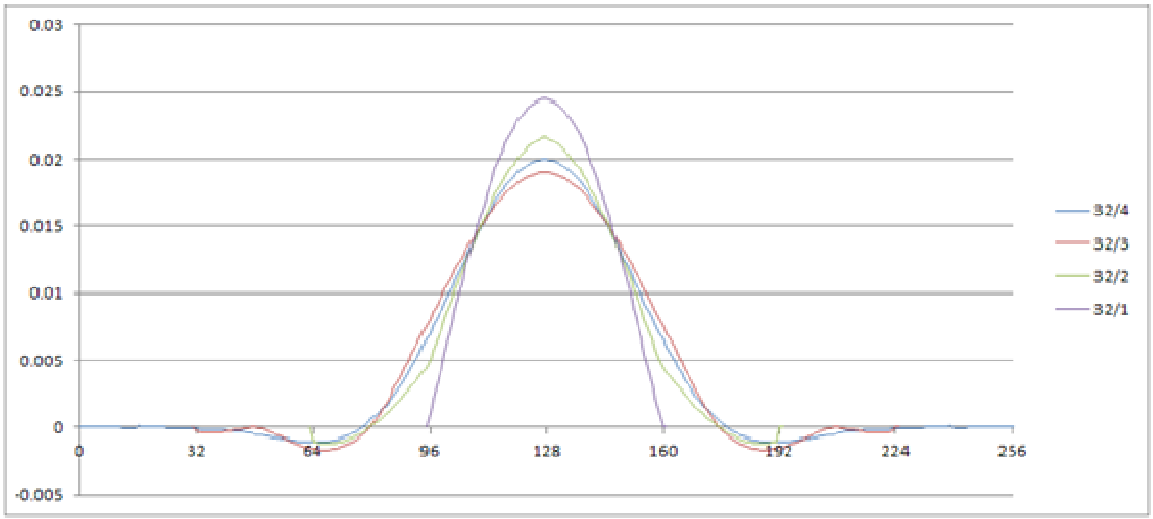


Figure 28 – ELT prototypes for $L=KM=2mM$ with M=32 and m=1, 2, 3, 4

Figure 28 illustrates the ELT prototypes for several values of m for M=32. m=1 corresponds to the case of the normal MDCT. The other values give an indication of the prototype shape when increasing the length. They tend to

reproduce a sinc function. However, the case m=2 (K=4) is limited and does not provide a significant improvement.

3.1.3 Low Delay Transform

In a normal MDCT processing with M frequency coefficients, the corresponding algorithmic delay is defined as $2M$ - 1. This delay represents the algorithmic delay associated with a filtering operation for a filter of length $2M$. The ELT uses a longer window for better performance at the cost of an additional algorithmic delay that depends on the factor K. In order to reduce the overall delay while keeping the advantage of the ELT, bi-orthogonal versions of the modulated filter banks have been considered. These filter banks are based on a decomposition of the filter bank structure into a discrete cosine transform of type IV (DCT_{IV}) and a cascade of pre-processing steps as described in equation (3.33). Based on a polyphase representation of the filter bank, the pre-processing can be decomposed in several basic matrices named maximum delay, zero delay and diagonal factor matrices. This general structure has been introduced in [Schuller 00].
As an introduction to this generalization, the following notations are used. First, the following convention is used. The zero delay matrices are defined as:

$$\mathbf{L}_i(z) = \mathbf{J} + \mathbf{diag}\left(l_0^i, \ldots, l_{M/2-1}^i, 0, \ldots, 0\right) z^{-1} \tag{3.36}$$

where $l_0^i, \ldots, l_{M/2-1}^i$ are real coefficients. The inverse matrix is noted:

$$\mathbf{L}_i^{-1}(z) = \mathbf{J} - \mathbf{diag}\left(0, \ldots, 0, l_{M/2-1}^i, \ldots, l_0^i\right) z^{-1} \tag{3.37}$$

It must be noted that cascading the zero delay matrix with its inverse does not introduce any additional delay. Hence, the zero delay matrices can be used to increase the filter length without any impact on the system delay.
The maximum delay matrix is defined by:

$$\mathbf{L}_i\left(z^{-1}\right) \cdot z^{-1} \tag{3.38}$$

and its inverse matrix is noted:

$$\mathbf{L}_i^{-1}\left(z^{-1}\right) \cdot z^{-1} \tag{3.39}$$

As opposed to the zero delay matrices, the cascade of the maximum delay matrix with its inverse introduces an additional delay of z^{-2}.
Finally, the diagonal matrix is defined by:

$$\mathbf{D} = \mathbf{diag}\left(d_0, \ldots, d_{M-1}\right) \tag{3.40}$$

with $d_0,\ldots,d_{M\text{-}1}$ being real coefficients. Based on this notation, a generalized presentation of the filter banks using the windowing and time domain aliasing stages (3.33) has been proposed by [Schuller 00]. The windowing and time domain aliasing stages are defined by:

$$\mathbf{WTDA}_a\left(z\right) = \prod_{j=0}^{\nu-1} \mathbf{L}_{\mu+\nu-1-j}\left(z\right) \cdot \prod_{i=0}^{\mu-1} \left(\mathbf{L}_{\mu-1-i}\left(z^{-1}\right) \cdot z^{-1}\right) \cdot \mathbf{D} \tag{3.41}$$

where ν is the number of zero delay matrices and μ is the number of maximum delay matrices. The inverse windowing and timed domain aliasing stage is then expressed by:

$$\mathbf{WTDA}_s\left(z\right) = \mathbf{WTDA}_a^{-1}\left(z\right) = \mathbf{D}^{-1} \cdot \prod_{i=0}^{\mu-1} \left(\mathbf{L}_i^{-1}\left(z^{-1}\right) \cdot z^{-1}\right) \cdot \prod_{j=0}^{\nu-1} \mathbf{L}_{\mu+j}^{-1}\left(z\right) \tag{3.42}$$

Based on this formulation of the windowing and time domain aliasing stages of the analysis and synthesis filter banks, [Schuller 00] has derived a wide range of low delay filter banks that are particularly relevant for low delay audio coding.

A specific case of the low delay transform is now considered as defined by [Schuller 00], with a window length $L=KM=4M$. This case has been used as it gives a good trade-off between a longer window for improved performance and a reasonable size in terms of design and complexity. The case of an asymmetric window defined in [Schuller 00] for low delay filter banks is studied. It should be noted that the modulation in the literature is different from the original MDCT basis functions and is defined by:

$$c_{k,n} = \cos\left(\frac{\pi}{M}\left(n - \frac{M}{2} + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right) \tag{3.43}$$

Concretely, the function is simply time-reversed with opposite sign compared to (3.34). The long asymmetric window is given for $K=4$, $L=KM$. An example of long window is given on Figure 29 with $L=KM=4\times512=2048$. In this example, $w(n)$ represents the synthesis window as used in the literature.

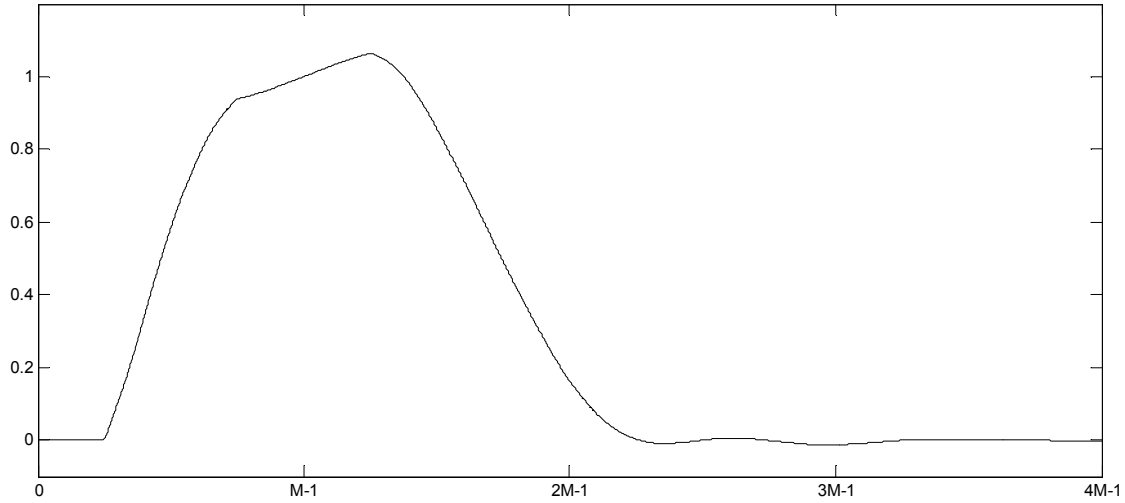


Figure 29 – Prototype of low delay synthesis window for $L=KM=4\times512=2048$.

The analysis filter bank or direct low delay transform corresponding to a long window can be written as:

$$X_{t,k} = -2 \cdot \sum_{n=-2M}^{2M-1} w(2M-1-n) \cdot x_{t,n} \cdot \cos\left(\frac{\pi}{M}\left(n-\frac{M}{2}+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right) \qquad (3.44)$$

for $0 \le k \le M$-1.
The synthesis filter bank or inverse low delay transform is then defined by:

$$\tilde{x}_{t,n} = -\frac{1}{M}\sum_{k=0}^{M-1} X_{t,k}.w(n).\cos\left(\frac{\pi}{M}\left(n-\frac{M}{2}+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right) \qquad (3.45)$$

for $0 \le n \le 4M$-1.
The reconstructed signal $\hat{x}_{t,n}$ is then obtained by overlap-add operation of the four elements coming from the three past blocks and the current block. Hence, the reconstructed signal can be expressed based the following equation:

$$\hat{x}_{t,n} = \tilde{x}_{t,n} + \tilde{x}_{t-1,n+M} + \tilde{x}_{t-2,n+2M} + \tilde{x}_{t-3,n+3M} \qquad (3.46)$$

for $0 \le n \le M$-1.
The analysis window is defined as the time reversed version of the synthesis window $w(n)$ for $0 \le n \le 4M$-1.

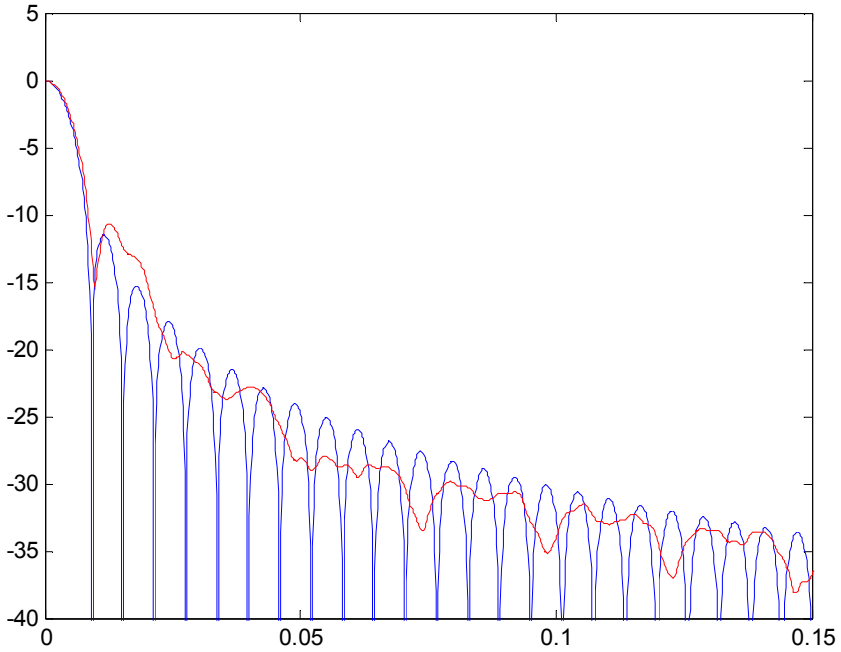


Figure 30 – Comparison of the frequency response of the sine window for the length $2M = 1024$ (blue) with the low delay window of length $L = 4M = 2048$ (red)

The comparison of the frequency responses of the traditional sine window of length $2M$ and the low delay new prototype window with length $4M$ is illustrated in Figure 30. The main lobe is slightly larger for the low delay window and the attenuation of the second and third lobes is slightly degraded compared to the sine window. However, the stop-band attenuation of the low delay window is optimized to offer similar performance.

It should be noted that the similarity in the construction of the low delay transform with MDCT allows using the same fast algorithms [Malvar 90, Duhamel 91]. Indeed, the longer prototype does not influence much the efficiency of those algorithms as the windowing operation is separated from the core transform which can use either an FFT or a DCT_{IV} implementation. Only the complexity of the windowing operations is slightly increased by the length of the window but the core of the algorithm is unchanged.

Malvar [Malvar 92b] estimates the complexity in terms of, respectively, number of additions and multiplications as:

- $\text{Mul}(M,K) = M/2\ (2K + \log2(M) + 3)$
- $\text{Add}(M,K) = M/2\ (2K + 3\log2(M) + 1)$

Hence, increasing the prototype from $2M$ to $4M$ leads to $M/2$ additional additions and multiplications.

3.2 Time Varying Transform

As introduced in 2.2.1.3, the human auditory system is sensitive to rapid change of energy in the time envelope of a sound. In order to avoid temporal unmasking artefacts (e.g. pre-echo), the simplest solution consists in reducing the transform size. For instance, one can choose a transform size of M=128 at 48 kHz. However, even if this reduces the problem introduced by the transient signals, it also dramatically reduces the efficiency of a coding system for stationary sounds. To overcome this problem, two main solutions have been studied.

- The first technique adapts the time/frequency resolution of the MDCT over time. It has been proposed in [Edler 89] to reduce the temporal support of the MDCT when a transient is detected. This technique requires an adaptation of the window length which is referred as *block* or *window switching*.
- A second technique uses frequency domain linear prediction to derive the time envelope in the frequency domain and is known as *temporal noise shaping* (TNS).

Those two techniques are widely used in audio coding standards as they significantly contribute to the quality of such audio coding schemes for transient signals. This characteristic makes them particularly attractive for this thesis which targets to adapt high quality audio coding to low delay applications.

3.2.1 Block switching for MDCT

In order to change the temporal support of the MDCT, a shorter window is used. The number of frequency components is then reduced, and thus both time and frequency resolutions are adapted to the signal content. As the MDCT can switch between transform sizes over time, this technique is known as block switching or window switching [Edler 89] and is related to the time varying filter bank theory.

Let us consider a transition from a MDCT of size M to a smaller size M_s. This configuration is widely used in MPEG audio coding such to isolate transients for signal exhibiting percussive sounds as shown in Figure 2. A long transform with M (i.e. M = 1024) is normally used for the audio signal which is considered as pseudo-stationary as illustrated in the first window $(0,\ldots,2M - 1)$ of Figure 31 and in Figure 32 (a). Transient sounds such as attacks (e.g. castanets, speech plosives) are processed with several successive short MDCT of smaller size M_s (i.e. eight short MDCT of size M_s =128, the AAC framing of 1024 is therefore kept) as is illustrated in Figure 31 after $2M$ samples and in Figure 32 (d).

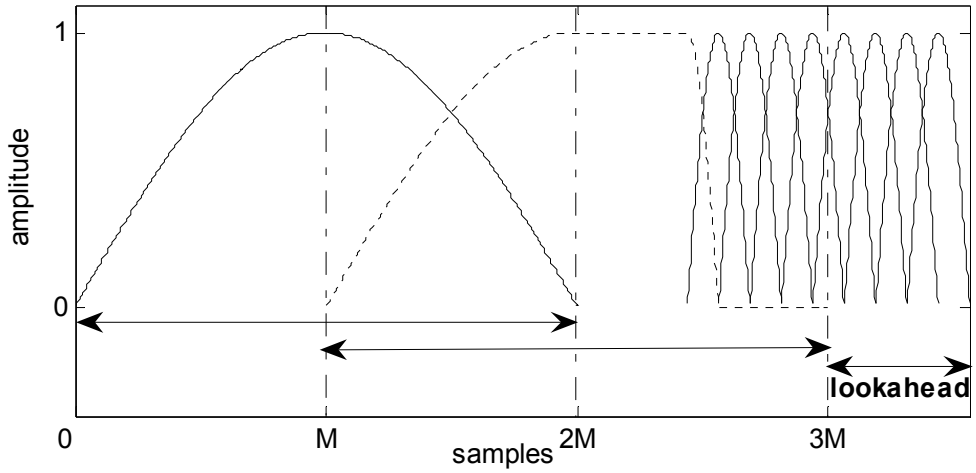


Figure 31 – Combination of windows: long window, transition window (dashed line), and eight short windows

In order to maintain the PR property, care has to be taken when the MDCT size changes. In [Edler 89], the author proposes a solution for maintaining PR using transition windows. During those transition frames, in attack anticipation, a transform of size M is performed with special asymmetric weighting windows. They are designed for the transition between long and short windows and vice versa. Those windows are illustrated on Figure 31, the window represented in dash line between samples M and $3M$ is the transition window from long to short size.

Transition windows are made of four portions:

1. In order to cancel the time domain aliasing with the preceding long window, the transition window uses the long weighting function in its first half.
2. The second portion contains a flat portion, i.e. a constant gain.
3. It is followed by the second half of the short weighting function.
4. Finally a zero tail is added in the zone handled by the short window sequence.

The first half of the short window is centred in the second half of the transition window to ensure that time aliasing components can be cancelled by the first short window.

The two transition windows are also shown on Figure 32 (b) and (c), which represent the transition between a long window and the series of short windows w_{LS} and between the series of short windows and a long window w_{SL} respectively. Using this simple construction method, the aliasing components are cancelled. The equations corresponding to the transition windows are given in equations (3.47) and (3.48):

$$w_{LS}(n)=\begin{cases}\sin\left(\left(n+\frac{1}{2}\right)\frac{\pi}{2M}\right) & ,0\le n\le M-1\\ 1 & ,M\le n\le \frac{3M-M_s}{2}-1\\ \sin\left(\frac{\left(n-\frac{3M-3M_s}{2}+\frac{1}{2}\right)\pi}{2M_s}\right) & ,\frac{3M-M_s}{2}\le n\le \frac{3M+M_s}{2}\\ 0 & ,\text{otherwise}\end{cases} \tag{3.47}$$

$$w_{SL}(n)=w_{LS}(2M-1-n) \tag{3.48}$$

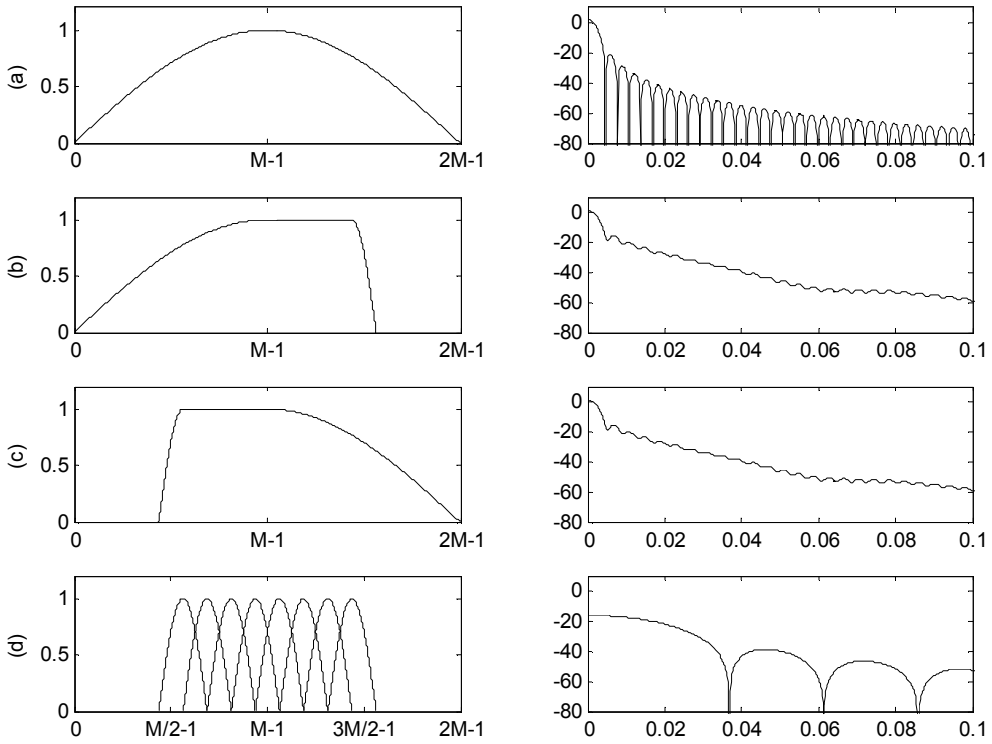


Figure 32 – Long window (a), Transition windows (b) and (c), and Short window

The transition windows, which are represented in Figure 32 (b) and (c), offer a lower stop-band attenuation compared to long window. However, it should be noted that the main purpose of those windows is to avoid temporally spreading of the quantization noise.

3.2.2 Look ahead and time delay for transform

Using such block switching method, transitions between long and short blocks have to be anticipated since transition windows need to be inserted prior to the series of short windows in order to keep the PR property. This

anticipation comes at the price of an additional delay due to the necessary look-ahead: in the example presented in Figure 31, the switching from long to short windows is based on the knowledge of $M/2+M_s/2$ samples ahead of the sample $3M$.
Suppose an attack is present just after sample $3M$ on Figure 31: this attack can be detected only when the frame containing the samples $3M$ to $4M$ is provided to the encoder. In order to have shorter windows in the sample range, a transition window in the $2M$ to $3M$ range is therefore required to ensure appropriate reconstruction. While this transient was unknown by the encoder, it is impossible to insert such transition window if the attack is not anticipated through a look-ahead buffer.

The block switching technique inherently increases the coding delay. As described in this section, this technique requires an additional delay added to a system that would only operate with long block transform. Depending on the position of attacks (or transients) in the input segment, two different scenarios must be identified. The first one does not necessarily require the introduction of an additional delay, whereas in the second scenario this minimum additional $M/2+M_s/2$ samples delay is absolutely necessary in order to obtain the best efficiency of the block switching technique.

In Figures 33 and 34, the possible timing scenario for window transitions is illustrated. Five frames are shown, a long window is first selected between 0 and $2M$-1, and then a transition window (long to short) spanning from M to $3M$-1, a short windows sequence and finally another transition window (short to long). A sharp attack is detected around sample t_{att}, as such it is processed by a short windows sequence to adapt to the signal transient nature.
In order to illustrate the two scenarios, the current frame is defined as the central frame on both Figures 33 and 34. In Figure 33, since the attack arises in the current frame (between samples $2M$ and $3M$), it can be directly detected and the transition window (long to short window) can adequately be inserted to anticipate this window switching. In this case, no look-ahead is necessary.

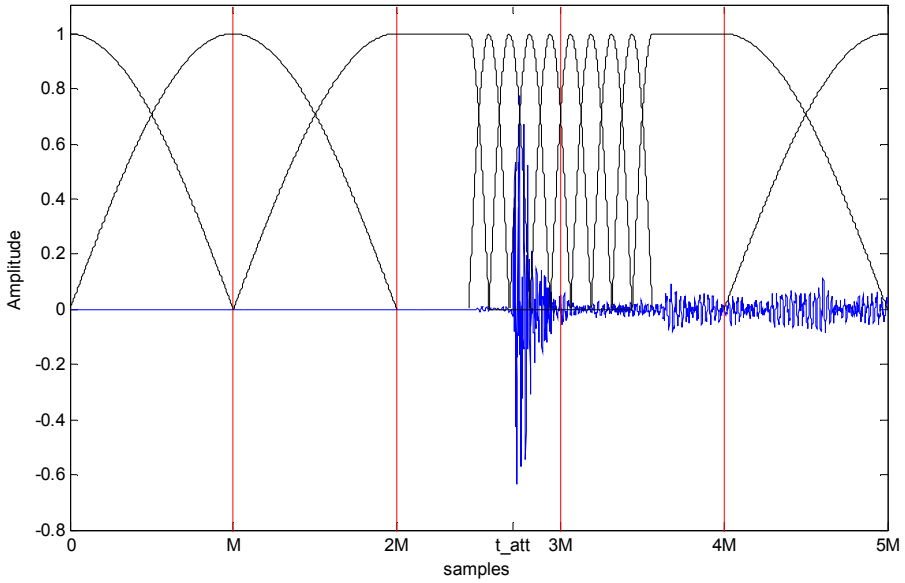


Figure 33 – Timing for transition window insertion: the attack arises at the end of the current frame; transition window can be selected when samples 2M to 3M are processed.

As opposed to Figure 33, Figure 34 presents the case where an attack occurs at the beginning of the next frame. In this second scenario, the processing algorithm must select the window which should be applied in the current frame, more precisely on the samples between M and $3M$, due to the overlap of the MDCT. In order to process the attack with a short windows sequence, the processing algorithm must anticipate the transient, and a transition window needs to be selected in the current frame. This cannot be done without the knowledge of the audio content for the following samples that leads to an additional look-ahead buffer. This look ahead buffer formally needs to be composed of at least $(M+M_s)/2$ samples.

This paragraph has presented the necessity to use a look-ahead buffer in order to detect the transient part of the signal in advance and to allow the insertion of a transition window in the MDCT processing, prior to the attack position t_{att}. This minimum look-ahead buffer is $(M+M_s)/2$ samples long in the optimum case when the attack detection algorithm can instantaneously detect a transient (in a sample by sample processing). This additional delay, which is due to the look-ahead, is not suitable for communication codecs where the overall delay is kept as low as possible to enhance the interactivity. Indeed, with such additional constraint of look-ahead, the total algorithmic delay of the system would be increase from $2M$ to $(5M+M_s)/2$, which is significant for low delay application.
Consequently, block switching has not been used in low delay perceptual audio coder in order not to introduce this additional source of delay. This

limitation results in a suboptimal quality compared to a coding scheme that would have been allowed to adapt the time frequency resolution.

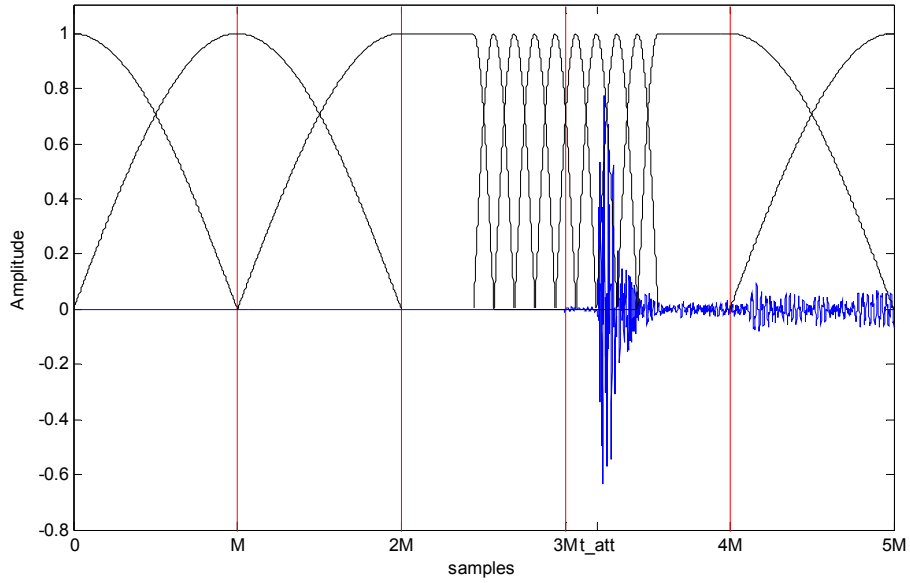


Figure 34 – Timing for transition window insertion: the attack arises in the next frame; without look-ahead buffer the transition window cannot be anticipated.

It should be noted that the transition from short to long windows does not introduce extra delay in the normal operating mode. Hence, only the transition from long to short window particularly needs to be adapted for low delay applications.

3.2.3 Temporal Noise Shaping

An alternative approach can be used in order to overcome the problem of pre-echo in transform coder. In [Herre 96] and [Herre 97], a solution has been proposed to reduce pre-echo artefacts based on the application of linear prediction in the frequency domain. For a transient signal, the spectrum is relatively flat, which reduces the efficiency of entropy coding techniques. In [Herre 96], the authors proposed to use the time-frequency duality of the linear prediction and to represent a time envelope adapted processing by applying a forward adaptive linear predictive coding (LPC) on the MDCT coefficient. This method is called Temporal Noise Shaping (TNS) as it allows a reshaping of the quantization noise in the time domain based on the linear prediction filter obtained in the frequency domain. The temporal structure of the quantization noise follows the signal more closely such that the quantization noise is shaped to lie in the most energetic part of the signal.

The TNS typically uses a Levinson-Durbin algorithm to compute the linear prediction filter coefficients (LPC) based on the autocorrelation function of the spectral coefficients as presented in [Makhoul 75]. Makhoul has pre-

sented the application of linear prediction to represent the spectral envelope. He has also shown that the LPC can be directly computed based on the DFT spectrum. However, the purpose was always to define the frequency envelope.
The filter order typically ranges between 4 and 12 to obtain a sufficiently reliable temporal envelope. In audio coding schemes, the TNS is usually used in combination with MDCT. As opposed to the use of the DFT, the MDCT introduces time domain aliasing. The use of LPC in the MDCT domain leads to a shaped quantization noise which appears mirrored in both left and right window half. In order to reduce this effect, one can use the low-overlap window as introduced in 3.1.1.1 which limits the length of the overlap region and then the impact of the time domain aliasing on the temporal envelope definition using a linear prediction filter. However, the low-overlap window offers a lower selectivity and stop-band attenuation which reduces the performance of the coding scheme for stationary signals.
In [Herre 97], the use of the TNS in combination with the MDCT is presented as a continuously signal-adaptive filter bank allowing to control the time and frequency structure of the quantization noise.

3.3 Conclusion

Chapter 3 has introduced the MDCT definition and its matrix notation. Moreover, the ELT and the low delay transform have been presented. They are based on longer prototypes and the latter one offers the possibility to reduce the algorithmic delay compared to the MDCT. The time-varying transforms have also been briefly described as they clearly contribute to the performance of audio coding in presence of transient signals. The block switching and TNS tools have been widely used in audio coding scheme to improve the quality of transient coding. The block switching offers a time-frequency resolution adaptation, but has not been used for low delay application as it requires an additional look-ahead to detect the attack and anticipate the resolution change. Nevertheless, the block switching would definitely contribute to improve the quality if it could be adapted to low delay audio coding.

Chapter 4

Advanced transform for low delay audio coding

In Chapter 3, the bases of the transform for audio coding have been introduced. The following sections present the main contributions of the thesis. The low delay block switching method is presented. It has been developed for improving transient signals in a low delay audio coding scheme. This method has then been extended to the low delay transform with a longer prototype.
The method has been generalized to seamless reconstruction of the transitions between all kinds of MDCT windows. This seamless reconstruction allows to change the transform window frame-by-frame without considering the previous or the following frame. This extension of the concept can be of course combined with the low delay block switching which is first introduced.
Finally, the relaxed definition of the perfect reconstruction property with MDCT led to the general definition of analysis and synthesis windows. From this general definition, the design of a new family of low delay windows for MDCT transform has been derived.

4.1 Low Delay Block Switching for MDCT

A contribution of this thesis is now presented. An adaptation of the block switching method has been developed for its application to low delay transform coding. This method is based on the traditional block switching that has been introduced in [Edler 89] and presented in 3.2.1. In order to overcome the problem of additional delay which is required for the transient detection, the proposed method uses a direct transition between long and

short windows for the direct MDCT and the perfect reconstruction is maintained in the inverse MDCT using a modified reconstruction method.
Consequently, the best quality can be achieved through block switching without requiring additional delay.

4.1.1 Low delay transition

In order to avoid additional look-ahead that increases the codec algorithmic delay and then generally prevents the use of block switching for low delay processing, a specific transition scheme is necessary. A general solution allowing the direct transition between two different MDCT sizes is presented. Using this method, the coding delay, introduced by the window switching method with the transient detection algorithm, is removed. This technique can be applied to low delay communication codecs such as MPEG-4 Low Delay AAC [Allamanche 99] or Enhanced Low Delay AAC [Schnell 08] as it will be described in Chapter 5. It is demonstrated that the additional look-ahead, which is usually required for the transient detection, can be avoided, as such permitting window switching even for low delay communication audio codecs.
The solution can be expressed as a post-processing operation in the inverse MDCT*, called compensation*, while the direct transform directly switches from long to short resolution, i.e. without transition window. The inverse transform is followed by this specific post-processing operation that removes the time aliasing components as shown in section 4.1.3. Consequently, PR can still be achieved.
The compensation windows, which replace the traditional synthesis window in the inverse MDCT to cancel the time domain aliasing terms is described in section 4.1.4 and the associated compensation algorithm is presented in 4.1.5. Finally, the complexity of the proposed method is evaluated in section 4.1.5.

4.1.2 Equivalent long transform for the shorter MDCT

Based on the MDCT matrix notation which has been introduced in 3.1.1.2, it is demonstrated that low delay block switching can be achieved. As given in equation (3.17), a frame t-1 is processed using a long window of size $2M$. The next frame (and the next transform), at frame t, is processed with eight short transforms of size M_s as done in the normal block-switching algorithm. The only difference is that the transition window step is skipped. The direct and inverse MDCT with short windows is similar to the long window processing of equation (3.11) and (3.16) with a reduced number of sub-band called M_s and applied eight times for processing the $2M$ samples of the past and current frames. In order to represent the eight short windows processing using a normal MDCT representation which is based on the M samples overlap between two consecutive windows, transform matrix

$\mathbf{P_s}$ is introduced. It is the equivalent M size matrix formulation for those eight short blocks. The transform matrix of eight short windows can be written in a **P** matrix fashion.

$$\tilde{\mathbf{x}}_t^{2M} = \mathbf{P_s}^{\mathbf{T}} \mathbf{P_s}\ \mathbf{x}_t^{2M} \tag{4.01}$$

As defined in equation (3.19), the direct and inverse transform operations using the eight short windows can be written as:

$$\mathbf{P_s}^{\mathbf{T}} \mathbf{P_s} = \begin{bmatrix} \mathbf{U_{s0}} & \mathbf{0} \\ \mathbf{0} & \mathbf{U_{s1}} \end{bmatrix} \tag{4.02}$$

where the two sub-matrices $\mathbf{U_{s0}}$ and $\mathbf{U_{s1}}$ are defined in equations (4.03) and (4.04). It must be noted that in both matrixes, the time domain aliasing is limited to a size M_s in the central part of the matrix. On the other hand, the signal can be directly reconstructed by the direct and inverse transform without the need of overlap and add operation for $(M\text{-}M_s)/2$ samples. The two parts of the transform matrix are given by:

$$\mathbf{U_{s0}} = \begin{bmatrix} \mathbf{0}_{\frac{M-M_s}{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{diag}\left(\mathbf{w_{s,0}}\right)\left(\mathbf{I}_{M_s} - \mathbf{J}_{M_s}\right)\mathbf{diag}\left(\mathbf{w_{s,0}}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\frac{M-M_s}{2}} \end{bmatrix} \tag{4.03}$$

and

$$\mathbf{U_{s1}} = \begin{bmatrix} \mathbf{I}_{\frac{M-M_s}{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{diag}\left(\mathbf{w_{s,}}_{M_s}\right)\left(\mathbf{I}_{M_s} + \mathbf{J}_{M_s}\right)\mathbf{diag}\left(\mathbf{w_{s,}}_{M_s}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{\frac{M-M_s}{2}} \end{bmatrix} \tag{4.04}$$

Where $\mathbf{w_s}$ is the short window used for the series of eight short MDCT transform and where $\mathbf{w_{s,0}}$ and $\mathbf{w_{s,}}_{\boldsymbol{M_s}}$ represents the first half and the second half of the short window respectively. The eight short windows are represented by an equivalent long window, encompassing a $2M$ time-frame, given in Figure 35. It should be noted that this representation highlights the equivalence between the series of short windows and the low-overlap window which is defined in equation (3.10) if the overlap size is chosen to be identical $M_s = M / 2$. The direct transition between a long sine window and eight short windows can also be seen as a direct transition between a long sine window and the low-overlap window of the same size.

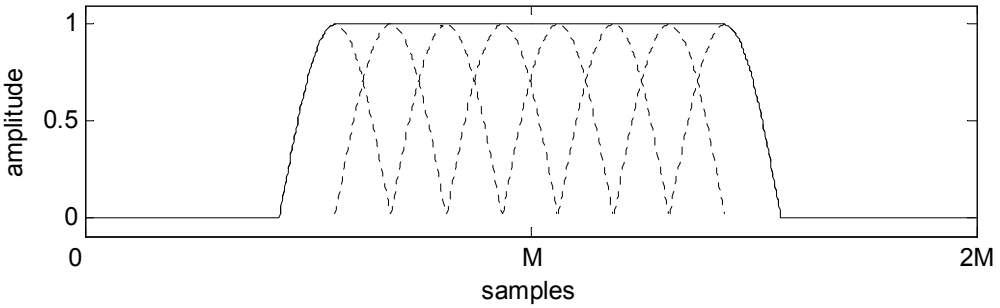


Figure 35 – Eight short windows, each of size $2M_s$ (dashed line), and the equivalent $2M$ size window (solid line).

4.1.3 Perfect reconstruction during resolution changes

A demonstration that even during direct transition from long to short transforms, the time aliasing can be suppressed using two appropriate weighting functions $\mathbf{w_1}$ and $\mathbf{w_2}$ applied after inverse transformation is now performed. This operation is called compensation.

It can be seen from equations (4.03) and (4.04) that the central (M - Ms) samples of the input vector $\mathbf{x}_t^{2M}$ are directly recovered from the combined operation of direct and inverse transform: both $\mathbf{U_{s0}}$ and $\mathbf{U_{s1}}$ contain a portion of identity matrix which provides this direct reconstruction of the input signal. Hence, only the (M + Ms)/2 first elements need a post-processing operation to ensure the perfect reconstruction through the time aliasing cancellation. Indeed, the overlap of the two consecutive blocks does not directly lead to the time aliasing cancellation since the direct transforms are not based on the same window and the time aliasing components are shaped differently. This mismatch between time aliasing components can be represented by:

$$\mathbf{U_1} + \mathbf{U_{s0}} \neq \mathbf{I}_M \tag{4.05}$$

However, if an appropriate set of weighting functions $\mathbf{w_1}$ and $\mathbf{w_2}$ is introduced, along with an anti–aliasing matrix $\mathbf{A}$, the PR can be obtained through:

$$\mathbf{diag}\left(\mathbf{w_1}\right)\mathbf{U_1} + \mathbf{diag}\left(\mathbf{w_2}\right)\mathbf{A}\mathbf{U_{s0}} = \mathbf{I}_M \tag{4.06}$$

The $M \times M$ matrix $\mathbf{A}$ aims at cancelling the time aliasing terms:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{J}_{\frac{\mathbf{M}-\mathbf{M_s}}{\mathbf{2}}} \\ \mathbf{0} & \mathbf{I}_{\mathbf{M_s}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\frac{\mathbf{M}-\mathbf{M_s}}{\mathbf{2}}} \end{bmatrix} \tag{4.07}$$

Equation (4.06) can be rewritten in a similar way to equations (3.28) and (3.29) to define the new PR conditions. Those two new PR equations lead the definition of the new post-processing weighting functions:

$$\begin{aligned}&\mathbf{diag}(\mathbf{w}_1)\mathbf{diag}(\mathbf{w}_M)\mathbf{J}_M\mathbf{diag}(\mathbf{w}_M)-\\&\mathbf{diag}(\mathbf{w}_2)\mathbf{diag}(\mathbf{w}_{\mathbf{s},0})\begin{bmatrix}\mathbf{0} & \mathbf{J}_{\frac{\mathbf{M+M_s}}{\mathbf{2}}}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{diag}(\mathbf{w}_{\mathbf{s},0})=\mathbf{0}\end{aligned} \tag{4.08}$$

and

$$\begin{aligned}&\mathbf{diag}(\mathbf{w}_1)\mathbf{diag}(\mathbf{w}_M)\mathbf{diag}(\mathbf{w}_M)+\\&\mathbf{diag}(\mathbf{w}_2)\mathbf{diag}(\mathbf{w}_{\mathbf{s},0})\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{\frac{\mathbf{M+M_s}}{\mathbf{2}}}\end{bmatrix}\mathbf{diag}(\mathbf{w}_{\mathbf{s},0})=\mathbf{I}_M\end{aligned} \tag{4.09}$$

This system is solved in $\mathbf{w_1}$ and $\mathbf{w_2}$ by a set of $2M$ linear equations with $2M$ unknown variables. As a result, the analytical expressions of $\mathbf{w_1}$ and $\mathbf{w_2}$ can be obtained as functions of $\mathbf{w}$ and $\mathbf{w_s}$ and more specifically as a function of the second half of the long window $\mathbf{w}_M$ and the first half of the short window $\mathbf{w}_{\mathbf{s},0}$.

4.1.4 Compensation windows

The two equations (4.08) and (4.09) are used to define the post-processing weighting functions performed after the inverse MDCT. This post-processing operation is decomposed in three parts.

The first part is used to cancel the aliasing generated by the long window and where there is no direct overlap with the short windows. In this region, the two weighting functions w_1 and w_2 are given by:

$$w_1(n)=\frac{1}{w^2(M+n)},\ w_2(n)=-\frac{w(n)}{w(M+n)} \tag{4.10}$$

for $0\le n\le\frac{(M-M_s)}{2}-1$.

As observed in (4.08), w_2 is applied on the time-reversed synthesized signal coming from the sequence of short windows.

The second part represents the direct overlap between the long and the first short window. Compensation is used to cancel the different weights which have been applied to the aliasing:

$$w_1(n)=\frac{w(n)w(M-1-n)-w_s\left(n-\frac{M-M_s}{2}\right)w_s\left(\frac{M+M_s}{2}-1-n\right)}{d(n)},$$

$$w_2(n) = \frac{w_s(M_s - 1 - m)}{d(n)} \tag{4.11}$$

for $\frac{M - M_S}{2} \le n \le \frac{M + M_S}{2} - 1$, with

$$d(n) = w(M-1-n)\left[w(M-1-n)\, w_s\left(\frac{M+M_s}{2} - 1 - n\right) + w(n)\, w_s\left(n - \frac{M-M_s}{2}\right)\right] \tag{4.12}$$

Finally, in the last region, the reconstruction is directly obtained from the short windows only.

It comes that $w_1(n)=0$ and $w_2(n)=1$ for $(M+M_s)/2 \le n \le M-1$. These compensation windows can be directly integrated in the synthesis process in order to reduce the complexity. In that case, new synthesis windows are used in case of direct transition between long and short windows.

4.1.5 Compensation algorithm

Using the sample notation of the overlap and add as defined in (3.4), the compensation algorithm of equation (4.06) is split in two parts. The first part relates to the reconstruction of the samples in the interval $0 \le n < (M - M_s)/2$ using the compensation windows as defined in (4.10), and is expressed by:

$$\hat{x}_{t,n} = w_{1,n}\tilde{x}_{t-1,n+M} + w_{2,n}\hat{x}_{t,M-1-n} \tag{4.13}$$

where $\tilde{x}_{t-1,n+M}$ are recovered from the previous long window transform, and $\hat{x}_{t,M-1-n}$ represents the reconstructed signal obtained from the eight overlapping short windows.

The second part of the algorithm deals with the samples $(M - M_s)/2 \le n \le (M + M_s)/2 - 1$. In this case, the reconstruction is based on the compensation windows defined in (4.11) and is given by:

$$\hat{x}_{t,n} = w_{1,n}\tilde{x}_{t-1,n+M} + w_{2,n}\tilde{x}^{s}_{t,n} \tag{4.14}$$

with $\tilde{x}^{s}_{t,n}$ representing the inverse transform of the first (over eight) short block.

The new synthesis windows can be defined as the multiplication of the normal synthesis windows by the compensation windows. These new windows allow to obtain a complexity similar to the normal synthesis. Figure

36 illustrates the different cases of transitions with the known windows (a) and (b), and the new synthesis windows with the compensation algorithm in (c). Figure 36 (a) represents the traditional block-switching configuration with the transition window between the long and the short windows. Figure 36 (b) shows the direct transition between long and short windows which is applied in the direct transform whereas Figure 36 (c) shows the equivalent windows used for the inverse transform. These windows directly replace the traditional synthesis windows in the inverse transform step and directly integrate the compensation windows.

It can be seen from equations (4.10) and (4.11), that the post processing integrates the cancellation of the windows which are normally used for the inverse transform. Hence, combining the inverse transform and the post processing is simple. These combined weighting functions lead to a complexity decrease compared to a separate post processing application. The computational complexity of the low delay block switching becomes identical to the normal block-switching which does not lead to complexity increase compared to long window synthesis.

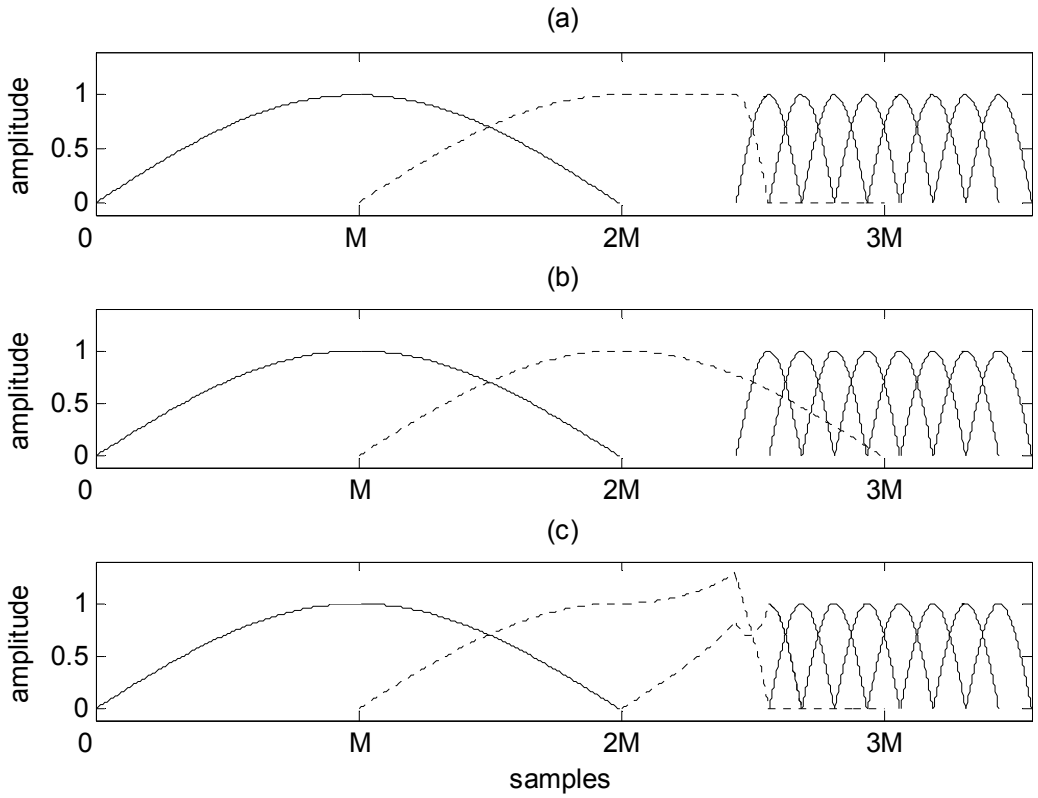


Figure 36 – Illustration of various window transitions. (a) Traditional window sequence: long window, long-short transition window (dashed line), eight short windows. (b) Direct transition between long (dashed line) and short (solid line) windows for the direct transform. (c) Compensation scheme for the inverse transform: in dashed line, the modified part of the long and first short window.

As described above, the proposed low delay block switching essentially relies on a compensation scheme in the inverse transform. Hence, the compensation scheme acts as the traditional window weighting and just replaces the original synthesis windows by the new weighting functions. It should be noted that no special care has to be taken to the analysis (encoding) stage, only the synthesis (decoding) side is affected by the proposed method. The encoder can select for the current block the most appropriate window (long or short) without the introduction of the transition window in between.

4.1.6 Low delay block switching behavior in audio coding

Some objective results of the performance of the low delay block switching are now presented. First, Figure 37 shows the window shape for the normal long window, the traditional transition window as introduced in 3.2.1 and finally the newly introduced synthesis window for the low delay block switching method are plotted on the left side. The corresponding frequency responses are represented on the right side. It can be seen from Figure 37 that the main difference between the transition window (b) and the low delay block switching synthesis window (c) lies between the samples M and $(3M+M_s)/2$. The peak, which is introduced in (c), results in a slightly worse frequency response of the synthesis window. Note that this frequency spreading only appears at the synthesis stage, it only affects the noise smearing and not the energy compaction at the encoder side.

The frequency responses of the transition window and the low delay block switching synthesis window are further superimposed in Figures 38 and 39. The second lobe attenuation is reduced by 2 dB for the new synthesis window. However, Figure 39 indicates that there is no strong performance difference between the two windows in the stop-band attenuation. It should be noted that one additional advantage of the low delay block switching is that the direct transform uses a normal sine window. Hence, the coding stage is performed based on a more accurate frequency representation of the audio signal. Indeed the sine window offers a better selectivity and a better stop-band attenuation which is particularly important for the allocation step where the perceptual relevance of each sub-band is used.

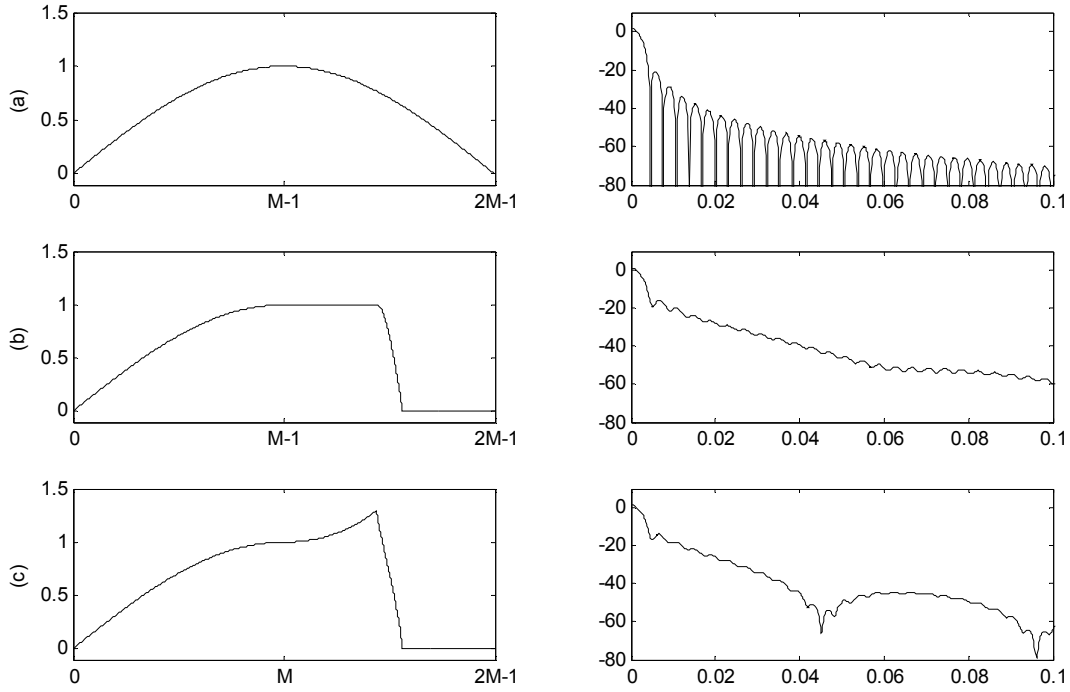


Figure 37 – Long sine window (a), Transition window (b) and Low delay block switching synthesis window (c)

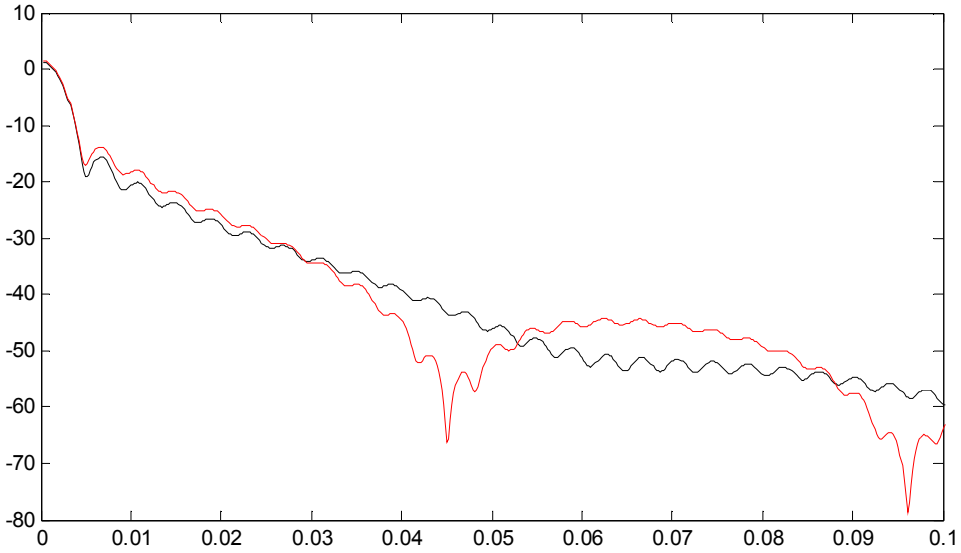


Figure 38 – Frequency responses (between the normalized frequencies 0 and 0.1) of transition window (in black) and low delay block switching synthesis window (in red)

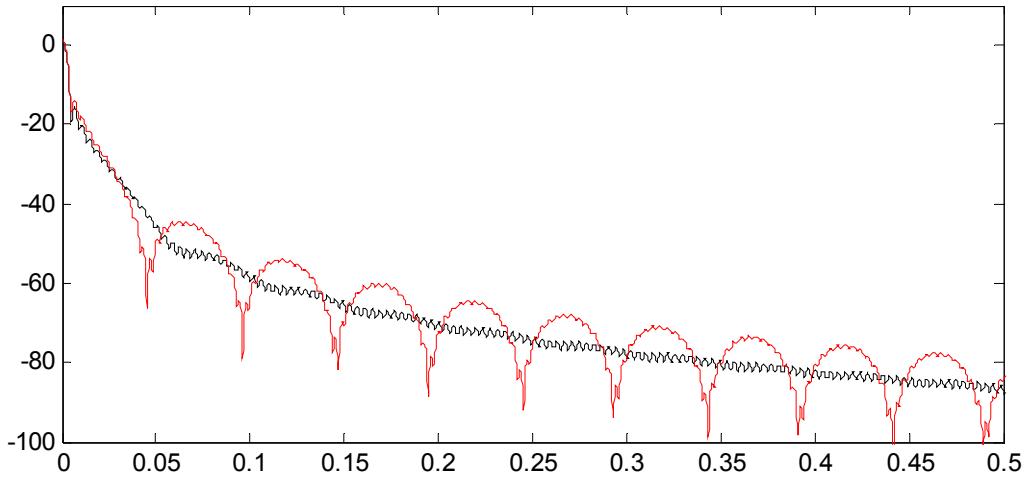


Figure 39 – Frequency responses (between the normalized frequencies 0 and 0.5) of transition window (in black) and low delay block switching synthesis window (in red)

In order to evaluate the influence of the quantization stage in the reconstruction of the audio signal using the low delay block switching, an experiment is conducted consisting in the superposition of white quantization noise to the transform coefficients and in performing the inverse transform followed by the overlap and add operation.
Figures 40 and 41 illustrate the results of this experiment for the long and the short windows respectively. It can be seen on Figure 40, that the temporal noise profile follows the synthesis window shape as shown on Figure 37 (c). In this experiment, no quantization noise has been introduced in the short windows, so we can then see the quantization noise brought by the long window shape after reconstruction. Figures 40 and 41 have been obtained by averaging the error after reconstruction over a large number of simulations with the injected quantization noise being a random noise.

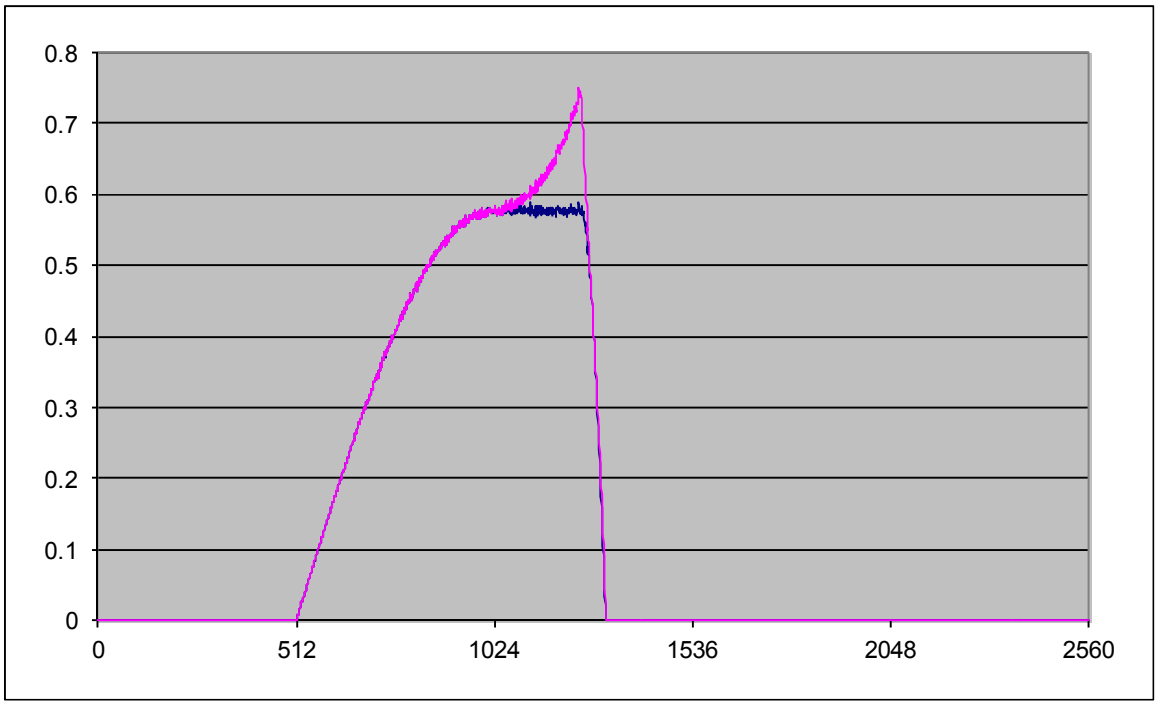


Figure 40 – Illustration of noise injection in long window for low delay block switching.

Figure 41 illustrates the noise profile for the short windows after reconstruction. The quantization noise introduced by each window is represented with a different colour. It can be seen that the fifth short window (labelled slot 4) is the first window to introduce some noise during the reconstruction. It is followed by the fourth (slot 3), the third (slot2), the second (slot 1) and finally the first short window. The noise energy increases with time to reach the maximum energy for the first window (slot 0).

It should be noted that the noise introduced by the short windows four and five have a very limited energy in the overlapping region which leads to very small perceptual quality impact. Only the three first windows have significant energy to potentially introduce some artefacts. However, the quantization noise of the first short windows will influence the audio quality and generate some pre-echo only if the transient lies in one of them. In that case, the transient being sufficiently early in the first part of block, a good block switching algorithm would avoid the use of the low delay block switching and introduce a transition window as traditionally used in block switching without the need of additional look-ahead. In case the transient is positioned in one of the window from 4 to 8, the low delay block switching is particularly adapted and the quantization noise which is introduced ahead of the transient is very limited, reducing the risk of perceived pre-echo artefacts.

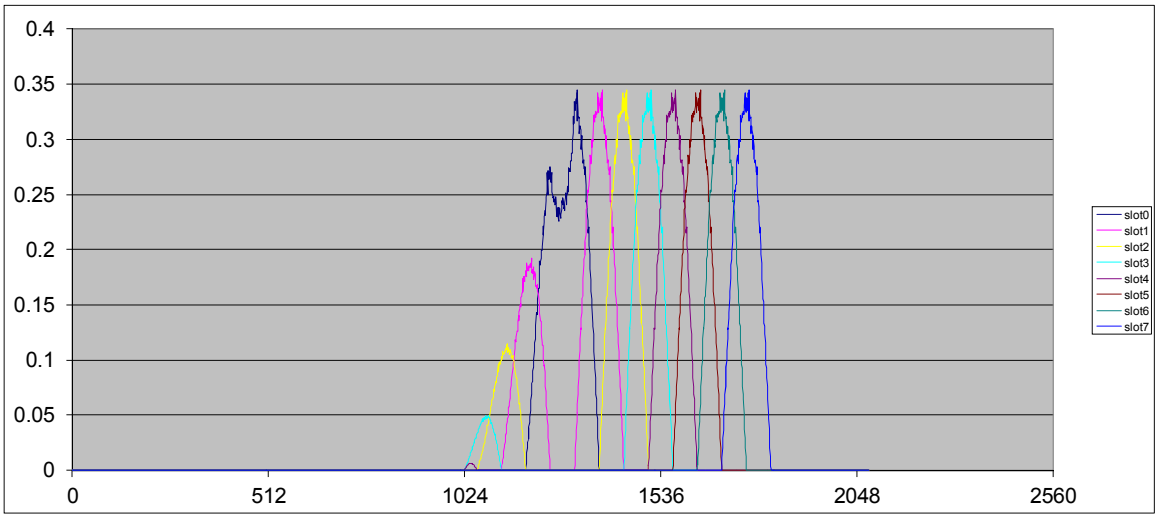


Figure 41 – Illustration of noise injection in short windows for low delay block switching.

4.2 Low Delay Block Switching for Low Delay Transform

The low delay block switching can also be adapted to the low delay transform and furthermore for switching between transform families. In this section, a possible transition between long and short window is presented, where the long window is defined as a low delay transform according to the design method introduced in 3.1.3. The size of the long window is defined as $L=KM=4M$ and the size of the short window is defined as $L_s=K_sM_s$ with $K_s=2$.

It can be noticed that the overlap factor K is different for the two block sizes. There is no particular interest to keep the same low delay transform prototype and overlap factor for the short block. The shorter length aims at providing a better time localization. Hence, improving the frequency resolution at the cost of this longer prototype would introduce the risk to spread the quantization noise on a longer temporal support in the transient region with the risk of unmasking this noise. As the main objective is to obtain a better temporal resolution, a different shape is selected. Hence our choice is a traditional symmetric short window with for instance $K_s=2$, $L_s=K_sM_s$. Following the example introduced in 3.1.3, the size of long prototype window is $KM=4\times512$ and the short window is $K_sM_s=2\times64$. The transition consists then in a direct transition between a long extended MDCT window and a traditional symmetric short MDCT window.

4.2.1 Short transform definition

The short window transform is defined based on the same modulation as the low delay transform, allowing to obtain compatible time aliasing. This change corresponds to a time-reverse modulation compared to the MDCT

modulation. The transform coefficients for the short blocks are then given by:

$$X_{t,k}^{m} = -2 \cdot \sum_{n=0}^{2M_s-1} x_{t,n+mM_s+(M-M_s)/2} \cdot w_s(n) \cdot \cos\left(\frac{\pi}{M_s} \left(n - \frac{M_s}{2} + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right) \quad (4.15)$$

for $0 \leq k \leq M_s$-1 and with $w_s(n)$ being the short symmetric analysis window. m represents the index of the short window in the group of eight short windows $m \in [0,7]$ as defined in the normal block switching 3.2.1.

The corresponding inverse transform is given for each short block by:

$$\tilde{x}_{t,n+mM_s+(M-M_s)/2} = -\frac{1}{M_S} \sum_{k=0}^{M_S-1} X_{t,k}^{m} . w_s(n) . \cos\left(\frac{\pi}{M_s} \left(n - \frac{M_s}{2} + \frac{1}{2} \right) \left(k + \frac{1}{2} \right) \right) \quad (4.16)$$

for $0 \leq n \leq 2M_s$-1.

The reconstructed signal $x_{t,n}$ is obtained by the overlap and add operation of two elements coming from two short inverse transforms (K_s=2). The reconstructed signal obtained by the overlap of two consecutive short blocks is given by the following equation:

$$\hat{x}_{t,n+\frac{M+M_s}{2}+mM_s} = \tilde{x}_{t,n+\frac{M+M_s}{2}+mM_s} + \tilde{x}_{t,n+\frac{M-M_s}{2}+(m+1)M_s} \quad (4.17)$$

for $0 \leq n \leq M_s$-1.

In this specific combination of short and long windows, the short window is defined as symmetric, which means that both analysis and synthesis windows are identical. Moreover the selected short window shape must satisfy the Perfect Reconstruction property. The usual sine window is then selected, and its definition is given by:

$$w_s(n) = \sin\left[\frac{\pi}{2M_S} (n + 0.5) \right] \quad (4.18)$$

for $0 \leq n \leq 2M_s$-1.

All the standard symmetric window shapes presented in 3.1.1 can of course be substituted to the sine window. The more accurate temporal resolution is obtained by the reduction of the block size. In order to still preserve the frequency selectivity of the short block transform, a sine or KBD windows should be used as short window.

4.2.2 Low delay transition with different overlap ratio

In this paragraph, the procedure used to determine the shape of the long and short synthesis windows in case of low delay transition between long and short windows is presented. This procedure is an extension of the previously presented transition between symmetric long and short windows. It leads to the insertion of transition windows.

Based on the low delay synthesis window w, the transition windows (LONG_START synthesis window $w_{synSTART}$ and LONG_STOP synthesis window $w_{synSTOP}$) are defined as:

$$w_{synSTART}(n) = \begin{cases} w(n) & ,0 \le n \le M-1 \\ 1 & ,M \le n \le \dfrac{3M-M_s}{2}-1 \\ \sin\left(\dfrac{\left(n-\dfrac{3M-3M_s}{2}+\dfrac{1}{2}\right)\pi}{2M_s}\right) & ,\dfrac{3M-M_s}{2} \le n \le \dfrac{3M+M_s}{2}-1 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.19}$$

$$w_{synSTOP}(n) = \begin{cases} 0 & ,0 \le n \le \dfrac{M-M_s}{2}-1 \\ \sin\left(\dfrac{\left(n+\dfrac{1}{2}\right)\pi}{2M_s}\right) & ,\dfrac{M-M_s}{2} \le n \le \dfrac{M+M_s}{2}-1 \\ 1 & ,\dfrac{M+M_s}{2} \le n \le M-1 \\ w(n) & ,M \le n \le 2M-1 \\ 0 & ,2M \le n \le 3M-1 \\ w(n) & ,3M \le n \le 4M-1 \end{cases} \tag{4.20}$$

The corresponding transition synthesis windows are presented in Figure 42. The transition analysis window are defined as $w_{synSTART} = w_{anaSTOP}$ and $w_{synSTOP} = w_{anaSTART}$.

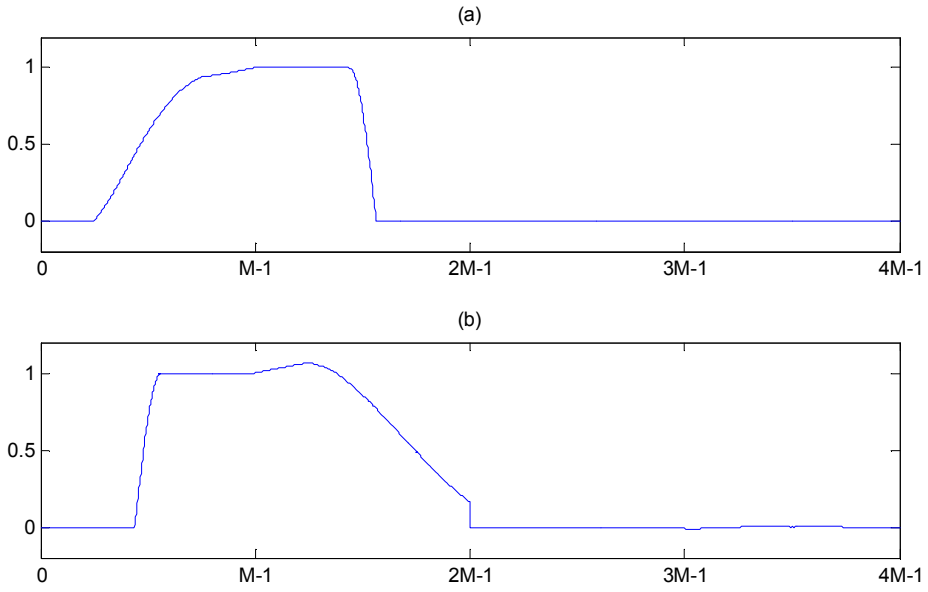


Figure 42 – LONG_START synthesis window $w_{synSTART}$ (a), LONG_STOP synthesis window $w_{synSTOP}$ (b)

In the following, the matrix notation is used as it represents the overlap operation in a compact fashion. Based on the previous definition of the direct transform in 3.1.3, the $4M \times M$ modulation matrix for the long transform is defined as:

$$\mathbf{T} = \begin{bmatrix} t_{0,-2M} & t_{0,1-2M} & \cdots & t_{0,2M-1} \\ t_{1,-2M} & t_{1,1-2M} & & \vdots \\ \vdots & & \ddots & \vdots \\ t_{M-1,-2M} & \cdots & \cdots & t_{M-1,2M-1} \end{bmatrix} \tag{4.21}$$

with the modulation function being defined as:

$$t_{k,n} = -2 \cdot \cos\left(\frac{\pi}{M}\left(n - \frac{M}{2} + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right) \tag{4.22}$$

for $0 \le k \le M$-1 and $-2M \le n \le 2M$-1.
The corresponding modulation matrix for the long inverse transform is then given by:

$$\mathbf{T}^{inv} = \begin{bmatrix} t^{inv}_{0,0} & t^{inv}_{0,1} & \cdots & t^{inv}_{0,M-1} \\ t^{inv}_{1,0} & t^{inv}_{1,1} & & \vdots \\ \vdots & & \ddots & \vdots \\ t^{inv}_{4M-1,0} & \cdots & \cdots & t^{inv}_{4M-1,M-1} \end{bmatrix} \tag{4.23}$$

with the modulation function for this inverse transform being defined as:

$$t_{n,k}^{inv} = -\frac{1}{M}\cdot\cos\left(\frac{\pi}{M}\left(n-\frac{M}{2}+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right)$$
$$t_{n,k}^{inv} = \frac{t_{k,n}}{2M} \tag{4.24}$$

for $0 \le n \le 4M\text{-}1$ and $0 \le k \le M\text{-}1$.

Following the convention as previously introduced in section 3.1.1.2, the notation of the input time signal of the direct transform is noted:

$$\mathbf{x}_t^{KM} = \left[x(tM), x(tM+1), ..., x(tM+KM-1)\right]^T \tag{4.25}$$

The reconstructed signal obtained after the consecutive direct and inverse transform operations without overlap and add is given by:

$$\tilde{\mathbf{x}}_t^{4M} = \mathbf{diag}(\mathbf{w}).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}).\mathbf{x}_{t-2}^{4M} \tag{4.26}$$

J.w represents the time-reversed version of the window **w** as expressed in the equations (3.44) and (3.45).
The matrix representing the combination of direct and inverse modulation is given by the product $\mathbf{T}^{inv}.\mathbf{T}$ which can be rewritten using the cosine functions orthogonality:

$$\mathbf{T}^{inv}.\mathbf{T} = \begin{bmatrix} -\mathbf{I}_M - \mathbf{J}_M & \mathbf{0} & \mathbf{I}_M + \mathbf{J}_M & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_M + \mathbf{J}_M & \mathbf{0} & \mathbf{I}_M - \mathbf{J}_M \\ \mathbf{I}_M + \mathbf{J}_M & \mathbf{0} & -\mathbf{I}_M - \mathbf{J}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M - \mathbf{J}_M & \mathbf{0} & -\mathbf{I}_M + \mathbf{J}_M \end{bmatrix} \tag{4.27}$$

Using the same example as previously used in equation (3.18) with M=4 and K=4, the matrix $\mathbf{T}^{inv}.\mathbf{T}$ takes the form given below. The two sub-matrixes identified in red are the components which represent the perfect reconstruction of the low delay transform. When (4.27) is used in equation (4.26) and after overlap and add of the four consecutive frames, all the sub-matrixes in black are cancelled and the components in red become equal to the identity matrix.

$$\mathbf{T}^{inv}.\mathbf{T} = \begin{bmatrix}
-1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix} \quad (4.28)$$

This example is used to illustrate the more complex time aliasing generated by the low delay transform. Equation (3.22) shows the time aliasing of the MDCT where the time aliasing is introduced based on only one frame for the first and second part of the window. It can be seen that due to the long prototype, the time aliasing terms of two frames are combined together, leading to a more complex aliasing cancellation operation. Based on these definitions, the low delay transition between a long asymmetric window for the low delay transform of size M with $K > 2$ and a short symmetric window for MDCT transform of size M_s with $K_s = 2$ can be derived.

In the normal case, the reconstructed signal obtained after the direct and inverse transform operation as well as overlap and add $\hat{\mathbf{x}}_t^M$ can be expressed as the sum of K components coming for the K consecutive inverse transforms and leading to the following equation:

$$\hat{\mathbf{x}}_t^M = \sum_{l=0}^{K-1} \tilde{\mathbf{x}}_{t-l}^{KM}\left[lM \ldots (l+1)M - 1\right] \quad (4.29)$$

$\tilde{\mathbf{x}}_t^{KM}$ is a vector with KM scalars as shown in (3.71). The values between [] indicate the first and the last scalar index in the vector. If $\hat{\mathbf{x}}_t^M$ is decomposed, it comes:

$$\hat{\mathbf{x}}_t^M = \sum_{l=0}^{K-1} \left[\mathbf{diag}(\mathbf{w}).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}).\mathbf{x}_{t-2-l}^{KM} \right]_{[lM \ldots lM+M-1]} \tag{4.30}$$

Now, considering a direct transition between long and short transforms, the reconstructed signal $\hat{\mathbf{x}}_t^M$ is expressed as follows:

$$\begin{aligned} \hat{\mathbf{x}}_t^M = & \sum_{l=2}^{K-1} \left[\mathbf{diag}(\mathbf{w}).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}).\mathbf{x}_{t-2-l}^{KM} \right]_{[lM \ldots lM+M-1]} + \\ & \left[\mathbf{diag}(\mathbf{w}_1).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}).\mathbf{x}_{t-3}^{KM} \right]_{[M \ldots 2M-1]} + \\ & \left[\mathbf{diag}(\mathbf{w}_2).\left[\left[\mathbf{0}_{2M} \quad \mathbf{P}_s^{\mathrm{T}}.\mathbf{P}_s \right] + \mathbf{C} \right].\mathbf{x}_{t-2}^{KM} \right]_{[0 \ldots M-1]} \end{aligned} \tag{4.31}$$

with $\mathbf{P}_s^{\mathrm{T}}.\mathbf{P}_s$ being defined as previously written in (3.50). In equation (3.77), the addition of K terms obtained from the K consecutive direct and inverse transform steps is still present. It should be noted that in that case the size of the matrix $\left[\mathbf{P}_s^{\mathrm{T}}.\mathbf{P}_s + \mathbf{C} \right]$ corresponding to the modification of the inverse transform step of the short windows is $2M \times 2M$, similarly to the normal low delay block switching.

Hence, in case of direct transition between different window sizes, the Perfect Reconstruction property is obtained if the following identity is verified:

$$\begin{aligned} & \sum_{l=2}^{K-1} \left[\mathbf{diag}(\mathbf{w}).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}) \right]_{[lM \ldots lM+M-1 \times 0 \ldots 4M-1]} + \\ & \quad \left[\mathbf{diag}(\mathbf{w}_1).\mathbf{T}^{inv}.\mathbf{T}.\mathbf{diag}(\mathbf{J}.\mathbf{w}) \right]_{[M \ldots 2M-1 \times 0 \ldots 4M-1]} + \\ & \quad \left[\mathbf{diag}(\mathbf{w}_2).\left[\left[\mathbf{0}_{2M} \quad \mathbf{P}_s^{\mathrm{T}}.\mathbf{P}_s \right] + \mathbf{C} \right] \right]_{[0 \ldots M-1 \times 0 \ldots 4M-1]} \\ & = \left[\mathbf{0}_M \quad \mathbf{0}_M \quad \mathbf{I}_M \quad \mathbf{0}_M \right] \end{aligned} \tag{4.32}$$

The modified long synthesis window $\mathbf{w}_1$ and long synthesis window $\mathbf{w}_2$, as well as the time inversion matrix $\mathbf{C}$ are then defined as a function of the modulations $\mathbf{T}^{inv}.\mathbf{T}$ and $\mathbf{P}_s^{\mathrm{T}}.\mathbf{P}_s$, and the analysis window $\mathbf{w}$.

The modified long synthesis window $\mathbf{w}_1$ and first short synthesis window $\mathbf{w}_{2s}$ are given in Figures 43 and 44 respectively. It should be noted that the synthesis windows have very similar shapes to the normal low delay block switching compensation windows.

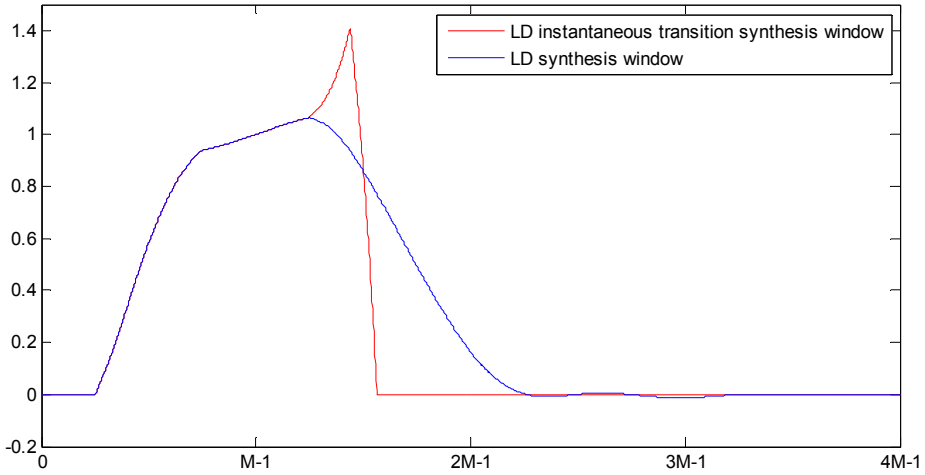


Figure 43 – Long synthesis window in normal operation (blue) and in case of low delay block switching $\mathbf{w}_1$ (red).

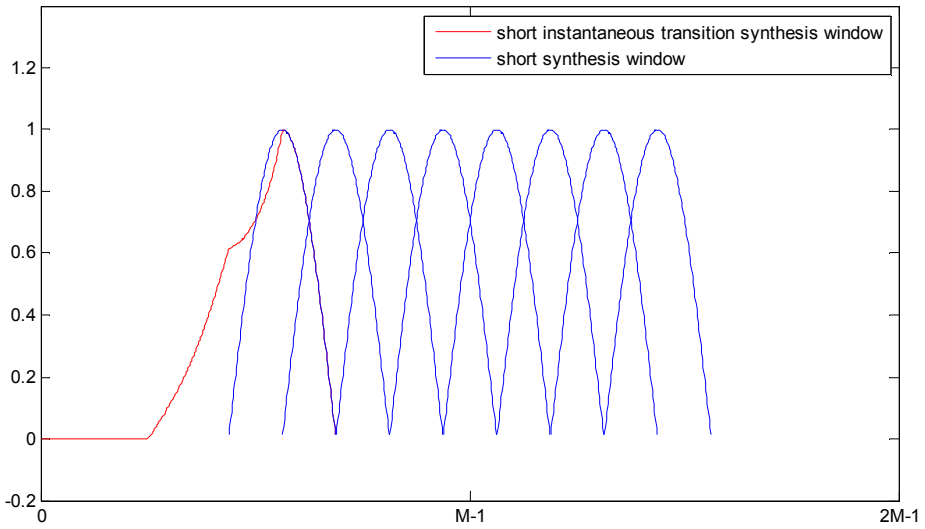


Figure 44 – Short synthesis window in normal operation (blue) and in case of low delay block switching $\mathbf{w}_{2s}$ (red).

4.2.3 Application of low delay block switching

Similarly to the low delay block switching illustrations presented in 4.1.6, this paragraph presents the window shapes and associated frequency responses in the context of low delay transform.

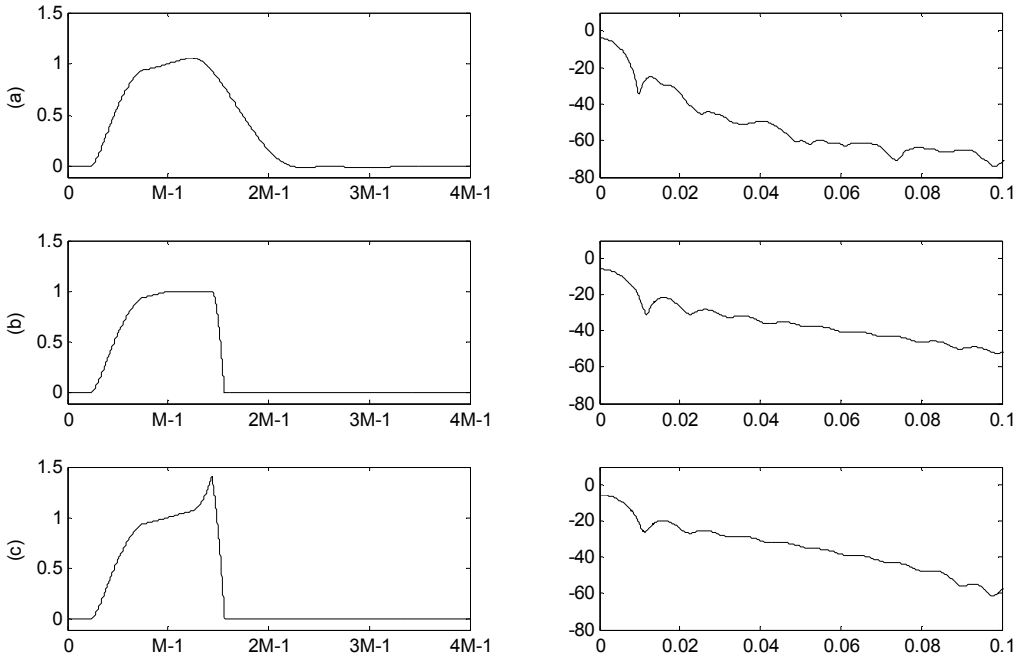


Figure 45 – Low delay synthesis window (a), Low delay synthesis window for normal transition between low delay and short sine window (b) and Low delay block switching synthesis window (c)

Figure 45 shows the low delay synthesis window (a), the synthesis window for the transition with short sine windows in case of traditional block switching (b), and finally the synthesis window obtained for the low delay block switching using low delay transform (c).

Figure 46 shows a comparison of the frequency responses between the three windows of Figure 45. Both transition windows offer lower performance in terms of stop-band attenuation compared to the low delay window. However, the performance of the traditional transition window is similar to the low delay block switching synthesis window with a second lobe attenuation which is reduced by 2 dB for the new synthesis window. As for the low delay block switching in MDCT case, it should be noted that the analysis is performed with the normal low delay window. As such the coding performance of the system is not degraded compared to the traditional block switching system. Actually, the fact that the analysis is done with the normal low delay window with better selectivity improves the coding stage.

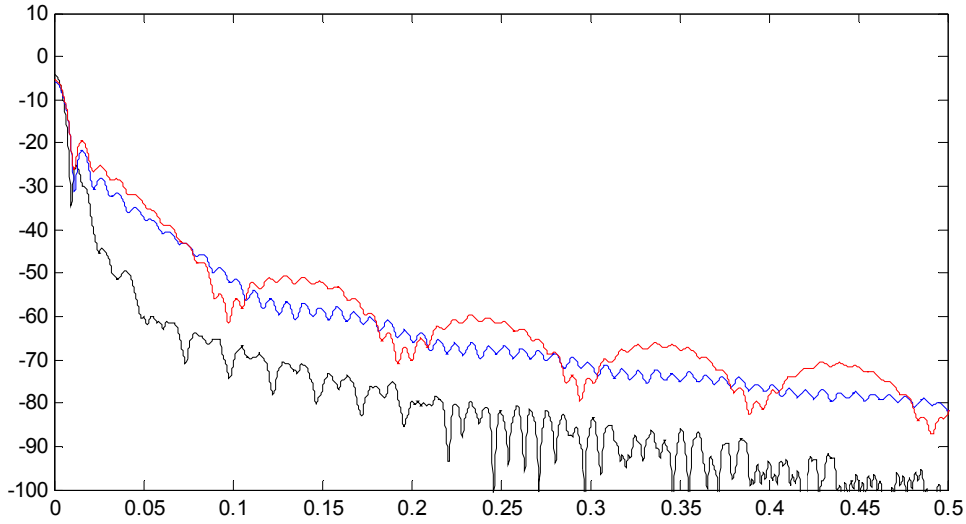


Figure 46 – Frequency responses (between 0 and 0.5) of low delay window (in black) low delay synthesis window for normal transition between low delay and short sine window (in blue) and low delay block switching synthesis window (in red)

4.3 Seamless reconstruction in MDCT

It has been seen in the previous sections that MDCT can be adapted to resolution changes with low delay block switching tool and perfect reconstruction can still be obtained even if non matching windows and transform types are used for transitions between two different frequency resolutions. In addition, the low delay transform and its application to low delay block switching has been introduced. The low delay window which is shown in Figure 31 has been extracted from an application to low delay audio coding. As explained in 3.1.3, it aims at solving the delay problem which was considered as inherent to the MDCT transform. In [Schnell 07], the authors justify the introduction of a new low delay filter banks by the unsolved problem of delay reduction with MDCT. According to the authors, the perfect reconstruction property cannot be achieved with traditional filter banks, like TDAC filter banks such as the MDCT, when trying to reduce their reconstruction delay. This is due to the fact that so far, the so-called paraunitary or orthogonal filter banks employ symmetric window functions and thus have a system delay identical to the window length minus one.

The purpose of this section is to introduce a new concept of MDCT transform and window design. This new MDCT window design method allows to obtain the perfect reconstruction even with non-matching windows and to design new low delay window compatible with the MDCT while maintaining the perfect reconstruction. A generalization of the concept, together with the low delay block switching algorithm presented in section 4.1, de-

fine a new framework for applications of MDCT transform in signal processing. In this section the focus is first set on the seamless reconstruction of the MDCT using non-matching windows and then in section 4.4, the delay reduction using some relaxation on the analysis and synthesis windows is introduced. It is shown that those relaxations lead to better low delay window prototypes and offer more flexibility in the design of MDCT window and MDCT processing system.

The MDCT definition provided in section 3.1.1 gives the most usual way to present the transform (see equation (3.1, 3.2)). The basis functions of the transform $p_{k,n}$ are defined as the multiplication of two components. The first one $w(n)$ is the window, also called filter bank prototype. And the second one $c_{k,n}$ represents the modulation functions. In the state of the art, only identical analysis and synthesis windows have been considered. This choice imposes the symmetry of the prototype as it will be described in this paragraph.
The conditions which have to be fulfilled by the window to achieve the Perfect Reconstruction have been reduced to the definition given in equation (3.5) and reminded below:

$$\begin{cases} w(n) = w(2M-1-n) \\ w^2(n) + w^2(M+n) = 1 \end{cases} \tag{4.33}$$

This set of equation leads to symmetric windows which are identical on both analysis and synthesis side as shown in Figure 47. It can be seen from this equation and the KBD example window provided in Figure 47 that any insertion of *zeros* in the first half of the synthesis window will automatically involve the insertion of the same number of *zeros* in the second half. Hence, reducing the delay implied the introduction of some *zeros* at both sides of the window and will automatically reduce dramatically the frequency resolution of the transform.
This set of constraints can however be relaxed to allow more flexibility on the window or filter bank prototype design. We will see in this section that this can be achieved by a redefinition of the Perfect Reconstruction equations.

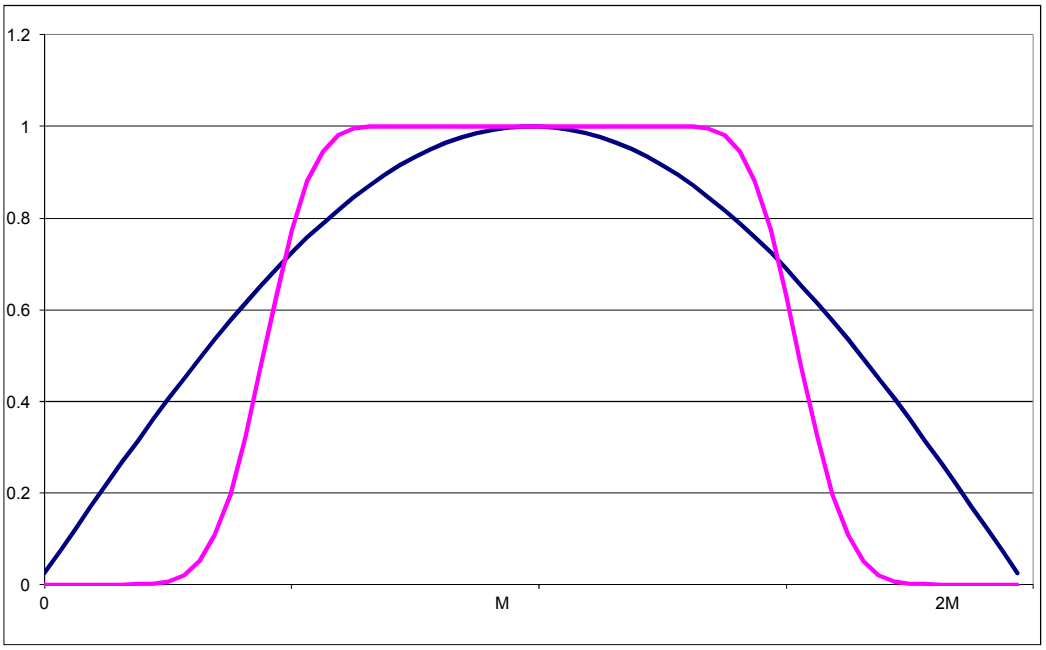


Figure 47 – Symmetric window for direct and inverse MDCT, the sine window (blue) and the Kaiser-Bessel derived (pink) windows are drawn

4.3.1 Relaxed Perfect Reconstruction equations

The MDCT transform is based on identical windows for the direct and the inverse transform. This constraint can be expressed by the following equation between the analysis $w_{ana}(n)$ and synthesis $w_{syn}(n)$ windows:

$$w(n) = w_{ana}(n) = w_{syn}(n) \tag{4.34}$$

4.3.2 Relaxation on the analysis window

The first relaxation consists in allowing the use of different analysis windows for two consecutive frames as introduced in 3.2.1. Hence the first and second halves of a window do not have to fulfil the symmetry constraint and are defined independently:

$$w_{ana}(n) \neq w_{ana}(2M - 1 - n) \tag{4.35}$$

where $2M$ is the size of the analysis window. Following this relaxation, the definition of the two halves of the consecutive analysis and synthesis windows will now be separated in two parts: the second half of the first analysis window $w_{ana1}(n+M)$ and the first half of the second analysis window $w_{ana2}(n)$. Following the same logic, the synthesis windows are defined in two parts: the second half of the first synthesis window $w_{syn1}(n+M)$ and the first half of the second synthesis window $w_{syn2}(n)$. However, we still have the constraint of equality between the half analysis window and its corre-

sponding half synthesis window ($w_{syn1}(n)$ = $w_{ana1}(n)$ and $w_{syn2}(n)$ = $w_{ana2}(n)$).
The rationale behind this separation between the two halves is that the time aliasing terms, which are introduced by the direct MDCT, are cancelled after inverse MDCT in the overlap-add over an M samples time frame (only one half of the corresponding windows) independently from the other halves of the two overlapping windows. The direct relation and constraint between the two halves of the legacy analysis window does not exist anymore if different windows can be used for consecutive frames.

This new window definition allows to define a set of windows that can be used with direct switching between different window shapes at the analysis side (direct transform).
Hence, at the encoding stage, the system can adapt the window shape to the audio input characteristics such to optimize a certain criterion depending on the input signal (e.g. maximization of the energy concentration or coding gain). The constraint on the first half of the analysis and synthesis window is imposed by the previous window and the selection of the best shape for the current frame imposes the window shape for the next window.

The relaxed Perfect Reconstruction equations define the relation between the second half of the first window and the first half of the following window. The relation between consecutive windows is now considered and the complete window is not defined directly. The new PR equations are then given by:

$$\begin{cases} w_{ana1}^2(n+M) + w_{ana2}^2(n) = 1 \\ w_{ana1}(2M-1-n)\,w_{ana1}(n+M) + w_{ana2}(M-1-n)\,w_{ana2}(n) = 0 \end{cases} \tag{4.36}$$

with $0 \le n \le M-1$.

If we consider the traditional solution which consists in imposing $w_{ana1}(n+M)=w_{ana2}(M-1-n)$ with $0 \le n \le M-1$, the solution which satisfies the PR equations will impose the symmetry of the windows as defined in the state of the art. However, in the normal operating mode of those perfect reconstruction conditions, one can select a set of windows allowing to modify windows over time. The simple example is the transition between a low overlap window and a long sine window as illustrated in Figure 48. The transition window is designed with two independent halves such to obtain the perfect reconstruction with previous and following windows.

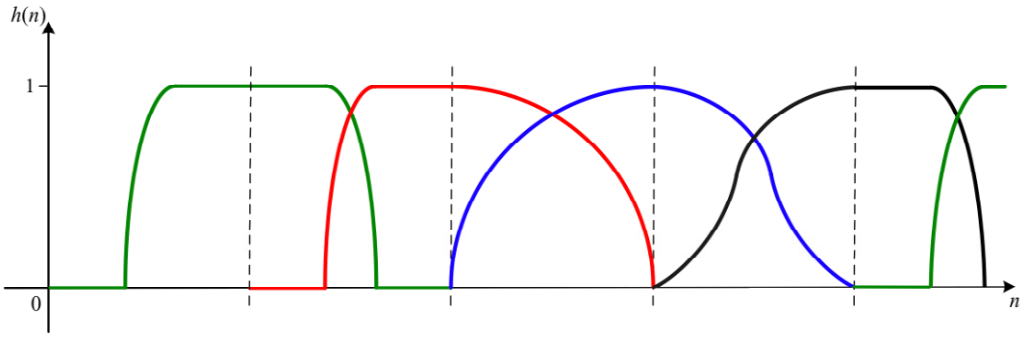


Figure 48 – Example of changing window shapes for two consecutive frames

Figure 48 illustrates this concept of independent definition of each half of a window in consecutive frames processing. The first and second halves of a window can be defined independently. This method is used in AAC for switching from sine windows to KBD shapes and vice versa.

4.3.3 Relaxation on the analysis/synthesis windows relationship

In order to really improve flexibility in the choice of the complete analysis window for each processed frame we introduce a second relaxation step corresponding to the biorthogonality introduced in [Cheung 95]. We propose to use a relaxation of both analysis and synthesis windows in the PR definition. This approach provides similar results to the polyphase approach introduced in [Schuller 00] while being based on a different representation. The proposed approach relies on the MDCT transform using a window representation, while [Schuller 00] uses a filter bank representation based the polyphase decomposition. However, as already stated, both approaches are equivalent. Here, following our window based representation, the analysis window will not have to be identical to the synthesis window. This relaxation can be expressed by:

$$w_{ana}(n) \neq w_{syn}(n) \tag{4.37}$$

Hence, the new Perfect Reconstruction must be written based on two halves analysis window definitions and two halves synthesis window definitions. Those equations are derived from equation (3.20) and (3.21) considering independent definitions of analysis and synthesis windows for the two parts of the reconstruction matrix. The PR equations are now defined by:

$$\begin{cases} w_{ana1}(n+M)w_{syn1}(n+M)+w_{ana2}(n)w_{syn2}(n)=1 \\ w_{ana1}(2M-1-n)w_{syn1}(n+M)+w_{ana2}(M-1-n)w_{syn2}(n)=0 \end{cases} \tag{4.38}$$

with $0 \le n \le M-1$.

From equation (4.38), the relation existing between the analysis and the synthesis windows is deduced. Using a matrix notation:

$$\begin{bmatrix} w_{ana1}(n+M) & w_{ana2}(n) \\ w_{ana1}(2M-1-n) & w_{ana2}(M-1-n) \end{bmatrix} \begin{bmatrix} w_{syn1}(n+M) \\ w_{syn2}(n) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4.39}$$

If the matrix determinant is not equal to zero for all n, which is expressed by:

$$D(n) = w_{ana1}(n+M)\,w_{ana2}(M-1-n) + w_{ana2}(n)\,w_{ana1}(2M-1-n) \neq 0 \tag{4.40}$$

The system of (4.39) can be inversed and leads to:

$$\begin{bmatrix} w_{syn1}(n+M) \\ w_{syn2}(n) \end{bmatrix} = \frac{1}{D(n)} \begin{bmatrix} w_{ana2}(M-1-n) & w_{ana2}(n) \\ w_{ana1}(2M-1-n) & w_{ana1}(n+M) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{4.41}$$

The two halves of the consecutive synthesis windows are then defined by:

$$\begin{cases} w_{syn1}(n+M) = \dfrac{w_{ana2}(M-1-n)}{D(n)} \\ w_{syn2}(n) = \dfrac{w_{ana1}(2M-1-n)}{D(n)} \end{cases} \tag{4.42}$$

with $0 \le n \le M-1$. Thus this definition of $w_{syn1}(n+M)$ and $w_{syn2}(n)$ will satisfy the PR conditions. It can be noticed that the two halves of the synthesis windows are simply obtained from the time-reversed version of the two halves of the analysis windows. The correction function $D(n)$ ensures that the complete system satisfies the PR conditions. It should be noted that $D(n)$ is a symmetric function as $D(n) = D(M\text{-}1\text{-}n)$. This limited number of correction coefficients shall be computed or stored for all possible transitions between non-matching analysis windows. Hence, only the synthesis windows are corrected to achieve the perfect reconstruction, the analysis system can independently select different window shapes for consecutive frames.

Figure 49 illustrates the use of two consecutive analysis windows which are not identical for the frame which is defined between samples M and $2M$. In Figure 49 (b), the corresponding synthesis windows are given.

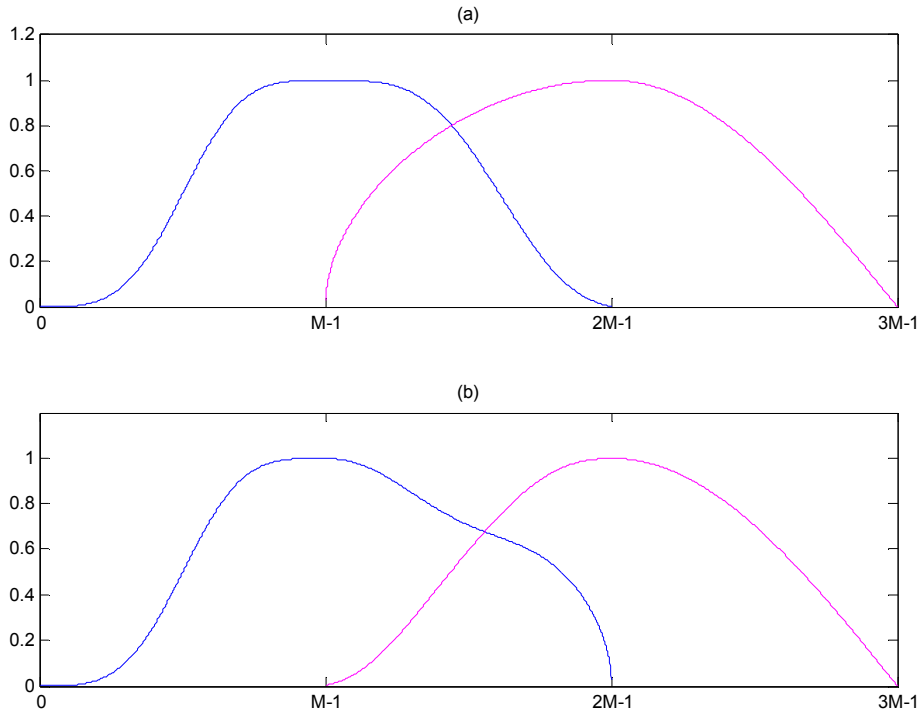


Figure 49 – Example of changing window shapes for two consecutive frames: (a) analysis windows – (b) synthesis windows

Based on the new definition of the Perfect Reconstruction conditions defined in this section, a transform coding algorithm with more flexibility can be designed. Indeed, each analysis window can be selected independently of the neighbour frames.
In addition, this new set of PR equations can be used to design a new family of asymmetric low delay windows which is described in 4.4.

4.4 Low delay MDCT window

In this section, a new method to define low delay windows based on the perfect reconstruction constraints is introduced using the seamless reconstruction method described in section 4.3.3. In order to reduce the delay associated with MDCT transform, one can use low overlap windows as defined in equation (3.10) and shown in Figure 25. This symmetric window is characterised by M_z coefficients at the beginning and at the end which are equal to zero. Thus, the combination of identical analysis and synthesis low overlap windows allows to reduce the total delay by $2M_z$ such that it is then defined by $2M - 2M_z - 1$. However, due to the reduced size of the window, the associated frequency selectivity is widely reduced. Overall, the performance of a system using the low overlap window is degraded compared to the sine window. Based on the newly introduced PR conditions, we propose a low delay window design algorithm.

4.4.1 Low delay window design

The solution which is defined in this section is based on the same relaxed PR conditions as described in equation (4.38). Hence the second halves of the analysis and synthesis windows are not constrained to be the time-reversed version of the first half. This is explicitly defined by $w_{ana1}(n+M) \neq w_{ana2}(M-1-n)$. As for the low overlap window, the reduction of delay is obtained by zeroing a portion of the analysis and synthesis windows. More specifically, $w_{ana1}(n+M)$, which is the second half of the analysis window, will contain M_z zeros in the range $M - M_z \leq n \leq M - 1$. As expressed before, $D(n)$ shall then be non zero and this condition is verified with $M_z < M/2$.
Following the definition of the synthesis window according to equation (4.42), the first M_z coefficients of the derived synthesis window are consequently equal to zero. It should be noted that this definition of the low delay window will advantageously limit the zeroed portion to the beginning of the synthesis window and to the end of the analysis window. However, using directly the equation (4.42) will not ensure that the analysis and synthesis windows are the time-reversed version of each other. Hence, the analysis and synthesis windows offer different frequency selectivity and noise shaping characteristics which give more flexibility in the system design but might not be suitable for an audio coding algorithm. Indeed, the time-reversed relation between analysis and synthesis windows allows to reduce the memory requirement of a communication codec.
In order to limit this effect and to be able to define a single prototype window that can be used for direct and inverse MDCT transform, we introduce a simple procedure which defines a new family of low delay windows. This second constraint is important for conversational applications as the encoder and decoder are both used in a terminal and using a single prototype for the analysis and synthesis windows allows to reduce the required memory to store the prototype. For instance, with large sampling frequency, the window size is usually selected between 640 and 1024 samples. A different window for encoder and decoder would lead to a significant memory increase.

First, the analysis window is chosen with the corresponding region with coefficients equal to zero according to the definition given above. Then, this initial analysis window is normalized by the squared root of the correction function $D(n)$.

A simple example of the design algorithm is provided below. This can be applied to generate the analysis or synthesis window. It should be noticed that the other window (synthesis and analysis respectively) is straightforwardly obtained as the time-reversed version of the initial one. This example is given starting with the design of a synthesis window.

As explained above, in an initialization step, a pre-synthesis window is defined with M_z zeros at the beginning. The general definition of the initialization window is given by:

$$w(n) = \begin{cases} h_{2M-M_z}(n) & ,0 \le n \le 2M - M_z - 1 \\ 0 & ,2M - M_z \le n \le 2M - 1 \end{cases} \tag{4.43}$$

with $h_{2M\text{-}Mz}$ being a shorter window of size $2M - M_z$. This window can be selected arbitrarily, and the sine, KBD or hanning windows are suitable. The initialization window can for instance be based on a reduced sine window with:

$$w(n) = \begin{cases} \sin\left[\dfrac{\pi}{2M - M_z}\left(n + \dfrac{1}{2}\right)\right] & ,0 \le n \le 2M - M_z - 1 \\ 0 & ,2M - M_z \le n \le 2M - 1 \end{cases} \tag{4.44}$$

This initial synthesis window is shown in blue on the Figure 50. The correction factors are then applied to this initialization. The correction function is defined by:

$$\Delta(n) = \sqrt{w(n)w(2M-1-n) + w(n+M)w(M-1-n)} \neq 0 \tag{4.45}$$

with $0 \le n \le M - 1$. And the final synthesis window is obtained by:

$$\begin{cases} w_{syn1}(n+M) = \dfrac{w(n+M)}{\Delta(n)} \\ w_{syn2}(n) = \dfrac{w(n)}{\Delta(n)} \end{cases} \tag{4.46}$$

with $0 \le n \le M - 1$. This final low delay synthesis window is shown in pink on Figures 50 and 51. It should be noticed that this synthesis window will generate a correction function $D(n) = 1$ for all n. Hence, the analysis window is directly obtained as the time-reversed version of the synthesis window. Figure 51 illustrates the analysis and synthesis windows obtained by the proposed low delay window design algorithm. We called this window the Asymmetric Low Delay (ALD) window.

It should be noted that in the original window definition given in equation (3.5), the number of degrees of freedom is limited to M due to the symmetry. However, with the ALD, the number of degrees of freedom is increased to $2M\text{-}2M_z$.

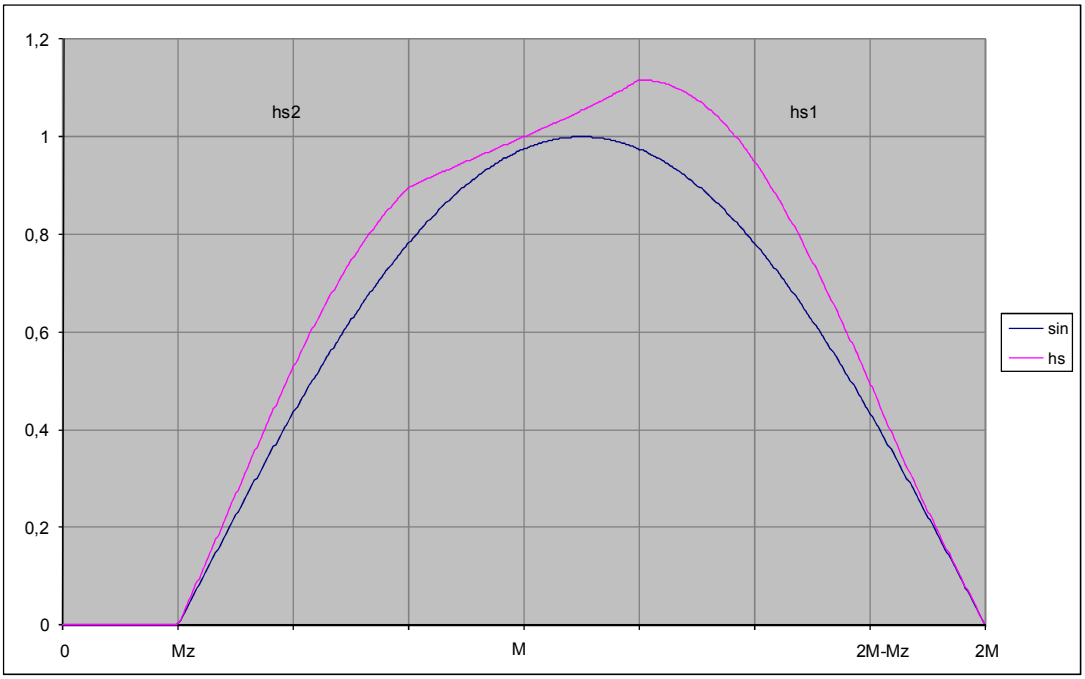


Figure 50 – Synthesis window initialization (blue) and final synthesis window after correction (pink)

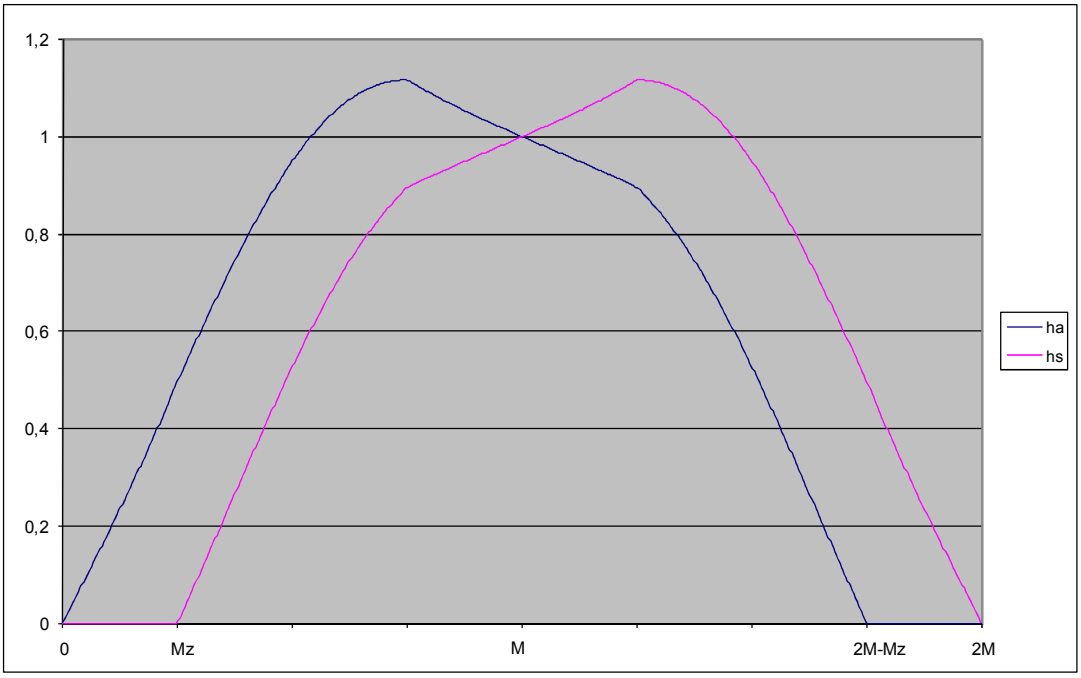


Figure 51 – Low delay analysis (blue) and synthesis (pink) windows

4.4.2 Discussion on the low delay MDCT window

In this section, we provide the results of the evaluation of the objective performance for the proposed low delay MDCT window. Two main measures are used for this purpose: the coding gain and time-frequency localization. The coding gain is defined in [Malvar 92b] as:

$$G = \frac{\sigma_x^2}{\left(\prod_{k=0}^{M-1} \sigma_{x_k}^2\right)^{1/M}} = \frac{\frac{1}{M}\sum_{k=0}^{M-1} \sigma_{x_k}^2}{\left(\prod_{k=0}^{M-1} \sigma_{x_k}^2\right)^{1/M}} \tag{4.47}$$

The numerator corresponds to the average energy, which is identical to the input signal variance due to the energy conservation of an orthogonal transform. The coding gain is usually given for an AR(1) signal with ρ = 0.95. In order to compute the coding gain of such signal, we fix $\sigma_x^2 = 1$. Then, based on the autocorrelation matrix which is given by:

$$\mathbf{R}_{xx} = \begin{bmatrix} 1 & \rho & \cdots & \rho^L \\ \rho & 1 & & \vdots \\ \vdots & & \ddots & \rho \\ \rho^L & \cdots & \rho & 1 \end{bmatrix} \tag{4.48}$$

The sub-band signal variance is derived by:

$$\sigma_{x_k}^2 = \mathbf{h}_k^T \mathbf{R}_{xx} \mathbf{h}_k \tag{4.49}$$

For biorthogonal transform, the adaptation of the coding gain computation is given in Annex D. The second measure is the time-frequency localization. Several definitions of the time-frequency localization measures have been proposed in the literature. All time-frequency localization measures are used to quantify the dispersion in time and frequency of a discrete time sequence or a continuous time function. Whatever the measure being used, concentrating the energy around the function or sequence centre increases the Time Frequency Localization. Temporal resolution (respectively frequency resolution) represents the time localization accuracy (respectively frequency localization) of the coefficients. These localizations are quite important in coding to shape the coding noise according to frequency masking for stationary signals or according to temporal masking for transient sounds. One commonly used definition of the Time-Frequency Localization for the filter design is given by:

$$\xi = \frac{1}{4\pi\sqrt{\sigma_t^2 \sigma_f^2}} \tag{4.50}$$

where σ_t^2 and σ_f^2 are the second order moments in time and frequency respectively.

Table 3 gives the comparison of the coding gain and time-frequency localization for several windows and transforms. For all the windows, the trans-

form size is M = 512. The best coding gains are obtained for the symmetric windows (Sine, ELT). It should be noted that the difference in performance between sine and ELT is not so significant. The two asymmetric windows offer lower performance in coding gain for AR(1) signal, this can be explained as illustrated in Annex B by the fact that the zeros introduced in the prototype desynchronize the analysis and synthesis band-pass filters. For the first sub-band, which are the most important for an AR(1) signal, this de-synchronization is particularly significant. In terms of time-frequency localization, the sine window offers the best performance, the two low delay windows providing different time/frequency resolution trade-off. The ELT is significantly worse due to the time spreading coming from the window length.

Window type	Coding Gain: G (dB)	TF localization ξ
Sine window	10.1095	1.102078264
ELT window (K=4)	10.1063	15.62724071
Low Delay Transform window (K=4)	9.4350	1.492345173
ALD window	9.2282	1.636096578

Table 3 – Comparison of the performance of the MDCT Sine window, MDCT ALD window, ELT and Low Delay Transform with M=512

Tables 4 and 5 compare the coding gains of different transform sizes with the theoretical maximum (defined for brick wall band pass filters). It can be seen that the sine window achieves almost the maximum coding gain for all sizes. As explained above, the coding gain performance of the ALD window is limited for the AR(1) signal and this is illustrated in Table 5. G Max corresponds to the theoretical maximum coding gain as given in [Malvar 92b].

M	G sine	G Max
32	10.0261	10.0279
64	10.0869	10.0872
128	10.1040	10.1040
256	10.1085	10.1085
320	10.1090	10.1090
512	10.1095	10.1095

Table 4 – Comparison of the performance of the MDCT sine window with the maximum theoretical coding gain

M	*G* ALD	*G* Max
32	9.1605	10.0015
64	9.2124	10.0678
128	9.2252	10.0985
256	9.2279	10.1079
320	9.2281	10.1086
512	9.2282	10.1091

Table 5 – Comparison of the performance of the MDCT ALD window with the maximum theoretical coding gain with $M_z = M/4$

The last experiment consists in evaluating the Segmental SNR of the database described in section 5.2 based on an optimal SNR allocation. Four window types have been compared and the results are given in Table 6. From those results, it can be seen that apart from the sine window which achieves the best Segmental SNR reconstruction, the Low Overlap and ALD windows offer similar results with a different trade-off between time and frequency resolution. The Low Overlap window being slightly shorter offers better performance for transient signals like found in speech, mixed content and some music signals. On the contrary, the ALD provides better frequency resolution which can be seen in the Stationary signal category. Finally, the low delay transform does not provide any improvement and gives even slightly worse results which are mainly due to the long prototype which strongly reduces the time resolution.

Category	Low Overlap Window	Sine Window	ALD Window	Low delay transform
Mixed Speech	61.34	62.02	60.97	58.41
Mostly Stationary	56.85	59.26	57.68	56.88
Music	55.34	55.67	54.66	54.29
Speech	53.06	53.28	52.13	51.98
Total	55.95	57.12	55.84	55.14

Table 6 – Comparison of the Segmental SNR (dB) performance of the MDCT ALD window with the maximum theoretical coding gain with M=512 and M/4 zeroes

Finally, Figure 52 provides the evolution of the coding gain for the ALD window which is designed based on the sine window as given in equations (4.44) and (4.46). Hence, for no delay reduction, the coding gain is equal to the $2M$ sine window performance. The performance is then degraded exponentially with the delay reduction. Hence, the selection of the adapted M_z for the ALD window depends on the required delay. M and M_z can allow to obtain the desired trade-off between the performance and delay reduction.

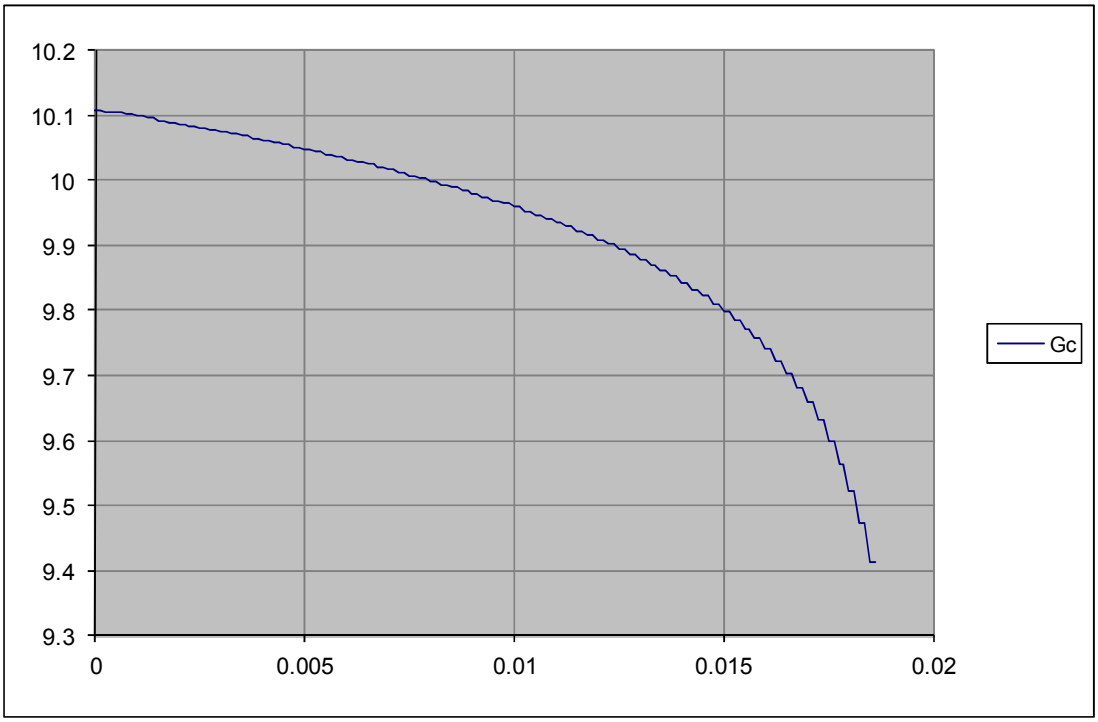


Figure 52 – Coding gain evolution with the delay reduction in ms (M_z)

4.5 Conclusion

In this chapter, the theoretical basis of the new low delay transform tools has been introduced. First, the low delay block switching allows a direct transition between long and short transform size without the introduction of transition windows. This tool has been adapted to the MDCT and the low delay transform defined in [Schuller 00]. This adaptation of the reconstruction process in time-varying transform has then been extended to cover all MDCT window transitions. By combining the relaxation of [Edler 89] and [Cheung 95], a new perfect reconstruction framework has been derived, leading to the introduction of more flexibility in a system based on the MDCT. This method is called seamless reconstruction. Finally, based on this method a new low delay window design procedure has been defined. Those new tools offer more flexibility in the selection and adaptation of the transform for low delay audio coding. It is expected that a significant quality improvement can be obtained when using those methods in a real system, leading to approaching the quality of well established audio coding standards while adding the low delay constraint.

Chapter 5

Application of the proposed filter bank design in low delay audio coding

In order to evaluate the proposed method for low delay block switching in real life low delay audio coding, several experiments based on Low Delay AAC (LD-AAC) and Enhanced Low Delay AAC (ELD-AAC) were conducted and are presented in the first part of this Chapter. The LD-AAC [Allamanche 99] is the initial low delay version of the AAC [ISO 09]. This extension of AAC has been standardized in 2000 with the goal to reduce the source of delay. For this purpose, the block switching has been removed and the frame length has been reduced to 512 or 480 samples. For a 48 kHz input signal, the minimum total algorithmic delay of the LD-AAC is then 20 ms if the bit reservoir is not used. The bit reservoir allows to hold part of the next frame's audio data in order to temporary change the effective bit rate. It is usually not used in low delay applications as it implicitly introduces an additional delay. The Enhanced Low Delay AAC (ELD-AAC) is the last evolution of the low delay version of the MPEG-4 Audio toolbox [ISO 09]. The ELD-AAC is based on the LD-AAC codec with the addition of the SBR module allowing to achieve a better quality for low bit rate and with the replacement of the MDCT by the Low Delay Transform as described in section 3.1.3 [Schnell 08]. The minimum total algorithmic delay of the ELD-AAC is 15 ms and is obtained when the SBR tool is not used at 48 kHz sampling frequency. With SBR, the algorithmic delay is increased to 31.3 ms, the SBR tool operating at this initial sampling frequency and the AAC core at half the sampling frequency.

The second part reports some experiments which have been conducted in order to evaluate the impact of the seamless reconstruction method when no delay constraint is specified. It means that only the ability to directly switch from one window to another window without transition is tested.

In the third part of the Chapter, the low delay window, which has been designed according to the proposed seamless reconstruction method, has been evaluated in the context of a scalable speech and audio coder standardized in ITU-T [ITU-T G.718 08][Vaillancourt 08]. In this coder, the ALD MDCT window has been evaluated compared to the normal sine window with subjective testing.

This work has been conducted in the course of MPEG standardization. In such a work, the encoder is not available and needs to be designed to demonstrate the performance of a proposed tool. Consequently, an extensive development work has been carried out in order to achieve a state-of-the-art quality for the MPEG AAC encoders which have been used for testing the proposed method. Besides the development of a high quality Low Delay AAC encoder which was the minimum requirement in order to start this evaluation, the new tools, which have been added for the validation of the low delay block switching, have been extensively tuned in this new context. For instance, the number and size of short windows have been investigated and the grouping algorithm for the short transform coefficients quantization has been adapted in the LD-AAC and ELD-AAC. But many tools have required a specific tuning for the low delay codecs.

5.1 Low delay block switching in MPEG low delay audio coding

In order to validate the proposed low delay block switching scheme in a real environment, it has been integrated in the low delay MPEG audio coding standards and several subjective listening tests have been carried out. In the following sections, the context of those experiments is presented before the quality assessment of the proposed approach with subjective listening tests.

5.1.1 A rationale for block switching in low delay audio coding

It is widely acknowledged that care has to be taken in order to deal with non stationary sounds in a transform audio coding scheme. For this purpose, two techniques have been standardized in MPEG-4: the Temporal Noise Shaping (TNS) as briefly introduced in 3.2.3 and the Block Switching as described in 3.2.1. These techniques are complementary and TNS can be used along with short windows in order to further improve the efficiency and quality.
TNS has the advantage of perceptually shaping the quantization noise in time in order to avoid pre-echoes.
The effect of block switching is to increase the time resolution. Consequently, a better quantization noise spreading with respect to temporal

masking effects can be achieved. Finally, block switching has the other advantage to increase the transform energy concentration property when dealing with transients.
Both tools are already part of the MPEG Advanced Audio Coding toolbox (AAC Main, and AAC Low Complexity) [ISO 09]. It has been shown that they were particularly suited when combined with Spectral Band Replication (SBR) in MPEG-4 High Efficiency-AAC, and especially in the case of transients [ISO 03]. These tools become particularly important in that context: when HE-AAC operates in a dual rate fashion, the internal sampling frequency for AAC becomes lower, 24 kHz being a widely used value. When lower rates are desired, the sampling rate can decrease down to 16 kHz.
In terms of temporal resolution, as the frame length for Low Delay (LD) AAC is 480 (or 512) samples, the quantization noise will be spread over the synthesis window support, that is $2 \times 480/24000 = 40$ ms.
Since block switching is not allowed for legacy LD AAC objects [Allamanche 99], it means that the transients are handled only by the TNS tool. The quantization noise spreading is too important especially for sounds that exhibit stationary periods of less than 10 ms, which is frequent, e.g. for speech signals.
All these arguments are in favour of reintroducing shorter windows, associated with the block switching technology in the Low Delay AAC and Enhanced Low delay AAC codecs such to improve the quality for non stationary sound items.

5.1.2 Application to MPEG-4 Low Delay AAC

MPEG-4 Low Delay AAC was standardized in 1999. This communication codec is generic in the sense that it is not specialized with respect to the audio content. Low Delay AAC retains the structure of standard AAC, with the two modifications being:
A shorter frame size: reduced from 1024 to 512 (respectively reduced from 960 to 480) samples resulting in a delay reduction by a factor of two.
Block switching is not retained for this scheme in order to avoid look-ahead buffers as illustrated in 3.2.2. This additional source of coding delay was consequently removed.

Reducing the frame size and removing the adaptive time frequency transform obviously lowers the audio quality. The experiment reported in section 5.2.2 gives an indication on how to quantify this degradation. This is the price to pay for having a codec applicable to conversational application. It was reported in [N3075 99] that an 8 kbit/s bit rate increase is necessary for LD-AAC to obtain the same quality as AAC. This was demonstrated at

32 and 64 kbit/s mono for LD-AAC which was compared to AAC operating at 24 and 56 kbit/s.

This thesis proposes to improve the Low Delay AAC quality using the low delay block switching tool. In order to do so, the standard AAC block switching procedure is reused: when a transient occurs, the transform size is divided by 8. Here the MDCT transform is reduced from 512 frequency components to 64. As such, higher temporal accuracy is ensured for non-stationary sounds (the quantization noise spreading is restrained to 5 ms).
In order to check whether block switching can improve LD-AAC with the same operating delay, the proposed compensation method for allowing direct switching without transition windows is applied. As mentioned before, doing so removes the look-ahead buffer which is essential for transform size decision.
Note that the necessary look-ahead is not negligible for communication codecs: for instance in LD-AAC, for a 24 kHz sampling frequency, the look-ahead would represent an addition of around 10 ms to the original algorithmic delay of 40 ms (for M=480).

5.1.3 Introduction to the low delay block switching in LD-AAC

In section 3.2.2, two examples where non-stationary sounds need special processing were discussed. In these two examples the encoder receives a new frame (called current frame) and has to transmit it instantaneously in order to minimize the delay. This means that the signal is considered as unknown after the current frame, that is, no look-ahead buffer can be used for shorter delays.
In the first example illustrated in Figure 33, an attack arising in the second half of the current frame (last quarter of the window) is considered. In this configuration, the encoder can switch to a transition window, the aim being to better concentrate the transient noise spreading in the second half, where the attack lies. Having used a long window at the previous frame, the encoder can select a long-short window for the current frame, anticipating that the next one would be processed using eight short windows.
In this first example, block switching can be used without any additional delay or look-ahead buffer.

In the second example given in Figure 34, the attack arises in the first half of the current frame (between $3M$ and $4M$). With a look-ahead buffer, an encoder would have anticipated this attack and would have switched to a transition window as soon as the previous frame. This way, the current frame would have been encoded using short windows.
Without look-ahead buffer, the current frame cannot be encoded using short windows in traditional block switching tool. The best that the block

switching tool can do is to introduce a transition window as illustrated in Figure 53.
However, as it has been demonstrated in 4.1, the perfect reconstruction can still be kept using the proposed low delay block switching method: a direct transition between a long window and eight short windows is permitted as illustrated on Figure 54.

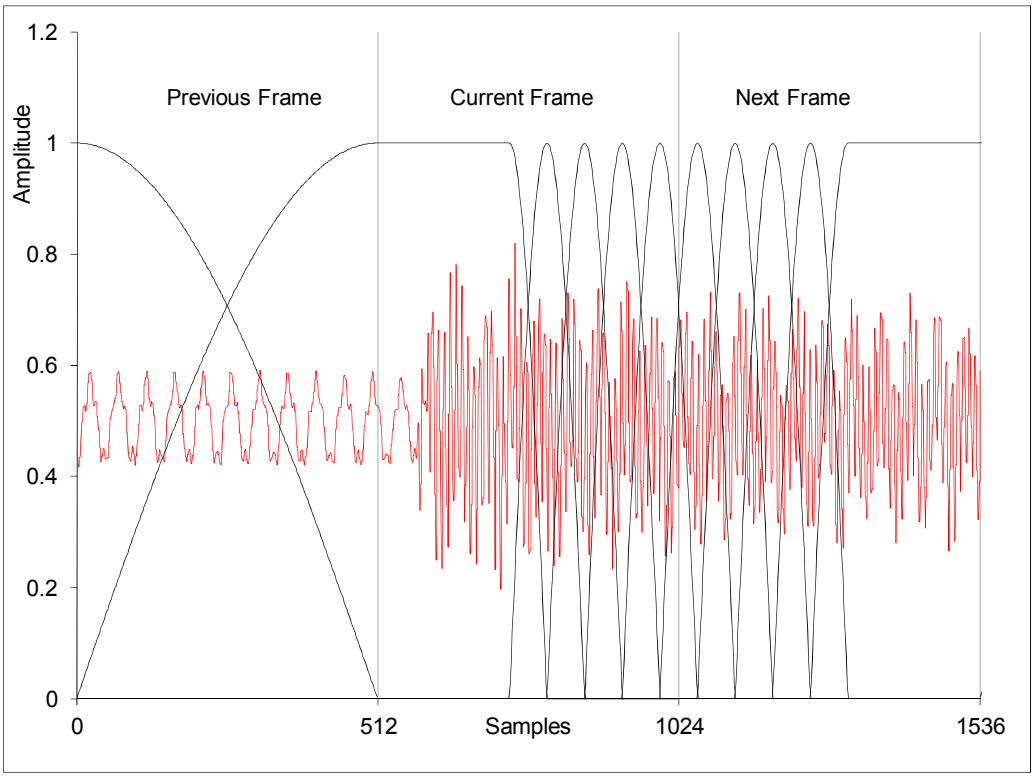


Figure 53 – Attack arising in the first half of the current frame

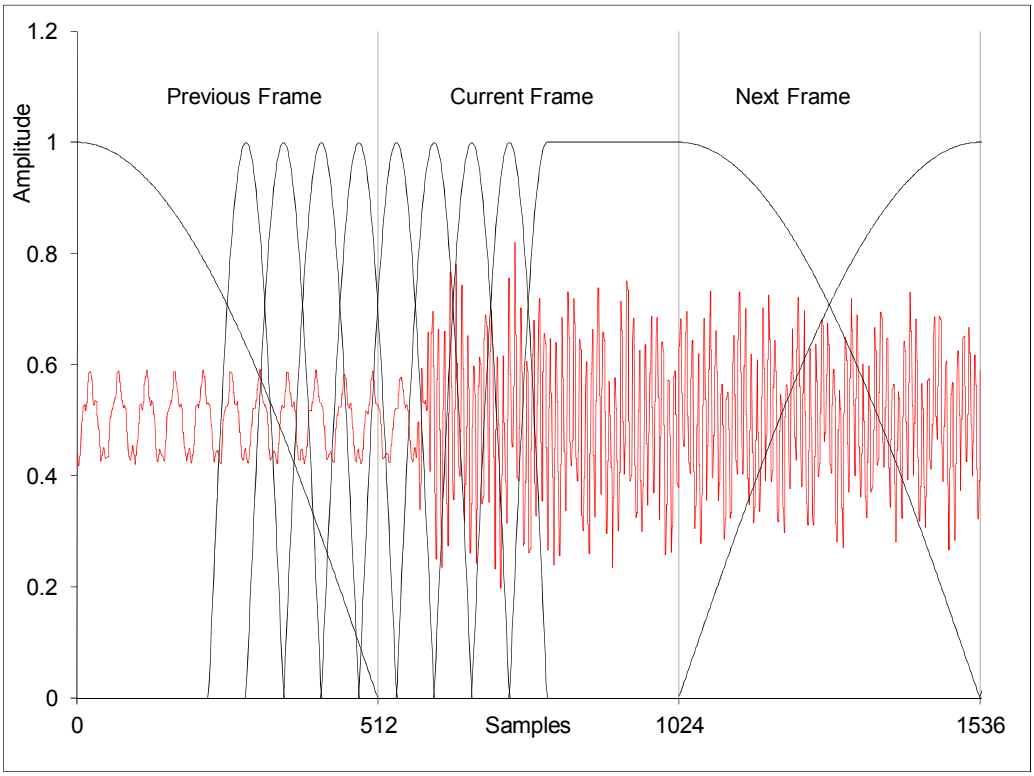
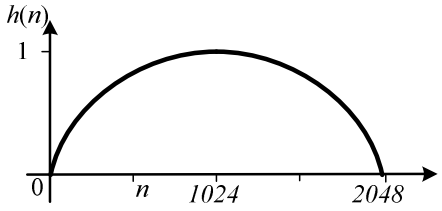
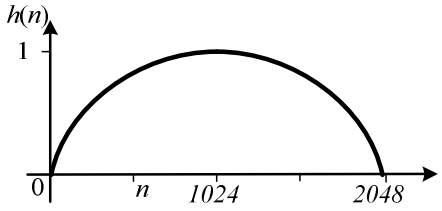
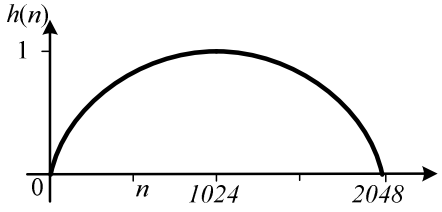
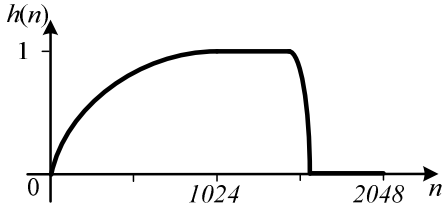
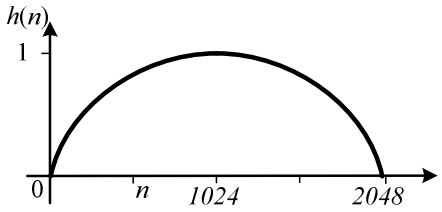
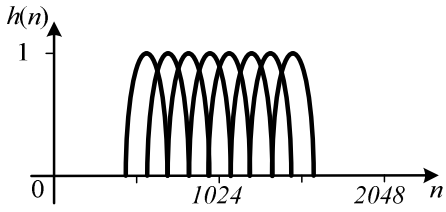
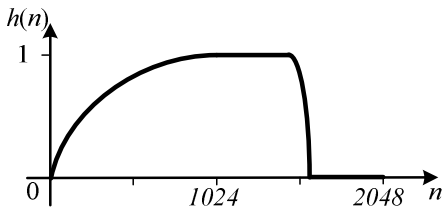
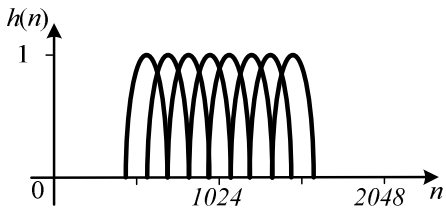
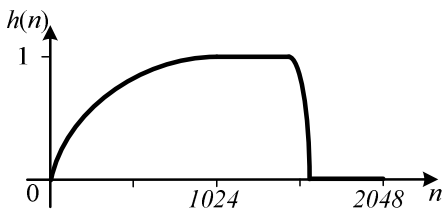
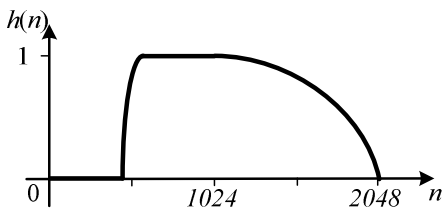
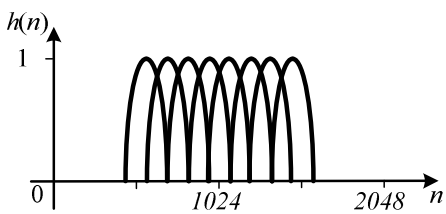
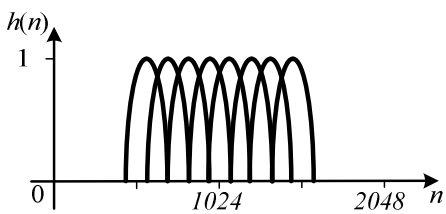
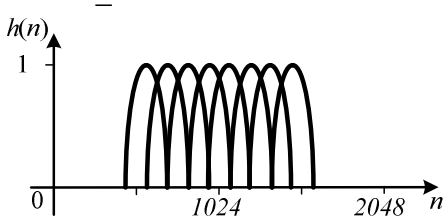
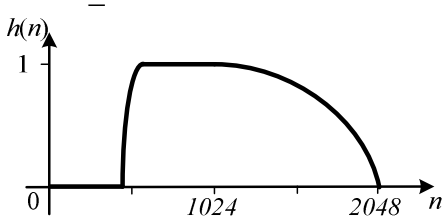


Figure 54 – Direct transition between long window and the eight short windows in low delay block switching

If short windows are directly used in the current frame, pre-echoes are avoided. Nevertheless that direct use of short windows means that transition windows cannot be inserted since the encoder could not anticipate the attack in the previous frame: the previous frame has already been emitted.
It is worth noting that direct switching from long to short is only needed for cases as illustrated in Figure 54. For the configuration described on Figure 33, the attack being detected in the current frame, the transition window can be used without additional delay.
Albeit theoretically feasible, direct switching from short to long windows without proper transition windows is not necessary for low delay audio coding as the look-ahead problem is not meaningful in that case. Table 7 lists the supported transition between windows. The transition noted in bold red indicates the new transition introduced by the low delay block switching.

Previous block	Current Block
LONG 	LONG 
LONG 	LONG_START 
LONG 	**EIGHT_SHORT** 
LONG_START 	EIGHT_SHORT 
LONG_START 	LONG_STOP 
EIGHT_SHORT 	EIGHT_SHORT 
EIGHT_SHORT 	LONG_STOP 

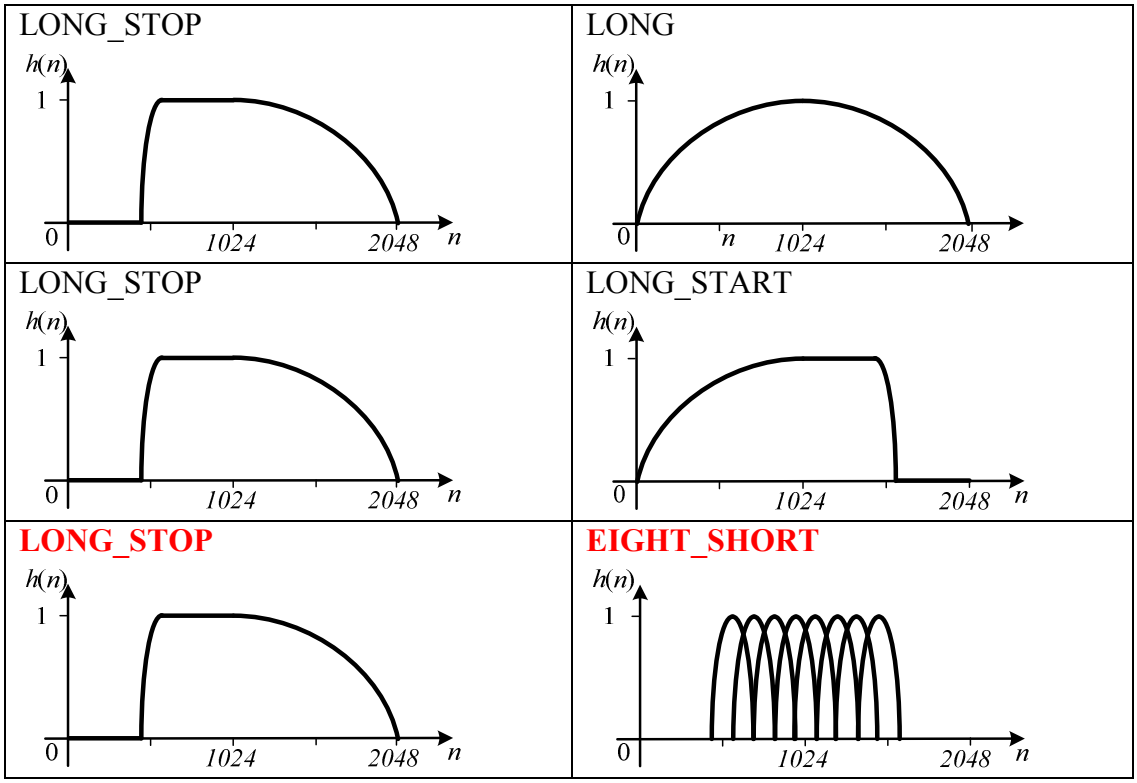


Table 7 – Supported transition between AAC window sequences

5.1.4 Quality assessment of the proposed low delay block switching

Formal subjective listening tests were organized in France Telecom R&D listening labs in order to assess the quality improvement obtained with the proposed approach. The MUSHRA methodology (ITU-R BS 1534-1 [ITU-R BS.1534-1 03]) was selected, since it is an appropriate subjective assessment method for that quality range. MUSHRA imposes the presence of anchors which are the original hidden signal and two band limited anchors. The original item is rated along with two low pass filtered versions with audio bandwidths of 3.5 kHz and 7 kHz respectively. The listeners were asked to grade the presented items according to their quality, ranking from bad to excellent. A total of 12 listeners performed the listening test. All listeners were experienced in MUSHRA listening test and some of them were experts in speech and audio coding or audio signal processing. The listening environment was as follows: Digigram VX222 sound card, 3Dlab DA converter and Stax headphones.

The experiment was based on the twelve traditional MPEG items that include critical material for speech, music, single and multiple instruments.

Item	Description	
es01	vocal (Suzanne Vega)	Singing Voice
es02	German speech	Male speaker
es03	English speech	Female speaker
si01	Harpsichord	Single instrument
si02	Castanets	Single instrument
si03	Pitch pipe	removed
sm01	Bagpipes	removed
sm02	Glockenspiel	Sur imposed bell sounds
sm03	Plucked strings	Sur imposed plucked strings
sc01	Trumpet solo and orchestra	removed
sc02	Orchestral piece	removed
sc03	Contemporary pop music	removed

Table 8 – Test items

Since the experiment deals with handling of transient part of the audio signal, only seven non-stationary items were retained from the test set. These items are described in Table 8 (items in grey represent the stationary items not used in the subjective evaluation). It has been verified that identical bitstreams were produces by the two codecs under test for the signals without detected transient.

The experiment was conducted at 32 kbit/s mono. The Low Delay AAC operates with the sampling frequency of 24 kHz in this mode. TNS was also activated for the sake of coding efficiency. The proposed method was compared to the reference quality MPEG-4 Low Delay AAC coder operating with the same configuration as France Telecom's encoder. The encoder under test was considered with and without low delay block switching. The purpose of incorporating the reference MPEG quality encoder was to ensure that the outcome of this experiment is reliable with respect to quality.

The systems under test are given in Table 9 below.

Codec	Description
ori	Hidden Reference
bw 3,5	3.5 kHz lowpass filtered anchor
bw 7.0	7 kHz lowpass filtered anchor
fhg	Reference Quality implementation (Fraunhofer) of LD-AAC at 32 kbit/s
ft	France Telecom implementation of LD-AAC at 32 kbit/s
ft prop	France Telecom implementation of LD-AAC at 32 kbit/s with the proposed low delay block switching scheme

Table 9 – Codecs under test for the LD-AAC with low delay block switching listening test

Figure 55 reports the outcome of the listening test. Mean and 95 % confidence intervals are reported in Table 10.

Conditions	Lower	Mean	Upper
ori	100.0	100.0	100.0
bw 3,5	11.7	14.0	16.4
bw 7.0	33.2	35.9	38.5
fhg	47.5	51.1	54.7
ft	48.3	51.5	54.7
ft prop	53.6	57.0	60.5

Table 10 – MUSHRA listening test scores for the assessment of LD-AAC with low delay block switching

An average improvement of more than 5 points on the MUSHRA scale is noticed: from 51.1 and 51.5 with AAC legacy systems, the grades become on average 57.0 with the proposed method. This test reveals that both implementations of the LD-AAC perform equally. We can therefore conclude that the improvement brought to our own implementation of the basic LD-AAC is meaningful. Figure 55 shows the listening test results on a per item basis: one can notice that the main source of improvement is for the castanet items (around 10 point of improvement) but also for some speech items such that es03 (also improved by about 10 Mushra points).

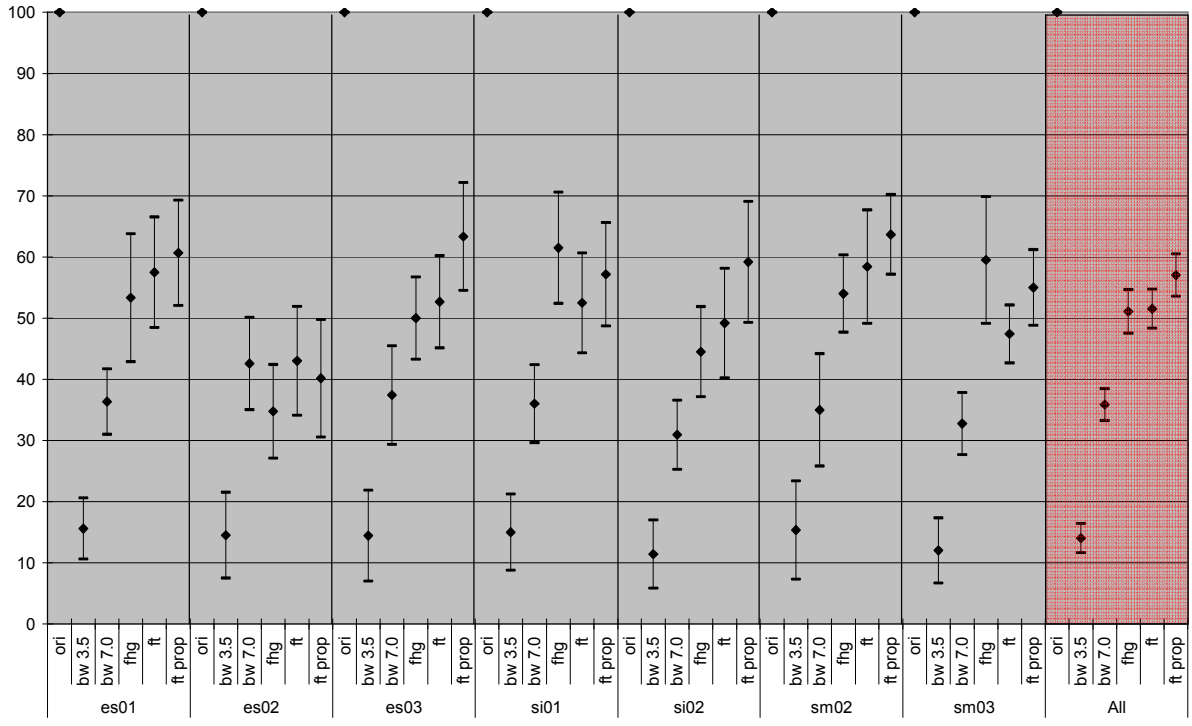


Figure 55 – MUSHRA listening test results for the assessment of LD-AAC with low delay block switching

Based on a per item analysis and on the average, the subjective listening test results demonstrate the benefit of having a block switching tool introduced in Low Delay AAC. None of the items is degraded when comparing the versions with and without the proposed low delay block switching tool.

When comparing France Telecom legacy AAC LD (ft) with the proposed enhancement using instantaneous block switching (ft prop), an improvement on average is indicated, though not statistically significant. This can be explained by two reasons: MUSHRA methodology tends to smear the difference in comparison with CMOS methodology. “ft” and “ft prop” systems are very similar: the only change is on the few attacks which are present in the audio signal, it means that around 90% of the signal is identical. This is the reason why listeners usually notice little difference between the two schemes. Here again, a CMOS methodology would have probably lead to a better discrimination of the differences.

In order to directly compare the systems with and without the proposed method, a differential MUSHRA analysis is used. The difference score between the proposed LD-AAC with low delay block switching and the initial LD-AAC (ft prop - ft) is computed per item and over all items. The 95% confidence interval is then computed per item as well as the mean difference score. The mean difference score and the associated 95% confidence intervals are given per item in Table 11. The rows highlighted in green represent the items for which a significant improvement has been observed.

Items	Lower	Mean	Upper
es01	-4.82	3.17	11.15
es02	-9.59	-2.83	3.92
es03	1.79	10.67	19.54
si01	0.53	4.67	8.80
si02	3.12	10.00	16.88
sm02	-3.98	5.25	14.48
sm03	0.81	7.58	14.36
Total	**2.65**	**5.50**	**8.35**

Table 11 – Differential MUSHRA listening test scores for the assessment of LD-AAC with low delay block switching

Figure 56 shows the corresponding graphical illustration of the listening test results using the differential MUSHRA scores.

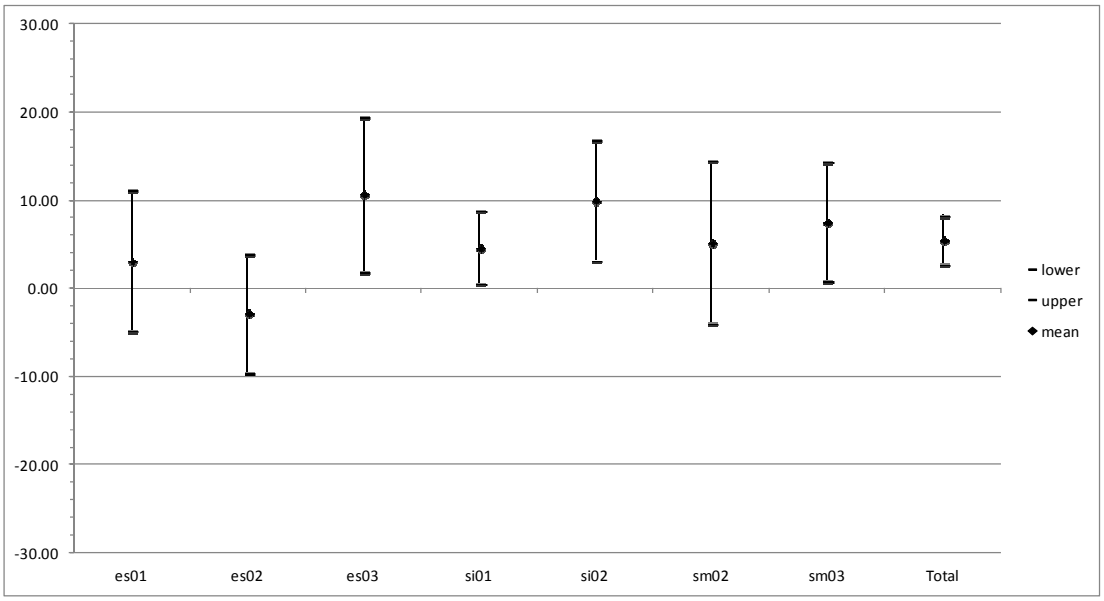


Figure 56 – Differential MUSHRA listening test results for the assessment of LD-AAC with low delay block switching

This leads to the conclusion that the low delay block switching tool associated with low delay AAC codec can improve the subjective quality even for a transform audio coder based on a reduced size MDCT. Additionally, this listening test validated the proposed implementation of the compensation algorithm since the subjects have not reported any specific artefacts.

5.1.5 Application to MPEG-4 Enhanced Low Delay AAC

The low delay block switching tool has been adapted to MPEG-4 Enhanced Low Delay AAC (ELD-AAC) as described in section 4.2. The main difference with the standard MPEG-4 LD-AAC comes from the filter bank which is based on a modified MDCT. This low delay transform is based on the transform described in section 3.1.3 [Schuller 00]. This section describes the integration of the low delay block switching into the MPEG-4 Enhanced Low Delay AAC and gives the performance assessment of the complete system with various signals.

The ELD-AAC is based on the low delay transform using a prototype length of $4M$ with M being equal to 480 or 512 (the prototype for M=512 is given on Figure 29). The number of zeroes introduced in the prototype is $M_z = M/4$, leading to a total delay reduction of 256 samples. The total delay of the low delay transform in ELD-AAC is then $2M - 2M_z = 3M/2$.

As described for the LD-AAC, the use of block switching algorithms would introduce additional algorithmic delay. This additional delay comes from the necessary look-ahead to detect the presence of a transient in the first half of the next frame. Based on this required look-ahead, the total algo-

rithmic delay of the low delay transform with block switching would be $3M/2 + M_z + M/2 + M_s/2 = (9M + 2M_s)/4$.

In order to avoid this delay increase, the low delay block switching tool has been integrated in the ELD-AAC codec. Based on the position of the attack, the encoder and decoder can use the normal transition window between a long window and short windows sequence. The two possible configurations of synthesis windows at the decoder with the presence of non-stationary sounds are presented in Figures 57 and 58.
In this first example, an attack arising in the second half of the current frame is considered. In this configuration, the encoder and decoder can switch to a transition window (see figure 57), the aim being to concentrate the transient noise spreading in the short window sequence in the second half of the frame.
Having used a long window at the previous frame, the encoder can select a long-short window for the current frame, the next ones being processed using the traditional eight short windows as for Low Complexity AAC.

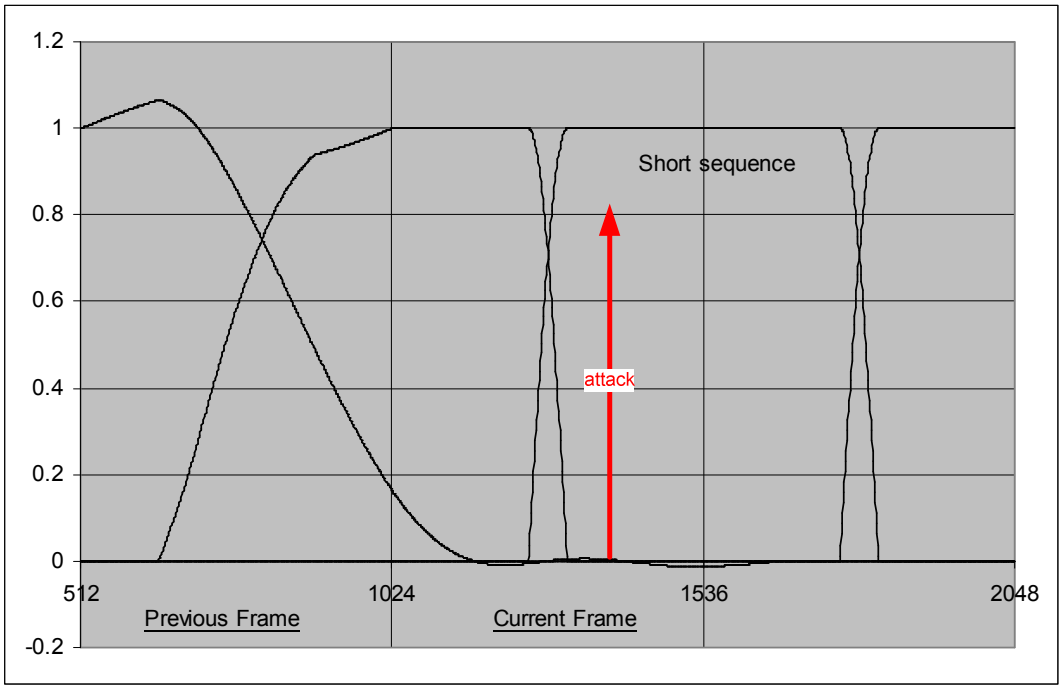


Figure 57 – First synthesis window sequence for low delay block switching in ELD-AAC

In this example, block switching can be used without any additional delay or look-ahead buffer.
In the second example, presented on Figure 58, the attack arises in the first half of the next frame. Without a look-ahead buffer, the transient can only be detected with the next frame and the current frame is processed with the long window. With the proposed low delay block switching, the window

sequence is shown in Figure 58. The short windows are directly used in the next frame and pre-echoes are avoided.

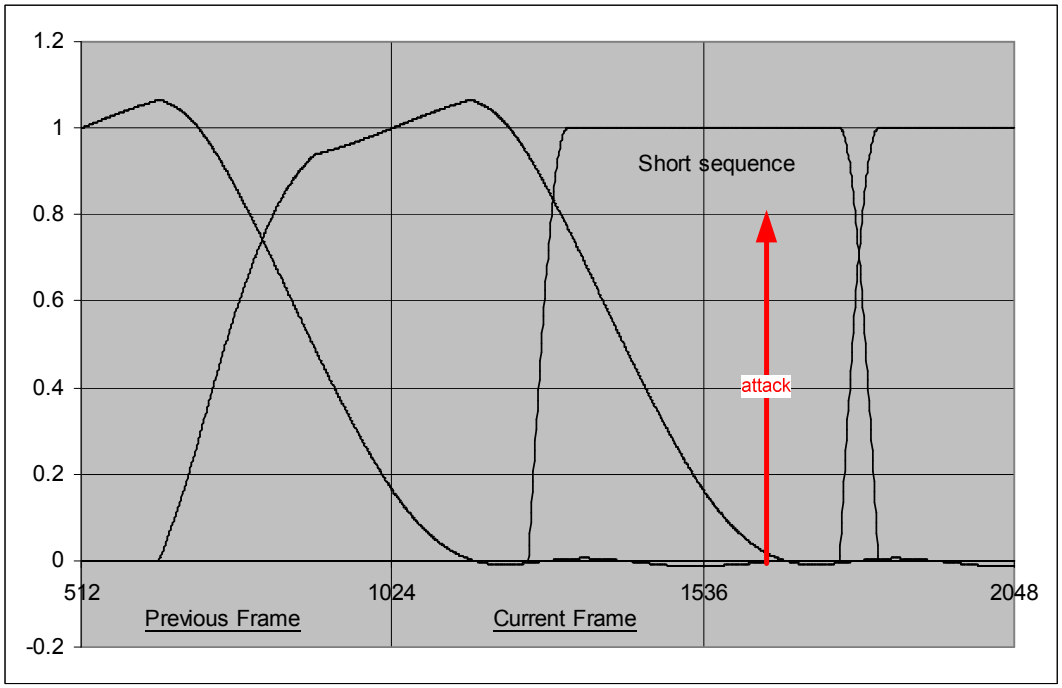


Figure 58 – Second synthesis window sequence for low delay block switching in ELD-AAC

Avoiding transition windows implies *a priori* that the time alias components cannot be suppressed. However, as demonstrated in section 4.2, the perfect reconstruction property of the complete system can still be preserved.

5.1.6 Implementation of perfect reconstruction with aliasing cancellation

The perfect reconstruction timing is somewhat different with low delay windows. Different portions of the frame need a particular processing. This is studied now.

Transition windows can be avoided at the encoder, which means that a direct switching from long to short windows can be decided when a transient is detected, without sacrificing the signal reconstruction. As it was mentioned previously, a different processing is made at the encoder and the decoder.

At the encoding side, short windows are applied directly after the long windows. At the decoder, short windows are applied without any special care just after the long windows, but a post processing has to be inserted in order to properly cancel the time aliasing and reconstruct the signal.

Figure 59 illustrates the zones which can be directly reconstructed from the two consecutive windows. When a long window is received (in dash line on Figure 59), the first M_z samples are set to zero due to the specific shape of the window. Using a similar framing as for LD-AAC, one can notice that, only a small portion of the frame cannot be directly reconstructed between sample $M_z = M/4$ (sample 1024 + 128 on Figure 59) and sample $(M+M_s)/2$ (1024 + 256 + 32).

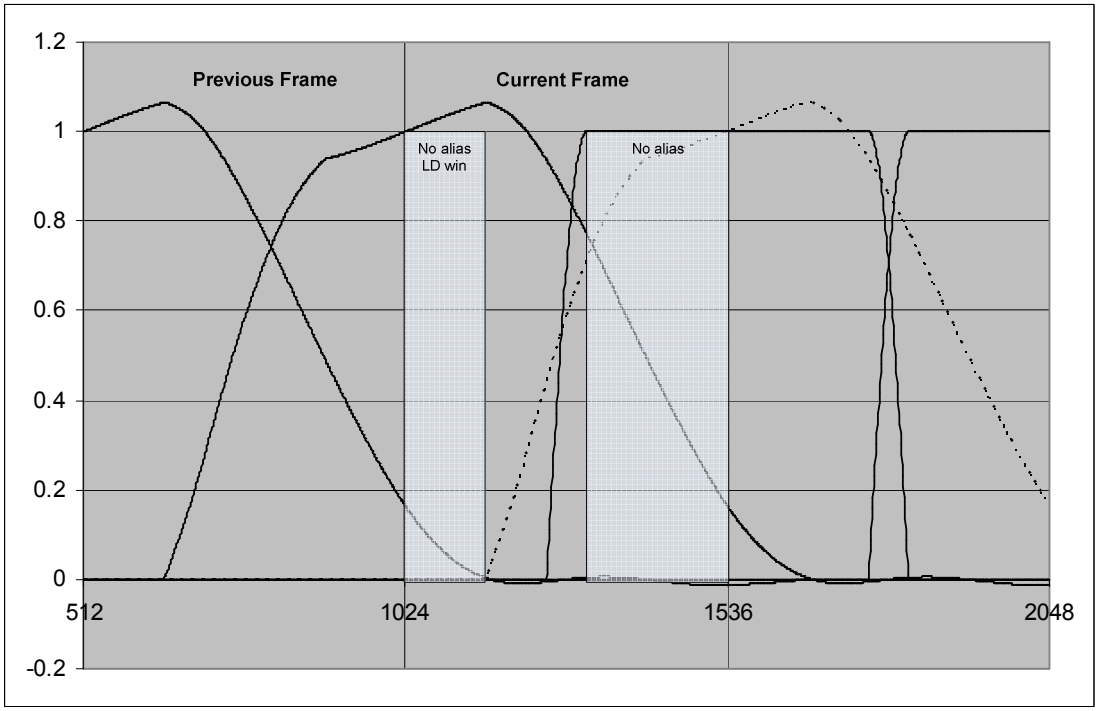


Figure 59 – Time alias free zones in low delay block switching reconstruction

The signal cannot be completely decoded since aliasing cancellation is not guaranteed due to the window mismatch between the two consecutive frames for both the encoder and the decoder. It can be noticed that the second half of the window is alias free if we consider more specifically the sample range going from $2M + (M + M_s)/2 = 1024 + 288$ to $3M - 1 = 1535$: the short windows properly reconstruct this portion of signal.

As a consequence, the time aliasing components, from the second half of the window that affects the first half, are perfectly known after the decoding of the short windows sequence. Therefore they can easily be removed from the first half. Moreover, some additional aliasing components are introduced at the analysis stage, due to the overlap factor of the low delay window. These components shall be removed to achieve a proper reconstruction. Hence, the aliasing suppression algorithm combines three components: alias components due to the longer windows, alias or reconstructed components coming from the short windows sequence, and finally some past samples that are time-aliased in the section to be reconstructed.

Figure 60 gives an overview of the components taken into account for the time aliasing cancellation. It is shown how the reconstruction algorithm is implemented in the MPEG-4 Enhanced Low Delay AAC context. The three temporal components are weighted and combined to achieve the perfect reconstruction.

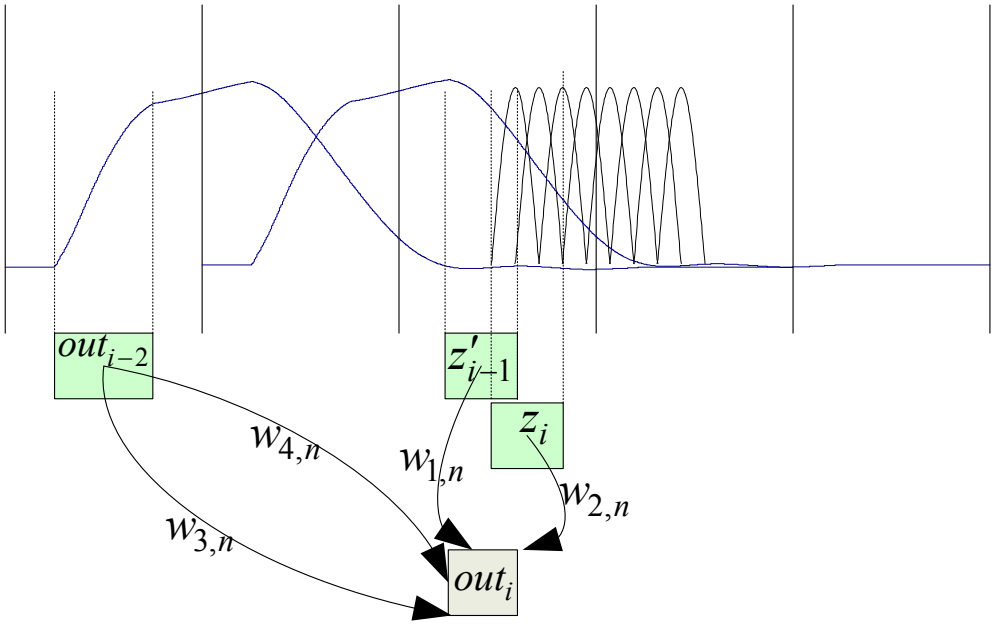


Figure 60 – Time components used for perfect reconstruction in ELD-AAC with low delay block switching

The time aliasing cancellation is obtained through the addition of several components as illustrated on Figure 60 with the use of compensation weighting functions (see section 4.2). Four weighting functions are defined. w_1 corresponds to the compensation of the long window synthesis. w_2 is the weight applied to the short window sequence compensation. w_3 and w_4 are the weighting functions applied to the past synthesized signal (two frames before). w_3 is applied to the decoded signal from the frame t-2 in the interval $M/4 \le n \le (M + M_s)/2 - 1$ in the direct order. w_4 is applied to the decoded signal from the frame t-2 in the interval $(M - M_s)/2 \le n \le 3M/4 - 1$ in the time-reversed order. This part of the past signal can be obtained from the decoded signal of the past frames.

The weighting functions are defined in two parts. The first part corresponds to the non-overlapping zone between the long window and the sequence of short windows. For $M/4 \le n \le (M - M_s)/2 - 1$, the weighting functions are defined by:

$$\begin{cases} w_{1,n} = \dfrac{1}{w_{LD}(M+n)\cdot w_{LD}(M-1-n)} \\ w_{2,n} = \dfrac{w_{LD}(n)}{w_{LD}(M-n-1)} \\ w_{3,n} = -\dfrac{w_{LD}(n)\,w_{LD}(4M-1-n)}{w_{LD}(M+n)\cdot w_{LD}(M-1-n)} \\ w_{4,n} = -\dfrac{w_{LD}(n)\,w_{LD}(3M+n)}{w_{LD}(M+n)\cdot w_{LD}(M-1-n)} \end{cases} \tag{5.1}$$

where w_2 is applied to the time-reversed synthesized signal coming from the sequence of short windows.

For $(M-M_s)/2 \le n \le (M+M_s)/2-1$ and $m = n-(M-M_s)/2$, the weighting functions are given by:

$$\begin{cases} w_{1,n} = \dfrac{\dfrac{w_s(M_s-1-m)}{w_{LD}(M+n)}}{w_{LD}(M-1-n)\,w_s(M_s-1-m)+w_{LD}(n)\,w_s(m)} \\ w_{2,n} = \dfrac{w_{LD}(n)-\dfrac{w_s(m)\,w_s(M_s-1-m)}{w_{LD}(M+n)}}{w_{LD}(M-1-n)\,w_s(M_s-1-m)+w_{LD}(n)\,w_s(m)} \\ w_{3,n} = \dfrac{-w_{LD}(n)\,w_{LD}(4M-1-n)\dfrac{w_s(M_s-1-m)}{w_{LD}(M+n)}}{w_{LD}(M-1-n)\,w_s(M_s-1-m)+w_{LD}(n)\,w_s(m)} \\ w_{4,n} = \dfrac{-w_{LD}(n)\,w_{LD}(3M+n)\dfrac{w_s(M_s-1-m)}{w_{LD}(M+n)}}{w_{LD}(M-1-n)\,w_s(M_s-1-m)+w_{LD}(n)\,w_s(m)} \end{cases} \tag{5.2}$$

The compensation weighting functions are plotted on Figures 61 and 62 for M = 512. One can notice that compensation is applied only for $M - M/4 - (M - M_S)/2 = 160$ samples of a 512 samples frame.

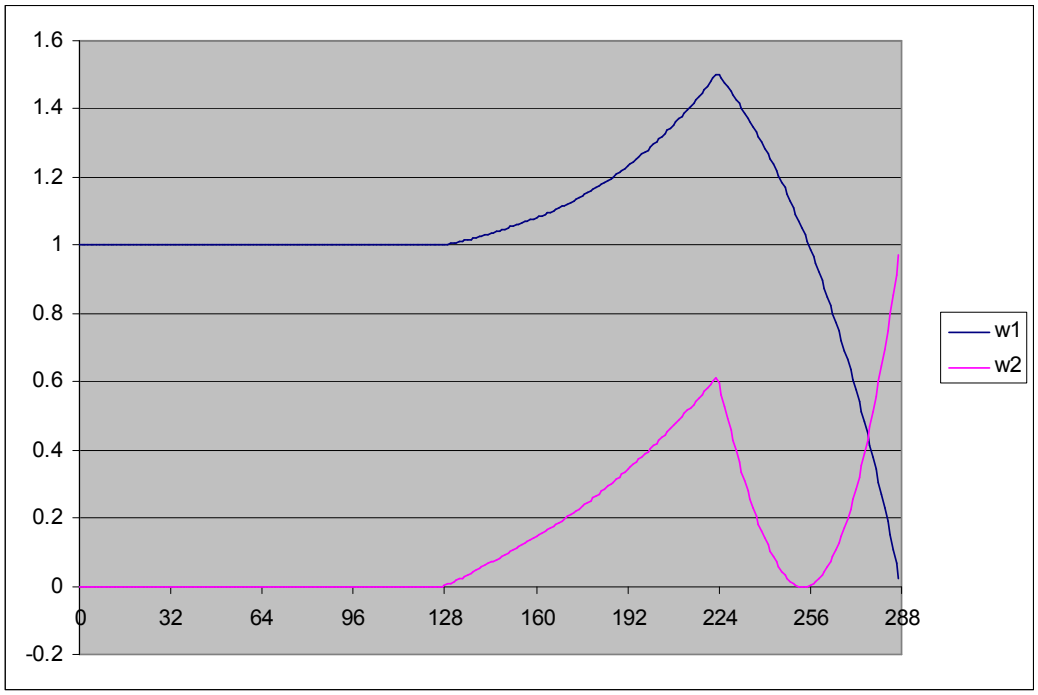


Figure 61 – Compensation weighting functions w_1 and w_2 for $M = 512$

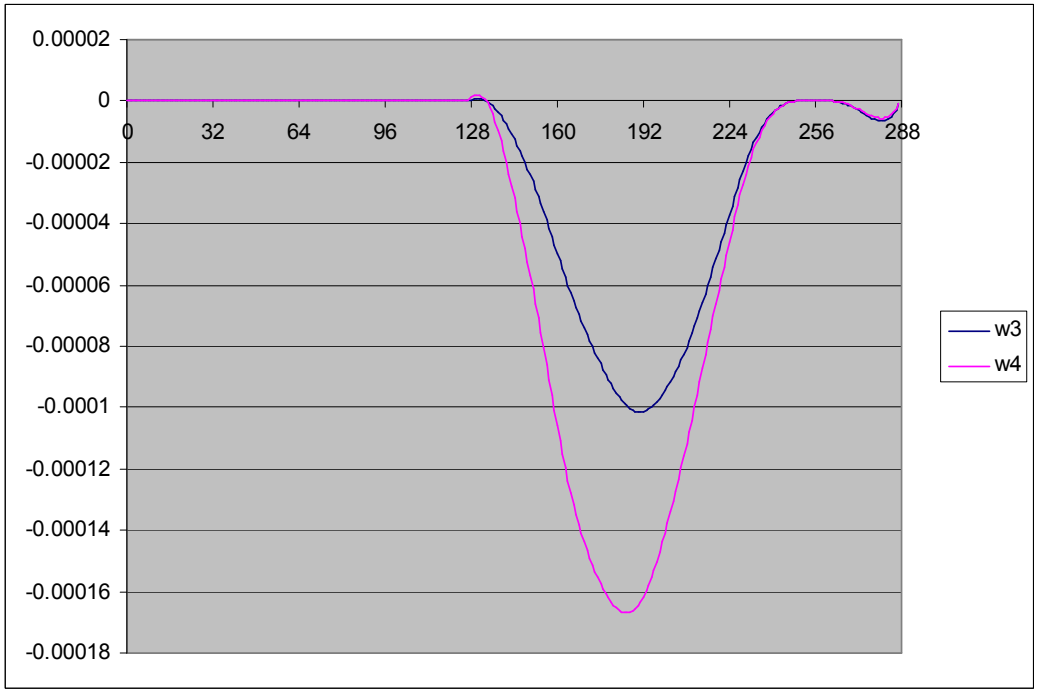


Figure 62 – Compensation weighting functions w_3 and w_4 for $M = 512$

5.1.7 Subjective evaluation of low delay block switching in ELD-AAC

Several listening tests have been conducted in order to evaluate the subjective quality impact of the proposed low delay block switching scheme in the context of MPEG-4 ELD-AAC for critical transient items, as well as for speech items.

5.1.8 Quality assessment with critical items

A first set of listening tests was performed using the MUSHRA methodology (as defined in the recommendation [ITU-R BS.1534-1 03]) in order to assess the performance of the proposed low delay block switching in the context of MPEG-4 Enhanced Low Delay AAC standardization. A total of 15 listeners were used.

Table 8 gives the list of 12 critical MPEG items used for the test. As explained previously, after verification, only 7 out of 12 items contain some transient parts, for which the low delay block switching tool can be activated.

The systems under test are given below:

Codec	Description
Ref	Hidden Reference
3,5 kHz	3.5 kHz lowpass filtered anchor
7 kHz	7 kHz lowpass filtered anchor
RM	Reference Quality implementation (Fraunhofer) of ELD-AAC at 32 kbit/s
FT	France Telecom implementation of ELD-AAC at 32 kbit/s
FT BS	France Telecom implementation of ELD-AAC at 32 kbit/s with the proposed low delay block switching scheme

Table 12 – Codecs under test for the ELD-AAC with low delay block switching listening test

Numerical and graphical results for the test showing the mean values and 95% confidence intervals are given on Figure 63 and Table 13. All the subjects were found reliable using the usual post screening procedure (3.5kHz, 7 kHz anchors and hidden reference being rated in ascending order and the hidden reference being graded above 90).

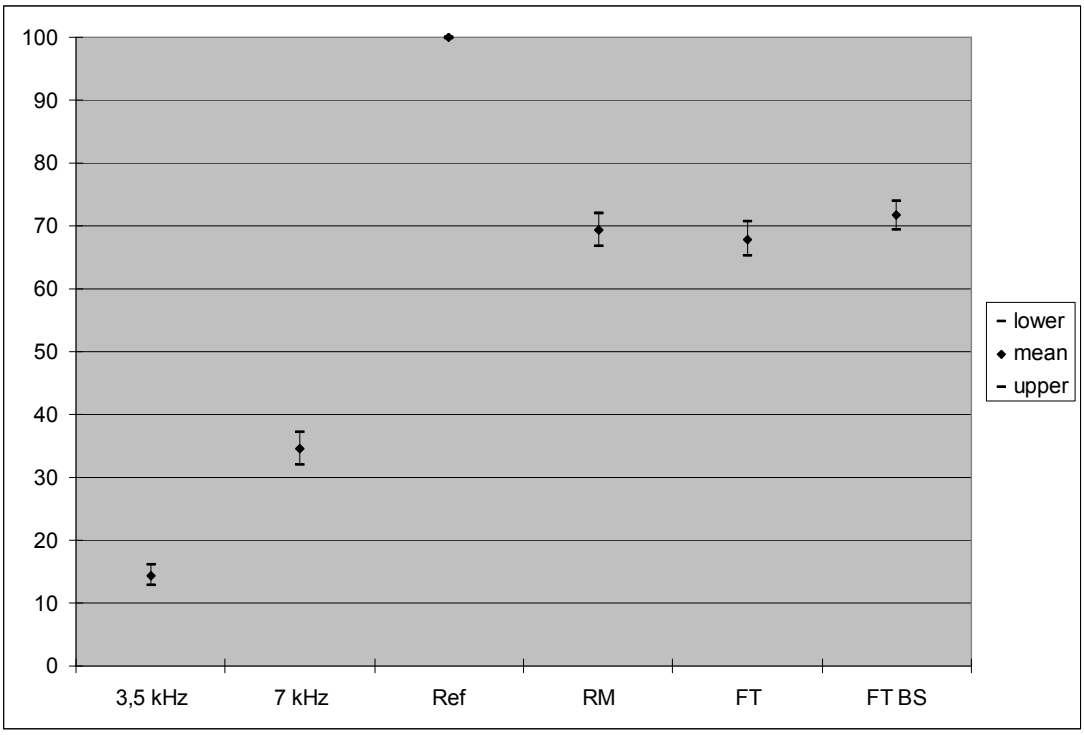


Figure 63 – Results over all items for the ELD-AAC with low delay block switching listening test

Condition	Lower	Mean	Upper
Ref	99.81	99.92	100.04
3.5 kHz	12.84	14.44	16.04
7.0 kHz	31.88	34.48	37.08
RM	66.65	69.28	71.91
FT	65.27	67.91	70.55
FT BS	69.38	71.64	73.91

Table 13 – Mean scores and 95% confidence intervals over all items for the ELD-AAC with low delay block switching listening test

The three systems under test present no statistical difference with overlapping confidence interval. However, the basic ELD-AAC implementation developed by France Telecom obtains a slightly lower average score over all items. On the contrary, the proposed system with low delay block switching tool obtains the best average score over all the tested systems. In order to have a better estimation of the benefit of the tool, Figures 64 and 65 show the MUSHRA scores per item. Figure 64 shows the results for the audio items containing transients and Figure 65 presents the results for the other audio items.

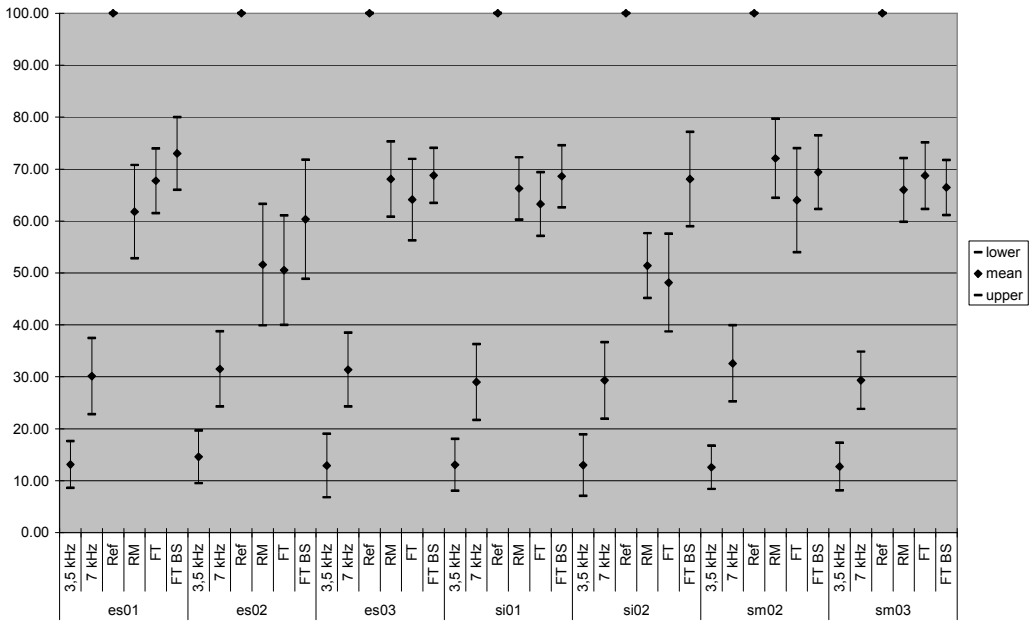


Figure 64 – Results for each of the 7 items with attacks

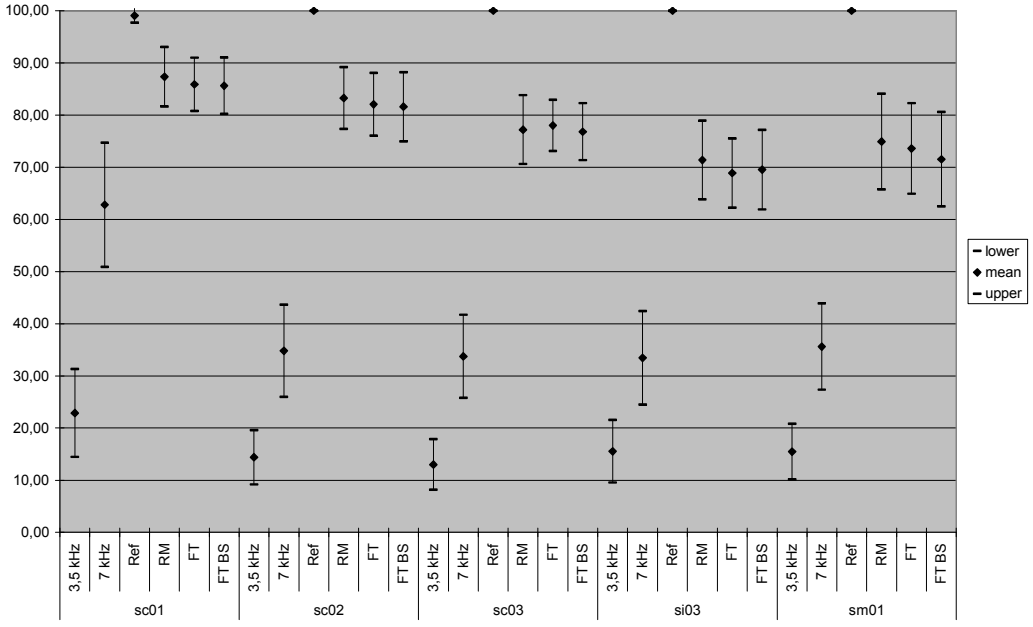


Figure 65 – Results for each of the 5 items without attacks

It can be seen on Figure 65 that for items without detected attacks or transients, there is no impact on the quality. As expected, the low delay block switching is not activated and systems FT and FT BS are identical. However, for items with transients, as shown on Figure 64, and especially for castanets (si02), the quality improvement is statistically significant. In order to better differentiate the systems, a Comparison Mean Opinion Score (CMOS) test has been conducted. This Reference/A/B scoring with the 7

points ITU comparison scale ("A is much better than B", better, slightly better, equal, slightly worse, worse, much worse are associated with scores from +3 down to -3) has bee carried out with a total of 8 expert listeners participating in this test. The 12 usual Core Experiments items of Table 12 were also used. Figure 66 shows the results for the audio items containing transients and Figure 67 presents the results for the other audio items.

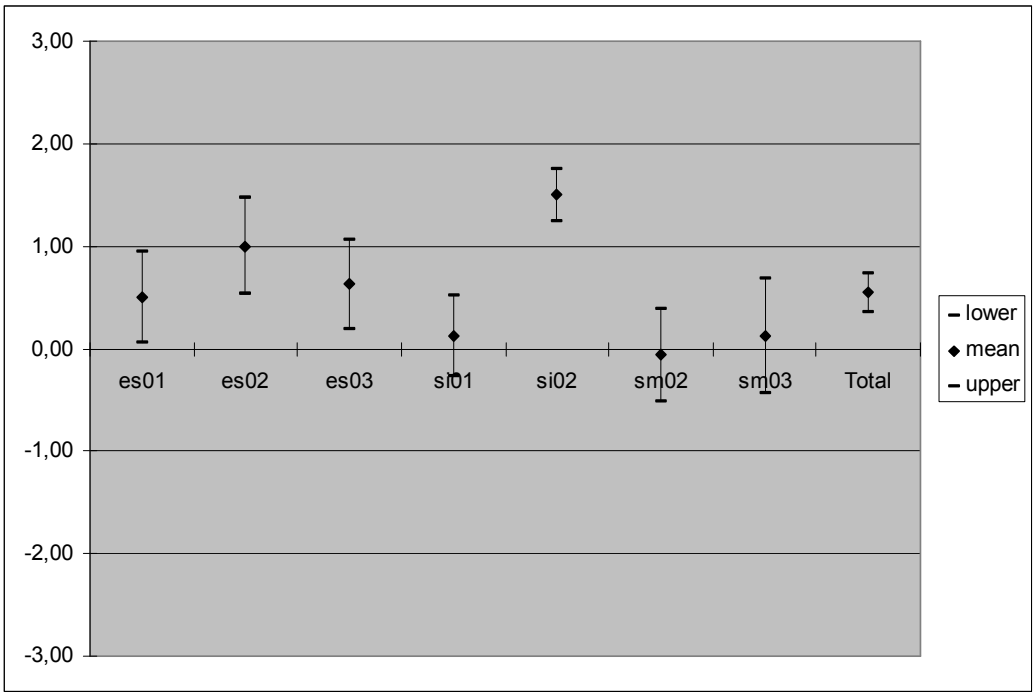


Figure 66 – CMOS listening test results for the 7 items with block switching

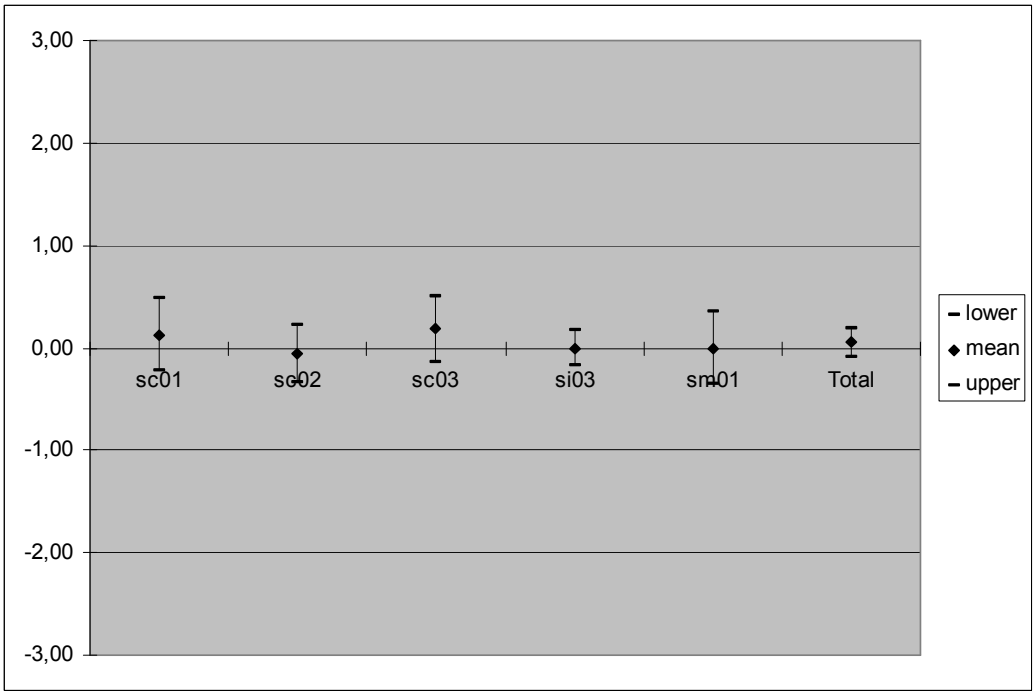


Figure 67 – CMOS listening test results for the 5 items without block switching

Figure 67 confirms that no difference can be perceived for audio items without transients. However, as shown on Figure 66 and in Table 14, with the proposed low delay block switching, the MPEG-4 ELD-AAC is significantly improved for 4 over 7 critical items, as well as on average over all items.

Items	lower	mean	upper
es01	0.06	0.50	0.94
es02	0.53	1.00	1.47
es03	0.19	0.63	1.06
sc01	-0.23	0.13	0.48
sc02	-0.34	-0.06	0.22
sc03	-0.13	0.19	0.51
si01	-0.27	0.13	0.52
si02	1.25	1.50	1.75
si03	-0.18	0.00	0.18
sm01	-0.36	0.00	0.36
sm02	-0.52	-0.06	0.39
sm03	-0.44	0.13	0.69
Total	**0.21**	**0.34**	**0.47**

Table 14 – Mean score with 95% confidence interval

The proposed low delay block switching adapted to MPEG-4 ELD-AAC offers good improvement for the twelve MPEG critical items. However, for the standardization process, it was requested to conduct one further listening test based on a speech database in order to demonstrate that for the targeted application which is audio communication and then with mainly speech content, the proposed technology could bring some value to the complete codec. Those results are introduced in the next section.

5.1.9 Quality assessment with speech items

In this third listening test, speech items have been used with a CMOS methodology. Table 15 gives the description of this speech database which was composed of several languages. Again, only two systems were tested in this experiment, the France Telecom implementation of the ELD-AAC (noted FT) which was shown to perform as good as the MPEG reference quality encoder provided by Fraunhofer IIS and the same codec with the addition of the low delay block switching tool (noted FT BS). The two codecs use the same SBR bitstreams provided by Fraunhofer IIS, only the core bitstream (AAC part) is modified. The bit rate is fixed (bit reservoir lower than 128 bits) at 32 kbit/s, with a frame length of 512 samples and 48/24 kHz sampling rates.

Title	Description
es04	male English
es05	Female German
nadib07	male Japanese
nadib08	male Tajik
nadib13	female French
nadib17	female German and background music
nadib20	female English and background music
nadib28	male French and background music
nadib39	male Japanese and guitar

Table 15 – Speech items test set

Figure 68 shows the results obtained in the CMOS test with the two systems. Positive scores indicate that the proposed low delay block switching can improve the quality. It can be seen that for 4 over 9 speech items, the quality is significantly improved and no degradations were observed. Moreover, on average over all speech items, the proposed technology also brings some value compared to the initial system.

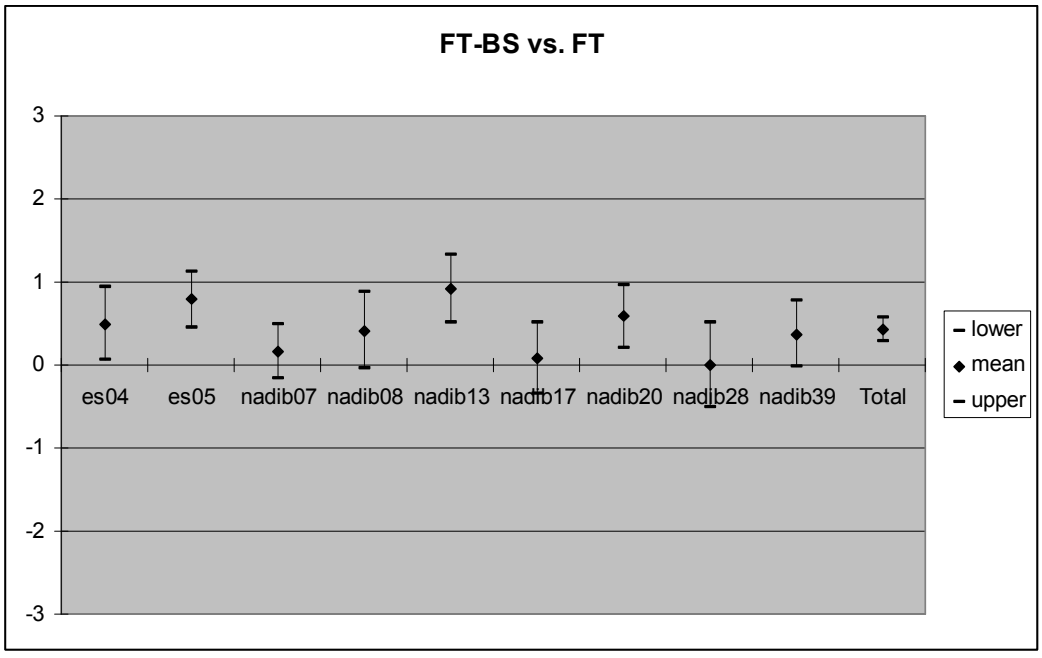


Figure 68 – CMOS listening test results for the 9 speech items with low delay block switching

Table 16 provides the mean scores and associated 95% confidence interval for the speech items and over all items. The rows highlighted in green show the speech items for which significant statistical improvements have been observed.

item	lower	mean	upper
es04	0.06	0.50	0.94
es05	0.46	0.79	1.12
nadib07	-0.16	0.17	0.49
nadib08	-0.04	0.42	0.87
nadib13	0.51	0.92	1.32
nadib17	-0.34	0.08	0.51
nadib20	0.21	0.58	0.95
nadib28	-0.50	0.00	0.50
nadib39	-0.03	0.38	0.78
Total	**0.29**	**0.43**	**0.57**

Table 16 – CMOS listening test scores for the 9 speech items with low delay block switching

5.1.10 Conclusion on low delay block switching in MPEG codecs

In 5.1, several listening tests which have been performed in the context of the MPEG standardization were presented. First, our adaptation of the low delay block switching tool to the MPEG-4 LD-AAC has shown very promising improvement for critical items with transients. Hence, it has been naturally adapted to the new codec ELD-AAC with the low delay transform. For both critical items and speech content, this modified version of the ELD-AAC has demonstrated a significant quality improvement. Even though the complexity of the proposed method is low (only 4 Multiplication-Addition over 160 samples per frame for M = 512), the low delay block switching tool has not been integrated in the MPEG-4 ELD-AAC standard.

5.2 Discussion on seamless reconstruction in MDCT

Based on the new framework offered by the seamless reconstruction technique described in 4.3, the impact on the overall performance of a coding scheme has been evaluated. Indeed, it was shown in 4.3.3 that both the length and shape of the MDCT analysis and synthesis windows can be adapted in time while allowing perfect reconstruction in the absence of quantization. In practical transform coding applications however, quantization plays a fundamental role in data compression, and hence some reconstruction error will inevitably be introduced. The seamless reconstruction method aims at allowing more flexibility in the selection of the optimal MDCT window sequence in order to minimize the effect of this error. In this paragraph, we present an experiment which has been conducted to

evaluate the benefit of the flexibility by comparing the performance based on an objective criterion. For this purpose, the segmental SNR was used. It is defined as:

$$SNRseg = \frac{10}{N} \sum_{t=0}^{N-1} \log_{10} \frac{\sigma_{x,t}^2}{\sigma_{d,t}^2} \tag{5.3}$$

with t being the index of the frame and N the number of frames. The segmental SNR computes the mean of the local SNRs over several segments of equal size N. In this way, any fluctuation in the instantaneous SNR values will be reflected in the final segmental SNR. It gives a more perceptually relevant measurement of the quality compared to the normal SNR and particularly exhibits the temporal performance if N is chosen sufficiently small. For this experiment, the quantization and bit allocation process are described in Annex C.

5.2.1 Experimental results

In order to evaluate the relevance of the seamless reconstruction, an experiment was conducted with the aim of achieving the best possible coding performance for a particular input signal. For this experiment, several sets of windows have been tested. For all test sets, based on the bit allocation presented in the previous section, the sequence of MDCT windows was determined. The results presented in this section provide a measure of the performance of various MDCT combinations in the form of segmental SNR values (in dB) taken over the complete reconstructed files following the selection and coding process. All tests were carried out at a bit rate of $R = 2$ bits per sample in order to keep the "high rate" assumption. It corresponds to a bit rate of 192 kbit/s in stereo for 48 kHz sources.

Practical audio coding systems must maintain a high level of performance over a wide variety of input signals. For this reason, the experiments were carried out on an extended set of 50 different audio files from the standard MPEG test set. These files were sampled at 48 kHz and were each around 15 seconds in length. To assist in the analysis of the results obtained, the files can be divided into the following distinct classes:

- Speech: 13 files
- Mixed Speech: 5 files
- Music: 12 files
- Mostly Stationary sounds: 20 files

The "Speech" category consists of clean speech signals (i.e. no other sounds present) and is therefore the most transient of the four groups. Conversely, "Mixed Speech" is composed of files featuring some background sounds in addition to voices, such as radio advertisement or an interview in

a busy street. These sounds introduce some more stationary regions to the largely transient speech signals.
As could be expected, the "Music" files contain a complex mixture of transient and stationary phenomena, making a priori prediction of the performance of a particular transform more difficult than for the other categories. The remaining signals (including great highland bagpipes and glockenspiel for example) are classified as "Mostly Stationary".
The first step of this experiment consists in verifying that the segmental SNR measure was reliable in that context. For this purpose, the results obtained for a simple case using fixed size MDCTs were obtained in order to verify that the segmental SNR method operates as expected.
In the second part of this experiment, the comparison of the performance obtained with several windows sets was performed. One of the windows sets introduces the seamless reconstruction. This windows set has been determined with the main objective to develop the best possible set of MDCT windows for a given number of combinations.

5.2.2 Validation of segmental SNR method

Table 17 displays the results obtained from a preliminary experiment using fixed length MDCTs with sine analysis and synthesis windows as defined in equation (3.6). For every transform size (where M is the number of transform coefficients) the mean of all the segmental SNR values in each category is shown. All values are in decibels (dB).

Category	$M = 128$	$M = 256$	$M = 512$	$M = 1024$	$M = 2048$
Speech	52.48	54.03	**54.90**	54.41	51.53
Mixed Speech	58.39	61.32	63.06	**63.83**	62.97
Mostly Stationary	52.37	56.26	59.82	62.90	**65.07**
Music	52.82	55.46	**56.47**	56.40	54.39
All Files	**53.11**	**55.99**	**58.06**	**59.23**	**58.78**

Table 17 – Segmental SNR (dB) achieved using fixed length sine windows

It can be seen that for the "Mostly Stationary" signals, the performance increases quite steadily with transform size, with the fixed MDCT of length $M = 2048$ yielding the highest segmental SNR. In contrast, the best result for the "Speech" class is achieved using a shorter transform where $M =$

512. Furthermore, the intermediate “Mixed Speech” files are best coded using MDCTs of length $M = 1024$, while for the complex “Music” category, very little difference in performance can be noted between $M = 512$ and $M = 1024$.
Indirectly it is found that the best transform size, for a fixed MDCT is around 1024: it is the choice made in audio coding for AAC for example.
Clearly these results also agree with the assumption that the adaptation of the window to the audio content would lead to improvement. This indicates that the segmental SNR method operates as expected, and can thus be used to provide meaningful comparisons when averaging the results of segmental SNR over a sufficiently large database.

5.2.3 Time segmentations using low overlap windows

Based on the assumption that the segmental SNR provides a good measure of the performance of the coding stage with a given window, a second set of windows has been introduced. This windows set is based on low overlap windows defined in equation (3.10) with window length between $M = 128$ and $M = 2048$. This particular set has been selected in order to avoid any mismatch between two consecutive transforms with different size. The low overlap windows set is defined based on the same size of overlapping region whatever the transform size. The window length range has been defined according to the Advanced Audio Coding (AAC) standard [Brandenburg 99] with the addition of length $M = 2048$ as it has been shown that a significantly higher segmental SNR can be achieved for stationary signals. Using this complete initial set, there are 5272 different ways to divide an input frame of 2048 samples using low overlap windows alone. In order to achieve practical windows set with a limited number of combinations, the combinations, which are employed most frequently, are finally selected in the windows set. This test was carried out for all 5272 combinations over a long “learning sequence” audio file of around 7200 frames. This learning sequence was created by concatenating 25 files taken from the four different categories so as to cover as wide variety of input signals as possible.

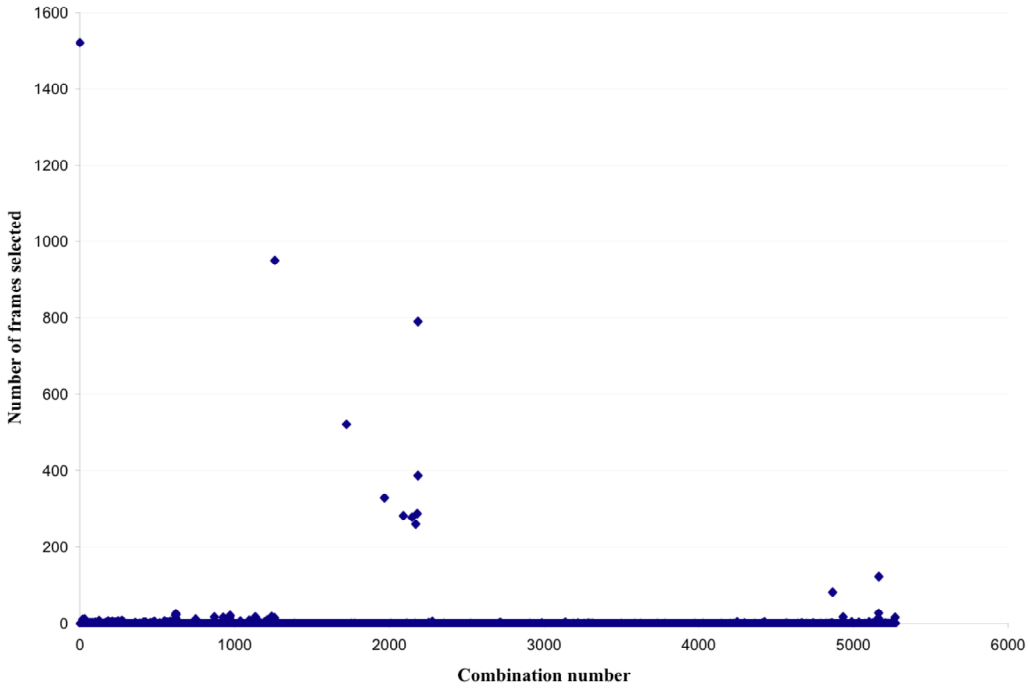


Figure 69 – Low overlap window combinations selected from 5272 set over learning sequence

The results of this experiment are presented in Figure 69, where the *x* axis displays the combinations numbered from 0 to 5271 and the *y* axis the number of frames for which each combination has been selected, representing the histogram of combinations. From this graph, it can be seen that some combinations are chosen far more frequently than others. To keep a windows set with a reasonable size, 12 combinations have been finally selected as shown on Figure 70. Those 12 combinations are by far the most frequent combinations used on figure 69 (more than 80% of the total frames).

Category	**Best sine window**	**Best low overlap window**	**Adaptive (12 combinations of low overlap windows)**
Speech	54.93 (*M* = 512)	54.56 (*M* = 1024)	**58.08**
Mixed Speech	64.06 (*M* = 1024)	62.92 (*M* = 1024)	**65.40**
Mostly Stationary	65.09 (*M* = 2048)	61.29 (*M* = 2048)	**62.11**
Music	57.56 (*M* = 512)	56.25 (*M* = 1024)	**59.26**
All Files	**60.54**	**60.38**	**60.71**

Table 18 – Segmental SNR (dB) for adaptive system using 12 combinations of low overlap windows

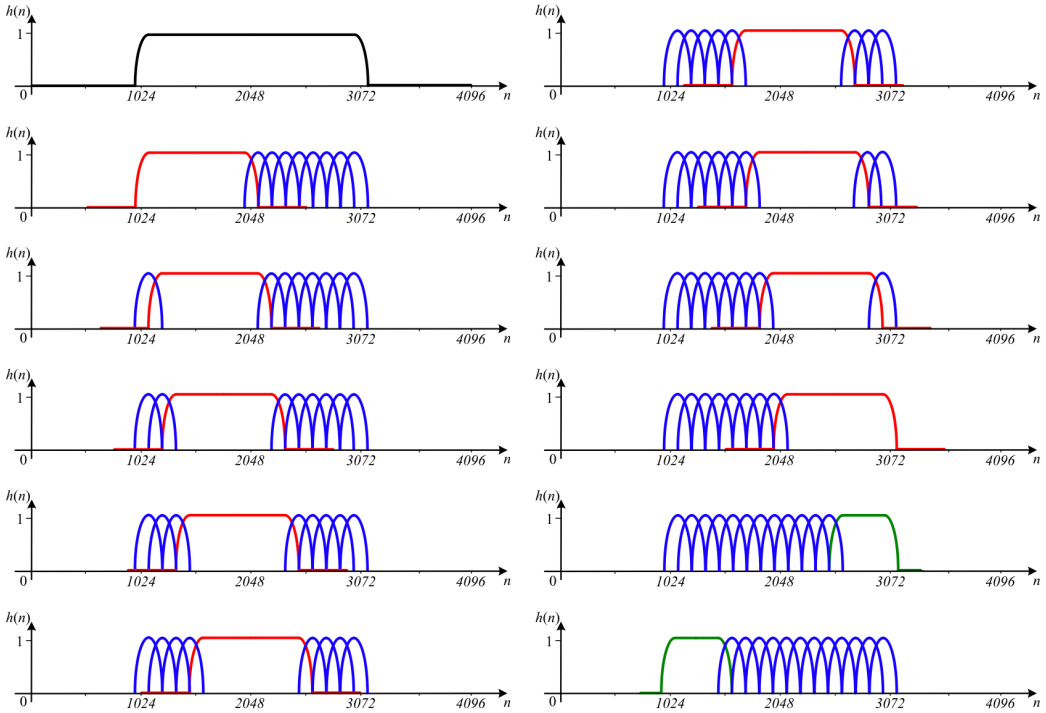


Figure 70 – The 12 selected low overlap window combinations

It has been observed that the mean segmental SNR of 62.92 dB can be achieved by coding the learning sequence with only these twelve combinations of the limited sub-set, only 0.19 dB less than the 63.11 dB attained using the full set of 5272 combinations.

Table 18 shows the performance of this sub-set compared to the best result attainable using a fixed size window (where the length of the highest performing transform is given in parenthesis) on the test database described in 5.2.1. This provides further evidence that significant improvements can be achieved through the application of this adaptive process using the combination set shown in Figure 70. This increase in performance is particularly noticeable between the fixed size low overlap window and the subset of low overlap window combinations in the more transient categories such as Speech or Music, for which more than 3 dB in average can be gained.

However, when compared to the best results which can be achieved using a fixed length sine window fixed size as represented in the first column of Table 18, different conclusion can be drawn. In the Speech category, a notable increase in performance (3.15 dB) can again be achieved with the low overlap combinations set due to the ability of the adaptive system to code the frequent speech transient sections with the shortest MDCT, whilst using longer transforms for the more stationary regions of the file, e.g. for voiced sounds. On the contrary, for the Mostly Stationary category, it can be seen that the use of a fixed size sine window of M = 2048 still provides an improvement of 2.98 dB over the adaptive low overlap system.

5.2.4 Definition and evaluation of the final windows set

Based on those preliminary results, sine windows are subsequently included to the combination set. By using the seamless reconstruction method discussed in 4.1 and 4.3.3, these additional windows can be incorporated without the need to design specific transition windows. Indeed the direct transition between long low overlap window and long sine window is allowed without having the corresponding transition window in the windows set. The final set which was used to evaluate the performance of the seamless reconstruction is presented in Figure 71. It should be noted that introducing all the necessary transition windows in the set would have first increased significantly the number of combinations in set and then the system complexity, given the closed loop window selection. Moreover, in order to anticipate the necessary inclusion of a transition in the window sequence, a huge look-ahead would have been necessary and the selection system would have necessarily been more complex as for each processed frame, several combinations of the current and future frame should be considered.

Category	**Best Low Overlap**	**Best Sine**	**12 Low Overlap Combinations**	**Final Set**
Speech	54.56	54.93	58.08	58.67
Mixed Speech	62.92	64.06	65.40	66.37
Mostly Stationary	61.29	65.09	62.11	66.32
Music	56.25	57.56	59.26	60.15
All Files	**60.38**	**60.54**	**60.71**	**62.85**

Table 19 – Segmental SNR (dB) for final windows set

Table 19 shows the results obtained using the final windows set, which includes both sine and low overlap windows. In can be seen that for the Speech category, this new set gives a further increase of 0.59 dB compared to the value obtained using the previous set.

However, it is in the Mostly Stationary category that this set proves most beneficial, providing a substantial increase of 4.21 dB in comparison to the low overlap windows set (1.23 dB higher than the fixed $M = 2048$ sine configuration.) This leads to an average improvement of 2.14 dB over all 50 test files. Hence, the seamless reconstruction MDCT method has shown a great potential in improving the flexibility of the MDCT window selection.

Over all categories, the final set has proven to offer the best performance in terms of segmental SNR over all other window combinations.

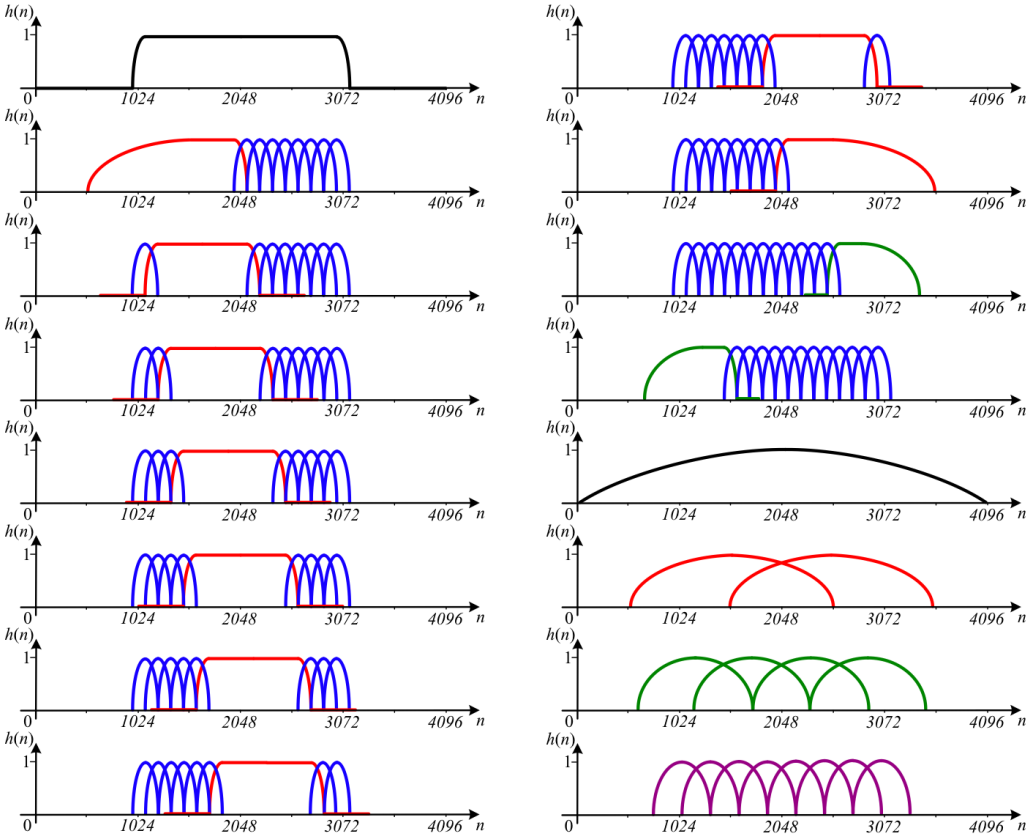


Figure 71 – Final experimental windows set

5.2.5 Comparison of final window combination set to AAC

Finally, the final set of window combinations, which has been evaluated in the previous section, has been compared to the current state-of-the-art AAC windows set (where the KBD window has been omitted) in order to determine the level of improvement. This windows set is shown in Figure 72. It basically consists in the long window $M = 1024$, the sequence of eight short windows with $M_S = 128$ and the associated transition windows, both with $M = 1024$. This windows set represents the necessary set for the block switching method described in section 3.2.1.

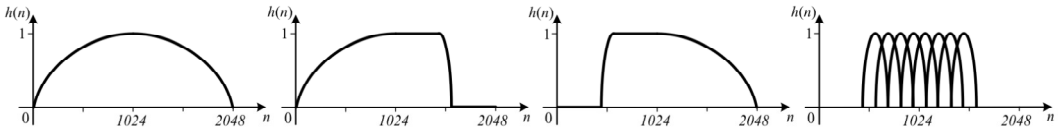


Figure 72 – Windows set used in MPEG 2/4 AAC

In addition to the standard AAC windows set and the final experimental set, an intermediate system has been tested. This system is also based on the AAC windows set. However in this case the seamless reconstruction

method was used to allow for non-standard transitions, such as those depicted in Figure 73. In that specific case, we can directly compare the benefit of the seamless reconstruction method with the state-of-the-art method using the same windows set.

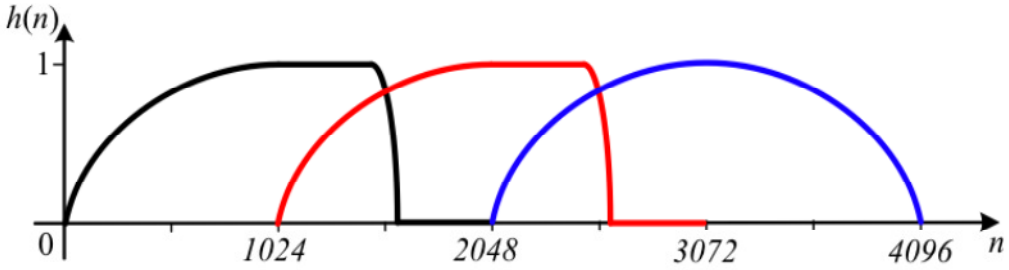


Figure 73 – Representation of non-standard AAC window transitions

The results obtained with these three configurations are displayed in Table 20. The first column in the table gives the segmental SNR values achieved by employing the AAC set using only standard window transmissions (i.e. the expected performance of a current AAC system). The results in the second column were also obtained using the AAC windows set, but this time allowing non-standard transitions and using the seamless reconstruction method. The last column gives the segmental SNR for the final experimental set which has been determined in 5.2.4.

Category	**AAC**	**Seamless AAC**	**Final Set**
Speech	57.48	58.09	**58.67**
Mixed Speech	65.16	65.57	**66.37**
Mostly Stationary	63.43	63.59	**66.32**
Music	58.82	59.24	**60.15**
All Files	**60.95**	**61.31**	**62.85**

Table 20 – Segmental SNR (dB) for final windows set compared to AAC

It is interesting to note that by permitting non-standard transitions an improvement can be achieved compared to the performance of the same windows set using standard TDAC reconstruction. This is most significant in the more transient category, with 0.61 dB being gained for the Speech class (the increase of 0.16 dB for stationary sounds is almost negligible). As ex-

pected, the final windows set provides a substantial improvement (2.73 dB) over the seamless AAC system for the Mostly Stationary signals, largely due to the inclusion of the M = 2048 transform. However in the Speech category, a more moderate but still noticeable increase of 0.58 dB was achieved as a result of the new windows combinations. Taking the mean value of all the results obtained for the 50 test files, the combination set proposed in this work can be said to provide an average increase of 1.9 dB over standard AAC, or 1.54 dB when the performance of the AAC windows set is enhanced using seamless reconstruction.

5.2.6 Summary

In this section, series of results were presented, leading to the development of a new MDCT windows set. This was shown to compare favourably with the set used in MPEG-2/4 AAC, especially for stationary signals. Moreover, it was shown that the performance of a particular windows set (in our case AAC windows set) can be improved by using the seamless reconstruction method to allow non-standard transitions between consecutive windows.

Taking the mean segmental SNR difference between the systems, a bit rate reduction can be estimated. Thus, considering the results of the comparison between the final windows set and that used in AAC, which were presented in Table 20, for the Mostly Stationary category, an improvement of 2.73 dB was noted. This implies a substantial reduction in bit rate of around 22 kbit/s representing around 11% reduction.
As mentioned before, while the increase in SNR using the new windows set is not as large for Speech signals, a gain of 0.58 dB was achieved. In a practical system, this would allow the bit rate to be reduced by approximately 5 kbit/s without sacrificing the quality of the reconstructed signal.
Again taking the mean of the SNR improvements for all 50 input files, a value of 1.54 dB was observed. The proposed windows set can thus be said to provide an equal level of signal quality to that attained by the AAC set whilst saving an average of 12 kbit/s in bit rate. From this it can be concluded that an efficient transform combination set has indeed been successfully developed based on the seamless reconstruction method.

5.3 Asymmetric Low Delay (ALD) window for ITU-T G.718

This section describes the subjective evaluation which was performed during the ITU-T G.718 development. This codec provides an embedded scalable solution for compression of 16 kHz sampled speech and audio signals at rates between 8 and 32 kbit/s. It has been designed to be robust to sig-

nificant rates of frame erasures or packet losses. It is composed of five layers. The two lower layers are based on Code Excited Linear Prediction (CELP) coding taking advantage of signal classification to use optimized coding modes. The higher layers encode the perceptually weighted error signal from lower layers using MDCT transform coding. Several vector quantization schemes are used to encode the MDCT coefficients to maximize the performance for both speech and music. The codec operates on 20 ms frames. During the ITU-T G.EV-VBR development phase, several investigations have been carried out in order to reduce the delay of the MDCT transform coding part. Some attempts were based on the division by two of the transform size. This first solution offered good performance for transient signal as it was just based on the use of shorter block size, but did not offer satisfactory quality for music in stationary parts or for voiced speech. The development of low delay MDCT window as described in 4.4 has been done to reduce the delay without sacrificing the efficiency and quality of the MDCT layers.

5.3.1 Introduction to ITU-T G.718

Figure 74 shows the structural block diagram of the encoder for wideband input (16 kHz sampling frequency).

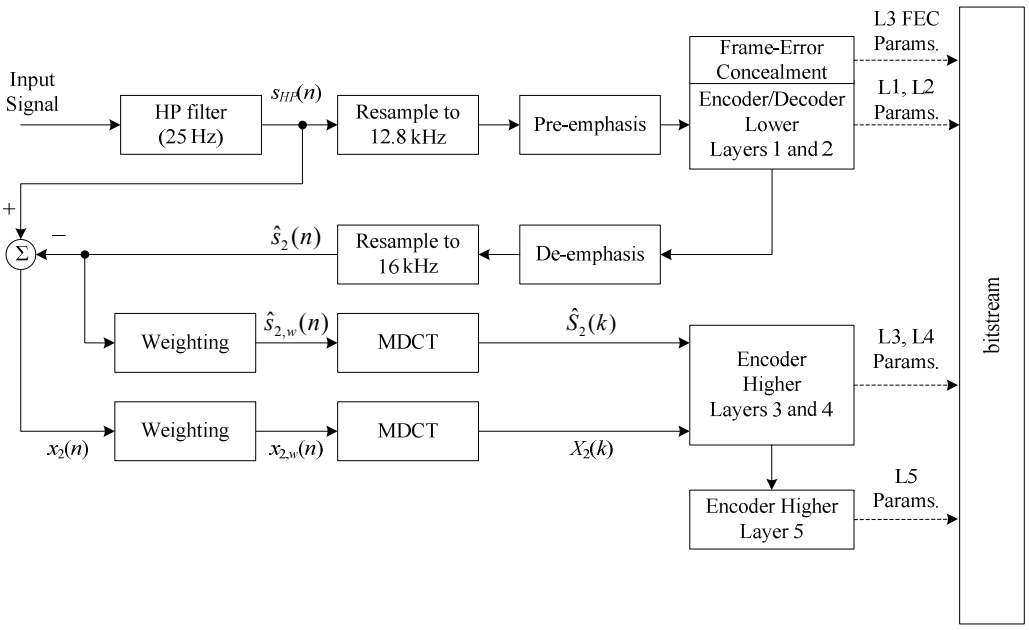


Figure 74 – Block diagram of the G.718 encoder

From Figure 74, it can be seen that while the two lower layers are pre-emphasized at 12.8 kHz, the three upper layers operate at the input sampling rate of 16 kHz. The core layers 1 and 2 use a classification based core layer with following modes: Unvoiced coding (UC), Voiced coding (VC), Transition coding (TC) and Generic coding (GC). The audio signal is modelled, using a CELP-based paradigm, by an excitation signal passing

through a linear prediction (LP) synthesis filter representing the spectral envelope. The excitation is adapted depending on the selected mode.
The codec has been designed with emphasis on performance in frame erasure conditions and several techniques limiting frame error propagation have been implemented. To further enhance the performance in frame erasure conditions, side information is sent in layer 3. This side information consists of class information for all coding modes. Previous frame spectral envelope information is also transmitted if the TC mode is used in the core-layer. For other core layer coding modes, phase information and the pitch-synchronous energy of the synthesized signal are sent. This allows better recovery of the excitation when a frame is lost.
Finally, the error resulting from the two first layers is further quantized in three transform coding layers.
The transform coding is based on the MDCT and performed at 16 kHz sampling frequency. As can be seen from Figure 74, the de-emphasized synthesis from core layers is resampled at 16 kHz. The resulting signal is then subtracted from the high-pass filtered input signal to obtain the error signal which is perceptually weighted and encoded every 20 ms in the transform domain. An asymmetric window, as shown in Figure 51, is used to reduce the delay associated to the transform coding stage from 20 to 10 ms while keeping the same number of frequency coefficients. Indeed, the delay associated with the framing (20 ms) is already taken into account by the core layers. Hence, only the additional delay due to the transform look-ahead is added to the core delay. The window is defined according to equations (4.44, 4.45 and 4.46) with M = 320 and the number of zero being $M_z = M/4$.

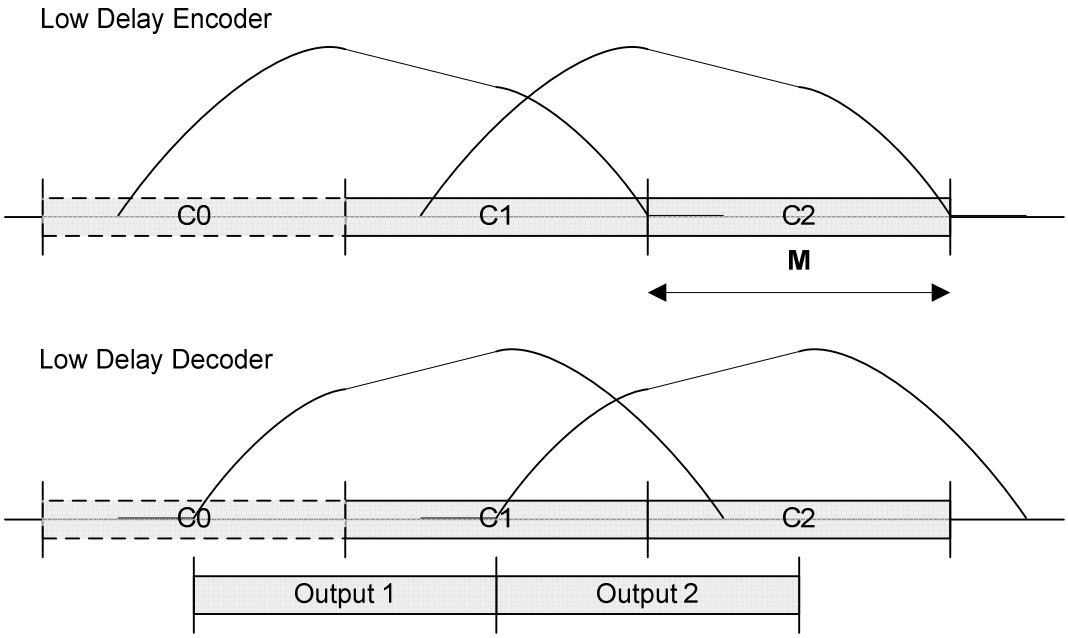


Figure 75 – Encoding and decoding timing with ALD window

Figure 75 shows the timing of the analysis and synthesis using the ALD window. Figure 76 illustrates the frequency response of the initial sine window with M = 320, the first low delay version which was based on two sine windows with M = 160 and finally the ALD window with M = 320. It can be seen that with the same delay reduction, the ALD offers better frequency performance than the M = 160 sine window. The first lobe provides a better selectivity than shorter window as well as better stop-band attenuation.

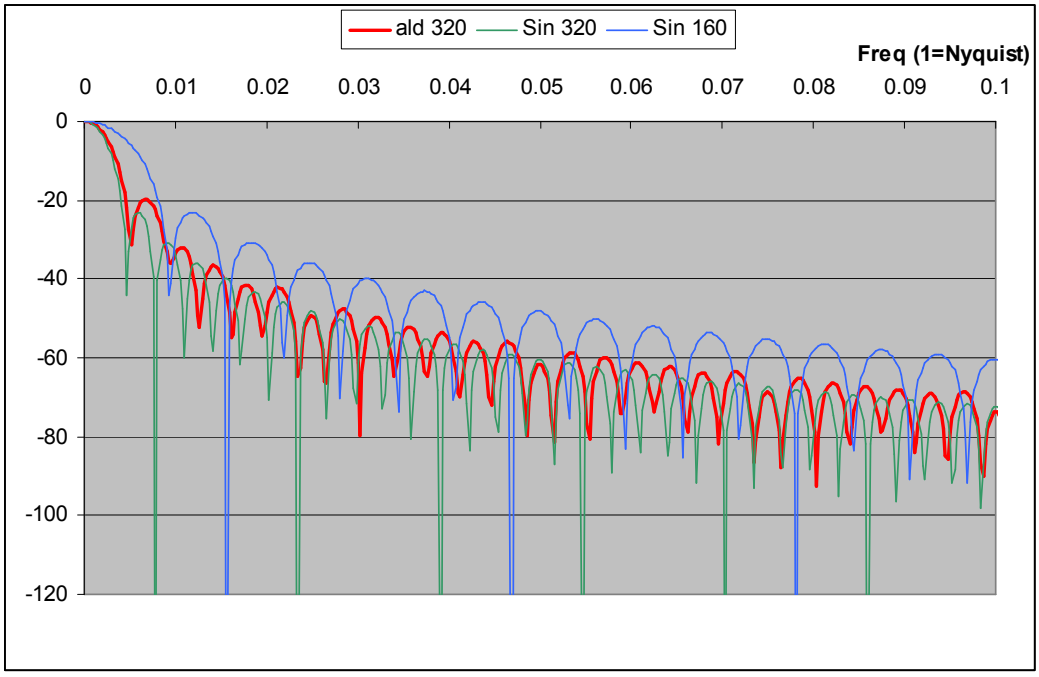


Figure 76 – Frequency responses of candidates MDCT windows for G.718

The MDCT coefficients are then quantized differently for speech and music dominant audio contents. The discrimination between speech and music contents is based on an assessment of the CELP model efficiency. More detailed descriptions of the codec can be found in [ITU-T G.718 08] and [Vaillancourt 08].

5.3.2 Evaluation of ALD window in G.718

In order to evaluate the benefit of the ALD window in the context of G.718, several listening tests have been carried out. The test methodology was an AB test without references, using the 7- points ITU scale (-3;+3). The tested database was composed of 6 clean speech items, 4 speech items with background noise (one item per background noise type: interfering talker, street, car, office) and 5 music items.

Figure 77 shows the results obtained at 16 kbit/s while comparing the ALD and sine window with M = 320 for both systems. While the ALD window offers 10 ms delay reduction, the performance over this database was very similar. No statistical differences can be observed. The same conclusion

can be drawn from Figure 78 which gives the results of the corresponding experiment when the codec operates at 32 kbit/s.

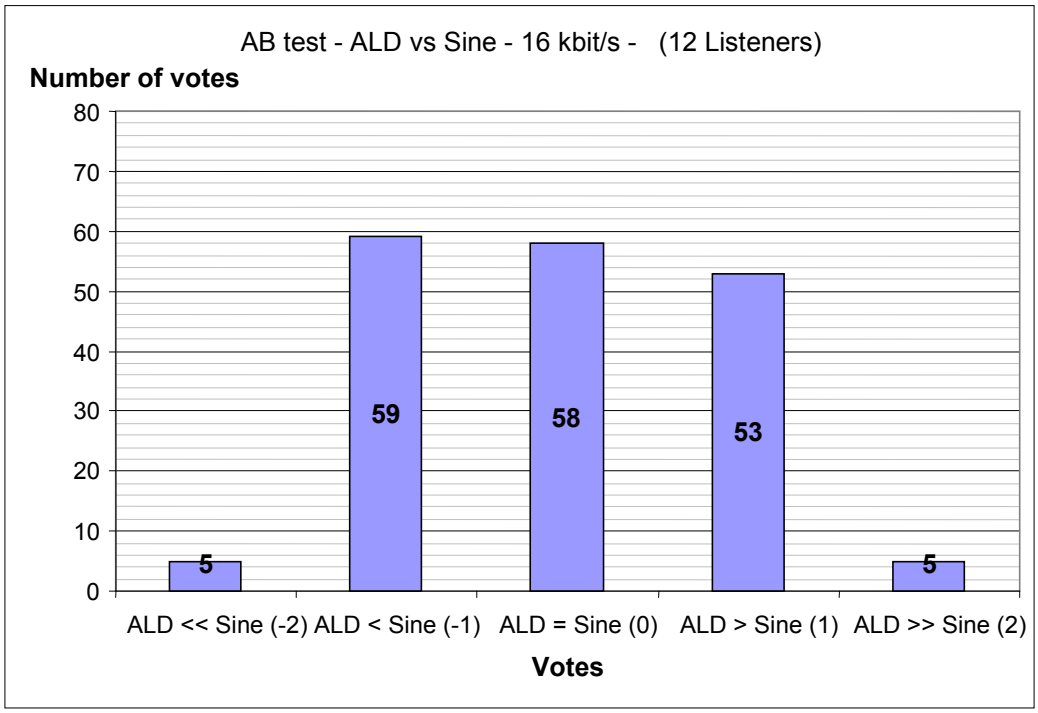


Figure 77 – AB listening test results for G.718 with ALD and Sine windows at 16 kbit/s

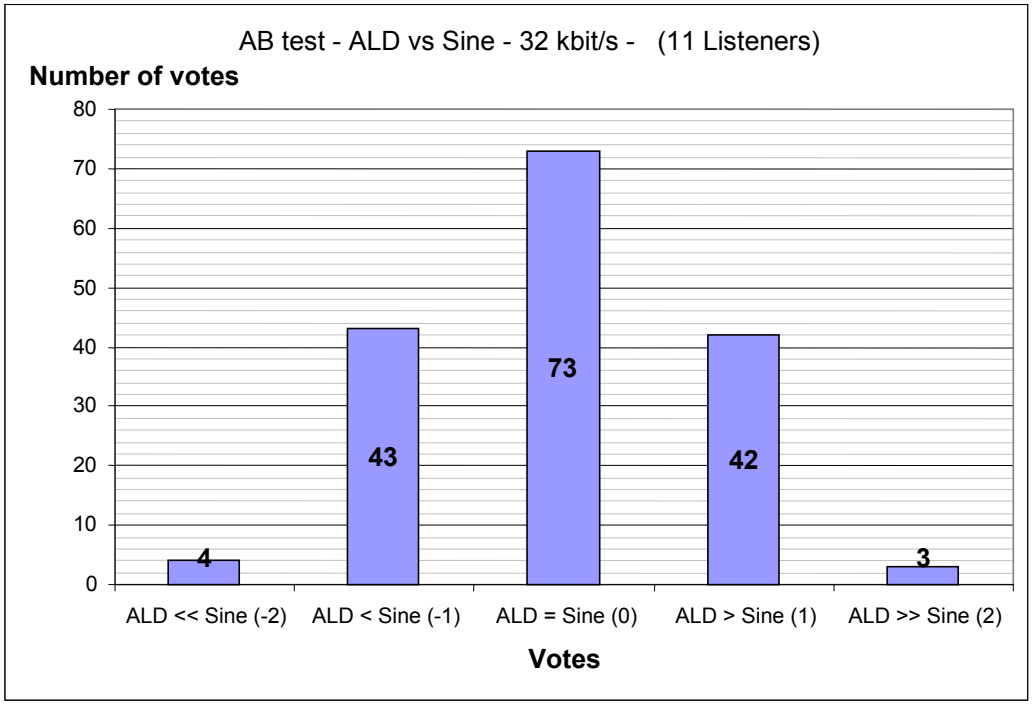


Figure 78 – AB listening test results for G.718 with ALD and Sine windows at 32 kbit/s

Finally, Figures 79 and 80 show the AB listening test results with frame erasure conditions. Figure 79 gives the results for G.718 operating at 16 kbit/s with 8% of frame loss rate, which is a quite important packet loss rate. Figure 80 gives the results for G.718 operating at 32 kbit/s with 5% of frame loss rate

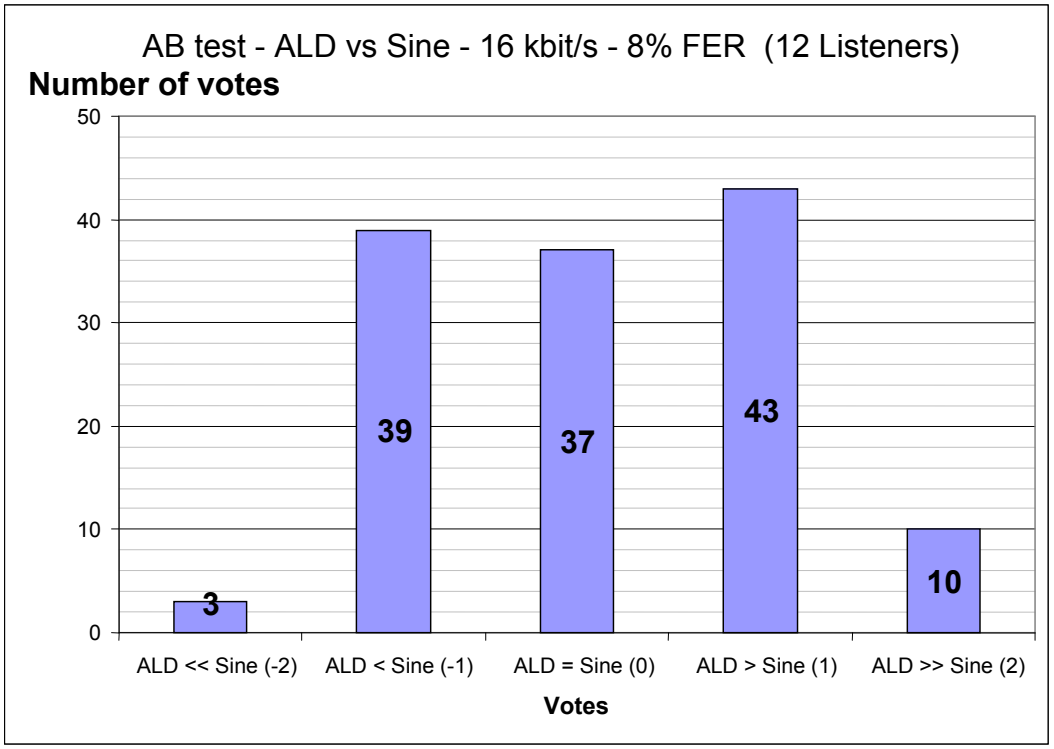


Figure 79 – AB listening test results for G.718 with ALD and Sine windows at 16 kbit/s with 8% packet loss

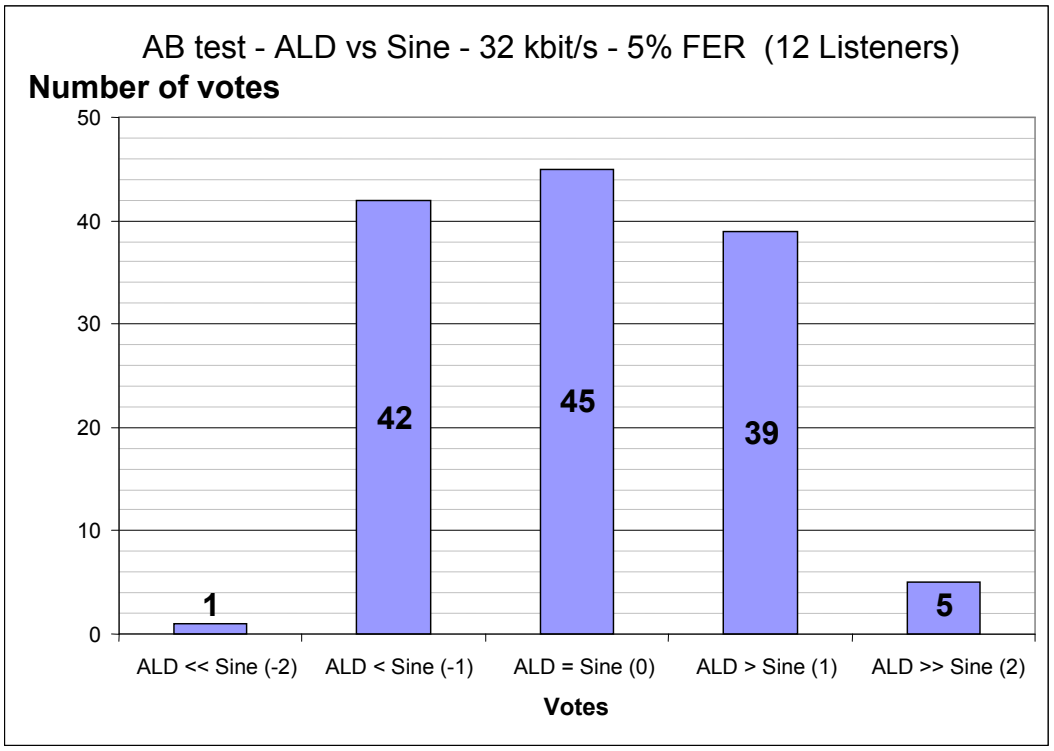


Figure 80 – AB listening test results for G.718 with ALD and Sine windows at 32 kbit/s with 5% packet loss

It can be seen that the frame error does not affect the quality of the codec even with the 10 ms delay reduction.

5.3.3 Conclusion on ALD window in G.718

The listening tests have demonstrated that the ALD window can dramatically reduce the delay of transform coding without degrading the quality with various audio signals. The delay reduction, which can be obtained with ALD window, provides more flexibility in the selection of the transform size for low delay application. Indeed, the ALD design method allows to reduce the total transform delay from 0 to theoretically M while keeping the perfect reconstruction. In practice, it has been seen that the initialization of $M/4$ zeros provides a good trade-off, ensuring a total delay reduction of $M/2$ without any significant performance degradation.

Chapter 6

Conclusion

In this work, an extension of the transform stage used for perceptual audio coding has been developed with a particular interest for the low delay application scenario. The low delay block switching and the low delay window design for MDCT are two components that have successfully been tested in low delay audio coding schemes.

6.1 Overview

The basic components of transform audio coding have been presented to give an overview of the main tools and the necessary techniques that are required such to develop a competitive audio coding scheme. The MDCT and related lapped transform (like ELT) are of course one of the main components as they are used in all the recent audio coding algorithms. The definition of the MDCT/ELT and its block size adaptation to the audio signal characteristics has been presented. The block switching tool has proven its benefit for transient sound coding, but was so far limited to coding schemes with less severe delay constraint.
The main contributions of this work have then been presented. It has been shown that the perfect reconstruction conditions of the MDCT can be relaxed and the obtained conditions provide more flexibility in the selection of the transform window, leading to the new framework called seamless reconstruction. Indeed, the low delay block switching and seamless reconstruction scheme offer the possibility to achieve the perfect reconstruction when switching between different MDCT resolutions or between non-matching windows.
Finally, the proposed MDCT extension has been implemented in a perceptual audio coder to evaluate the real improvement in terms of subjective and objective quality.

6.2 Thesis achievement

The proposed low delay block switching tool has been adapted to the MDCT and the low delay transform. This proposal gives the possibility to improve significantly the subjective quality of low delay perceptual audio coders for transient signals, such as speech signals. This low delay block switching has been successfully integrated in the MPEG low delay audio coding algorithms: LD-AAC and ELD-AAC. A significant improvement of the quality for transient signals has been demonstrated. Several listening tests have been performed during the standardization process of the ELD-AAC. All of them have demonstrated a significant quality improvement for transient sounds and speech content, while keeping the same maximum complexity for the decoder. This improvement has been acknowledged by most of the experts in the MPEG Audio Group. However, it has been considered as not sufficient to adopt the tool in the MPEG-4 ELD-AAC standard.

The seamless reconstruction method has been used to build a simple audio coding system with an extended MDCT windows set and the ability to keep the perfect reconstruction whatever window sequence is selected. This system has demonstrated a good improvement compared to a normal AAC windows set. Moreover, this experiment has combined the seamless reconstruction and the arbitrary switching among transform sizes. It was shown that a significant quality improvement can be expected by defining a new audio coding scheme based on this new extended MDCT toolbox.
Finally, the seamless reconstruction has been exploited to define a new family of MDCT windows ensuring low delay. The proposed procedure offers a simple definition of an Asymmetric Low Delay window which can significantly reduce the algorithmic delay associated with the transform while keeping the performance close the widely used sine window. This ALD window design has been used to reduce the delay of the transform coding layers in ITU-T G.EV-VBR leading to the standardization of a new scalable speech and audio codec ITU-T G.718. In this context, the ALD has demonstrated its ability to preserve most of the attractive properties of the MDCT. Indeed, the ALD provides perfect reconstruction, overlapping of blocks, critical sampling, good frequency selectivity and retain standard fast implementations.

This work has been conducted with a close relation to the ITU-T and MPEG standardization development. Several MPEG contributions have been produced to present the proposed low delay block switching tool [MPEG 07a,b,c,d,e] and the proposed solution has been described in [Virette 08] and [Philippe 08]. The low delay window design method has

successfully been adopted by ITU-T for the G.718 [ITU-T 08] [Vaillancourt 08].

6.3 Perspective

In this work, the low delay block switching and the ALD window have been developed to improve the audio quality while targeting a low algorithmic delay. They have been successfully adapted to low delay audio coders and demonstrated a significant improvement. The low delay block switching is obviously efficient for transient signals, while the ALD window offers good performance as a particularly interesting trade-off between frequency selectivity and temporal resolution. However, more work would be necessary to exhibit the window design criterion/perceptual quality relationship.

The seamless reconstruction from which the ALD window family has been derived is not necessarily limited to low delay applications. The first objective evaluation of this method has shown the potential improvement that can be obtained by extending the windows set of a coding scheme such as the one used in AAC. It is expected that this method implemented in a real transform audio coding scheme would definitely contribute to increase the flexibility of the window selection process. Increasing the transform length and permitting more signal dependent transform selection would certainly improve the quality if adapted to a coding scheme such as AAC.

Overall, the proposed MDCT toolbox extends the audio transform coding framework, with an increased flexibility in the selection of the transform in the delay-quality space.

Author's Bibliography

[MPEG 07a] P. Philippe, D. Virette, "Proposed Core Experiment for Enhanced Low Delay AAC", ISO/IEC JTC1/SC29/WG11 MPEG, M14237, Marrakech, Morocco, January 2007.

[MPEG 07b] P. Philippe, D. Virette, "Updated description for AAC ELD instantaneous block switching CE", ISO/IEC JTC1/SC29/WG11 MPEG, M14520, San José, USA, April 2007.

[MPEG 07c] P. Philippe, D. Virette, "Additional information for AAC ELD instantaneous block switching CE", ISO/IEC JTC1/SC29/WG11 MPEG, M14720, Lausanne, Switzerland, July 2007.

[MPEG 07d] P. Philippe, D. Virette, "Listening test results on block Switching Core Experiment for ELD-AAC", ISO/IEC JTC1/SC29/WG11 MPEG, M14978, Shenzhen, China, October 2007.

[MPEG 07e] P. Philippe, D. Virette, "Proposed changes to ELD AAC", ISO/IEC JTC1/SC29/WG11 MPEG, M14979, Shenzhen, China, October 2007.

[ITU-T 08] J. Hagqvist, J. Gibbs, M. Jelinek, J. Svedberg, J. Stachurski, H. Ehara, L. Zhang, D. Virette, "Proposed draft for new Recommendation G.VBR-EV "Frame error robust narrowband and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s"", ITU-T SG16 – C 481, Geneva, Switzerland, April 2008.

[Virette 08] D. Virette, B. Kövesi, P. Philippe, "Adaptive time-frequency resolution in modulated transform at reduced delay", Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP'08), pp. 3781-3784, Las Vegas, USA, April 2008.

[Philippe 08] P. Philippe, D. Virette, B. Kövesi, "Adaptive time-frequency resolution in modulated transform at reduced delay", 124th AES Convention, preprint 7333, Amsterdam, The Netherlands, May 2008.

[Vaillancourt 08] T. Vaillancourt, M. Jelínek, A. E. Ertan, J. Stachurski, A. Rämö, L. Laaksonen, J. Gibbs, U. Mittal, S. Bruhn, V. Grancharov, M. Oshikiri, H. Ehara, D. Zhang, F. Ma, D. Virette, S. Ragot, "ITU-T EV-VBR: A Robust 8-32 kbit/s Scalable Coder for Error Prone Telecommunications Channels," 16th EUSIPCO, Lausanne, Switzerland, August 25-29, 2008.

[Patent 07a] D. Virette, P. Philippe, B. Kövesi, "Low-delay transform coding using weighting windows",WO 2008/081144 A2, January 2007.

[Patent 07b] D. Virette, P. Philippe, "Transform-based coding/decoding, with adaptive windows",WO 2009/081003 A1, December 2007.

Bibliography

[3GPP 99] 3GPP TS 06.71, “Adaptive Multi-Rate speech processing functions; General description”, 1999.

[3GPP 99] 3GPP TS 26.171, “Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description”, 2002.

[Allamanche 99] E. Allamanche, R. Geiger, J. Herre, T. Sporer, “MPEG-4 Low Delay Audio Coding Based on the AAC Codec”, 106th AES Convention, Munich, Germany, May 1999.

[Adoul 87] J. P. Adoul and C. Lamblin. “A comparison of some algabraic structures for CELP coding of speech,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1953–1956, 1987.

[Bellanger 76] M. G. Bellanger, G. Bonnerot, and M. Coudreuse. Digital Filtering by Polyphase Network: Application to Sample-Rate Alteration and Filter Banks. IEEE Trans. Acoust., Speech, Signal Processing, 24:109 – 114, April 1976.

[Blauert 97] J. Blauert, “Spatial Hearing: The Psychoacoustics of Human Sound Localization”, MIT Press, Cambridge, USA, 1997.

[Bosi 99] M. Bosi, "Filter Banks in Perceptual Audio Coding," AES 17th Conference on high-quality Audio Coding, September 2-5, 1999, Florence, Italy.

[Brandenburg 99] K. Brandenburg, "MP3 and AAC explained", Proc. AES 17th International Conference: High-Quality Audio Coding, Florence, Italy, 1999 September 2-5.

[Breebaart 04] J. Breebaart, S. van de Par, A. Kohlrausch and E. Schuijers, "High-quality parametric spatial audio coding at low bit rates" Proc. 116th AES Conv., Berlin, Mai 2004.

[Breebaart 05a] J. Breebaart, S. van de Par, A. Kohlrausch and E. Schuijers, "Parametric Coding of Stereo Audio", EURASIP Journal on Applied Signal Processing, 2005:9, 1305-1322.

[Breebaart 05b] J. Breebaart, J. Herre, C. Faller, J. Roden, F. Myburg, S. Disch, H. Purnhagen, G. Hotho, M. Neusinger, K. Kjorling and W. Oomen, "MPEG spatial audio coding / MPEG surround: overview and current status" Proc. 119th AES Conv., New York, USA, 2005.

[Briand 06a] M. Briand, D. Virette et N. Martin, "Parametric representation of Multichannel Audio based on Principal Component Analysis", Proc. 120th AES Conv., Paris, 2006, preprint 6813.

[Briand 06b] M. Briand, D. Virette et N. Martin, "Parametric Coding of Stereo Audio based on Principal Component Analysis", 9th Int. Conf. on Digital Audio Effects DAFx'06, Montréal, 2006.

[Cheung 95] S. Cheung, J. S. Lim, "Incorporation of Biorthogonality into Lapped Transforms for Audio Compression", Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3079–3082, 1995.

[Chinen 11] T. Chinen, Y. Yamamoto, M. Hatanaka, H. Honma, M. Nishiguchi, "Finalization of CE on PVC for SBR", ISO/IEC JTC1/SC29/WG11 M19256, Daegu, Korea, January 2011.

[Collen 02] P. Collen, "Techniques d'enrichissement de spectre des signaux audionumériques", Ph.D. Thesis, Ecole Nationale Supérieure des Télécommunications, 2002.

[Conway 93] Conway J.H., Sloane N.J.A., "Sphere packings, lattices, and groups", Springer, 1993.

[Dietz 02] M. Dietz, L. Liljeryd, K. Kjörling et O. Kunz, "Spectral Band Replication, a novel approach in audio coding", Proc. 112th AES Conv., Munich, 2002, preprint 5553.

[Duhamel 91] Duhamel P., Mahieux Y. and Petit J.-P., "A Fast Algorithm For The Implementation of Filter Banks Based on Time Domain Aliasing ancellation," Int. Conf. on Acoust., Speech, Signal Proc., pp. 2209-2212, 1991.

[Edler 89] Edler B.: "Codierung von Audiosignalen mit übberlappender Transformation und adaptiven Fensterfunktionen", Frequenz 1989.

[Ekstrand 01] P. Ekstrand, "Aliasing reduction using complex-exponential modulated filterbanks", EP 1641120A2, April 02, 2001.

[Ekstrand 02] P. Ekstrand, "Bandwidth extension of audio signals by spectral band replication ", Proc.1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002), Leuven, Belgium, November 15, 2002.

[Faller 02] C. Faller, F. Baumgarte, "Binaural cue coding: a novel and efficient representation of spatial audio", Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP'02), vol. 2, pp. 1841-1844, Orlando, USA, Mai 2002.

[Faller 04] C. Faller, "Parametric Coding of Spatial Audio", Ph.D. Thesis, Ecole Polytechnique Fédérale de Lausanne, 2004.

[Fielder 96] L. D. Fielder, M. Bosi, G. A. Davidson, M. Davis, C. Todd, and S. Vernon" AC-2 and AC-3: Low Complexity Transform-Based Audio Coding," in N. Gielchrist and C. Grewin (ed.), Collected Papers on Digital Audio Bit-Rate Reduction, AES 1996, pp. 54-72.

[Fletcher 40] Fletcher H., "Auditory patterns", Review of Modern Physics, 1940.

[Gersho 92] Gersho A. and Gray R. M., "Vector Quantization and Signal Compression," Kluwer Academic Publishers, 1992.

[Grill 99] B. Grill, "The MPEG-4 General Audio Coder", 17th AES International Conference, Florence, Italy, September 2-5, 1999.

[Herre 92] J. Herre, K. Brandenburg et E. Eberlein, "Combined Stereo Coding", 93rd AES Conv., San Francisco, 1992, preprint 3369.

[Herre 94] J. Herre, K. Brandenburg and D. Lederer, "Intensity stereo coding", 96th AES Conv., Amsterdam, 1994, preprint 3799.

[Herre 96] J. Herre, J. D. Johnston, "Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS)", 101st AES Conv., Los Angeles, November 8-11 1996, preprint 4384.

[Herre 97] J. Herre, J. D. Johnston, "Exploiting Both Time and Frequency Structure in a System That Uses an Analysis/Synthesis Filterbank with High Frequency Resolution", 103rd AES Conv., New York, September 26-29 1997, preprint 4519.

[Herre 08] Herre, J., Kjorling, K., Breebaart, J., Faller, C., Chong, K. S., Disch, S., Purnhagen, H., Koppens, J., Hilpert, J., Rödén, J., Oomen, W., Linzmeier, K. and Villemoes, L., "MPEG Surround – The ISO/MPEG standard for efficient and compatible multi-channel audio coding", J. Audio Eng. Soc. 56 No 11, 932-955, 2008.

[Huffman 52] D. Huffman, "A Method for the Construction of Minimum Redundancy Codes", Proceedings of the IRE, 40:1098–1101, September 1952.

[ISO 92] ISO/IEC JTC1/SC29/WG11, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5Mbit/s", MPEG-1, ISO/IEC 11172, 1992.

[ISO 03] ISO/IEC JTC1/SC29/WG11, "Coding of audio-visual objects", MPEG-4, ISO/IEC 14496-3:2001/Amd 1, "Bandwidth extension", 2003.

[ISO 09] ISO/IEC JTC1/SC29/WG11, "Coding of audio-visual objects", MPEG-4, ISO/IEC 14496-3 Fourth Edition, "Part 3: Audio", 2009.

[ITU-R BS.1534-1 03] ITU-R BS.1534-1, "Method for the subjective assessment of intermediate quality level of coding systems", 2003.

[ITU-T G.718 08] ITU-T G.718, "Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s", 2008.

[ITU-T G.722.1 99] ITU-T G.722.1, "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss", 2005.

[ITU-T G.722.1C 05] ITU-T G.722.1 Annex C, "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss, Annex C - 14 kHz mode at 24, 32, and 48 kbit/s", 2005.

[ITU-T G.729 96] ITU-T G.729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)", 1996.

[ITU-T G.729.1 06] ITU-T G.729.1, "G.729 based Embedded Variable bit-rate coder: An 8-32 kbit/s scalable wideband coder bitstream interoperable with G.729", 2006.

[Iwakami 96] Iwakami N., Moriya T., "Transform-Domain Weighted Interleave Vector Quantization (TwinVQ)", 101st AES Conv., Los Angeles, November 8-11 1996, preprint 4377.

[Jayant 84] Jayant N. S., Noll P., "Digital Coding of Waveforms: Principles and Applications to Speech and Video," Prentice Hall, 1984.

[Johnston 92] J.D. Johnston et A. Ferreira, "Sum-difference stereo transform coding", Proc. IEEE Int. Conf. Acoustics, Signal Processing, vol.2, pp. 569-572, San Francisco, USA, Mars 1992.

[Kataoka 93] A. Kataoka, T. Moriya, S. Hayashi, "An 8-bit/s speech coder based on conjugate structure CELP", ICASSP 93, April 27-April 30, vol. 2

[Koilpillai 92] R.D. Koilpillai, P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction", IEEE Transactions on Signal Processing, 40(4):770-783, April 1992.

[Linde 80] Linde, Y., Buzo, A., Gray, R.M., An Algorithm for Vector Quantizer Design, IEEE Transactions on Communications, vol. 28, pp. 84–94, 1980

[Lloyd 57] S.P. Lloyd "Least Squares Quantization in PCM's", Bell Telephone Laboratories Paper, Murray Hill, NJ, 1957

[Makhoul 75] J. Makhoul. "Linear prediction: A tutorial review". Proceedings of the IEEE, 63 (5):561–580, April 1975.

[Malvar 90] Malvar, H.S: "Lapped transforms for efficient transform/subband coding," IEEE Trans. Acoust., Speech, Signal Processing, vol. 38, pp. 969–978, June 1990.

[Malvar 91] Malvar, H.S: "Extended lapped transforms: fast algorithms and applications", ICASSP 1991, 14-17 April 1991 Page(s):1797 - 1800 vol.3.

[Malvar 92a] Malvar, H.S: "Extended lapped transforms: Properties, applications and fast algorithms", IEEE Transactions on Signal Processing, 40(11):2703-2714, November 1992.

[Malvar 92b] H. S. Malvar, "Signal Processing with Lapped Transforms," Artech House, Inc. 1992.

[Moore 03] Moore B.C.J., "An Introduction to the Psychology of Hearing", Academic Press, 5th edition, 2003.

[Moreau 95] Moreau N., "Techniques de compression des signaux," Masson 1995.

[N3075 99] R. Sperschneider, F. Feige, S. Quackenbush, " Report on the MPEG-4 Audio Version 2 Verification Test ", ISO/IEC JTC 1/SC 29/ WG11, MPEG N3075, December 1999. (http://mpeg.chiariglione.org/working_documents/mpeg-04/audio/mpeg-4_audio_v2_ver.zip)

[Nagel 09] F. Nagel, S. Disch, N. Rettelbach, "A phase vocoder driven bandwidth extension method with novel transient handling for audio codecs", 126th AES Convention , Munich, Germany, May 2009.

[Neuendorf 09] 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, B. Grill, "Unified Speech and Audio Coding Scheme for High Quality at Low Bit Rates", International Conference on Acoustics, Speech and Signal Processing (ICASSP'2009), Taipei, TAIWAN April 19-24, 2009.

[Paliwal 91] Paliwal K.K. and Atal B.S., “Efficient vector quantization of LPC parameters at 24 bits/frame”, Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 661-664, May 1991.

[Philippe 01] P. Philippe, P. Collen and J-B. Rault, "Description of the France Telecom proposal for the Call for Evidence", ISO/IEC JTC1/SC29/WG11 M6721l, Pisa, Italy, January 2001.

[Princen 86] J. P. Princen, A. B. Bradley, "Analysis/Synthesis filter Bank Design Based on Time Domain Aliasing Cancellation," IEEE Trans. on ASSP, Vol. ASSP-34, No. 5, October 1986.

[Princen 87] J. P. Princen, A. Johmson, A. B. Bradley, "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation", International Conference on Acoustics, Speech and Signal Processing (ICASSP 1987), 1987.

[Ragot 07] S. Ragot, B. Kövesi, R. Trilling, D. Virette, N. Duc, D. Massaloux, S. Proust, B. Geiser, M. Gartner, S. Schandl, H. Taddei, Y. Gao, E. Shlomot, H. Ehara, K. Yoshida, T. Vaillancourt, R. Salami, M.S. Lee, D.Y. Kim, "ITU-T G.729.1: An 8-32 kbit/s scalable coder interoperable with G.729 for wideband telephony and Voice over IP," Proc. ICASSP, Honolulu, Hawaii, USA, April 2007.

[Rothweiler 83] Rothweiler J. H., "Polyphase Quadrature Filters - A New Subband Coding Technique," IEEE Int. Conf. On Acoust., Speech, Signal Proc., pp. 1280-1283, Boston 1983.

[Sabin 82] Sabin, M. J.; Gray, R. M., “Product code vector quantizers for speech waveform coding”, Globecom '82, Miami, Nov 29-Dec 2, 1982, Vol. 3.

[Salomon 00] D. Salomon. Data compression: the complete reference. Springer-Verlag, New York, 2nd edition edition, 2000.

[Schnell 07] M. Schnell, R. Geiger, M. Schmidt, M. Multrus, M. Mellar, J. Herre, G. Schuller, "Modified Discrete Cosine Transform – Its Implications for Audio Coding and Error Concealment", Journal of Audio Eng. Soc., Vol. 51, No 1/2 2003

[Schnell 08] M. Schnell, M. Schmidt, M. Jander, T. MAlbert, R. Geiger, V. Ruoppila, P. Ekstrand, B. Grill, " MPEG-4 Enhanced Low Delay AAC — A New Standard for High Quality Communication ", 125th AES Conv., Preprint 7503, 2008

[Schroeder 85] M. R. Schroeder and B. S. Atal. Code-Excited Linear Prediction (CELP): High-quality speech at very low bit rates. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 937–940, 1985.

[Schuller 00] Gerald D. T. Schuller, Tanja Karp: “Modulated Filter Banks with Arbitrary System Delay: Efficient Implementations and the Time-Varying Case”, IEEE Transactions on Signal Processing, Vol. 48, No. 3, March 2000.

[Shannon 48] Shannon C. “A mathematical theory of communication”, Bell System Technical Journal, vol. 27, p. 379-423 and 623-656, July and October, 1948.

[Shimada 04] Shimada O., Nomura T., Takamizawa Y., Serizawa M., Tanaka N., Tsushima M., Norimatsu T., Seng C.K., Hann K.K., Hong N.S., "A low power SBR algorithm for the MPEG-4 audio standard and its DSP implementation", AES 116th Convention, Berlin, Germany, 2004 May 8-11.

[Shlien 97] Shlien S., "The modulated lapped transform, its time-varying forms, and its applications to audio coding standards", IEEE Transactions on Speech and Audio Processing, Volume 5, Issue 4, July 1997 Page(s):359 – 366.

[So 07] So S., Paliwal K.K. "A comparative study of LPC parameter representations and quantisation schemes for wideband speech coding", Digital Signal Processing, Elsevier, Volume 17, Issue 1, January 2007 Page(s):114 – 137.

[Vaidyanathan 93] P. P. Vaidyanathan, "Multirate Systems and Filter Banks," Prentice Hall, Englewood Cliffs, NJ, 1993.

[Vaillancourt 08] T. Vaillancourt, M. Jelínek, A. E. Ertan, J. Stachurski, A. Rämö, L. Laaksonen, J. Gibbs, U. Mittal, S. Bruhn, V. Grancharov, M. Oshikiri, H. Ehara, D. Zhang, F. Ma, D. Virette, S. Ragot, "ITU-T EV-VBR: A Robust 8-32 kbit/s Scalable Coder for Error Prone Telecommunications Channels," 16th EUSIPCO, Lausanne, Switzerland, August 25-29, 2008.

[Wang 03] Wang Y., Vilermo M.: "Modified Discrete Cosine Transform – Its Implications for Audio Coding and Error Concealment", Journal of Audio Eng. Soc., Vol. 51, No 1/2 2003

[Witten 87] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression”, Commun. ACM 30, 6 June 1987.

[Zwicker 07] Zwicker E., H. Fastl, "Psychoacoustics - Facts and Models," Springer, 2007.

Annex A

In this Annex, we present the polyphase decomposition of the MDCT according to the definition of the polyphase representation given in 2.1.3.3 and the MDCT definition introduced in 3.1.1. This polyphase representation of the MDCT can be used to define the perfect reconstruction constraint.

A.1 Polyphase decomposition of MDCT

As introduced in 2.1.3.3, the analysis filter bank, $H_k(z)$ can be written into a sum of M terms, known as polyphase decomposition:

$$H_k(z) = \sum_{l=0}^{M-1} z^{-l} E_{k,l}\left(z^M\right) \tag{A.1}$$

Where $E_{k,l}(z)$ are the terms of the polyphase component matrix. The polyphase components are expressed with the analysis filter impulse responses $h_{a,k,n}$ for sub-band k k and coefficient n. For the MDCT, taking into account a transform of size M with impulse response of size $2M$, the polyphase component matrix is written:

$$\mathbf{E}(z) = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) & \cdots & E_{0,M-1}(z) \\ E_{1,0}(z) & E_{1,1}(z) & \cdots & E_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{0,M-1}(z) & E_{1,M-1}(z) & \cdots & E_{M-1,M-1}(z) \end{bmatrix} \tag{A.2}$$

Based on the impulse response, this matrix can be expressed by:

$$\mathbf{E}(z) = \begin{bmatrix} h_{a,0,0} + z^{-1}h_{a,0,M} & h_{a,0,1} + z^{-1}h_{a,0,M+1} & \cdots & h_{a,0,M-1} + z^{-1}h_{a,0,2M-1} \\ h_{a,1,0} + z^{-1}h_{a,1,M} & h_{a,1,1} + z^{-1}h_{a,1,M+1} & \cdots & h_{a,1,M-1} + z^{-1}h_{a,1,2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{a,M-1,0} + z^{-1}h_{a,M-1,M} & h_{M-1,1} + z^{-1}h_{M-1,M+1} & \cdots & h_{a,M-1,M-1} + z^{-1}h_{a,M-1,2M-1} \end{bmatrix} \tag{A.3}$$

which leads to the following definition which separates the contribution of the two frames:

$$\mathbf{E}(z)=\begin{bmatrix} h_{a,0,0} & h_{a,0,1} & \cdots & h_{a,0,M-1} \\ h_{a,1,0} & h_{a,1,1} & \cdots & h_{a,1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{a,M-1,0} & h_{a,M-1,1} & \cdots & h_{a,M-1,M-1} \end{bmatrix} + z^{-1}\begin{bmatrix} h_{a,0,M} & h_{a,0,M+1} & \cdots & h_{a,0,2M-1} \\ h_{a,1,M} & h_{a,1,M+1} & \cdots & h_{a,1,2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{a,M-1,M} & h_{a,M-1,M+1} & \cdots & h_{a,M-1,2M-1} \end{bmatrix} \tag{A.4}$$

Using the definition of the basis functions $c_{k,n}$ given in equation (3.2):

$$c_{k,n}=\sqrt{\frac{2}{M}}\cos\left(\frac{\pi}{M}\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\right) \tag{A.5}$$

with $0\le n\le 2M-1$, $0\le k\le M-1$ and the prototype (or window) for the analysis filters of length $2M$, we have:

$$\begin{aligned}\mathbf{E}(z)=&\begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,M-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1,0} & c_{M-1,1} & \cdots & c_{M-1,M-1} \end{bmatrix}\begin{bmatrix} h_{a,0} & 0 & \cdots & 0 \\ 0 & h_{a,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{a,M-1} \end{bmatrix} \\ &+z^{-1}\begin{bmatrix} c_{0,M} & c_{0,M+1} & \cdots & c_{0,2M-1} \\ c_{1,M} & c_{1,M+1} & \cdots & c_{1,2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1,M} & c_{M-1,M+1} & \cdots & c_{M-1,2M-1} \end{bmatrix}\begin{bmatrix} h_{a,M} & 0 & \cdots & 0 \\ 0 & h_{a,M+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{a,2M-1} \end{bmatrix}\end{aligned} \tag{A.6}$$

We introduce the notation:

$$\mathbf{C_0}=\begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,M-1} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1,0} & c_{M-1,1} & \cdots & c_{M-1,M-1} \end{bmatrix} \tag{A.7}$$

and

$$\mathbf{C_M}=\begin{bmatrix} c_{0,M} & c_{0,M+1} & \cdots & c_{0,2M-1} \\ c_{1,M} & c_{1,M+1} & \cdots & c_{1,2M-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M-1,M} & c_{M-1,M+1} & \cdots & c_{M-1,2M-1} \end{bmatrix} \tag{A.8}$$

for the two parts of the modulation matrix. The window parts of the polyphase definition are given by:

$$\mathbf{h}_{\mathbf{a,0}} = \begin{bmatrix} h_{a,0} & 0 & \cdots & 0 \\ 0 & h_{a,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{a,M-1} \end{bmatrix} \tag{A.9}$$

and

$$\mathbf{h}_{\mathbf{a},1} = \begin{bmatrix} h_{a,M} & 0 & \cdots & 0 \\ 0 & h_{a,M+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{a,2M-1} \end{bmatrix} \tag{A.10}$$

Hence, we obtain:

$$\mathbf{E}(z) = \mathbf{C}_0\mathbf{h}_{\mathbf{a,0}} + z^{-1}\mathbf{C}_1\mathbf{h}_{\mathbf{a,1}} \tag{A.11}$$

Based on the same notation, for the synthesis filter bank, we have the following definition of the polyphase representation:

$$\mathbf{R}(z) = \mathbf{h}_{\mathbf{s,0}}\mathbf{C}_1^{\mathrm{T}} + z^{-1}\mathbf{h}_{\mathbf{s,1}}\mathbf{C}_0^{\mathrm{T}} \tag{A.12}$$

where $h_{s,n}$ is the synthesis prototype filter (or window) with length $2M$.

A.2 Perfect reconstruction with MDCT

The perfect reconstruction of the MDCT is ensured if the complete system defined by the analysis and synthesis filter bank verifies the following conditions:

$$\begin{aligned} \mathbf{R}(z)\mathbf{E}(z) &= \left(\mathbf{h}_{\mathbf{s,0}}\mathbf{C}_1^{\mathrm{T}} + z^{-1}\mathbf{h}_{\mathbf{s,1}}\mathbf{C}_0^{\mathrm{T}}\right)\left(\mathbf{C}_0\mathbf{h}_{\mathbf{a,0}} + z^{-1}\mathbf{C}_1\mathbf{h}_{\mathbf{a,1}}\right) \\ &= \mathbf{h}_{\mathbf{s,0}}\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_0\mathbf{h}_{\mathbf{a,0}} + z^{-1}\left[\mathbf{h}_{\mathbf{s,0}}\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1\mathbf{h}_{\mathbf{a,1}} + \mathbf{h}_{\mathbf{s,1}}\mathbf{C}_0^{\mathrm{T}}\mathbf{C}_0\mathbf{h}_{\mathbf{a,0}}\right] + z^{-2}\mathbf{h}_{\mathbf{s,1}}\mathbf{C}_0^{\mathrm{T}}\mathbf{C}_1\mathbf{h}_{\mathbf{a,1}} \end{aligned} \tag{A.13}$$

with

$$\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_0 = \mathbf{C}_0^{\mathrm{T}}\mathbf{C}_1 = \mathbf{0} \tag{A.14}$$

and

$$\mathbf{h}_{s,0}\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1\mathbf{h}_{a,1} + \mathbf{h}_{s,1}\mathbf{C}_0^{\mathrm{T}}\mathbf{C}_0\mathbf{h}_{a,0} = \mathbf{I}_{\mathrm{M}} \tag{A.15}$$

Those two equations represent the polyphase notation of equations (3.28) and (3.29). If the conditions are verified, the application of analysis and synthesis filter banks lead to the prefect reconstruction of the input signal:

$$\mathbf{R}(z)\mathbf{E}(z) = z^{-1}\mathbf{I}_M \tag{A.16}$$

The MDCT is then a perfect reconstruction system introducing one frame delay (*M* samples).
Based on this condition for perfect reconstruction, one can write the synthesis filter bank as a function of the analysis filter bank. This is obtained using the following equation:

$$\mathbf{R}^{-1}(z)\mathbf{R}(z)\mathbf{E}(z) = z^{-1}\mathbf{R}^{-1}(z) \tag{A.17}$$

which results in the direct relation between analysis and synthesis:

$$\mathbf{R}^{-1}(z) = z\mathbf{E}(z) \tag{A.18}$$

Annex B

B.1 Low delay transition window

Figures 81 and 82 show the comparison between the low delay prototypes used in the MPEG-4 ELD-AAC and the transition windows (LONG_START and LONG_STOP synthesis windows) as represented on Figure 42.

Figure 81 – ELD-AAC prototype (blue), LONG_START synthesis window (red)

Figure 82 – ELD-AAC prototype (blue), LONG_STOP synthesis window (red)

For reference, we draw the comparison between start window based on sine long window and eld window (red). Their responses are very similar.

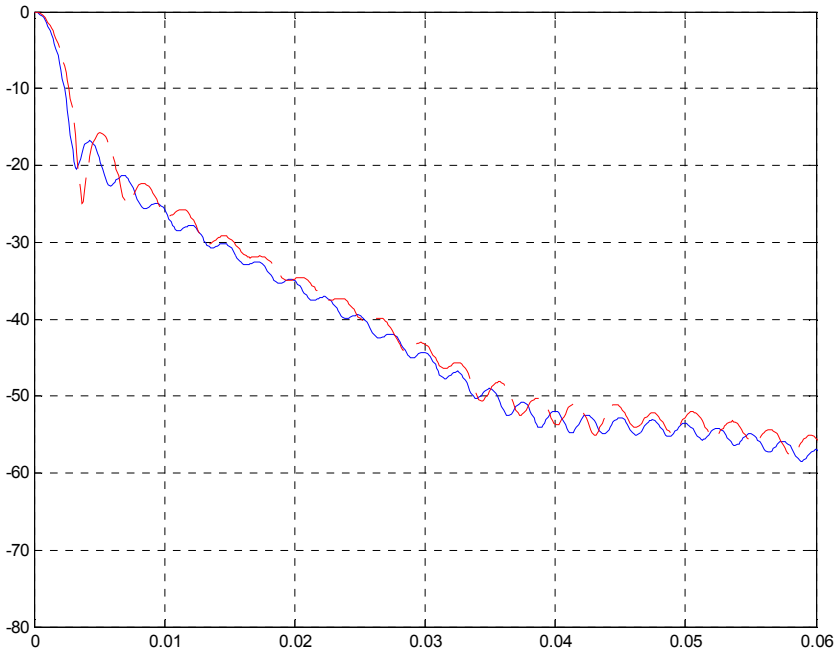


Figure 83 – LONG_START sine window (blue), LONG_START ELD-AAC synthesis window (red)

B.2 Frequency response of ALD window

Figures 84 and 85 represent the frequency response of the first four bands for the analysis and synthesis transform respectively. All the figures of section B.2 are obtained with M=64.

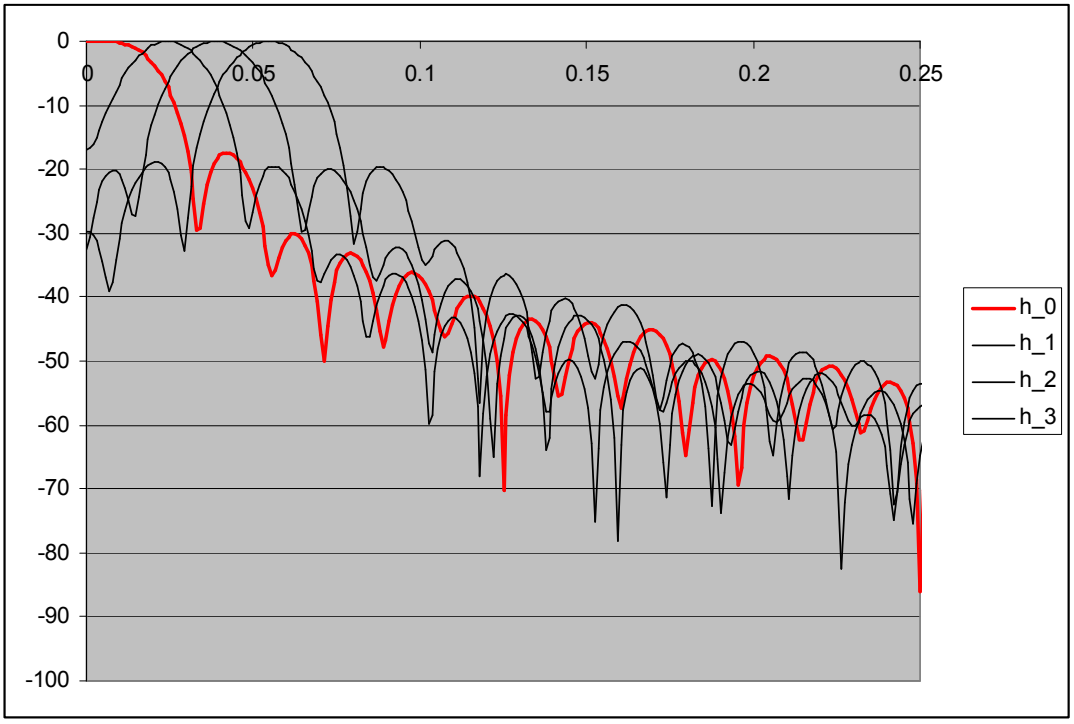


Figure 84 – First 4 sub-band filters of the analysis ALD transform

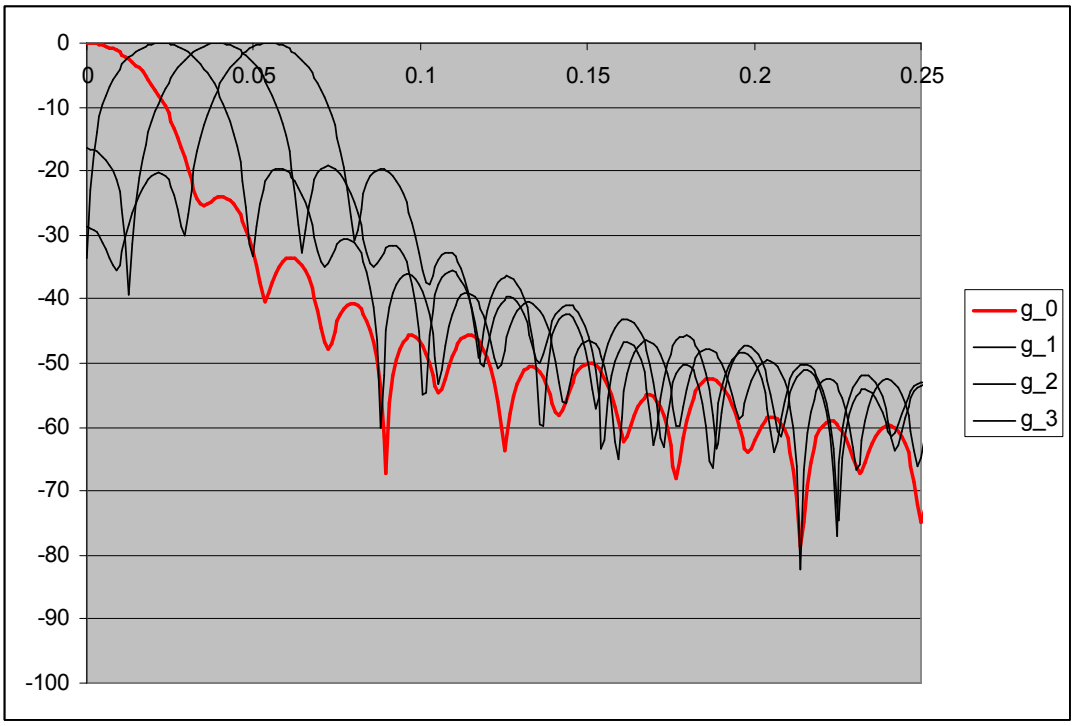


Figure 85 – First 4 sub-band filters of the synthesis ALD transform

Figure 86 shows the same four bands of the MDCT transform with sine window.

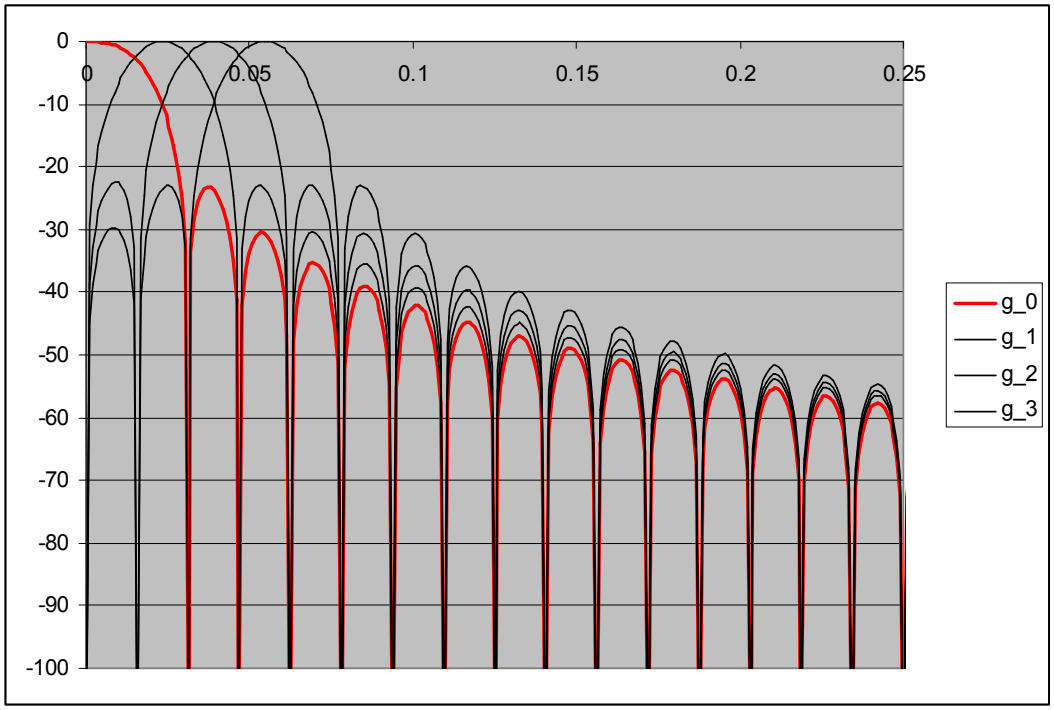


Figure 86 – First 4 sub-band filters of the MDCT transform with sine window

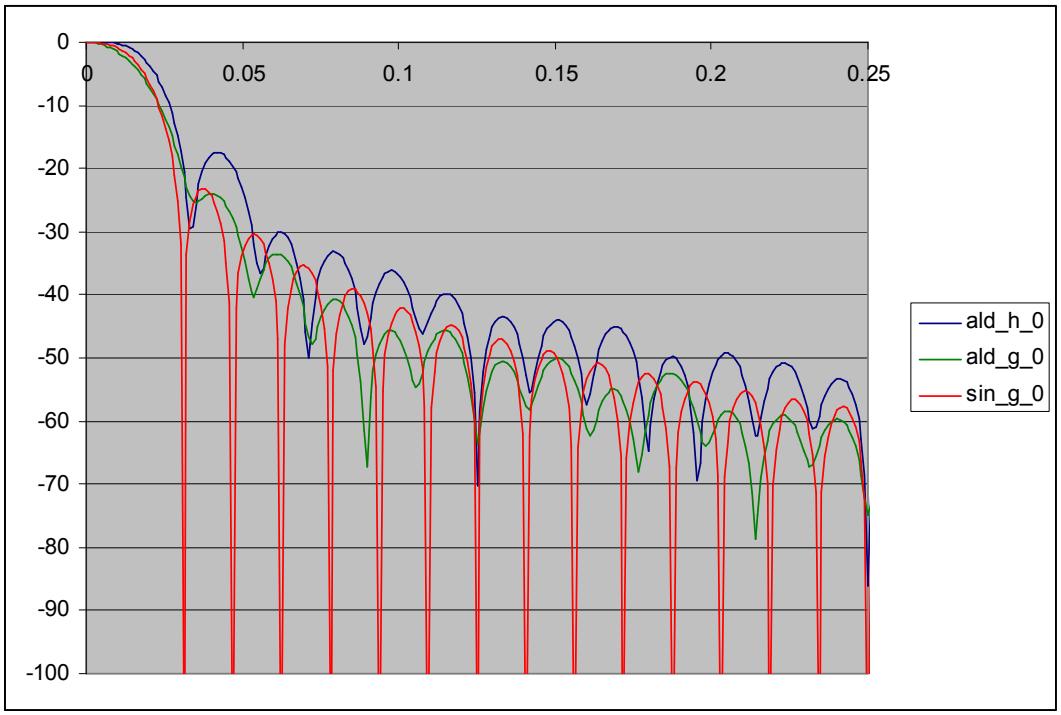


Figure 87 – First band filter for analysis ALD transform (blue), synthesiss ALD transform (green), MDCT transform with sine window (red)

Figures 88 to 91 represent the impulse responses of the sub-band filter corresponding to the ALD window MDCT transform. It can be seen that the asymmetric property of the window introduces more significant differences for low frequency band where the asymmetric window will strongly modified the modulation. This characteristic is important as it affect the theoretical coding gain which relies on a AR process of first order which has strong low frequency components.

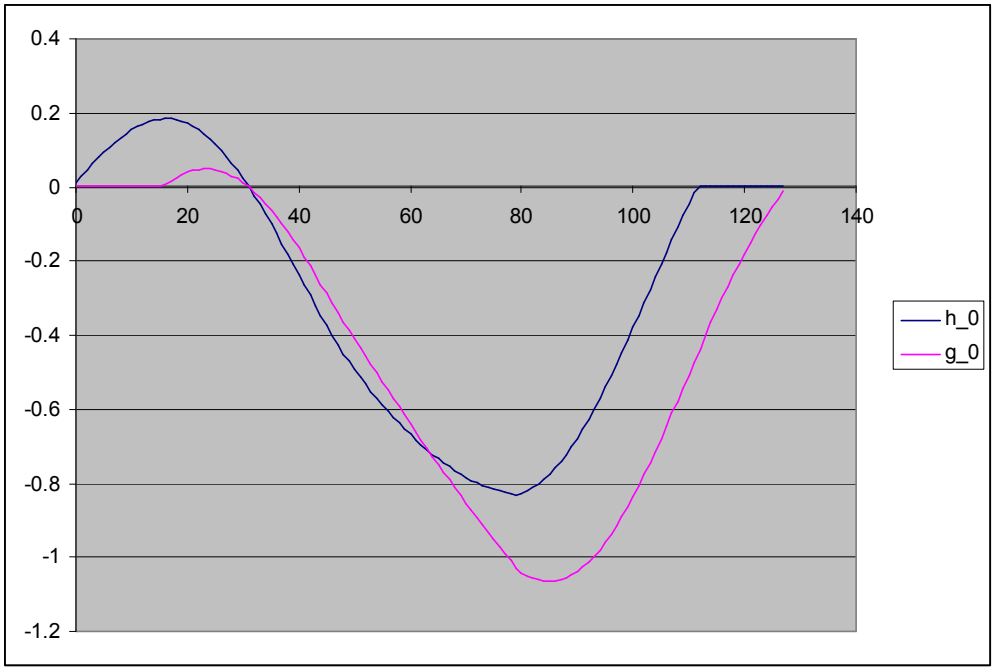


Figure 88 – Analysis (blue) and synthesis (pink) ALD transform (band 0)

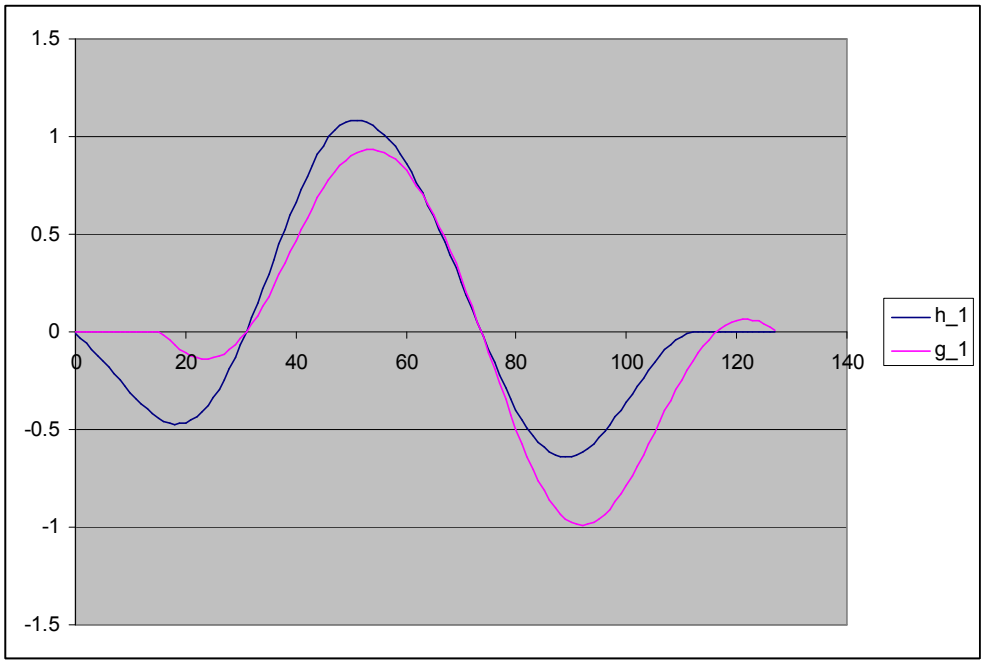


Figure 89 – Analysis (blue) and synthesis (pink) ALD transform (band 1)

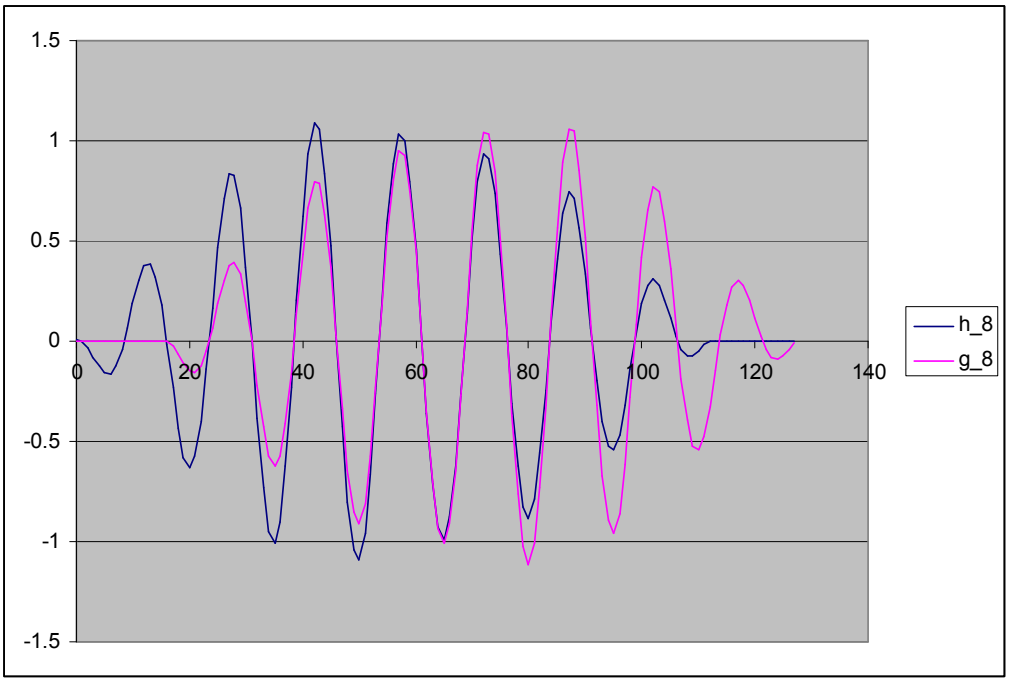


Figure 90 – Analysis (blue) and synthesis (pink) ALD transform (band 8)

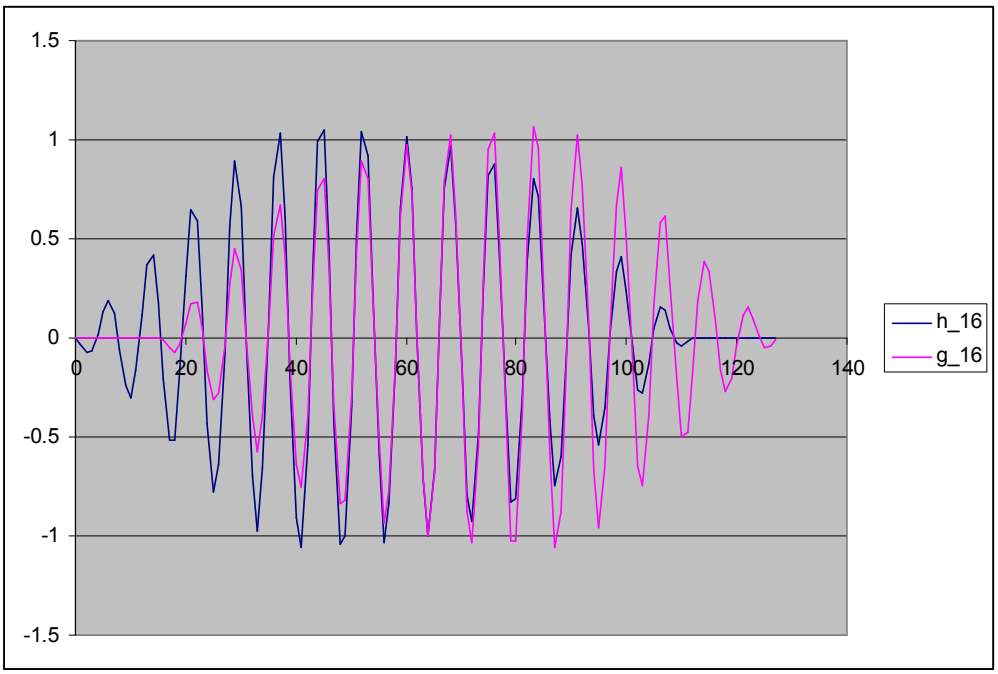


Figure 91 – Analysis (blue) and synthesis (pink) ALD transform (band 16)

Annex C

C.1 Rate-distortion cost function for quantization error

In order to model the effect of a scalar quantizer according to 2.2.2.1 in an audio coding system, the rate-distortion function must be defined. The quantization error is usually expressed by the MSE as given in equation (2.33). It can be rewritten as:

$$\sigma_d^2 = \mathrm{E}\left[\left(x - \mathrm{E}[x]\right)^2\right] = \sum_k \int_{I_k} \left(x - \hat{x}_k\right)^2 P_x(x)\,dx \tag{C.1}$$

where $P_x(x)$ is the probability density function (PDF) of the quantizer input x and I_k denotes the k^{th} quantization interval. For the mid-tread staircase quantisation characteristic illustrated in Figure 12, this is given by:

$$I_k : \left\{ kq - \frac{q}{2} < x \le kq + \frac{q}{2} \right\} \tag{C.2}$$

leading to a quantizer output of $\hat{x}_k = kq$. The bit rate R is then approximated by the entropy E as given in equation (2.36) and can then be written:

$$R = E = \sum_k P_k \log_2 P_k \tag{C.3}$$

where P_k is the probability of an occurrence of output $\hat{x}_k$. In practice the input PDF will generally be unknown. It can however be approximated as a generalised Gaussian distribution as shown in Figure 83, which for a normalisation factor c is given by:

$$P_x(x) = ce^{\left(\frac{|x|}{\sigma_X^2}\right)^\gamma} \tag{C.4}$$

with σ_X^2 denoting the frequency domain signal variance of the input frame, given by:

$$\sigma_X^2 = \frac{1}{N}\sum_{i=0}^{N-1} X_i^2 \tag{C.5}$$

This representation of the input PDF assumes a mean of zero, and requires an estimation of the exponent γ which describes the steepness of the generalised Gaussian curve.

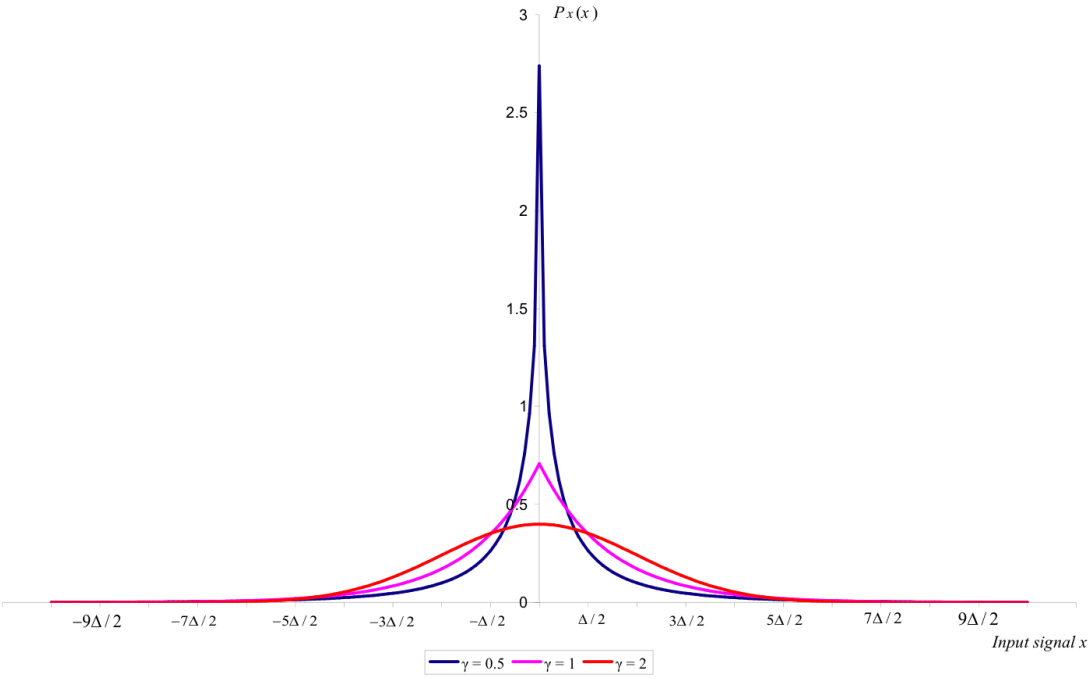


Figure 92 – Generalized Gaussian probability density function $P_x(x)$

If q is assumed to be sufficiently small (or conversely R sufficiently high) such that the probability is constant over each step, equation (C.1) can be approximated by equation (2.37) and the rate R becomes:

$$R = E = e(x) - \log_2 q \tag{C.6}$$

where $e(x)$ is the differential entropy given by:

$$e(x) = \frac{1}{\gamma \ln 2} - \log_2\left(\frac{\gamma}{2\Gamma(1/\gamma)} \cdot \frac{1}{\sigma_X}\sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}}\right) \tag{C.7}$$

As shown in section 2.2.2.1, the quantization error can be expressed as a function of the differential entropy and the rate, (2.38) can be written as:

$$\sigma_d^2 = \frac{1}{12} 2^{2e(x)} 2^{-2R} \tag{C.8}$$

In order to express the quantisation error as a function of R and σ_X^2, the quantizer performance factor ε^2 can be defined such that the final RD function is given by:

$$\sigma_d^2 = \varepsilon^2 2^{-2R} \sigma_X^2 \tag{C.9}$$

This uses the following expression which is dependent on the signal variance σ_X^2:

$$\varepsilon^2 = \frac{2^{2e(x)}}{12\sigma_X^2} \tag{C.10}$$

However, it can be shown that ε^2 depends only on the exponent γ, and is given by:

$$\varepsilon^2 = \frac{\left(\Gamma\left(1/\gamma\right)\right)^3 e^{2/\gamma}}{3\Gamma\left(3/\gamma\right)\gamma^2} \tag{C.11}$$

Thus by approximating the input PDF as a generalised Gaussian distribution, the cost function (C.9) relating the reconstruction error σ_d^2 and the rate R can be obtained.

C.2 Optimal Bit Allocation and Error Variance

This section examines the theory required to determine the optimal allocation of the average bit rate R, both in time (i.e. R_t) and in the sub-band domain (R_{SB}). Firstly, in order to establish which sub-bands should be allocated the highest number of bits, the signal variance of each sub-band must be computed as follows:

$$\sigma_{X_{SB}}^2 = \frac{1}{K_{SB}} \sum_{k=j}^{j+K_{SB}} X_k^2 \tag{C.12}$$

where j is the index of the first element in the sub-band SB, and K_{SB} denotes the sub-band width. It has been shown in (C.9) that for sufficiently high R, the quantization error introduced in each sub-band can be approximated by the equation:

$$\sigma_{d_{SB}}^2 = \varepsilon^2 2^{-2R_{SB}} \sigma_{X_{SB}}^2 \tag{C.13}$$

with a constant quantizer performance factor ε^2. The number of bits R_{SB} allocated to a particular sub-band is given by the optimal allocation paradigm:

$$R_{SB} = R_t + \frac{1}{2}\log_2 \frac{\sigma_{X_{SB}}^2}{\sigma_{GMt}^2} \tag{C.14}$$

where R_t denotes the bit rate allocated to each time segment .The quantity σ_{GMt}^2 is the weighted geometric mean of all sub-band variances in transform block, given by:

$$\sigma_{GMt}^2 = \left(\prod_{SB=0}^{N-1} \sigma_{X_{SB}}^{2K_{SB}} \right)^{\frac{1}{\sum_{SB=0}^{N-1} K_{SB}}} \tag{C.15}$$

As a consequence of the optimal allocation equation (C.14) the error variance will be constant over all sub-bands, thus the actual noise to be injected in each transform is:

$$\sigma_{qt}^2 = \varepsilon^2 2^{-2R_t} \sigma_{GMt}^2 \tag{C.16}$$

giving a total error variance for each combination of:

$$\sigma_{qc}^2 = \frac{1}{M} \sum_{t=0}^{T-1} M_t \sigma_{qt}^2 \tag{C.17}$$

where M_t is the transform length. To determine R_t, the Lagrange multiplier method can be used to minimise:

$$J(\lambda) = \sigma_{qc}^2 + \lambda MR = \sigma_{qc}^2 + \lambda \sum_{t=0}^{T-1} M_t R_t \tag{C.18}$$

which leads to the following results:

$$R_t = \frac{1}{2}\log_2 \left(\frac{\varepsilon^2 \sigma_{GMt}^2}{\lambda M} \right) \tag{C.19}$$

and

$$\lambda = 2^{\left(\frac{\sum_{t=0}^{T-1} M_t \log_2 \left(\frac{\varepsilon^2 \sigma_{GMt}^2}{M} \right) - 2MR}{M} \right)} \tag{C.20}$$

Then in order to provide the optimal allocation in situations where one or more time segments contain no information, and hence require no bits to be allocated, the λ factor can be adjusted as follows:

$$\lambda = 2^{\left(\frac{\sum_{t=0}^{T-1} M_t \log_2 \left(\frac{\varepsilon^2 \sigma_{GMt}^2}{M} \right) - 2MR}{\sum_{t=0}^{T-1} M_t} \right)} \tag{C.21}$$

summing over transforms where $\sigma_{GMt}^2 \neq 0$. In this way the average bit rate R can be variably distributed between each R_t and R_{SB} so as to minimize the quantization error for each input frame.

Annex D

$\sigma_{x,k}^2$	Signal variance in sub-band k
$\sigma_{q,k}^2$	Quantization noise variance in sub-band k
$\sigma_{r,k}^2$	Quantization noise variance in sub-band k after synthesis filter k
ε_*^2	Constant representing quantizer characteristics
r_k	Number of bits per sample in sub-band k
$\gamma_{xx}\left(e^{j\omega}\right)$	Power spectral density of signal x
$h_{k,n} \rightleftharpoons H_k\left(e^{j\omega}\right)$ vector $\mathbf{h}_k$	Analysis filter impulse response k
$g_{k,n} \rightleftharpoons G_k\left(e^{j\omega}\right)$ vector $\mathbf{g}_k$	Synthesis filter impulse response k
$\sigma_{g_k}^2 = \frac{1}{2\pi}\int_{-\pi}^{+\pi}\left\|G_k\left(e^{j\omega}\right)\right\|^2 d\omega = \sum_{n=0}^{L-1} g_{k,n}^2$	Power of synthesis filter k
M	Number of sub-bands
L	Filter length
R	Bit rate

$\sigma_{q,k}^2 = \varepsilon_*^2 2^{-2r_k}\sigma_{x,k}^2$ represents the quantization noise in a sub-band k.

$\sigma_{x,k}^2 = \int_{-\pi}^{+\pi}\gamma_{xx}\left(e^{j\omega}\right)\left|H_k\left(e^{j\omega}\right)\right|^2\frac{d\omega}{2\pi} = \mathbf{h}_k^T\mathbf{R}_{xx}\mathbf{h}_k$ is the energy in the sub-band k

$\sigma_{r,k}^2 = \frac{1}{M}\int_{-\pi}^{+\pi}\gamma_{q,k}\left(e^{j\omega}\right)\left|G_k\left(e^{j\omega}\right)\right|^2\frac{d\omega}{2\pi}$

With the assumption that the quantization noise is a white noise in the related sub-band:

$\sigma_{r,k}^2 = \sigma_{q,k}^2\int_{-\pi}^{+\pi}\left|G_k\left(e^{j\omega}\right)\right|^2\frac{d\omega}{2\pi M}$

$\sigma_{g_k}^2 = \int_{-\pi}^{+\pi}\left|G_k\left(e^{j\omega}\right)\right|^2\frac{d\omega}{2\pi M} = \frac{1}{M}\sum_{n=0}^{L-1} g_{k,n}^2$

$\sigma_{r,k}^2 = \frac{1}{M}\sigma_{q,k}^2\sigma_{g_k}^2$

The total quantization noise is expressed by

$$\boxed{\sigma_q^2 = \frac{1}{M}\sum_{k=0}^{M-1}\sigma_{q,k}^2\sigma_{g,k}^2}$$

$$\sigma_q^2 = \frac{\varepsilon_*^2}{M}\sum_{k=0}^{M-1}2^{-2r_k}\sigma_{x,k}^2\sigma_{g,k}^2$$

With the constraint

$$R = M\cdot r = \sum_{k=0}^{M-1} r_k$$

The problem of optimal bit allocation can be defined with a lagrangian:

$J(\lambda) = \sigma_q^2 + \lambda\left(\sum_{k=0}^{M-1} r_k - R\right)$ must be minimized.

$$J(\lambda) = \frac{\varepsilon_*^2}{M}\sum_{k=0}^{M-1}2^{-2r_k}\sigma_{x,k}^2\sigma_{g,k}^2 + \lambda\left(\sum_{k=0}^{M-1} r_k - R\right)$$

Using the derivative:

$$\frac{dJ(\lambda)}{d\lambda} = \left(\sum_{k=0}^{M-1} r_k - R\right) = 0 \rightarrow\ R = \sum_{k=0}^{M-1} r_k$$

$$\frac{dJ(\lambda)}{dr_k} = -2\ln 2\,\frac{\varepsilon_*^2}{M}2^{-2r_k}\sigma_{x,k}^2\sigma_{g,k}^2 + \lambda = 0$$

The number of bits per sub-band is expressed by

$$r_k = -\frac{1}{2}\log_2\left[\frac{\lambda M}{2\ln 2\,\varepsilon_*^2\sigma_{x,k}^2\sigma_{g,k}^2}\right]$$

$$r_k = -\frac{1}{2}\log_2\left[\lambda M\right] + \frac{1}{2}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \frac{1}{2}\log_2\left[2\ln 2\,\varepsilon_*^2\right]$$

We have thus

$$R = \sum_{k=0}^{M-1} r_k = \sum_{k=0}^{M-1} -\frac{1}{2}\log_2\left[\lambda M\right] + \frac{1}{2}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \frac{1}{2}\log_2\left[2\ln 2\,\varepsilon_*^2\right]$$

$$2R = -M\log_2\left[\lambda M\right] + \sum_{k=0}^{M-1}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \sum_{k=0}^{M-1}\log_2\left[2\ln 2\,\varepsilon_*^2\right]$$

$$\log_2\left[\lambda M\right] = \frac{1}{M}\sum_{k=0}^{M-1}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \log_2\left[2\ln 2\,\varepsilon_*^2\right] - \frac{2}{M}R$$

And the number of bits per sub-band becomes:

$$r_k = -\frac{1}{2}\left[\frac{1}{M}\sum_{k=0}^{M-1}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \log_2\left[2\ln 2\,\varepsilon_*^2\right] - \frac{2}{M}R\right] + \frac{1}{2}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \frac{1}{2}\log_2\left[2\ln 2\,\varepsilon_*^2\right]$$

$$r_k = -\frac{1}{2M}\sum_{k=0}^{M-1}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right] + \frac{1}{M}R + \frac{1}{2}\log_2\left[\sigma_{x,k}^2\sigma_{g,k}^2\right]$$

$$r_k = \frac{1}{M} R - \frac{1}{2M} \log_2 \left[\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right] + \frac{1}{2} \log_2 \left[\sigma_{x,k}^2 \sigma_{g,k}^2 \right]$$

$$\boxed{r_k = r + \frac{1}{2} \log_2 \left[\frac{\sigma_{x,k}^2 \sigma_{g,k}^2}{\left(\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right)^{1/M}} \right]}$$

We have then

$$2^{-2r_k} = \frac{\left(\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right)^{1/M}}{\sigma_{x,k}^2 \sigma_{g,k}^2} \cdot 2^{-2r}$$

The distortion after reconstruction is expressed by:

$$\sigma_q^2 = \frac{\varepsilon_*^2}{M} \sum_{k=0}^{M-1} \frac{\left(\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right)^{1/M}}{\sigma_{x,k}^2 \sigma_{g,k}^2} \cdot 2^{-2r} \sigma_{x,k}^2 \sigma_{g,k}^2$$

$$\sigma_q^2 = \varepsilon_*^2 2^{-2r} \left(\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right)^{1/M}$$

The coding gain in case of biorthogonal transform is then given by:

$$\boxed{G_c = \frac{\sigma_x^2}{\left(\prod_{k=0}^{M-1} \sigma_{x,k}^2 \sigma_{g,k}^2 \right)^{1/M}}}$$

Abstract: In recent years there has been a phenomenal increase in the number of products and applications which make use of audio coding formats. Among the most successful audio coding schemes, the MPEG-1 Layer III (mp3), the MPEG-2 Advanced Audio Coding (AAC) or its evolution MPEG-4 High Efficiency-Advanced Audio Coding (HE-AAC) can be cited.

More recently, perceptual audio coding has been adapted to achieve coding at low-delay such to become suitable for conversational applications. Traditionally, the use of filter bank such as the Modified Discrete Cosine Transform (MDCT) is a central component of perceptual audio coding and its adaptation to low delay audio coding has become an important research topic. Low delay transforms have been developed in order to retain the performance of standard audio coding while reducing dramatically the associated algorithmic delay.

This work presents some elements allowing to better accommodate the delay reduction constraint. Among the contributions, a low delay block switching tool which allows the direct transition between long transform and short transform without the insertion of transition window. The same principle has been extended to define new perfect reconstruction conditions for the MDCT with relaxed constraints compared to the original definition. As a consequence, a seamless reconstruction method has been derived to increase the flexibility of transform coding schemes with the possibility to select a transform for a frame independently from its neighbouring frames. Finally, based on this new approach, a new low delay window design procedure has been derived to obtain an analytic definition for a new family of transforms, permitting high quality with a substantial coding delay reduction.

The performance of the proposed transforms has been thoroughly evaluated, an evaluation framework involving an objective measurement of the optimal transform sequence is proposed. It confirms the relevance of the proposed transforms used for audio coding. In addition, the new approaches have been successfully applied to the recent standardisation work items, such as the low delay audio coding developed at MPEG (LD-AAC and ELD-AAC) and they have been evaluated with numerous subjective testing, showing a significant improvement of the quality for transient signals. The new low delay window design has been adopted in G.718, a scalable speech and audio codec standardized in ITU-T and has demonstrated its benefit in terms of delay reduction while maintaining the audio quality of a traditional MDCT.

Keywords: Low delay audio coding – Transform coding – Block switching – MDCT – Seamless reconstruction – Low delay window design