Perceptual Coding of Narrowband Audio Signals

Hossein Najafzadeh-Azghandi

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

April 2000

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This thesis is dedicated

To the memory of my father,

To my mother,

And to those who strive to bring prosperity to ALL human beings.

Abstract

New applications such as Internet broadcast and communications, consumer multimedia products, digital AM broadcast and satellite networks are emerging. Those applications require moderate audio quality without annoying artifacts at bit rates below 16 kbit/s. Although speech coders provide high speech quality at bit rates around 8 kbit/s, they perform poorly when encoding audio signals. In this thesis, we present a novel transform coding paradigm based on the characteristics of the human hearing system. The proposed encoder, i.e., Narrowband Perceptual Audio Coder (NPAC), can accommodate a wide range of narrowband audio inputs without annoying artifacts at bit rates down to 8 kbit/s.

NPAC employs a variety of algorithms to remove the perceptually irrelevant parts and statistical redundancies of the input signal. The new algorithms used in NPAC include a perceptual error measure in training the codebooks and selecting the best codewords, perceptually-based bit allocation algorithms and an adaptive predictive scheme to vector quantize the scale factors.

The proposed encoder has moderate complexity and delivers good quality for narrowband audio inputs at around 1 bit/sample. Informal subjective tests have been conducted to compare the performance of NPAC with an 8 kbit/s commercially-available audio coder. The tests results show that NPAC performs better for both music and speech inputs.

Résumé

Des nouvelles technologies telles que la diffusion par Internet, la diffusion AM numérique, et les réseaux satellites deviennent de plus en plus populaires et constituent la base de plusieurs nouvelles applications et produits multimédias. La réussite de ces produits sur la marché dépend de la qualité des signaux audio et vidéo ainsi que de la largeur de bande utilisée. Pour le signal audio, il est désirable que le débit soit en bas de 16 kbit/s tout en offrant une qualité acceptable, c'est-à-dire sans de distorsion remarquable.

Il est à noter que certains codeurs de parole permettent de transmettre le signal de parole au débit de 8 kbit/s avec une très bonne qualité. Toutefois, puisque ces codeurs profitent de la structure particulière de la parole, ils ne peuvent pas offrir la même qualité audio pour d'autres signaux comme la musique.

Dans cette thèse, nous présentons une philosophie d'encodage des signaux audio qui tient compte de la structure du système auditif. Le codeur proposé se nomme Codeur Audio Perceptuel à bande Étroite (CAPE). CAPE permet d'encoder plusieurs types de signal audio à bande étroite au débit de 8 kbit/s sans de distorsion remarquable.

Plusieurs nouveaux algorithmes sont utilisés dans CAPE afins d'éliminer la redondance statistique ainsi que la partie sans importance perceptuel du signal d'entrée. Parmi les nouveautés de CAPE, il y a une mesure d'erreur perceptuelle qui est utilisée lors de l'entraînement des tableaux de quantification, et pour la sélection du meilleur vecteur de ces tableaux lors de l'encodage. De plus, l'allocation des bits pour les gains du spectre dans différentes bandes de fréquence se fait par un algorithme adaptatif et prédictif qui tient compte de l'importance perceptuel de ces gains.

Notre codeur a une complexité moyenne. Il permet d'encoder les signaux audio à bande étroite avec une très bonne qualité en utilisant seulement 1 bit par échantillon à une fréquence d'échantillonnage de 8 kHz. Nous avons comparé informellement la qualité subjective de CAPE avec un codeur audio commercial opérant au même débit. Les résultats indiquent que la performance de CAPE est supérieure pour la musique et la parole.

Acknowledgments

I would like to express my deepest gratitude to my supervisor Prof. Peter Kabal for his support and guidance during the course of my Ph.D. study at McGill University. Prof. Kabal's kind treatment of his students is highly appreciated.

The financial support provided by Prof. Kabal from the Canadian Institute for Telecommunications Research, and from other funds is gratefully acknowledged.

My thanks go to my fellow graduate students of the TSP lab for their genuine friendship. Moreover, I thank my dear friends Syavosh Zadissa and Michael Godbout for the French translation of the thesis abstract.

I am deeply indebted to my parents for all the sacrifices they have made to bring me up to the point where I am today. I wish I could be as caring to my fellow human beings as they have been.

Contents

List of Figures

List of Tables

[1]Jan. 2001: Corrections to Table 2.1

List of Acronyms

ASPEC	Adaptive Spectral Perceptual Entropy Coding
AAC	Advanced Audio Coding
ADPCM	Adaptive Differential Pulse Code Modulation
AM	Amplitude Modulation
CELP	Code-Excited Linear Prediction
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DPCM	Differential Pulse Code Modulation
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GLA	Generalized Lloyd Algorithm
IDFT	Inverse Discrete Fourier Transform
KBD	Kaiser-Bessel-Derived
KLT	Karhunen-Loève Transform
LOT	Lapped Orthogonal Transform
LSF	Line Spectral Frequency
MDCT	Modified Discrete Cosine Transform
MLT	Modulated Lapped Transform
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MSB	Most Significant Bit
NMR	Noise-to-Mask Ratio
NPAC	Narrowband Perceptual Audio Coder
PAC	Perceptual Audio Coder
PCM	Pulse Code Modulation
SFM	Spectral Flatness Measure
SMR	Signal-to-Mask Ratio
SNR	Signal-to-Noise Ratio
TNS	Temporal Noise Shaping
VQ	Vector Quantization

Chapter 1

Introduction

Audio compression (coding) is concerned with the efficient transmission or storage of audio data with good perceptual quality. Audio files require a lot of bandwidth (or memory) for transmission (or storage). For instance, an audio signal sampled at 8 kHz and using 16 bits for each sample produces a data rate of 128 kbit/s. However, we will show that it is possible to reduce the data rate to less than 10 kbit/s while maintaining acceptable quality of the compressed signal.

Audio coding algorithms has been employed in many applications including digital broadcasting, personal communication systems, Internet and multimedia communication systems [1, 2]. The increasing traffic in wireline and wireless networks calls for high compression efficiency in order to better utilize the capacity of existing resources.

Users of communication systems require high quality reproduction of all signals that can be presented to a common carrier. Therefore, there is a need for bandwidth efficient coding of variety of sounds including speech, music and multiple simultaneous speakers. Such signals need to be efficiently represented (good quality at low rates) for transmission over wireless (e.g., cell phones) or wireline (e.g., telephony or Internet) networks. Traditional speech coders designed specifically for speech signals, achieve compression by utilizing models of speech production based on the human vocal tract. However, these traditional coders are not effective when the signal to be coded is not human speech but some other signals such as music. These other signals do not have the same typical characteristics as human speech. As well, production of sound from these other signal sources can not be modelled on mathematical models of the human vocal tract. As a result, traditional speech coders

often have uneven results for non-speech signals. For example, for many traditional coders music-on-hold is coded with annoying artifacts.

In this thesis in order to accommodate a wide variety of audio data, we take into consideration the characteristics of the final receiver, i.e., the human hearing system. This means that any part of the audio signal which is not sensed by the auditory system is considered *perceptually irrelevant* and should be discarded. When the perceptually irrelevant information is removed, the audio encoder can operate at much lower bit rates and still provide good sound quality. An audio coding scheme which minimizes the perceptible distortion is called a *perceptual coding* algorithm. Perceptual audio coders use models of the *auditory masking* phenomena (will be discussed in Chapter 2) to identify the inaudible (perceptually irrelevant) parts of audio signals.

Upper Bound for Operating Bit Rates of Perceptual Audio Coders

According to Shannon's rate-distortion theory, a source signal which is transmitted at a bit rate below its entropy must have some distortion (the distortion level depends on the bit rate) [3]. However, in perceptual coding we have to consider only the audible part of the distortion. Johnston [4] has proposed a new concept called *perceptual entropy* as the minimum bit rate for transmission of audio signals such that no perceptible difference between the original and coded signal is perceived, i.e., *transparent coding*. Based on the perceptual entropy criterion, it is possible to transmit audio signals without any perceptible distortion at a rate much lower than that predicted by Shannon's theory based on the MSE criterion. One of the important implications of this new concept is that a signal can be transparently coded without a high Signal-to-Noise Ratio (SNR). Therefore in audio coding, an SNR is not a good measure to judge the quality of the reconstructed signal.

1.1 Motivation for Low Rate Coding of Narrowband Audio

Although a lot of research have been done on high quality coding of wideband audio signals over the past decade, new applications such as Internet broadcasting, consumer multimedia products, narrowband digital AM broadcasting and satellite networks are emerging. For those applications moderate audio quality without annoying artifacts at low bit rates below 16 kbit/s is adequate [5, 6, 7].

For some applications either the number of users is huge (e.g., Internet) or the available bandwidth is limited (e.g., satellite and radio communications, slow modems). For instance, for Internet broadcast, all users should be accommodated including those with a rather slow connection, i.e., modems with rates of 14.4 and 28.8 kbit/s.

As another application, the WorldStar[1] satellite system provides a data rate of 8 kbit/s per channel [8]. Since the cost per transmission channel is quite high, it is desirable to compress the audio signal to 8 or 16 kbit/s (for monaural signals) [5].

One important recent application is "Narrowband Digital Broadcasting" (NADIB) which is a project sponsored by the European Union for digital audio broadcasting in AM frequency bands [9]. A verification testing for an AM digital audio broadcasting application has shown that higher quality compared to that of analog AM techniques can be achieved in the same bandwidth with digital techniques [10].

In 1992, the International Multimedia Association (IMA) made a recommendation for coding of narrowband audio signals sampled at 8 kHz to be used in computers and multimedia systems. According to that recommendation, 8-bit PCM (i.e., 64 kbit/s) or 4-bit ADPCM (i.e., 32 kbit/s) algorithms are specified for compression of 8 kHz audio data [11]. The specified bit rates are very high for many applications. Currently available general purpose audio coders operate at bit rates above 16 kbit/s (e.g., ITU G.726 audio standard). On the other hand speech coders operating at bit rates lower than 16 kbit/s are not suitable for encoding audio signals. This implies a gap between the operating bit rates of state-of-the-art narrowband speech coders (8 kbit/s and below) and low bit rate audio coders operating at around 16 kbit/s. We believe that a proper coding paradigm using different coding tools based on the characteristics of the human hearing system can fill the gap and accommodate a wide range of narrowband audio inputs without annoying artifacts at low rates down to 8 kbit/s.

1.2 Objective of Our Research

In this thesis we have concentrated on perceptual coding of narrowband audio data. The input audio signal is band-limited from 50 Hz to 3.6 kHz, sampled at 8 kHz, and represented with 16 bit linear PCM.

The goal of this thesis is to develop a coding structure which allows the compression of

[1] WorldStar is a trademark of WorldSpace, Inc.

narrowband audio signals at low bit rates down to 8 kbit/s while producing reconstructed signals without annoying artifacts. Note that in most commercial narrowband audio coders such as RealAudio[2] and MPEG-4, the user should specify whether the source is music or speech, and then a different encoder is used for each type.

We propose a new transform audio coding structure based on the characteristics of the human hearing system. New algorithms are introduced in different blocks of the coder including a perceptually-based error measure, perceptually-trained VQ, adaptive perceptually-based bit allocation algorithms, adaptive predictive VQ of the scale factors, prototype window design for the MDCT, a window switching mechanism taking into account the asymmetrical characteristics of temporal masking effects.

Our coding system has moderate complexity and a software implementation of the coder written in the C programming language runs in real time on a computer using a 450 MHz Pentium II processor. The algorithmic delay[3] of this coder is 30 msec, which is reasonable for most applications. We have focused on the compression of the audio signals; the robustness of the resulting bit stream against channel effects has not been investigated. However, since the operating bit rate of the coder is around 8 kbit/s, channel coding can be added and still the total bit rate would be quite low.

Although the proposed coder belongs to the family of perceptual audio coders, we have to point out certain distinctions between this coder and high rate wideband audio coders. While the goal of high rate audio coding is to achieve transparent or near transparent quality of wideband audio with a 7–20 kHz bandwidth [12, 13, 14, 15, 16], our goal is to achieve moderate audio quality, i.e., without annoying artifacts. State-of-the-art high rate audio coders spend around 1.5 bits per sample to reproduce high quality audio. Additionally, since important spectral features of natural audio signals are located between 300–5000 Hz [17], in high rate audio coders most bits are spent on that frequency band. In fact, in high rate audio coding, the average number of bits per sample for low frequencies (0.3–5 kHz) is considerably more than 1.5 bits per sample. In our coder, we spend 1 bit per sample for the frequency band 50–3600 Hz. Here we make a trade-off between the bit rate and the quality of the reconstructed signal and hence expect moderate audio quality.

[2]RealAudio is a trademark of RealNetworks, Inc.

[3]Algorithmic delay is the length of a block of data plus the lookahead.

1.3 Audio Coding Techniques

Audio coders can be roughly grouped into three classes: *parametric* coders (also known as *source* coders), *hybrid* coders and *waveform* coders. Parametric coders estimate and transmit the parameters characterizing a particular sound source. Those parameters are used to reconstruct a signal which sounds similar to the original sound. The waveform of the reconstructed signal is not necessarily similar to that of the original. Parametric coders usually operate at very low bit rates at the expense of the quality and naturalness of the reconstructed signal. On the contrary, waveform coders operate at higher bit rates and try to match the waveform of the compressed signal to that of the original. Hybrid coders use techniques from both parametric and waveform coding and provide better quality at higher data rates (compared to parametric coders). In the following, we briefly describe different classes of audio coding.

1.3.1 Parametric Coders

Parametric coders or source coders model the signal source with a few parameters. For speech, there is a good source model based on the mechanism of speech production. In the model, the vocal tract is modeled as a time-varying filter which is excited with either a white noise source (for unvoiced speech) or a sequence of impulses separated by the pitch period (for voiced speech). Parametric speech coders operate at around 2 kbit/s or below and deliver synthetic quality.

For general audio signals, a new and very promising trend called *object-based audio* coding or *structured audio* coding is emerging. That is a part of the MPEG-4 audio standard which is used to encode audio data at bit rates of 0.1 to 10 kbit/s [10, 18]. In an object-based audio encoder, the input signal is first decomposed into audio objects which can be described by appropriate source models and represented by corresponding sets of model parameters. For each object the model parameters are estimated, coded, and transmitted. In the decoder for each audio object, a signal is synthesized from the decoded model parameters. The decoder outputs the sum of all object signals.

1.3.2 Hybrid Coders

Hybrid codecs attempt to fill the gap between waveform and parametric coders. Waveform coders provide good quality for narrowband audio at bit rates around 16 kbit/s; on the other hand, parametric coders operate at very low bit rates but cannot provide natural quality.

Hybrid coders (including Analysis-by-Synthesis coders), similar to parametric coders, extract and transmit the parameters of the audio signal. Moreover, a compressed error signal which is the difference between the reconstructed signal from the extracted parameters and the original signal is also transmitted. This way the reconstructed signal waveform becomes close to the original waveform.

In the field of speech coding, the most successful hybrid scheme has been the Code-Excited Linear Predictive (CELP) paradigm. Many variations of CELP coders have been standardized including [1, 19] G.723.1 operating at 6.3/5.3 kbit/s, G.729 operating at 8 kbit/s, G.728 a low delay coder operating at 16 kbit/s and all the digital mobile telephony encoding standards including [20, 21, 22, 23] GSM, IS-54, IS-95 and IS-136. Figure 1.1 shows a simple block diagram of a CELP coder. It is based on the modelling of speech production; two synthesis filters are used to introduce short and long term correlation among the speech samples. The parameters of the filters are determined through minimizing a perceptually weighted difference between the original and reconstructed signal. Although CELP coders give high quality speech at bit rates below 8 kbit/s, due to differences between general audio signals and speech, they perform poorly for non-speech signals.

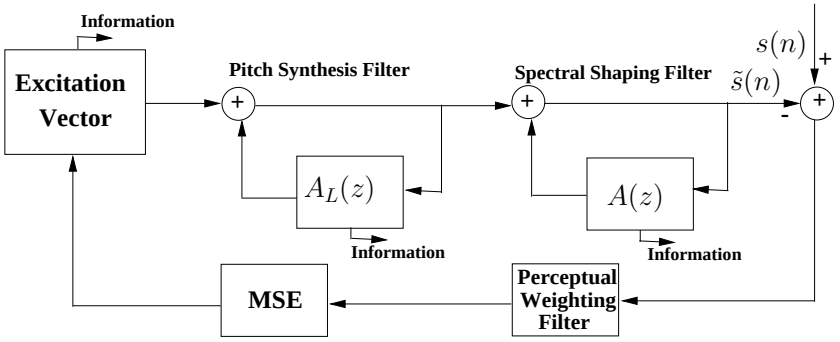


Fig. 1.1 Basic blocks of a CELP coder, adapted from [19].

For general audio signals, object-based analysis-by-synthesis schemes have been recently proposed [24, 25]. Hybrid schemes function similar to parametric coders with the difference

that a compressed residual signal is also transmitted to the receiver in order to enhance the quality of the reconstructed audio signal.

1.3.3 Waveform Coders

Waveform coders try to produce a reconstructed signal whose waveform is as close as possible to the original. Since there are no appropriate source models for general audio signals, waveform coders have been the best choice to encode audio signals. They deliver high quality for different inputs such as music [26] albeit often at the expense of high bit rates. From a signal representation perspective, waveform coding schemes can be divided into different classes: time domain and frequency domain algorithms.

Time Domain Coders

Time domain coders perform the coding process on the time samples of the audio data. The well known coding methods in the time domain are [1] Pulse Code Modulation (PCM), Adaptive Pulse Code Modulation (APCM), Differential Pulse Code Modulation (DPCM), Adaptive Differential Pulse Code Modulation (ADPCM), Delta Modulation (DM), Adaptive Delta Modulation (ADM) and Adaptive Predictive Coding (APC). In the following, we briefly describe some important coding schemes in the time domain.

Pulse Code Modulation

Pulse Code Modulation (PCM) is a widely used form of waveform coding. For audio, a linear PCM scheme typically spends 16 bits to quantize each time sample. There are also two slightly different nonuniform PCM algorithms (ITU G.711 standard), i.e., μ-law (American standard) and A-law (European standard), which logarithmically quantize audio samples with 8 bits per sample. Note that the logarithmic quantizer has been designed to provide a uniform SNR for different talker levels. Although PCM provides high quality audio, the required bit rate is quite high.

DPCM and ADPCM Coders

In Differential Pulse Code Modulation (DPCM), instead of the time samples, the difference between the original and predicted signal is quantized. At the decoder the quantized

difference signal is added to the predicted signal to give the reconstructed signal. This scheme is based on the assumption that audio samples are correlated enough such that the error signal, defined as the difference between the audio samples values and the predicted values, has a lower variance than the original audio signal. Consequently, we expect to quantize the error signal with fewer bits than the original signal.

An enhanced version of DPCM is Adaptive Differential Pulse Code Modulation (ADPCM) in which the predictor and quantizer are adapted to local characteristics of the input signal. There are a number of ITU standards based on ADPCM algorithms for narrowband speech and audio coding: G.721 operating at 32 kbit/s, G.723 operating at 40 and 24 kbit/s, G.726 and G.727 operating at 40, 32, 24 and 16 kbit/s. The complexity of ADPCM coders is fairly low.

Frequency Domain Coders

Frequency domain coders carry out the compression on a frequency representation of the input signal. Compared to time domain coders, frequency domain coders usually provide better quality at the expense of higher complexity [26]. Other advantages of frequency domain coders include the ability to encode different parts of the frequency spectrum independently and using adaptive bit allocation schemes to shape the quantization noise based on perceptual principles.

Since the reproduced signal will be perceived by the hearing system, in order to reduce the data rate, the auditory masking effects can be incorporated into the coding structures. Therefore, frequency domain waveform coders are the proper choice for perceptual audio compression since the auditory masking properties are well modeled in the frequency domain.

Frequency domain coders are divided into two groups: *subband coders* and *transform coders*. Subband coders employ a few bandpass filters (i.e., filterbank) to split the input signal into a number of bandpass signals (subbands signals) which are coded independently. At the receiver the subband signals are decoded and summed up to reconstruct the output signal. The ITU has a standard on subband coding (i.e., G.722 audio coder [27]) which encodes wideband audio signals (7 kHz bandwidth sampled at 16 kHz) for transmission at 48, 56, or 64 kbit/s.

In transform coding, a fast transformation is used to convert blocks of the input signal

into a large number of frequency coefficients. Transform coding is the proper paradigm for perceptual audio coding due to the following reasons [28]:

- Good transforms compact the signal energy into a few transform coefficients which allows many transform coefficients to be set to zero without affecting the quality.
- Suitable transforms produce decorrelated coefficients allowing for efficient quantization of the transform coefficients.
- Appropriate transforms can provide good frequency resolution which is required to model the auditory masking effects.
- Using perceptually-based distortion measures is possible.
- Fast transform techniques are available.

Pioneer work on transform coding was done in the late 1970s by Zelinsky and Noll [29, 30], Tribolet and Crochiere [31]. Although that work was mainly intended for coding of speech in the frequency domain, it is the basis of almost all state-of-the-art audio coders. Concerning the perceptual aspect of audio coders, the work published by Schroeder *et al* [32] described the use of the auditory masking effects in the coding paradigms. That work has been the starting point for a large amount of work on perceptual coding of speech and audio signals.

1.3.4 Perceptual Audio Coding

Fig. 1.2 shows a general block diagram of perceptual audio coders working in the frequency domain. Perceptual audio coders employ a transform to decompose the input signal into spectral components. The auditory masking threshold is calculated using the signal spectrum. The transform coefficients are quantized and coded using the masking threshold. In the last step, the encoded transform coefficients are multiplexed with the side information to produce a bit stream. In the next chapters we will describe different blocks in a perceptual audio coder.

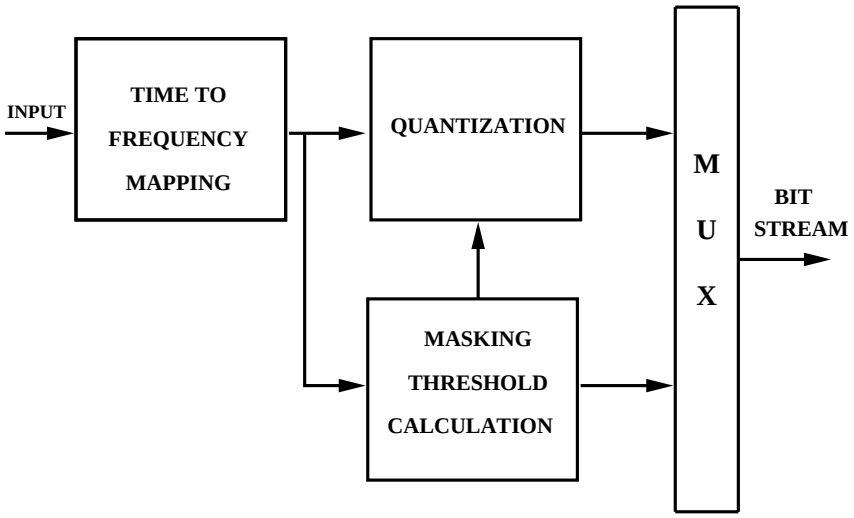


Fig. 1.2 General block diagram of a perceptual coder working in the frequency domain.

1.4 Thesis Contributions

In this work we take on the challenge of designing a universal coding structure to accommodate narrowband audio inputs at bit rates comparable to existing speech coders. To accomplish this goal, we have developed a new audio coding structure based on the characteristics of the human hearing system. The proposed coder, which is referred to as the *Narrowband Perceptual Audio Coder (NPAC)*, provides moderate quality for narrowband (4 kHz bandwidth) audio inputs at bit rates down to 8 kbit/s [33, 34, 35]. The proposed coder employs a number of different coding techniques which are described in this thesis. The emphasis has been on using the human auditory mechanism, especially the masking effects in different parts of the coder to reduce the bit rate while delivering acceptable quality.

To accomplish our goal, the proposed NPAC employs a variety of perceptual-based algorithms to remove the perceptually irrelevant parts of the input signal in addition to statistical redundancies.

The original features of the proposed coder are divided into three categories as follows.

Time-to-Frequency Transformation

- The transform coefficients are non-uniformly divided into 17 subbands in accordance with the Bark scale to correspond to the frequency analysis that occurs in the ear.
- A prototype window for the MDCT has been designed. The frequency response of the ear has been considered in the design procedure.
- A window switching method has been employed to reduce the spread of the quantization noise caused by large attacks. The non-symmetrical characteristics of the temporal masking effects has been considered in the switching criterion.
- A new procedure for designing prototype windows for Modulated Discrete Cosine Transforms (MDCT) has been derived from the Chebyshev polynomial. A number of windows similar to the KBD window (used in the MPEG audio standard) have been designed using the proposed procedure. This procedure provides two tuning parameters which allow to control the temporal and frequency characteristics of the resulting windows while KBD windows have only one parameter.

Masking Models

- An algorithm has been developed to map the masking thresholds in the DFT domain into the masking thresholds corresponding to the Modulated Discrete Cosine Transform (MDCT) coefficients.
- A model for the temporal masking effects has been developed and incorporated into the NPAC coder.

Quantization Algorithms

- A perceptual distortion measure has been introduced to take into account only the audible part of the quantization noise.
- A perceptually-based vector quantization method, which employs the proposed perceptual distortion measure in populating the codebooks, is utilized. The same distortion measure is used to select the best codewords from the codebooks in the process of coding.

- A refinement to the new distortion measure has been made to shape the quantization noise inside the critical bands.
- In order to reduce the required memory to store the codebooks, a number of methods have been used to design a single embedded codebook for each critical band.
- Perceptually-based bit allocation algorithms have been proposed and investigated. One of them is based on the Signal-to-Mask Ratio (SMR) while the others are based on the distribution of the audible energy.
- A new adaptive VQ system has been employed to quantize the scale factors. In order to reduce the complexity, predictor matrices with a few non-zero diagonals have been designed.
- A variable rate coding scheme is suggested based on the SMR-based bit allocation algorithm.

1.5 Outline of The Thesis

This thesis is organized into 6 chapters. Chapter 2 is concerned with the human auditory masking. Starting with a brief overview of the human hearing system, Chapter 2 discusses the processing of sounds in the ear with an emphasis on the nonuniform frequency analysis of the input stimuli by the basilar membrane. The critical band concept is investigated followed by the discussion about the auditory masking phenomena and a number of related models.

In Chapter 3, we discuss lapped transforms and their importance to audio coding. A thorough analysis of lapped transforms is given and the conditions for perfect reconstruction of the output signal are obtained in a matrix form. The Modulated Lapped Transform (MLT) which is a special case of lapped transforms is analyzed. This is followed by a comparison of orthogonal and biorthogonal lapped transforms. The role of the prototype window in the MLT performance is investigated and an optimization procedure for designing a desirable prototype window is presented. Finally window switching is described as a method of reducing pre-echo artifacts.

Chapter 4 begins with a short overview of quantization methods used in audio coders. Then a number of widely used state-of-the-art audio coders are briefly described followed

by an overview of the MPEG audio standards.

In Chapter 5, we introduce the proposed Narrowband Perceptual Audio Coder (NPAC). The functionality and algorithms for each module is described. A comparison is made between the performance of NPAC and two coders, i.e., RealAudio and the G.729 coders.

Chapter 6 concludes the thesis by providing a summary of our work followed by some remarks about the proposed coder. Finally, we make a number of suggestions for further work in the field of narrowband perceptual audio coding.

Chapter 2

Human Auditory Masking

The hearing system converts sound waves into mechanical energy and finally into electrical impulses perceived by the brain. It consists of the ear, auditory nerve fibers and a part of the brain. Figure 2.1 shows a simplified structure of the peripheral part of the human hearing system, i.e., the ear.

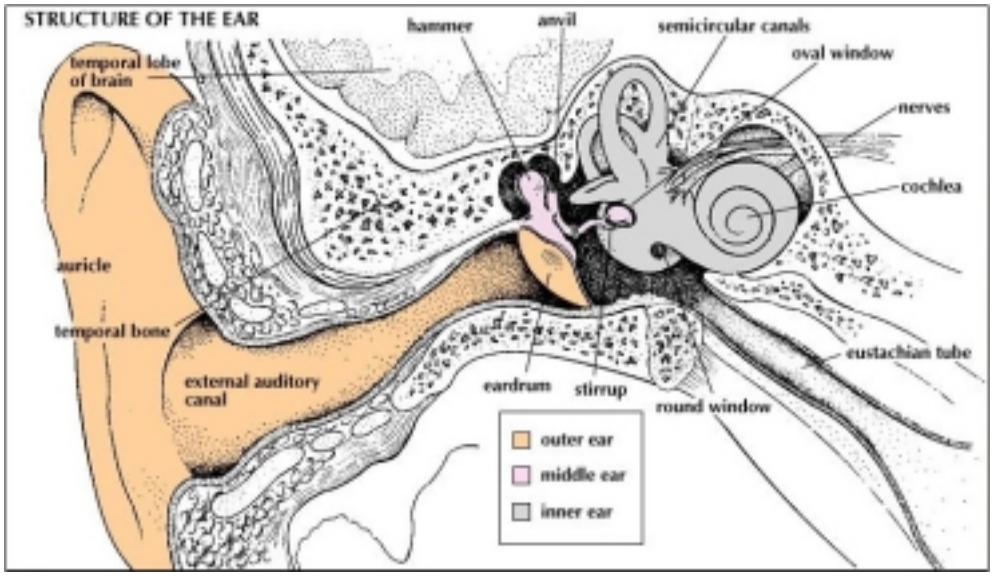


Fig. 2.1 Simplified structure of the ear, from [36].

The ear contains three parts, i.e., the outer ear, the middle ear and the inner ear. The structure and the role of each part in perceiving sounds are discussed as follows.

2.1 Outer Ear

The outer part of the ear consists of the pinna (auricle), the ear canal (external auditory meatus) and the eardrum (tympanic membrane) [37]. The pinna collects sounds, i.e., air pressure waves in the air which are amplified and conveyed by the ear canal to the eardrum. The ear canal is like a tube which is sealed in one end. It encloses a column of air which resonates at around 3 kHz, which enhances the intelligibility of speech [2]. The resonance of the air inside the ear canal increases the sound pressure level by a factor of 10 at the eardrum for a range of frequencies from 2 kHz to 5.5 kHz [38]. The sound pressure makes the eardrum vibrate. This way the sound energy is converted into the mechanical energy.

2.2 Middle ear

The middle ear is an air-filled space containing the three smallest bones in the human body, i.e., the ossicles, including the hammer (malleus), anvil (incus) and stirrup (stapes). As it is shown in Fig. 2.2 these bones form a system of levers which vibrate along with the eardrum. This vibration amplifies the sound and carries it to the inner ear via the oval window.

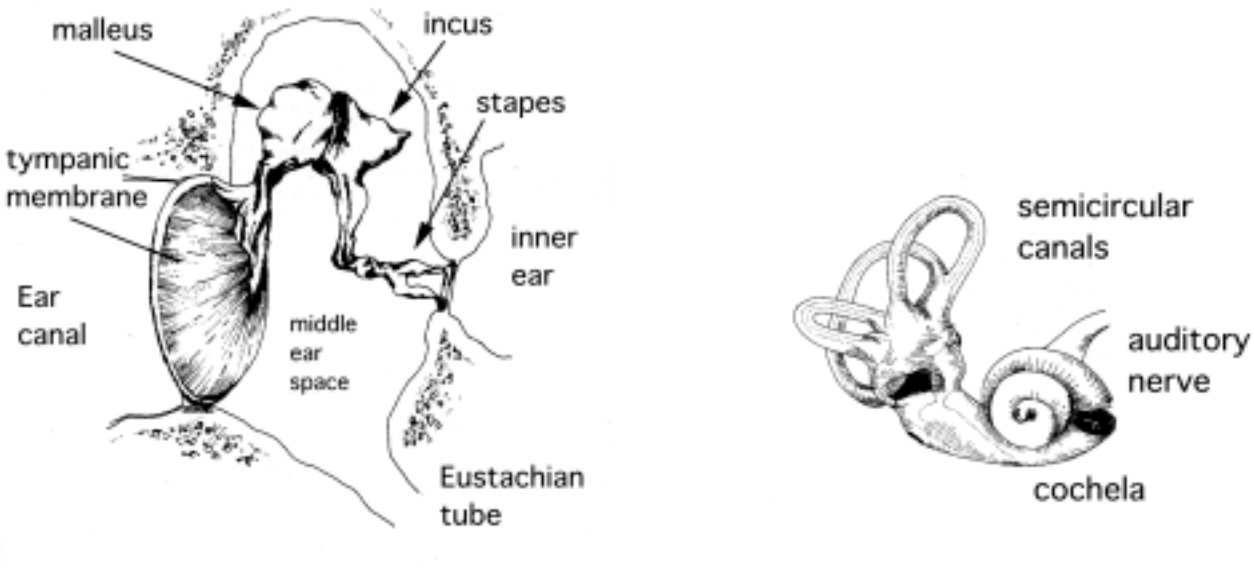
There are some tiny muscles in the middle ear which protect the ear against very large vibrations caused by loud sounds. When the sound level exceeds a certain level, these tiny muscles function in two ways to protect the inner ear. One set of the muscles contracts to limit the movement of the hammer which attenuates the vibrations passing through the middle ear. Some other muscles contract to keep the stirrup away from the oval window in order to weaken the vibration passed to the inner ear [40].

In addition to the aforementioned functions the middle ear filters out low frequency sounds in noisy environments and decreases one's sensitivity to his own speech [40].

Another part of the middle ear system is the eustachian tube which equalizes the air pressure in the middle ear.

2.3 Inner Ear

The inner ear has a great role in both hearing and the body balance. The hearing organ is a bony cone-shaped spiral called cochlea which is filled with fluids. Figure 2.2 shows the



(a) Middle ear

(b) Inner ear

Fig. 2.2 Structure of the middle and inner parts of the ear, adapted from [39].

shape of the inner ear.

2.3.1 Cochlea

The Cochlea is the part of the inner ear which converts incoming vibrations from the middle ear into the electrical impulses. Although the modelling of the cochlea functions has been an active research area for many years, there are still ambiguities in its mechanism such as the frequency selectivity of the auditory system and the nonlinear behavior of the cochlea.

The cochlea, which is smaller than the tip of a little finger, is divided along its length by two membranes; Reissner's membrane (vestibular membrane) and the basilar membrane. It contains many parts including the basilar membrane and the organ of Corti which play important roles in hearing. Figure 2.3 shows the cochlea cross section.

The vibration in the middle ear is passed to the inner ear by the stirrup which moves in and out of the inner ear through the oval window. The oval window is 15 to 30 times smaller than the eardrum which amplifies the pressure inside the cochlea [42]. The pressure

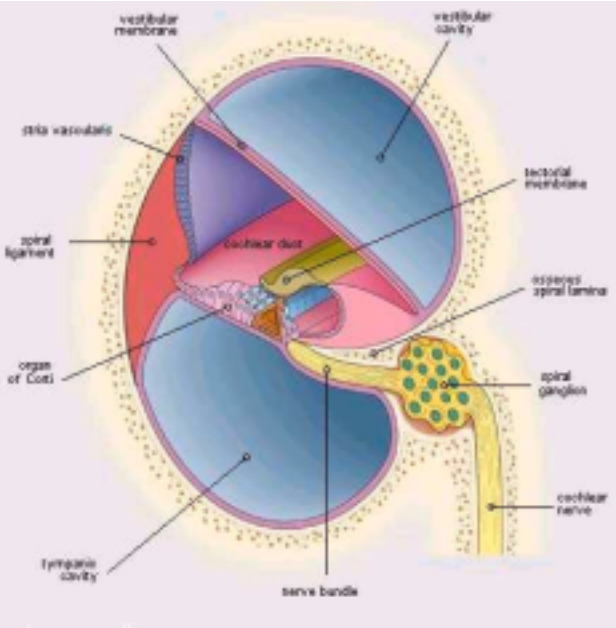


Fig. 2.3 Cochlea cross section [41].

change makes the basilar membrane move up and down which is sensed by a collection of cells called the *organ of Corti*. In addition to the signal detection and energy conversion, the cochlea is able to compress the dynamic range of input signals. The dynamic range of the hair cells is about 40–60 dB, whereas the range of audible sound pressure levels is about 100 dB [37].

Since the basilar membrane and the organ of Corti play a great role in perception of acoustic events, we describe their functions in more details.

2.3.2 Basilar Membrane

The basilar membrane extends along the length of the cochlea. It is narrow and stiff at the end near the middle ear and wider and less stiff at the other end. Its physical properties strongly affect the response of the basilar membrane to different stimuli.

The basilar membrane reacts to the pressure change in the fluids inside the cochlea. The pressure change in the cochlea fluids is mainly due to the stirrup movements and also vibrations reaching the skull from other sources. The response to a single frequency input takes the form of a wave which travels the length of the membrane. The wave stops at a specific region of the basilar membrane (corresponding to different frequencies) along the

length of the membrane where the greatest vibration of the membrane occurs. In fact, each point on the basilar membrane is tuned to a specific frequency, with a spatial gradient of about 0.2 octave/mm [43]. Due to physical characteristics of the basilar membrane, for high frequencies the maximum amplitude of the travelling wave occurs near the base of the basilar membrane but for low frequencies the wave travels further along the length of the basilar membrane. Hence each region along the basilar membrane has the greatest response to a distinct frequency component of the sound spectrum.

In fact the basilar membrane performs a frequency to place transformation. This way, the basilar membrane behaves like a spectrum analyzer. The amplitude and the location of the vibration peak is sensed by the sensory hair cells (which will be discussed later). The location of the vibration peak is important because it determines which nerve fibres will send signals to the brain. Since the auditory nerve fibers are very finely tuned, the brain can identify the frequency of the input signal. The important point is that weak local activities on the basilar membrane are ignored by the brain and hence are not perceived, i.e., will be masked [2].

For a steady sinusoidal input, each point on the basilar membrane vibrates at the same frequency but with different amplitudes and phases [40]. When the basilar membrane is stimulated with two steady inputs with different frequencies, depending on how close the frequencies are, the basilar membrane shows different behavior. If the input frequencies are far apart, then there will be two distinct maximum peaks of displacement on the basilar membrane. In the case that the input frequencies are close enough, then there will be only one broad maximum on the pattern of vibration and the tones cannot be resolved by the basilar membrane. The frequency resolution of the basilar membrane is higher at low frequencies compared to high frequencies. This fact can be explained by considering the physical properties of the basilar membrane. For frequencies above 500 Hz, the position on the basilar membrane which is excited the most by a given frequency varies approximately with the logarithm of the frequency [37]. Also for that range of frequency, the bandwidth of the vibration for a steady stimuli is approximately proportional to the center frequency. These two characteristics of the basilar membrane explain the frequency resolution of the hearing system.

2.3.3 Organ of Corti

The organ of Corti converts the mechanical movements of the basilar membrane into electrical impulses. These pulses which are carried by the auditory nerve fibers to the brain contain information about the frequency, the intensity and the timbre[1] [40].

It contains the sensory hair cells which are arranged in multiple rows and rest on the basilar membrane. Auditory nerve fibers are connected to the base of the hair cells. Figure 2.4 shows a cross section of the organ of Corti.

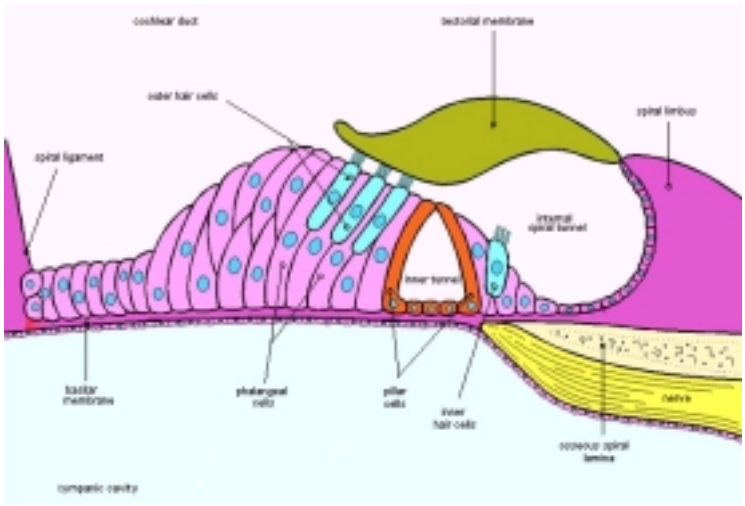


Fig. 2.4 The organ of Corti cross section [41].

There are two types of hair cells in the organ of Corti. The inner hair cells, which are completely surrounded by the inner phalangeal cells and arranged as a single row along the basilar membrane, deliver electrical impulses to the brain [40]. The outer hair cells, which are arranged in 3 to 5 parallel rows, receive neural signals from the brain [40]. The hairs at the top of the outer hair cells make contact with the tectorial membrane when the basilar membrane moves up and down. When the basilar membrane moves, it excites the inner hair cells, which leads to the generation of electrical impulses in the neurons of the auditory nerve.

Hair cells vibrate at the frequency of the strongest stimulation in a local region; therefore they ignore weaker stimulations [2]. This property of the hearing system is the physiological

[1] Timbre is the attribute of a sound that allows us to differentiate between two sounds of the same pitch, intensity and duration [40].

basis of the simultaneous making phenomenon which will be discussed later.

2.4 Critical Bands

Auditory perception is based on a *critical band* analysis in the inner ear. A critical band is the bandwidth around a center frequency beyond which subjective responses of the hearing system abruptly change [40]. The importance of the critical bands comes from the fact that the hearing system discriminates between energy in and out of a critical band. Additionally, the simultaneous masking property of the hearing system, which will be discussed later, is related to the critical bands.

The physiological basis of the critical bands is not clear. However, the existence of the critical bands is certainly related to the function of the basilar membrane [40]. Each point on the basilar membrane is tuned to a frequency called the *characteristic frequency*; that is the place at which the travelling wave caused by a stimulus reaches its maximum amplitude [40]. We can assume a non-ideal bandpass filter, which is referred to as an *auditory filter*, centered at each characteristic frequency [44]. The effective bandwidth of the bandpass auditory filter is defined as the critical bandwidth. Each critical band corresponds a length of 1.3 mm (according to [2]) or 1.5 mm (according to [32]) on the basilar membrane. Moore [40] defines a critical band as the Effective Rectangular Band (ERB) which is the bandwidth of an ideal bandpass filter centered at any frequency (the area under the squared-magnitude of the ideal filter equals that of the auditory filter centered at that frequency). According to Moore each ERB covers 0.9 mm on the basilar membrane. Note that there is no border between the critical bands and a band can be specified for any point on the basilar membrane.

The bandwidth of the critical bands was first measured by Fletcher in the 1940's. According to his experiment, in order to measure the bandwidth of a critical band centered at any frequency, we make a tonal signal inaudible by a narrowband noise centered at that frequency. If we increase the bandwidth of the noise, the level of the inaudible sinusoid can be increased. When the bandwidth of the noise increases above a certain value, i.e., the critical bandwidth, the level of the sinusoid input remains almost constant. Figure 2.5 shows the threshold level as a function of the noise bandwidth. This experiment is based on a few assumptions [40]: the hearing system contains a bank of overlapping bandpass linear filters, the listener is assumed to perceive only the output signal of the auditory filter

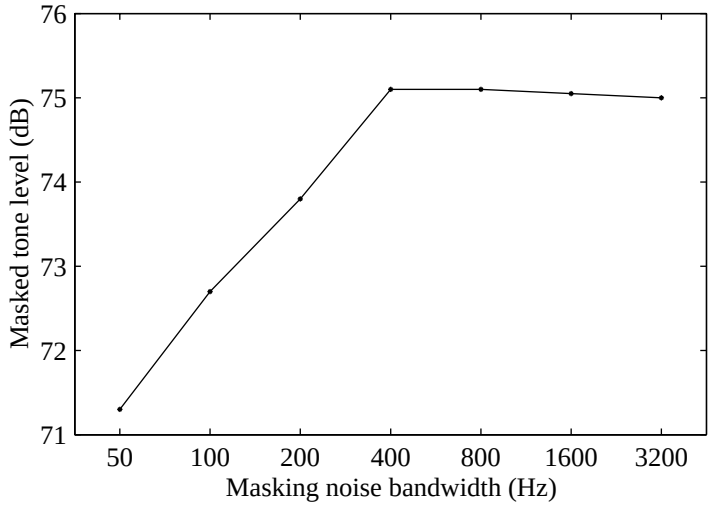


Fig. 2.5 Threshold of a just audible 2 kHz test tone [44].

centered at the tonal signal frequency and the threshold of the signal is only determined by the power of the noise at the output of the auditory filter. The power of the noise at the output of the bandpass filter determines the threshold of the test tone. Since the bandpass filter is tuned to the frequency of the sinusoid, the tonal signal will be passed but a great part of the noise will be removed (when the noise bandwidth is greater than the critical bandwidth). The part of the noise which passes through the filter has a remarkable effect on making the tonal signal inaudible. In this experiment, increases in noise bandwidth result in higher noise power at the output of the bandpass filter as long as the noise bandwidth is less than the filter bandwidth. However, when the noise bandwidth exceeds the bandpass filter bandwidth, further increase in noise bandwidth will have a little effect on the output noise power. This bandwidth, at which the signal threshold stops increasing, is the critical bandwidth.

Experiments have shown that the width of the critical bands is narrower at low frequencies. In fact, the signal is processed in the inner ear on a nonlinear scale called the *Bark scale* (Bark is the unit of perceptual frequency and a critical band has a width of one Bark). Therefore, as shown in Fig. 2.6, the peripheral section of the hearing system can be modeled as a nonuniform filterbank consisting of bandpass filters with bandwidths equal to critical bandwidths. About 75% of the critical bands are below 5 kHz and hence the hearing system receives more information from low frequencies. Approximately, the critical

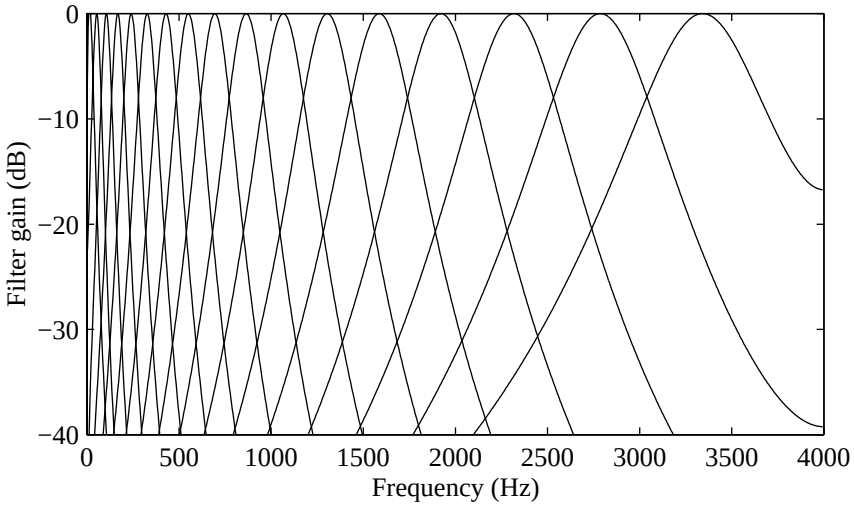


Fig. 2.6 Example of the nonuniform auditory filterbank.

bandwidth is 100 Hz up to 500 Hz. Above 500 Hz, the critical bandwidth is approximately 20% of the center frequency [17]. There is a relation between the distance along the basilar membrane and its frequency resolution. Considering the fact that a length of 1.5 mm on the basilar membrane represents approximately 1 Bark, near the top (far from the middle ear) a length of 0.1 mm represents a frequency difference of about 7 Hz whereas near the base 0.1 mm represents 450 Hz.

Many analytical expressions have been proposed in the literature to relate the critical band number z (in Bark) to frequency f (in Hz). Schroeder *et al* in [32] propose the following formula

$$f = 650 \sinh(z/7). \tag{2.1}$$

Zwicker proposes the following [37]

$$z = 13 \arctan(0.00076 f) + 3.5 \arctan(f/7500)^2. \tag{2.2}$$

The bandwidth of each critical band as a function of center frequency can be approximated by [37]

$$\text{Critical Bandwidth} = 25 + 75(1 + 1.4(f/1000)^2)^{0.69}. \tag{2.3}$$

Glasberg and Moore [40] proposed the following relation

$$\text{number of ERBs} = 21.4 \log_{10}(0.00437f + 1). \tag{2.4}$$

The ERB as a function of the frequency is given by [40]

$$\text{ERB} = 24.7(0.00437f + 1). \tag{2.5}$$

An example of the critical bands covering a range of 3.7 kHz is listed in Table 2.1.

Table 2.1 List of the critical bands covering a range of 3.7 kHz [2].

Band No.	Center Frequency (Hz)	Bandwidth (Hz)
1	50	0–100
2	150	100–200
3	250	200–300
4	350	300–400
5	450	400-510
6	570	510–630
7	700	630–770
8	840	770–920
9	1000	920–1080
10	1170	1080–1270
11	1370	1270–1480
12	1600	1480–1720
13	1850	1720–2000
14	2150	2000–2320
15	2500	2320–2700
16	2900	2700–3150
17	3400	3150–3700

2.5 Auditory Masking

Masking is one of the important characteristics of the hearing system by which a weaker audio signal becomes inaudible by a louder signal occurring simultaneously or close in time [17]. In daily life we observe many examples of the simultaneous masking. For

example during a conversation in a very noisy environment, one needs to raise his voice in order to be understood.

The masking phenomena reflect the limited frequency and temporal resolutions of the hearing system. The mechanism of auditory masking, and the human hearing system in general, is not well understood. Although there are physiological explanations for the masking phenomena of the hearing system, it is almost impossible to develop comprehensive theories only based on physiological data. In order to obtain enough data, one needs to have access to the inner parts of the human hearing system such as the cochlea, basilar membrane, nerve fibers, the brain without damaging them. In order to overcome this barrier, we have to rely on psychoacoustic data collected through subjective tests.

There are different masking effects with different mechanisms; *Simultaneous masking* occurs when the masker and the maskee (masked signal) are presented to the hearing system at the same time. In addition to simultaneous masking, time domain masking phenomenon, referred to as *temporal masking*, occurs when the masker and the maskee are presented close in time, but not simultaneously. The phenomenon of masking a signal which occurs before the beginning of the masker is called *premasking* or *backward masking*. Another form of the temporal masking, which is referred to as *postmasking* or *forward masking*, happens if the masked signal occurs after the end of the masker.

In audio coding the masker is the original input signal and the maskee is either the quantization noise or weak components of the input signal. The masking phenomena can be exploited to reduce the bit rate, especially if a large number of the spectral components of the signal are masked. From a bit allocation point of view, the quality of the reconstructed signal will be enhanced by assigning bits to spectral components based on perceptual criteria. By properly shaping the quantization noise spectrum, we can make it less audible than a noise with the same energy but without noise shaping. In coding the spectral components, if scalar quantizers are used, the optimal step size for each scalar quantizer provided by the masking threshold such that the quantization noise lies below the masking threshold.

For almost all audio signals many spectral components are below the masking threshold and can be discarded. From our experience, on average for music and voiced speech, more than 50% of the transform coefficients are masked. In order to test the masking properties of the hearing system, the masking threshold for some speech and audio signals was calculated. After replacing the masked coefficients by zero, there was no perceptual difference between

the original and reconstructed signal. In fact the difference between the original and the reconstructed signal is not perceivable because it is masked by the signal itself. If we (intentionally) raise the level of masking to have about 20% of spectral components above the masking threshold, the quality of the reconstructed signal is still good. It implies the importance of coding the perceptually significant spectral components such that they are reproduced precisely.

2.5.1 Simultaneous Masking

Simultaneous masking occurs when two stimuli are *simultaneously* presented to the auditory system and one of them is made inaudible by the other. Physiological evidence reveals that the simultaneous masking is caused due to the function of the basilar membrane and the hair cells.

There are two theories for the mechanism of simultaneous masking [40]. One theory suggests that the masker produces a great amount of activities on the basilar membrane such that any activity caused by a weaker signal may become undetectable. In fact the hair cells detect the strongest vibration in any local region (critical band) along the basilar membrane. The simultaneous masking pattern is well predicted by this theory which models the hearing system as a bank of linear filters. The second theory, which is highly nonlinear, suggests that the masker suppresses the activity which the masked signal would produce if there is no masker. Based on this theory, the neural response to a tone at the characteristic frequency of that neuron might be suppressed by a tone which does not produce any neural activity in that neuron. Many researchers believe that the first theory plays the dominant role in the mechanism of the simultaneous masking.

Although the physiologically-based theories mentioned above explain the mechanism of the simultaneous masking phenomenon, the analytical masking models are developed using psychoacoustic data. In the following, we briefly present some psychoacoustic findings about the simultaneous masking properties of hearing.

To determine the masking pattern (curve) of a simple stimulus, the masker is fixed and the test signal (maskee) varies. The masking threshold at any frequency is the level of the test signal when it is just inaudible. Figure 2.7 shows the approximate masking curves due to narrowband noises centered at 1 kHz and 4 kHz with a level of 60 dB. As it is observed, the maximum of the masking curves depends on the center frequency. The peak of the

masking curve is 2–6 dB below the excitation level [37]. Note that the dashed curve in Fig. 2.7 shows the *threshold of hearing in quiet*; that is the minimum level at which the ear can detect a tone at a given frequency. This curve is measured by subjective tests. Listeners usually show different thresholds in quiet and therefore an average is taken as the threshold of hearing. The following formula expresses the threshold in quiet at frequency f (in Hz) [45],

$$T_q = 3.64(f/1000)^{-0.8} - 6.5\exp(-0.6(f/1000 - 3.3)^2) + 10^{-3}(f/1000)^4 \quad \text{dB}. \tag{2.6}$$

Based on psychoacoustic experiments, although the lower slope of the masking curve is almost independent of the masker level, the upper slope (towards the higher frequencies) depends on the level of the masker. As it is shown in Fig. 2.8 the higher the excitation level the lower the upper slope is. This nonlinear behavior of the hearing system is attributed to the saturation of the outer hair cells in the cochlea at high levels [40].

Note that the nature of the masker as being noise-like or tonal has an impact on the masking curve. For instance, the maximum of the masking curve due to a single tone is more sharp (peaky) [37]. Additionally the distance between the masker level and the masking threshold is greater for tonal signals.

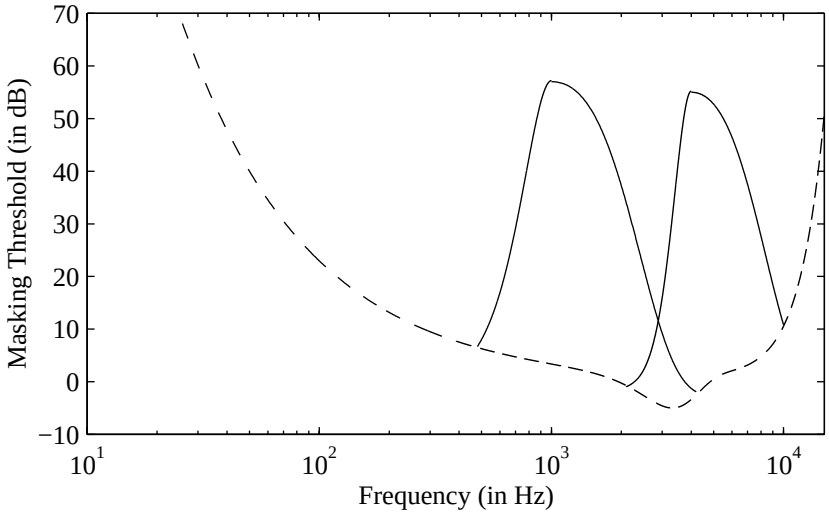


Fig. 2.7 Masking curves due to two narrowband noises centered at 1 kHz and 4 kHz. Dashed curve shows the absolute threshold of hearing.

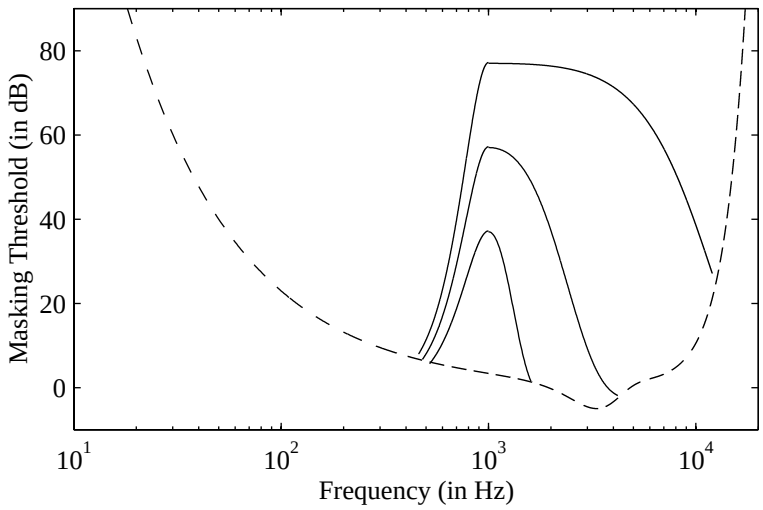


Fig. 2.8 Masking curves due to a narrow band noise centered at a 1 kHz tone with level (top to bottom): 80, 60, 40 dB.

2.5.2 Simultaneous Masking Models

Many analytical models have been proposed in the literature to calculate the simultaneous masking curve. We briefly discuss the following steps which are almost common among the models and differences are mostly about their parameters. A few masking models will be presented afterward.

Transformation from Frequency to Critical Band Scale

The linear frequency is mapped into the critical band scale. This is done to emulate the function of the basilar membrane to find a representation of the signal spectrum similar to that presented to the inner ear. As mentioned in previous sections, different analytical expressions have been proposed (based on the psychoacoustic measurements) to relate the linear frequency (in Hertz) to the critical band rate (in Bark).

Calculation of the Excitation Pattern

The energy distribution(*excitation pattern*) along the basilar membrane is calculated. In fact the excitation pattern is the distribution of the energy of the travelling wave along the basilar membrane due to a stimulus. For a complex sound, the excitation pattern for each spectral component is found. The shape of the excitation pattern caused by a single

spectral component is called the *spreading function*. Based on the psychoacoustic findings, the spreading function is a function of the frequency and the level of the masker. In almost all masking models a triangular shape (on a critical band scale) is assumed for the spreading function. However different slopes of the function on both sides have been reported in the literature [46, 44, 37, 47, 48, 49]. From psychoacoustic data, the masking pattern show steep slopes on the low frequency side (of the masker) of 80–240 dB/octave (for a tonal masker) and 55–190 dB/octave for a narrowband noise masker [40]. The slope on the high frequency side becomes less with increasing the masker level. Note that in transform audio coding, since each block of the input signal is multiplied by a window, the power spectrum of the signal will be spread due to the convolution in the frequency domain. It seems that we have to consider this effect when calculate the excitation pattern[2]. However since the slopes of the spreading function are found from psychoacoustic data and moreover there is no exact values for those slopes, we might ignore the windowing effects.

If a linear model is assumed for the inner ear, the global excitation pattern is found by convolving the energy spectrum (on a critical band scale) with a fixed spreading function (independent of the frequency and level of the masker).

In simultaneous masking models, different spreading functions have been used. The following spreading function proposed by Schroeder *et al* [32] is used by Johnston [51],

$$\mathrm{SpFn}(z) = 15.81 + 7.5(z + 0.474) - 17.5(1 + (z + 0.474)^2)^{0.5}, \tag{2.7}$$

where z is the critical band number in Bark. This spreading function is independent of the masker and has slope of 25 dB/Bark on the low frequency side of the masker and -10 dB/Bark on the high frequency side.

Terhardt [45] proposed a triangular spreading function with a fixed slope of 27 dB/Bark on the lower frequency side and $-24-(230/f)+0.2L$ dB/Bark on the higher frequency side (f and L are the frequency (in Hz) and the level (in dB) of the masker). This spreading function, contrary to Schroeder's, depends on the masker level and frequency.

Calculation of the Global Masking Curve

The masking threshold is found by subtracting an offset (masking index) from the excitation level. The masking index depends on the spectral structure of the masker. We will discuss

[2]Soulodre in [50] has discussed this effect.

this issue for different masking models in the following sections.

Zwicker suggests that if the variation of the excitation level due to a masker alone and the excitation caused by the masker and another signal is less than 1 dB, the second signal becomes inaudible [37]. This amount of variation , i.e., 1 dB is fixed regardless of the masker, meaning that the excitation level due to the maskee should be at least 6 dB below that of the masker. In reality, the nature of the masker as being tonal or noise-like has an impact on the masking threshold. For instance the masking threshold due to a narrow band noise is higher than a tonal signal with the same power. Moore [44] suggests that a value of 0.1 dB for the variation of the excitation level is a good criterion to make sure that the maskee will be inaudible (the excitation due to the maskee will be 16 dB below the excitation of the masker.). For a noise masker, Moore assumes a masking offset of 4 dB [40].

The global masking pattern is estimated by a superposition of the individual masking patterns. There is no clear rule to superpose the individual masking patterns. As a first approximation the hearing system is modelled as a overlapping linear bandpass filters. By this assumption, the global masking pattern is determined by summing up the individual masking thresholds. Many psychoacoustic masking models are based on this assumption [15, 51, 52]. However, some psychoacoustic experiments suggest that a nonlinear model of the additivity of the individual masking thresholds better fits the hearing system [53]. A linear summation of the individual masking thresholds results in a lower global threshold than that obtained by a nonlinear model. The final step is to make sure that the masking threshold is above the threshold of hearing.

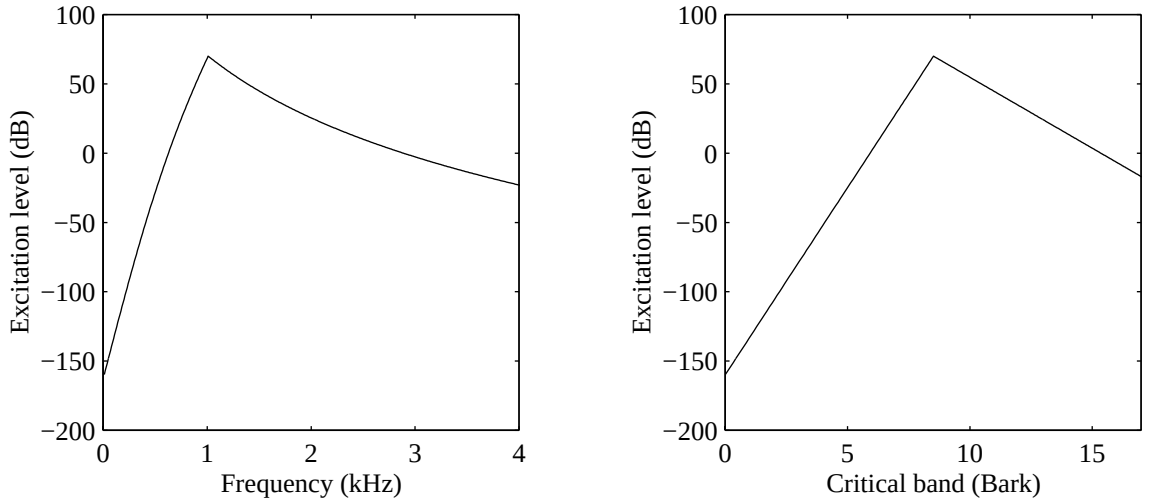
Terhardt's Masking Model

This model proposed in [45] assumes that the masking pattern produced by a pure tone is triangular in shape on the critical band rate scale. The upper slope of the masking pattern which depends on the frequency and the sound level of the masker is given by

$$s_u = -24 - \frac{230}{f} + 0.2L \quad \text{dB/Bark}, \tag{2.8}$$

where f is frequency of the masker in Hz and L is the level of the masker in dB. The lower slope of the masking pattern is independent of the masker level and is set to 27 dB/Bark. Fig 2.9 shows the excitation pattern produced by a 1 kHz tone with a sound level of 70 dB

versus the frequency and Bark. The masking level is 2–6 dB below the excitation level.



(a) Linear frequency scale.

(b) Bark scale.

Fig. 2.9 Excitation pattern produced by a 1 kHz tone.

The excitation level due to several maskers at frequency j is assumed to be additive and given by

$$L_{ex}(z_j) = 10\log_{10}(\sum_{i=1,i\neq j}^{I} 10^{[L_i - s(z_i - z_j)]/20})^2 \quad \text{dB}, \tag{2.9}$$

where I is the number of the spectral components, z_i and z_j are the Bark values of the i-th and j-th frequencies, L_i is the sound level at frequency i and s is the slope of the masking pattern, i.e., $s = 27$ dB if j is less than i and otherwise it will be found from Eq. 2.8. In order to take into consideration the masking effect of the noise inside critical band j, the intensity of the noise around component j should be added to $L_{ex}(z_j)$. The noise intensity P_n is found by summing up the sound intensities of those spectral components which fall in the critical band centered at Bark z_j except the five central components (the components at j and two neighboring components at each side). Additionally, the threshold of hearing is added to the masking power. The final masking power is given by

$$P_{\text{mask}}(z_j) = 10\log_{10}[(\sum_{i=1,i\neq j}^{I} 10^{[L_i - S(z_i - z_j)]/20})^2 + P_n + 10^{T_q/10}]\text{dB}, \tag{2.10}$$

where T_q is the threshold of hearing in quiet. As mentioned earlier, the masking threshold lies 2–6 dB below the masking power.

In [54] two following formulas have been given to calculate the masking threshold from the masking power

$$m(j) = L_{ex}(z_j) - 0.8(14.5 + \lfloor z_j \rfloor)\text{dB}, z_j \leq 14, \tag{2.11}$$

$$m(j) = L_{ex}(z_j) - 0.8(42.5 - \lfloor z_j \rfloor)\text{dB}, z_j \geq 15, \tag{2.12}$$

where $\lfloor z_j \rfloor$ denotes the integer part of z_j. In this approach the factor 0.8 and the conservative estimate of the offset at high frequencies are to make up for the lack of accuracy in estimating the nature of the signal, i.e., the tonality factor.

In Terhardt's masking model the discrimination between noisy and tonal spectral components is very approximate. Moreover, there is no frequency-dependent formula to calculate the masking threshold from the masking power. Another problem with this model is that the masking threshold at each spectral component is calculated due to other components, whereas the masking effect of the component itself contributes to the masking threshold.

Johnston's Masking Model

This model was proposed by Johnston [51] based on the work by Schroeder *et al* [32]. In order to calculate the masking threshold, the power in each critical band is found; then the Bark power spectrum will be spread over all critical bands through convolving the Bark spectrum with the following spreading function

$$\text{SpFn(z)} = 15.81 + 7.5(z + .474) - 17.5(1 + (z + 0.474)^2)^{0.5}, \tag{2.13}$$

where z is the separation in critical bands. This spreading function is independent of the level and frequency of the masker.

For noise-masking-tone, the masking threshold will be 5.5 dB below the spread spectrum. For tone-masking-noise the masking threshold will be $(14.5 + i)$ dB below the spread spectrum, where i is the critical band index. In order to determine the nature of the signal as being tone-like or noise-like, the spectral flatness measure which is defined as follows is

used

$$\text{SFM} = 10\log_{10}(\frac{G_m}{A_m}) \quad \text{dB}, \tag{2.14}$$

where G_m and A_m are the geometric mean and arithmetic mean respectively. Then the tonality factor is defined as follows

$$a = \min(\frac{\text{SFM}}{\text{SFM}_{\text{max}}}, 1), \tag{2.15}$$

where SFM_{max} corresponds to a signal which is assumed to be a pure tone and is set to -60 dB; a zero value for SFM represents noise. To find the masking threshold the following offset is subtracted from the spread spectrum (in dB)

$$O(i) = a(14.5 + i) + 5.5(1 - a). \tag{2.16}$$

Finally the masking threshold is compared with the threshold of hearing to make sure that it is not below the threshold of hearing.

MPEG Masking Models

Two psychoacoustic models are given in the MPEG standard [15, 55]. The output of both psychoacoustic models is a signal-to-mask ratio for each subband or a group of subbands to be used in bit allocation.

Psychoacoustic Model 1

The input block of the audio signal is multiplied by a Hanning window and transformed to the frequency domain using a 512-point FFT (Layer I) or a 1024-point FFT (Layer II). The output of the FFT is used to determine the tonal and noise-like components by finding the local peaks. This discrimination is important as there is a difference between the masking threshold due to a tonal or noise-like component. The masking threshold is calculated for each component above the threshold in quiet. The masking threshold at frequency i ($z(i)$ in Bark) due to a masker component at frequency j ($z(j)$ in Bark) with the sound pressure level $L(z(j))$ is given by

$$T[z(j), z(i)] = L(z(j)) + \Upsilon(z(j)) + \mathcal{M}\big(z(j), z(i)\big) \text{ dB}, \tag{2.17}$$

where $\mathcal{M}$ is the masking function characterized by different lower and upper slopes defined as

$$\begin{aligned}
\mathcal{M} &= 17(\Delta z + 1) - \big(0.4L(z(j)) + 6\big) \quad \text{dB}, && -3 \le \Delta z < -1 \\
\mathcal{M} &= (0.4L(z(j)) + 6)\Delta z \quad \text{dB}, && -1 \le \Delta z < 0 \\
\mathcal{M} &= -17\Delta z \quad \text{dB}, && 0 \le \Delta z < 1 \\
\mathcal{M} &= -(\Delta z - 1)\big(17 - 0.15L(z(j))\big) - 17 \quad \text{dB}, && 1 \le \Delta z < 8
\end{aligned} \tag{2.18}$$

where $\Delta z = z(j) - z(i)$. For reducing the complexity, the masking effect of any masker is not considered outside the range $-3 < \Delta z < 8$. In Eq. 2.17, $\Upsilon(.)$ is the masking index which is different for tonal and non-tonal maskers. For tonal maskers

$$\Upsilon(z(j)) = -1.525 - 0.275z(j) - 4.5 \quad \text{dB}, \tag{2.19}$$

and for nontonal maskers

$$\Upsilon(z(j)) = -1.525 - 0.175z(j) - 0.5 \quad \text{dB}. \tag{2.20}$$

The global masking threshold at frequency i, $T_g(i)$, is obtained by adding the masking threshold due to each masker to the threshold in quiet,

$$T_g(i) = 10\log_{10}(10^{(T_q(i)/10)} + \sum_j 10^{(T[z(j),z(i)]/10)}), \tag{2.21}$$

where T_q is the threshold of hearing in quiet. In this model the masking threshold due to each tonal component is calculated. All non-tonal components in a critical band are summed to form a single non-tonal masker for each critical band. The index number of the non-tonal component is set to the spectral line nearest to the geometric mean of the critical band.

Psychoacoustic Model 2

This model is based on the model developed by Schroeder *et al* [32] and very similar to the model proposed by Johnston [51]. The main difference between this masking model and the MPEG psychoacoustic model 1 is that instead of a binary classification of the spectral components, which is not consistent with the mechanism of the hearing system, each component is continuously labeled between two limits. A tonality factor is found for

each band based on the predictability of the current spectral line from the corresponding two previous components.

AAC Psychoacoustic Model

The psychoacoustic model used in the Advanced Audio Coding (AAC) [52] standard is very similar to the MPEG psychoacoustic model 2. The only difference is that the offset value for the tone-masking-noise is 18 dB for all bands.

NPAC Simultaneous Masking Model

Since, in our coding paradigm (NPAC) [35], the MDCT is employed to decompose the input signal, we have modified the model proposed by Johnston [51] to calculate the masking threshold corresponding the MDCT coefficients. The model is more suitable for an MDCT-based coder and also discriminates between different frequency bands to calculate the masking index (offset).

We briefly point out the modification made to Johnston's model. Starting with an FFT of the input frame, the calculation of the masking threshold consists of the same steps up to finding the tonality factor. In contrast to the previous model in which the spectral flatness measure is used to identify the nature of the whole frame as being tone-like or noise-like, we take another approach based on the predictability of the transform coefficients in each critical band. Note that most audio signals have a noise-like structure at high frequencies despite the fact that they may have a strong harmonic structure at low frequencies. Considering this fact, it would be more accurate to identify the nature of the spectrum locally at different critical bands. The tonality factor will be calculated for each Bark using

$$\tilde{X}^{(j)} = 2X^{(j-1)} - X^{(j-2)}, \tag{2.22}$$

where $\tilde{X}^{(j)}$ is a linear prediction of the current subvector based on the observation of previous subvectors $X^{(j-1)}$ and $X^{(j-2)}$. The relative prediction error is calculated

$$e = \frac{\parallel X^j - \tilde{X}^{(j)} \parallel}{\parallel X^j \parallel + \parallel \tilde{X}^j \parallel}. \tag{2.23}$$

The relative prediction error will be converted to the tonality factor according to [15]

$$a = \min(1, \max(-0.3 - 0.43\log(e), 0)). \tag{2.24}$$

The following offset for each critical band is subtracted from the log spread Bark spectrum to find the masking threshold

$$O(i) = a(14.5 + i) + 5.5(1 - a), \tag{2.25}$$

where i is the index of the critical band. Like the previous model, a comparison of the masking threshold with the absolute threshold of hearing is made. Since the masking threshold is calculated based on the DFT of the input frame, it is not accurate to use this masking threshold for the MDCT coefficients. Instead, we consider the following relationship between the DFT and MDCT to find a more accurate masking threshold for MDCT coefficients,

$$C(k) = \sqrt{2/M}|S(k)|\cos\big(\frac{2\pi n_0(k+0.5)}{N} - \angle S(k)\big), \tag{2.26}$$

where $S(k)$ is the Fourier transform of the modulated windowed input signal, $C(k)$ is the MDCT, $n_0 = (M+1)/2$, M and N are the number of samples in the frequency and time domain, respectively. If m_{DFT} is the masking threshold corresponding to the kth DFT coefficient, then in order to have the same Signal-to-Mask Ratio (SMR) at any coefficient in the DFT and MDCT domain, the following relation should hold

$$C^2(k)/m_{\text{MDCT}} = |S(k)|^2/m_{\text{DFT}}. \tag{2.27}$$

Considering the relation between the MDCT and DFT, we find the masking threshold for the kth MDCT coefficient [35],

$$m_{\text{MDCT}} = \frac{2}{M} m_{\text{DFT}} \cos^2\big(\frac{2\pi n_0(k+0.5)}{N} - \angle S(k)\big). \tag{2.28}$$

2.6 Temporal Masking

Temporal masking occurs when the masker and the maskee are not presented to the hearing system at the same time. The temporal masking characteristic of the hearing system is asymmetric, meaning that the backward masking effect is much less than the forward masking. Backward masking is effective about 5 msec before the occurrence of a strong stimuli, whereas forward masking lasts up to 200 msec [2]. An example of the postmasking is less audibility of low energy consonants following a high energy vowel. Figure 2.10 shows the temporal masking pattern due to a short burst of a tonal signal [2].

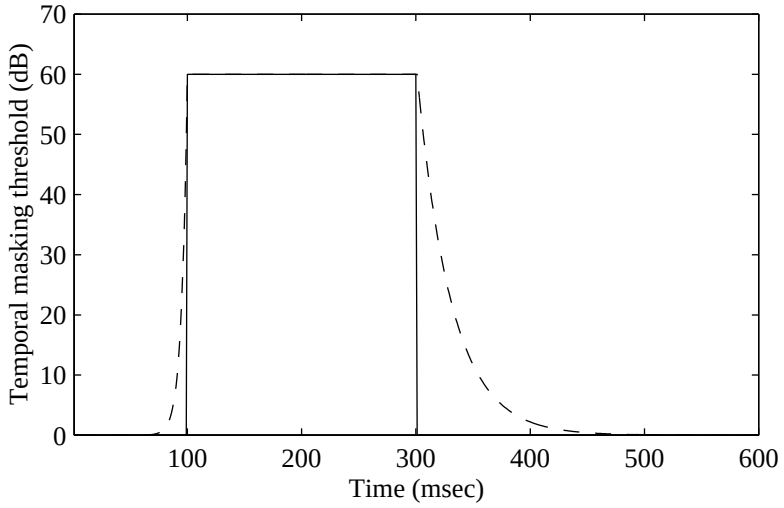


Fig. 2.10 The temporal masking pattern (dashed curve) due to a short burst of a tonal signal starting at 100 msec and ending at 300 msec [2].

Although psychoacoustic experiments reveal the temporal masking effects, this phenomenon is not well understood. Temporal masking effects suggest that the brain might integrate sound over a short time interval or perhaps the brain simply processes loud sounds faster than soft sounds [2].

Moore suggests that the following different phenomena contribute to the forward masking effects which occur after the end of a masker [40].

- Temporal overlap of the basilar membrane responses to different stimuli might play a role in the temporal masking. This phenomenon contributes to temporal masking for about 10 msec after the end of the masker.

- Short term neural fatigue at higher neural levels might reduce the perception of the activity of the maskee which occurs after the masker.

- The neural activity as a response to the masker persists at higher levels after the end of the masker. This activity masks the activity produced by the maskee. This effect is also suggested to occur at stages higher than the auditory nerve.

2.6.1 Temporal Masking Model

Of the two forms of temporal masking, backward masking is more vague and also far less important. Therefore we just focus on the more prominent form of the temporal masking, i.e., forward masking. Any forward masking model is based on psychoacoustic experiments which reveal the following findings about that phenomenon [40]

- The forward masking level (in dB) is approximated by a linear function of the logarithm of the time interval between the end of the masker and the onset of the maskee. The level of the forward masking decays to zero (regardless of the masker level) after almost 200 msec.

- Forward masking is affected by the frequency of the masker and the maskee.

A lot of research has been done on the temporal masking of the hearing system and there are a few analytical expressions which approximate that effect [50, 56, 57, 58, 59, 46].

We have developed the following model based on the model proposed in [50] as it takes both the effect of the frequency and the level of the masker into account

$$m_{\mathrm{t}}(f, L) = \alpha + \beta \exp(-f/\gamma), \tag{2.29}$$

where m_{t} is the temporal masking in dB, L is the masker level in dB, f is the frequency in Hz and α, β, γ are three parameters to be found from experimental data. In [50] three expression have been fitted to the experimental data for α, β, γ. In this work, we consider the temporal masking if the level of the masker is more than 30 dB. Based on this assumption and the data given in [50] we have found the following expression for the above-mentioned

parameters:

$$\begin{aligned}\alpha &= 0.001L^2 + 0.2267L + 17.7142,\\ \beta &= -0.0047L^2 + 1.2256L - 24.32548,\\ \gamma &= -0.0002L^4 + 0.0546L^3 - 5.4685L^2 + 234.7411L - 3325.0350.\end{aligned} \tag{2.30}$$

Note that the data reported in [50] indicate the level of the temporal masking at 20 msec after the masker. Although the time interval between successive frames in our coder is 15 msec, and hence the temporal masking level will be underestimated using this formula, we have chosen to use it in order to prevent any overmasking of the transform coefficients. In our coding scheme, we calculate the temporal masking for each critical band. In doing so we assume that all the energy in each subband is concentrated in the center frequency (except the first band for which we set f to 100 Hz) and the sound level is due to the contribution of all the coefficients in the band. This way, for each frame of transform coefficients we calculate the masking threshold at 17 frequencies. If the masking threshold is greater than the sound level in any band, we assume that all the coefficients in that band are masked. If the transform coefficients are not completely masked, the masking threshold will be equally divided between the coefficients. We have examined the accuracy of this model by subject tests and noticed no difference between the original signals and processed signals where the temporally-masked transform coefficients were set to zero.

2.7 Combined Masking Threshold

A combined masking threshold is computed by considering the effect of both temporal and simultaneous masking thresholds. A lot of research has been done to find how to combine these two different phenomena [50]. One way to deal with this problem is to linearly sum the masking levels. According to some experiments this model is not appropriate; and therefore another model called *power-law* has been proposed in the literature as follows [50, 60, 53]

$$m_{\text{net}} = (m_1^p + m_2^p)^{(1/p)}, \tag{2.31}$$

where m_{net} is the net masking level due to two masking levels m_1 and m_2. A value of 0.3 for parameter p is found to be the best match to experimental data [50].

Chapter 3

Signal Decomposition Using Lapped Transforms

The first stage of transform coding is to decompose blocks of the input signal into its frequency components. In designing the transform, we have to consider the following design goals.

(a) Perfect Reconstruction

The spectral decomposition should be invertible, i.e., the transform should be *perfect reconstruction*. This refers to signal decomposition from which the original signal can be exactly recovered in the absence of quantization [61]. This has the advantage that all noise which is added in the coding/decoding process is generated by the quantizer. Since the noise source is known, it can be controlled so that it is masked by the signal. It is also desirable that the transform and its inverting process both maintain a high degree of frequency selectivity in order to accurately compute the auditory masking pattern.

(b) Critical Sampling

The analysis system should be *critically sampled* [61]; i.e., the number of transform coefficients per time is the same as the input sample rate. Critical sampling ensures that subsequent stages of the audio coder are not required to operate at a higher sample rate than the input sample rate. Although non-critically sampled systems allow more flexibility

in designing the filterbank, they have a higher sample rate at the output of the analysis stage than the input sample rate.

(c) Good Frequency and Temporal Resolution

The bandwidth of each bandpass filter (in the filterbank) should be equal to or narrower than the width of the narrowest critical band, i.e., 100 Hz. This makes it easy to control the perception of the quantization noise. At the same time, the analysis interval for the filterbank should be small enough to avoid introducing noise components over an interval such that (temporal) masking constraints are violated. In general, most uniformly spaced filterbanks cannot meet both of these constraints because of the large variation in width of the critical bands with frequency. Moreover, a high frequency resolution is desirable to take advantage of the transform gain, which is the frequency domain equivalent of the prediction gain. (The transform gain is higher for signals with a non-flat spectrum.)

3.1 Block Transforms

Before computing the transform of a given signal $x(n)$, we must group its samples into blocks. Referring to $\boldsymbol{x}$ as one of these blocks, the transform of $\boldsymbol{x}$, $\boldsymbol{X}$, is computed by

$$\boldsymbol{X} = \mathbf{T}\boldsymbol{x},$$

where $\mathbf{T}$ is the transformation matrix. In order to reconstruct $\boldsymbol{x}$ from $\boldsymbol{X}$, $\mathbf{T}$ must be invertible. Each choice of $\mathbf{T}$ leads to a different transform. For compression purposes, $\mathbf{T}$ should compact the energy of each block of data into a few coefficients in the transform domain. In transform coding, instead of quantizing the samples in the time domain, we perform the quantization on the transform coefficients by allocating more bits to the coefficients containing higher energy. Besides energy compaction (in the sense that the energy is concentrated in only a few coefficients), the transform coefficients should be uncorrelated. The Karhunen-Loeve transform (KLT) is the optimal transform. This is because the KLT is the orthogonal transform that will produce a set of uncorrelated coefficients. Moreover, the KLT maximizes the energy compaction in $\boldsymbol{X}$. Although the KLT is an ideal choice in signal compression systems, it is seldom used in practice since it is signal dependent. One of the major disadvantages of block transforms is the problem of *block edge effects* that we

will discuss later.

3.2 Lapped Transforms

The basic motivation behind the development of lapped transforms comes from one of the major disadvantages of traditional block transforms, i.e., *block edge effects*, which are discontinuities in the reconstructed signal [61, 62]. In transform coding, we start by transforming a block of N samples of the input signal. The transform coefficients are then quantized and transmitted. At the receiver, the inverse transform is computed and the reconstructed signal block is appended to the output. Because of the independent processing of each block, some of the quantization errors will produce discontinuities in the signal. Due to the block edge effects, there will be an audible periodically occurring noise in the reconstructed signal. One of the most computationally efficient approaches towards the reduction of blocking effects is prefiltering and postfiltering [61]. The filtering techniques have the disadvantage of reducing the coding gain and producing a lowpass effect around the boundaries.

In lapped transforms, the basis functions are longer than the length of the transform. In this way, the basis functions from one block and its neighboring block overlap. In addition to reduction of block edge effects, a lapped transform can achieve significant improvements in the coding gain G_{TC} (which is discussed later), when compared to standard block transforms [61].

3.2.1 Analysis of Lapped Transforms

With a lapped transform, we map an input block of N samples into M transform coefficients. Since we want to have the same sample rate at the input and output of the analysis stage, we compute M new transform coefficients for every new M input samples. In this way there will be an overlap of $N - M$ samples in the computation of consecutive blocks. The idea of a lapped transform is shown in Fig. 3.1.

Here we use a matrix notation to analyze a lapped transform with a 50% overlap between successive blocks of the input signal. Looking at Fig. 3.1, we transform the first block $\boldsymbol{x}^{(1)}$ of the input signal by

$$\boldsymbol{X}^{(1)} = \mathbf{H}\boldsymbol{x}^{(1)}, \tag{3.1}$$

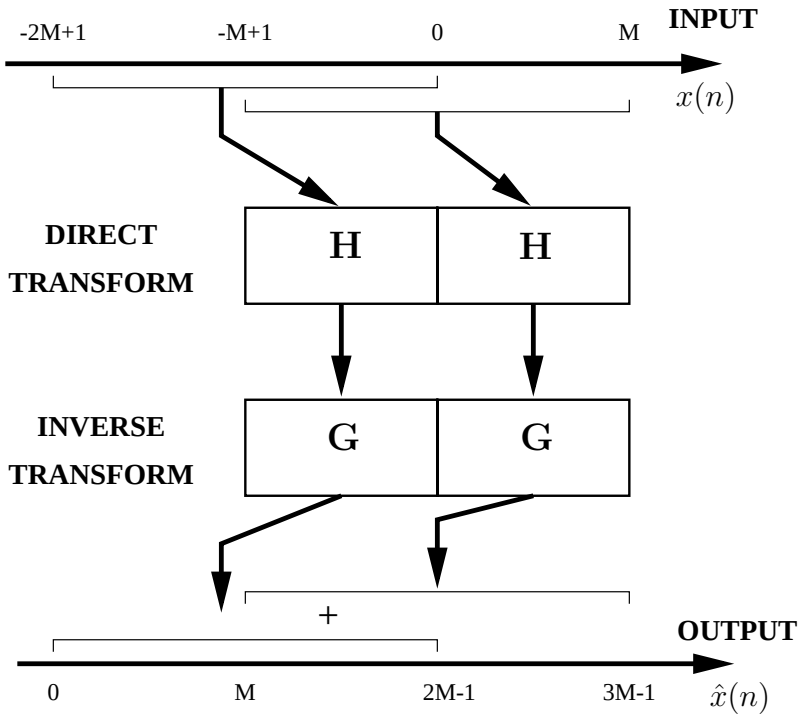


Fig. 3.1 Signal processing with a lapped transform with a 50% overlap between successive frames [61].

where $\mathbf{H}$ is an $M \times 2M$ analysis matrix, $\boldsymbol{x}^{(1)}$ is a column vector containing the following $2M$ samples of the input signal

$$\boldsymbol{x}^{(1)} = [x(-2M+1) \dots x(0)]^t, \tag{3.2}$$

and $\boldsymbol{X}^{(1)}$ is the vector of M transform coefficients. The $2M \times M$ synthesis matrix $\mathbf{G}$ transforms $\boldsymbol{X}^{(1)}$ back into the time domain;

$$\boldsymbol{y}^{(1)} = \mathbf{G}\boldsymbol{X}^{(1)}. \tag{3.3}$$

Since the algorithmic delay[1] for this transformation is $2M-1$ samples, $\mathbf{y}^{(1)}$ contains the following samples,

$$\boldsymbol{y}^{(1)} = [y(0) \dots y(2M-1)]^t. \tag{3.4}$$

[1]Algorithmic delay is the length of the block of data plus the lookahead. Note that there is no lookahead in this transform.

The next block of data, which contains M samples from the previous block, goes through the same steps to obtain

$$\boldsymbol{X}^{(2)} = \mathbf{H}\boldsymbol{x}^{(2)}, \tag{3.5}$$

and

$$\boldsymbol{y}^{(2)} = \mathbf{G}\boldsymbol{X}^{(2)} = [y(M) \dots y(3M-1)]^t. \tag{3.6}$$

As there is 50% overlap between the input blocks, we have the same amount of overlap between $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$. In order to have perfect reconstruction, the sum of the overlapping parts of $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ should equal the corresponding part of the input signal. Note that the algorithmic delay is $2M-1$ samples, therefore for perfect reconstruction we must have

$$[\hat{x}(M) \dots \hat{x}(2M-1)] = [x(-M+1) \dots x(0)]. \tag{3.7}$$

In order to express the left side of Eq. 3.7 in terms of the input signal and the analysis and synthesis matrices, we rewrite Eq. 3.1 as follows

$$\boldsymbol{X}^{(1)} = [\mathbf{H_1} \quad \mathbf{H_2}] \begin{bmatrix} \boldsymbol{x_a}^{(1)} \\ \boldsymbol{x_b}^{(1)} \end{bmatrix} = \mathbf{H_1}\boldsymbol{x_a}^{\mathbf{(1)}} + \mathbf{H_2}\boldsymbol{x_b}^{(1)}, \tag{3.8}$$

where $\mathbf{H_1}$ and $\mathbf{H_2}$ are two $M \times M$ square matrices containing the first and second M columns of the analysis matrix $\mathbf{H}$; $\boldsymbol{x_a}^{(1)}$ and $\boldsymbol{x_b}^{(1)}$ contain the first and second M elements of $\boldsymbol{x}^{(1)}$. Also we rewrite Eq. 3.3 as follows

$$\begin{bmatrix} \boldsymbol{y_a}^{(1)} \\ \boldsymbol{y_b}^{(1)} \end{bmatrix} = \begin{bmatrix} \mathbf{G_1} \\ \mathbf{G_2} \end{bmatrix} \boldsymbol{X}^{(1)}. \tag{3.9}$$

Therefore,

$$\boldsymbol{y_a}^{(1)} = \mathbf{G_1}\boldsymbol{X}^{(1)}, \qquad \boldsymbol{y_b}^{(1)} = \mathbf{G_2}\boldsymbol{X}^{(1)}, \tag{3.10}$$

where $\boldsymbol{y_a}^{(1)}$ and $\boldsymbol{y_b}^{(2)}$ are the first and second half of $\boldsymbol{y}^{(1)}$; $\mathbf{G_1}$ and $\mathbf{G_2}$ are two $M \times M$ square matrices containing the first and second M rows of the synthesis matrix $\mathbf{G}$. Similarly we obtain

$$\boldsymbol{y_a}^{(2)} = \mathbf{G_1}\boldsymbol{X}^{(2)}, \qquad \boldsymbol{y_b}^{(2)} = \mathbf{G_2}\boldsymbol{X}^{(2)}. \tag{3.11}$$

We express the desired part of the reconstructed signal as the sum of the overlapping parts

of $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$ as follows

$$[\hat{x}(M)\dots\hat{x}(2M-1)]^t = \boldsymbol{y_b}^{(1)} + \boldsymbol{y_a}^{(2)}. \tag{3.12}$$

By considering Eq. 3.8 and Eq. 3.10, we obtain the followings

$$\boldsymbol{y}_b{}^{(1)} = \mathbf{G_2H_1}\boldsymbol{x_a}^{(1)} + \mathbf{G_2H_2}\boldsymbol{x_b}^{(1)}, \tag{3.13}$$

and similarly

$$\boldsymbol{y_a}^{(2)} = \mathbf{G_1H_1}\boldsymbol{x_a}^{(2)} + \mathbf{G_1H_2}\boldsymbol{x_b}^{(2)}. \tag{3.14}$$

By Combining Eq. 3.13 and Eq. 3.14, we obtain the desired segment of the output as follows

$$\boldsymbol{y_b}^{(1)} + \boldsymbol{y_a}^{(2)} = \mathbf{G_2H_1}\boldsymbol{x_a}^{(1)} + \mathbf{G_2H_2}\boldsymbol{x_b}^{(1)} + \mathbf{G_1H_1}\boldsymbol{x_a}^{(2)} + \mathbf{G_1H_2}\boldsymbol{x_b}^{(2)}. \tag{3.15}$$

Since we have

$$\boldsymbol{x_b}^{(1)} = \boldsymbol{x_a}^{(2)} = [x(-M+1)\dots x(0)]^t, \tag{3.16}$$

in order to have perfect reconstruction (see Eq. 3.7), we must have in Eq. 3.15

$$\mathbf{G_2H_1} = \mathbf{G_1H_2} = \mathbf{0_M}, \tag{3.17}$$

and

$$\mathbf{G_1H_1} + \mathbf{G_2H_2} = \mathbf{I_M}, \tag{3.18}$$

where $\mathbf{0_M}$ is an $M \times M$ zero matrix and $\mathbf{I_M}$ is an $M \times M$ identity matrix.

Perfect reconstruction requires that Eq. 3.17 and Eq. 3.18 be satisfied. However in audio coding a high coding gain is very desirable. Therefore, the analysis matrix $\mathbf{H}$ must compact the energy of each frame of the input signal into a few transform coefficients.

If we look at the analysis matrix $\mathbf{H}$ from a filtering point of view, each row of that matrix represents the impulse response of a bandpass FIR filter. Therefore in order to achieve a high coding gain, the frequency response of each row should ideally resemble the frequency response of an ideal lowpass filter with little leakage into other bands. There are efficient ways to design the analysis matrix such as the Modulated Lapped Transform (MLT) which will be discussed later. One special case of lapped transform is when $\mathbf{G} = \mathbf{H}^t$.

In that case the perfect reconstruction conditions become

$$\mathbf{H_2}^t\mathbf{H_1} = \mathbf{H_1}^t\mathbf{H_2} = \mathbf{0_M}, \tag{3.19}$$

$$\mathbf{H_1}^t\mathbf{H_1} + \mathbf{H_2}^t\mathbf{H_2} = \mathbf{I_M}. \tag{3.20}$$

This special case is referred to as a *Lapped Orthogonal Transform (LOT)* [63]. Eq. 3.19 requires that $\mathbf{H_1}$ and $\mathbf{H_2}$ be orthogonal. It means that the overlapping parts of the basis functions are orthogonal. Eq. 3.20 implies that the rows of the analysis matrix $\mathbf{H}$ form a set of orthonormal basis functions.

Comment on the Analysis and Synthesis Matrices

From Eq. 3.12, we conclude that

$$\text{Range}(\mathbf{H_1}) \subseteq \text{Null}(\mathbf{G_2}), \tag{3.21}$$

where Range[2] and Null[3] denotes the range space and the null space of a matrix. Eq. 3.21 leads to

$$\text{rank}(\mathbf{H_1}) \leq \dim(\text{Null}(\mathbf{G_2})), \tag{3.22}$$

and similarly,

$$\text{rank}(\mathbf{H_2}) \leq \dim(\text{Null}(\mathbf{G_1})). \tag{3.23}$$

Note that for any matrix $\dim(\text{Range}(.)) = \text{rank}\ (.)$. Eq. 3.18 implies that

$$\text{rank}(\mathbf{G_1H_1} + \mathbf{G_2H_2}) = \text{rank}(\mathbf{I_M}) = \mathbf{M}. \tag{3.24}$$

For any $M \times M$ matrix such as $\mathbf{H_1}$ we have

$$\text{rank}(\mathbf{H_1}) + \dim(\text{Null}(\mathbf{H_1})) = M. \tag{3.25}$$

From Eq. 3.24, we conclude that

$$\text{rank}(\mathbf{G_1H_1} + \mathbf{G_2H_2}) \leq \text{rank}(\mathbf{H_1}) + \text{rank}(\mathbf{H_2})). \tag{3.26}$$

[2]For matrix $\mathbf{H}$, $\text{Range}(\mathbf{H}) = \{\boldsymbol{y}|\exists \boldsymbol{x} : \mathbf{H}(\boldsymbol{x}) = \boldsymbol{y}\}$.
[3]For matrix $\mathbf{H}$, $\text{Null}(\mathbf{H}) = \{\boldsymbol{x}|\mathbf{H}(\boldsymbol{x}) = \mathbf{0}\}$.

By considering Eqs. 3.22, 3.23, 3.25 and 3.26, we get

$$M \leq M - \text{rank}(\mathbf{G_1}) + M - \text{rank}(\mathbf{G_2}), \tag{3.27}$$

and then

$$\text{rank}(\mathbf{G_1}) + \text{Range}(\mathbf{G_2}) \leq M. \tag{3.28}$$

On the other hand, intuitively the dimension of the null space of the synthesis matrix $\mathbf{G}$ must be zero, meaning that the synthesis matrix should map a vector to a zero vector only if that vector is a zero vector too. Therefore we must have

$$\text{rank}(\mathbf{G_1}) + \text{rank}(\mathbf{G_2}) \geq \text{rank}(\mathbf{G}) = M. \tag{3.29}$$

Eq. 3.28 and Eq. 3.29 lead to

$$\text{rank}(\mathbf{G_1}) + \text{rank}(\mathbf{G_2}) = M. \tag{3.30}$$

Since $\mathbf{G_1}$ and $\mathbf{G_2}$ map into an M dimensional space, Eq. 3.30 implies that they map into two orthogonal subspaces which span the entire M dimensional space.

The analysis matrix $\mathbf{H}$ should (ideally) produce M uncorrelated transform coefficients corresponding to each block of the input signal. This means that the rank of $\mathbf{H}$ should equal M. For sub-matrices $\mathbf{H_1}$ and $\mathbf{H_2}$, we have

$$\text{rank}(\mathbf{H_1}) + \text{rank}(\mathbf{H_2}) \geq \text{rank}(\mathbf{H}) = M. \tag{3.31}$$

On the other hand from Eqs. 3.22, 3.23, 3.25 and 3.30, we can conclude

$$\text{rank}(\mathbf{H_1}) + \text{rank}(\mathbf{H_2}) \leq M. \tag{3.32}$$

Finally we find the same relationship between the sub-matrices of the analysis matrix as we have for the synthesis matrix;

$$\text{rank}(\mathbf{H_1}) + \text{rank}(\mathbf{H_2}) = M. \tag{3.33}$$

The Perfect Reconstruction (PR) conditions are the only constraints that must be satisfied.

However, the relations between the sub-matrices provide us with some insight into lapped transforms. Moreover, those relations might be used as the constraints for an optimization procedure to design a lapped transform.

3.2.2 Filterbank Representation of a Lapped Transform

Any transform including a lapped transform can be represented by a filterbank structure [64, 65, 66]. The finite impulse responses of the analysis filters are the time reversed of the rows of the analysis matrix **H** [61]. The finite impulse responses of the synthesis filters are the columns of the synthesis matrix **G** [61]. Fig. 3.2 shows a block diagram of a lapped transform.

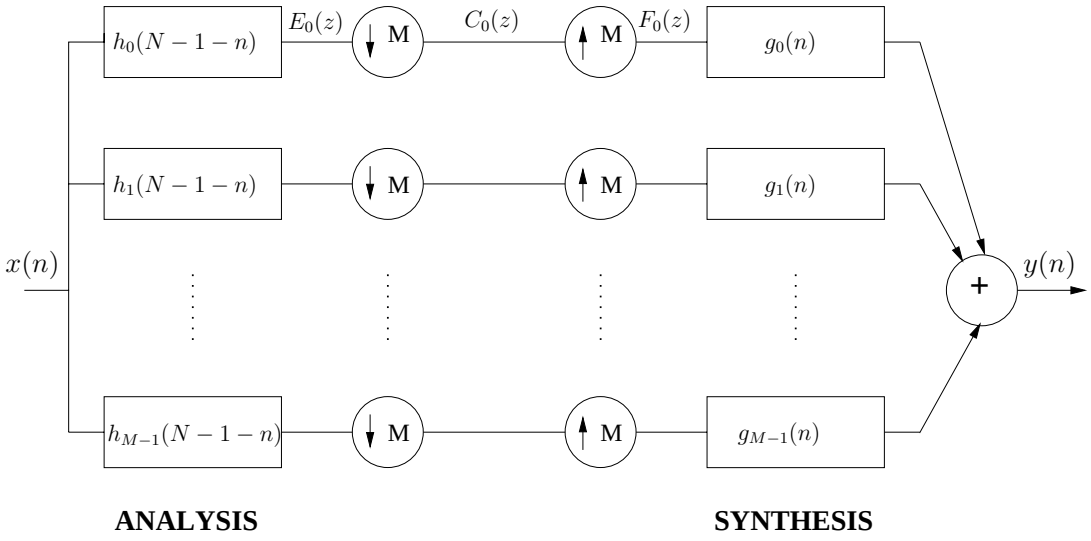


Fig. 3.2 Filterbank representation of a lapped transform.

In order to find the relationship between the z transform of the input $x(n)$ and the output $y(n)$ of the filterbank, we analyze the first branch of the filterbank. Since we have a similar structure for all branches, the input signal goes through the same processing (with different filters) in different paths. In Fig. 3.2

$$E_0(z) = z^{-(N-1)} X(z) H_0(z^{-1}). \tag{3.34}$$

After downsampling by M, we get

$$C_0(z) = \frac{1}{M}\sum_{k=0}^{M-1} E_0(e^{\frac{-j2\pi k}{M}} z^{\frac{1}{M}}). \tag{3.35}$$

After upsampling by M, $F_0(z)$ is given by

$$F_0(z) = C_0(z^M) = \frac{1}{M}\sum_{k=0}^{M-1} E_0(e^{\frac{-j2\pi k}{M}} z). \tag{3.36}$$

By plugging Eq. 3.34 into Eq. 3.36, we get

$$F_0(z) = \frac{1}{M} z^{-(N-1)} \sum_{k=0}^{M-1} e^{\frac{j2\pi k(N-1)}{M}} X_0(e^{\frac{-j2\pi k}{M}} z) H_0(e^{\frac{j2\pi k}{M}} z^{-1}). \tag{3.37}$$

Finally $Y(z)$ is given by

$$Y(z) = \sum_{i=0}^{M-1} G_i(z) F_i(z), \tag{3.38}$$

$$Y(z) = z^{-(N-1)} \sum_{i=0}^{M-1} \Big(\frac{1}{M} \sum_{k=0}^{M-1} e^{\frac{j2\pi k(N-1)}{M}} H_i(e^{\frac{j2\pi k}{M}} z^{-1}) G_i(z) \Big) X_i(e^{\frac{-j2\pi k}{M}} z). \tag{3.39}$$

As seen in Eq. 3.39, the filterbank output is a combination of the delayed version of the input signal and many other terms. For a block transform with perfect reconstruction (without any overlap between successive frames) it is possible to recover $x(n)$ from $y(n)$ if we satisfy the following conditions; for intersymbol interference (ISI) cancellation,

$$\sum_{i=1}^{M-1} H_i(e^{\frac{j2\pi k}{M}} z^{-1}) G_i(z) = 0, \tag{3.40}$$

and in order to have no amplitude and phase distortion

$$H_0(z^{-1}) G_0(z) = M. \tag{3.41}$$

The above-mentioned conditions cannot be satisfied for lapped transforms as the main idea of a lapped transform is to construct the output signal by overlapping and adding the

inverse transform of successive blocks of data.

3.3 Modulated Lapped Transforms

Modulated Lapped Transforms (MLT), which are also known as Modified Discrete Cosine Transforms (MDCT), proposed by Princen and Bradley [67, 68], form a family of lapped transforms that is generated from modulations of a low-pass prototype filter. The basis functions of MLT have lengths equal to $N = 2M$, where M is the number of subbands. Perfect reconstruction can be achieved with appropriate choices of the phase of the modulated cosine function and the low-pass prototype window.

The main advantage of an MLT filterbank is that it can be computed efficiently. The MLT basis functions are defined by [67]

$$h(n)\sqrt{2/M}\cos\big((n+\frac{M+1}{2})(k+0.5)\frac{\pi}{M}\big), \tag{3.42}$$

where $k = 0, 1, ..., M-1$, $n = 0, 1, ..., 2M-1$ and $h(n)$ is the low-pass prototype (also referred to as window). Fig. 3.3 shows the magnitude frequency response of an MLT generated by modulating a half-sine window.

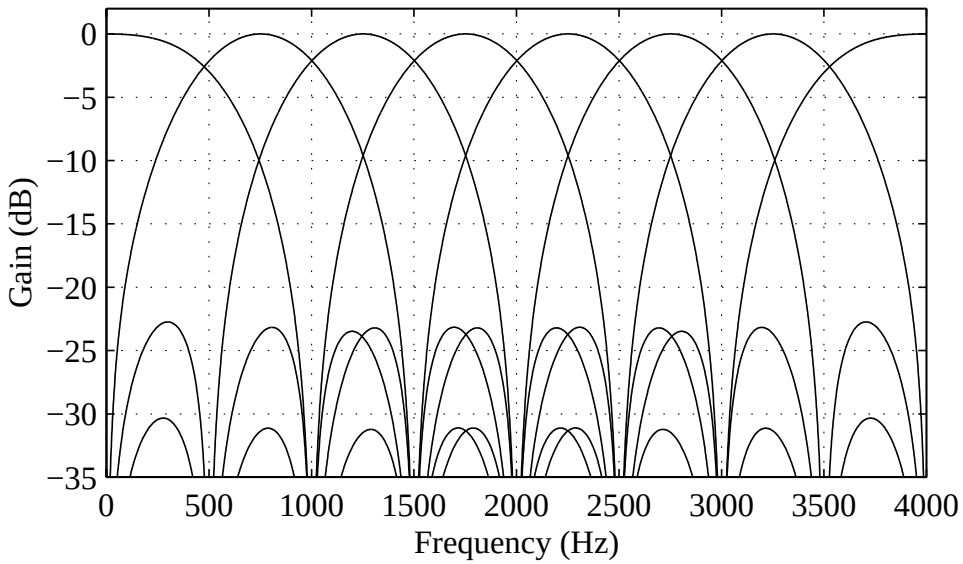


Fig. 3.3 Magnitude frequency response of a modulated lapped transform ($M = 8$).

3.3.1 Perfect Reconstruction Conditions for an MDCT

We analyze an MDCT when two different windows are used to generate the analysis and synthesis filterbanks. Then we obtain the perfect reconstruction conditions for the specific case when only one window is used for the analysis and synthesis filterbanks.

The MDCT of a block of the input signal $x(n)$ is given by

$$X(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{N-1} h(n) \cos\Big((n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M} \Big) x(n), \tag{3.43}$$

where $h(n)$ is the analysis window, M is the number of subbands and $k = 0, 1, ..., M-1$. Note that the length of each frame of data is $N = 2M$. The inverse MDCT is given by

$$y(n) = g(n) \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} X(k) \cos\Big((n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M} \Big), \tag{3.44}$$

where $g(n)$ is the synthesis window. Substitute Eq. 3.43 into Eq. 3.44 to obtain

$$y(n) = \frac{2}{M} g(n) \sum_{k=0}^{M-1} \sum_{m=0}^{N-1} h(m)x(m) \cos\Big((m+\frac{M+1}{2})(k+\frac{1}{2})\frac{\pi}{M} \Big) \cos\Big((n+\frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M} \Big), \tag{3.45}$$

$$y(n) = \frac{1}{M} g(n) \sum_{m=0}^{N-1} h(m)x(m) \sum_{k=0}^{M-1} \Big(\cos\Big((m-n)(k+\frac{1}{2})\frac{\pi}{M} \Big) + \cos\Big((m+n+M+1)(k+\frac{1}{2})\frac{\pi}{M} \Big) \Big) \tag{3.46}$$

$$\begin{aligned} y(n) = {} & \frac{1}{M} g(n) \sum_{m=0}^{N-1} h(m)x(m) \sum_{k=0}^{M-1} (\cos\Big((m-n)(k+\frac{1}{2})\frac{\pi}{M} \Big) \\ & + \frac{1}{M} g(n) \sum_{m=0}^{N-1} h(m)x(m) \sum_{k=0}^{M-1} \cos\Big((m+n+M+1)(k+\frac{1}{2})\frac{\pi}{M} \Big). \end{aligned} \tag{3.47}$$

Now we consider two different cases; for $0 \le n \le M-1$ (the first half of $y(n)$), $y(n)$ is zero except for $m = n$ and $m = M-1-n$. We can easily show that for this case, $y(n)$ becomes

$$y(n) = g(n)h(n)x(n) - g(n)h(M-1-n)x(M-1-n), \qquad n = 0, ..., M-1. \tag{3.48}$$

For $M \leq n \leq 2M-1$ (second half of $y(n)$), $y(n)$ is zero except for $m = n$ and $m = 3M-1-n$; hence

$$y(n) = g(n)h(n)x(n) + g(n)h(3M-1-n)x(3M-1-n), \qquad n = M, ..., 2M-1. \tag{3.49}$$

In Fig. 3.1 the desirable segment ($\hat{\boldsymbol{x}}_r$) of the reconstructed signal is given by

$$\hat{\boldsymbol{x}}_r = \boldsymbol{y_a}^{(2)} + \boldsymbol{y_b}^{(1)}. \tag{3.50}$$

Since $\boldsymbol{y_a}$ and $\boldsymbol{y_b}$ have different time references, we take the beginning of $\boldsymbol{y_b}$ as the time reference. Therefore

$$\hat{\boldsymbol{x}}_r = \boldsymbol{y}^{(1)}(n+M) + \boldsymbol{y}^{(2)}(n), \qquad n = 0, ..., M-1, \tag{3.51}$$

$$\begin{aligned}\hat{\boldsymbol{x}}_r =& g(n+M)h(n+M)\boldsymbol{x}^{(1)}(n+M) + g(n+M)h(2M-1-n)\boldsymbol{x}^{(1)}(2M-1-n) \\ &+ g(n)h(n)\boldsymbol{x}^{(2)}(n) - g(n)h(M-1-n)\boldsymbol{x}^{(2)}(M-1-n).\end{aligned} \tag{3.52}$$

We have

$$\hat{\boldsymbol{x}}_r = \boldsymbol{x}^{(1)}(n+M) = \boldsymbol{x}^{(2)}(n), \qquad n = 0, ..., M-1, \tag{3.53}$$

and also

$$\boldsymbol{x}^{(1)}(2M-1-n) = \boldsymbol{x}^{(2)}(M-1-n), \qquad n = 0, ..., M-1. \tag{3.54}$$

Therefore in order to achieve perfect reconstruction, the following conditions must be satisfied

$$\begin{aligned}&h(n)g(n) + h(n+M)g(n+M) = 1, \\ &g(n)h(M-1-n) - g(n+M)h(2M-1-n) = 0.\end{aligned} \tag{3.55}$$

If we use the same window for the analysis and the synthesis stages, the transform is called *Modulated Lapped Orthogonal Transform (MLOT)*. For this case if we use a symmetrical

window $h(n)$, the perfect reconstruction conditions become

$$\begin{aligned} h^2(n) + h^2(n+M) &= 1, \\ h(n) &= h(N-1-n). \end{aligned} \tag{3.56}$$

Comment on the Output of the Synthesis Filterbank

As we saw earlier, the output of the synthesis filterbank $y(n)$ is given by

$$\begin{aligned} y(n) &= g(n)h(n)x(n) - g(n)h(M-1-n)x(M-1-n), \qquad n = 0, ..., M-1, \\ y(n) &= g(n)h(n)x(n) + g(n)h(3M-1-n)x(3M-1-n), \qquad n = M, ..., 2M-1. \end{aligned} \tag{3.57}$$

Since $y(n)$ is a combination of $x(n)$ and the time reversed version of $x(n)$, it is not possible to recover $x(n)$ from $y(n)$. It is obvious that when we map a $2M$ dimensional input signal into an M dimensional transform vector, we lose some information. However, the elegance of a lapped transform is that we can restore the lost information by adding the overlapping parts of the successive output vectors of the synthesis filterbank. A segment of the output signal equals the sum of the first part of the current output and second half of the previous output of the synthesis filterbank. As we see in Eq. 3.57, the first half of the current output is $g(n)h(n)x(n)$ **minus** the time reversed version of the first half of the corresponding block of the input signal, i.e., $x(M-1-n)$, multiplied by $g(n)h(M-1-n)$. On the other hand the second half of the previous output vector equals $g(n)h(n)x(n)$ **plus** the time reversed version of the second half of the corresponding block of the input signal, i.e., $x(M-1-n)$, multiplied by $g(n)h(M-1-n)$. Since there is 50% overlap between the input blocks of data, the first half of the current block of the input signal is exactly the same as the second half of the previous block of the input signal. Therefore in constructing the output signal (via an overlap-add method), it is possible to cancel the time reversed terms by using appropriate analysis and synthesis windows.

As an example, Fig. 3.4 shows the process of constructing a segment of the output signal for a flat input $x(n) = 1$.

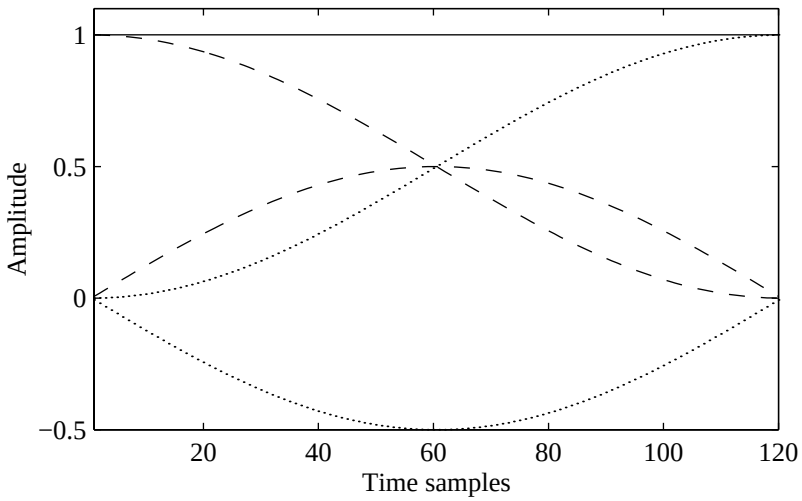


Fig. 3.4 Dashed curves: the first and second terms of the second half of the synthesis filterbank output due to the first block of data, Dotted curves: the first and second terms of the first half of the synthesis filterbank output due to the second block of data, Solid line: the segment of the reconstructed signal.

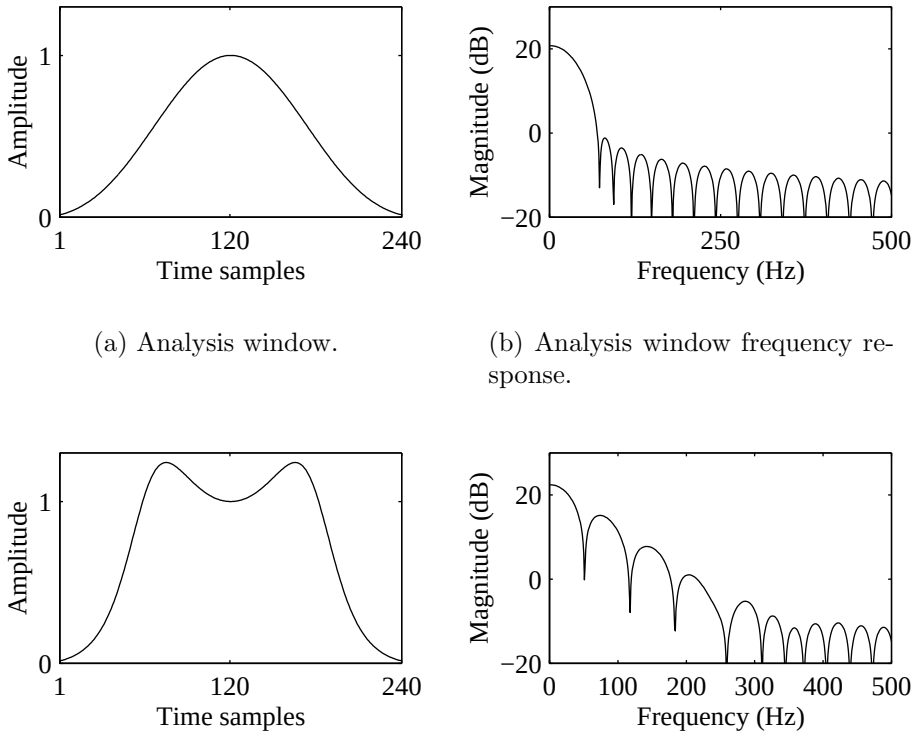
3.3.2 Orthogonal versus Biorthogonal Modulated Lapped Transforms

A modulated filterbank is generated by modulating a single prototype lowpass filter in case of an Lapped Orthogonal Transform (LOT) or two prototype lowpass filters for the analysis and synthesis stages in case of a Lapped Biorthogonal Transform (LBT) [69, 70, 71].

In order to make a choice between the two options, we have to consider the functions of the analysis and synthesis filterbanks. The analysis filterbank is required to decompose the input signal and delivers approximately uncorrelated transform coefficients. Moreover, the energy of the input signal should be compacted into a few transform coefficients. These requirements imply that the analysis filterbank should approximate an ideal filterbank. On the other hand the synthesis window should *smoothly* go to a small value at the boundaries in order to reduce block edge effects. Moreover the synthesis filterbank (generated using the synthesis window) should suppress or attenuate the out-of-band quantization noise which requires a good filtering performance. Therefore in designing the analysis and synthesis windows we face conflicting requirements.

We can make a compromise and choose an identical window for both analysis and synthesis (that is the orthogonal case). If the emphasis is only on obtaining a high coding gain or less block edge effects (like in image coding) then we employ a biorthogonal lapped

transform in which two different windows are used. One window is optimized at the expense of the other window characteristics. The perfect reconstruction conditions (see Eq. 3.56) on an N point window for an orthogonal transform leaves $N/4$ degrees of freedom in designing the prototype window. However, if two separate windows are used for the analysis and synthesis filterbanks, we have $N/2$ degrees of freedom to design a symmetrical analysis window. This results in an analysis window with a better frequency response. Nevertheless, in order to have a perfect reconstruction analysis/synthesis system, the synthesis window will be found using Eq. 3.55. Figure 3.5 shows the analysis and synthesis windows and their frequency responses of a lapped biorthogonal transform. As we see, the synthesis window



(a) Analysis window.

(b) Analysis window frequency response.

(c) Synthesis window.

(d) Synthesis window frequency response.

Fig. 3.5 Analysis and synthesis windows for a lapped biorthogonal transform.

does not smoothly approach zero which is a desirable characteristic of the synthesis window in order to reduce block edge effects. Moreover, it is obvious that a better analysis window

leads to a worse synthesis filterbank.

3.3.3 Windows for Modulated Lapped Orthogonal Transforms

Any window which satisfies the perfect reconstruction conditions can be used to generate the filter bank. However, to obtain a high coding gain, the frequency response of the window should approximate an ideal lowpass filter.

For an ideal M-band filterbank, the bandwidth of each bandpass filter should be $\frac{f_s}{2M}$, where f_s is the sampling frequency. Since the filterbank is generated by modulating a prototype lowpass filter (which is the frequency response of the window), the bandwidth of the prototype lowpass filter should be $\frac{f_s}{4M}$.

For a fixed length FIR filter, we have to trade the width of the main lobe versus the stopband attenuation. In some coding schemes such as AC-2 and AC-3 [72] and the AAC [52] a Kaiser-Bessel-Derived (KBD) window is used. This window is defined as follows

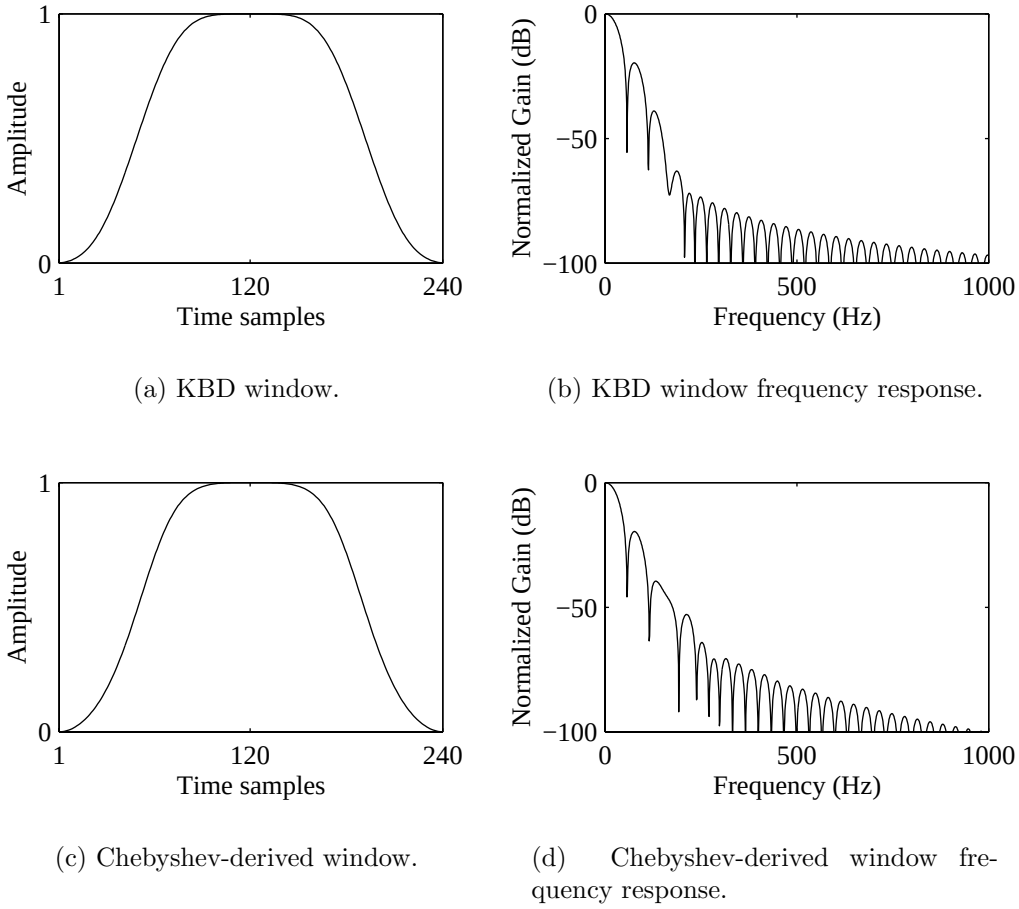
$$h_{\mathrm{KBD}}(n) = \sqrt{\frac{\sum_{i=0}^{n} \mathcal{W}(i)}{\sum_{i=0}^{N-1} \mathcal{W}(i)}} \quad , \qquad n = 0, ..., N/2 - 1, \tag{3.58}$$

where N is the length of the window and $\mathcal{W}(i)$ is the Kaiser-Bessel kernel window function defined as follows

$$\mathcal{W}(i) = \frac{I_0\left(\pi\nu\left(1 - (\frac{i-N/4}{N/4})^2\right)\right)}{I_0(\pi\nu)}, \tag{3.59}$$

where I_0 is the modified zero order Bessel function of the first kind and ν is the parameter of the window. Figure 3.6 shows the KBD window and its frequency characteristic for $\nu = 6$. As we can see the KBD window shows a very good stopband attenuation at the cost of a larger transition band.

We can design windows similar to the KBD window with similar stopband attenuation. A family of windows derived from the Chebyshev polynomial (type 1) is presented in Appendix B. The Chebyshev-derived window has two parameters by which we can adjust the window shape and the frequency response of the resulting prototype lowpass filter. Fig. 3.6 shows the Chebyshev-derived window and its frequency response.



(a) KBD window.

(b) KBD window frequency response.

(c) Chebyshev-derived window.

(d) Chebyshev-derived window frequency response.

Fig. 3.6 KBD window with parameter 4 and a Chebyshev-derived window with parameters (2,1.3) and the frequency responses.

Optimization Procedure for Window Design

In designing a window for a modulated transform, we have to compromise between the main lobe bandwidth, transition bandwidth and the stopband attenuation. In order to make a trade-off between the selectivity and the stopband rejection of the filter bank, we consider the frequency selectivity of the hearing system. Therefore the transition bandwidth can be increased in favor of a higher stopband rejection. For instance, in our audio coder the number of the transform coefficients for each block of input data is 120. Hence, the bandwidth for each frequency bin of an ideal filterbank should be $4000/120 \approx 33.3$ Hz. Since the narrowest critical bandwidth is 100 Hz, we do not need to have a frequency resolution better than the hearing system. This fact gives us some freedom to design the prototype lowpass filter. Note that the window is the impulse response of the prototype lowpass filter. Therefore we use an optimal window and an optimal prototype lowpass filter interchangeably.

We take a combination of time and frequency constraints to optimize the window. We try to design the prototype lowpass filter whose frequency response in the passband and stopband approximates the frequency response of an ideal lowpass filter. We also consider a transition band for the lowpass filter.

The optimization procedure which is similar to [73] is performed as follows

$$\begin{aligned} & h(n) = \operatorname{argmin} \sum_{k=0}^{\frac{N_F}{2}+1} W(k)(H_{\text{ideal}}(k) - H(k))^2 \\ & \text{subject to} \\ & h(2M-1-n) = h(n) \\ & h^2(n) + h^2(n+M) = 1, \end{aligned} \tag{3.60}$$

where N_F is the Fourier transform length, H is the normalized DFT of the window $h(n)$, and M is the number of transform coefficients of the MDCT, that is half the length of the window. H_{ideal} is the DFT of the ideal lowpass filter defined as follows

$$H_{\text{ideal}}(k) = \begin{cases} 1, & 0 \le k < k_p \\ 0, & k_p \le k, \end{cases} \tag{3.61}$$

where k_p is the edge of the transition band. For an N_F point DFT and M MDCT co-

efficients, each ideal bandpass filter is represented by $\frac{N_F}{2M}$ points of the DFT. Since the prototype lowpass filter generates the filterbank, its bandwidth is half the bandwidth of each bandpass filter. Therefore the passband of the ideal lowpass filter is represented approximately by $\frac{N_F}{4M} + 1$ points, meaning that $k_p \approx \frac{N_F}{4M} + 1$.

W is a weighting function which gives different weights to the passband, transition band and the stop band. Note that we give more weight to the stopband to reduce the leakage between bands. W is defined by

$$W(k) = \begin{cases} 1, & 0 \leq k < k_p \\ 0, & k_p \leq k < k_s \\ 100, & k_s \leq k, \end{cases} \tag{3.62}$$

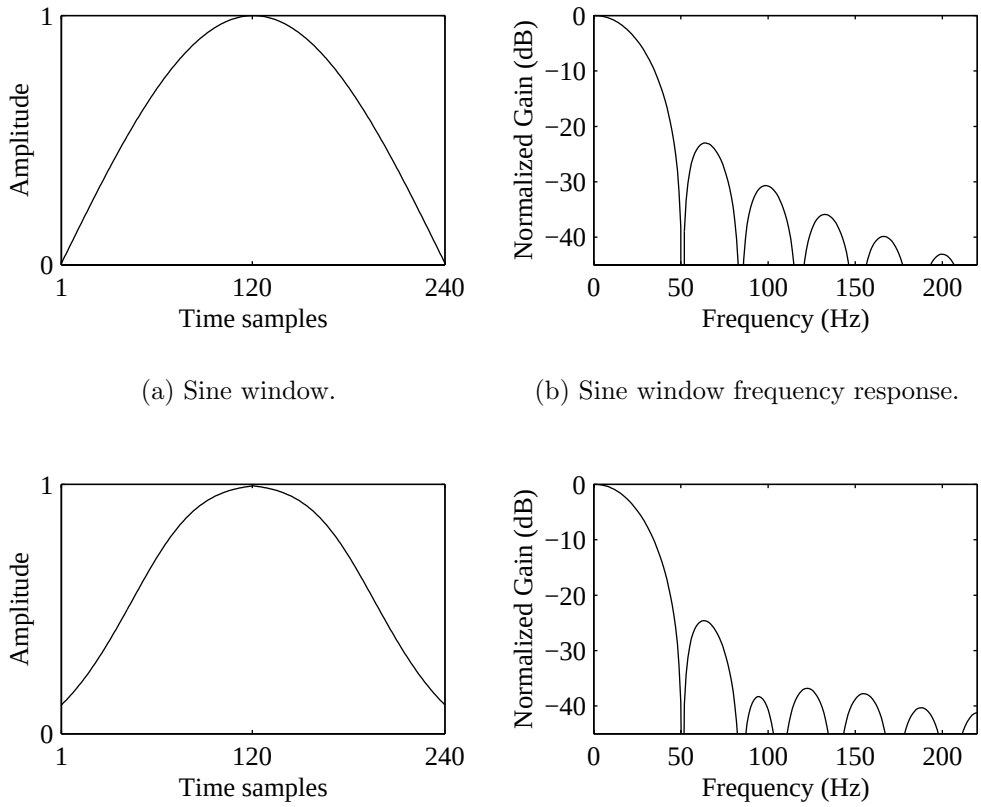
where k_s is the edge of the stop band. In order to set a value for k_s, we refer to our discussion above. We assume that the width of the transition band can be larger than a critical band. Since the critical bandwidths are frequency dependent, we take a value of 200 Hz for the transition band. For a sampling rate of 8000 Hz and an N_F point DFT, the transition width becomes $\frac{N_F}{40}$ and therefore in Eq. 3.62, k_s is set to $k_p + \frac{N_F}{40}$.

We compare a window designed using the optimization procedure and a sine window which is widely used in audio coding. Fig. 3.7 shows the sine window and the designed window. As seen in the Figure, the designed window shows a smoother transition at the boundaries. Moreover, the stopband attenuation of the designed window is higher.

3.3.4 Coding Performance of Transform Coding

In transform coding, instead of quantizing the samples of the signal with a desired number of bits per sample (which is referred to as PCM), we perform the quantization on the transform coefficients. It is well known that a lower mean-square error will result from quantizing the transform coefficients [3]. Assuming scalar quantizers, the reduction in transform coding mean square error over PCM is given by [74]

$$G_{TC} = \frac{\frac{1}{M}\sum_{k=0}^{M-1} \sigma_k^2}{\sqrt[M]{\prod_{k=0}^{M-1} \sigma_k^2}}, \tag{3.63}$$



(a) Sine window.

(b) Sine window frequency response.

(c) Designed window.

(d) Designed window frequency response.

Fig. 3.7 Comparison of the sine window and the designed window.

which is referred to as the *transform coding gain*; (σ_k^2 is the variance of the kth transform coefficient). The denominator is the geometric mean of the transform coefficient variances which is minimized by the KLT. Therefore, the KLT is the optimal transform for transform coding. By using lapped transforms, we can achieve higher coding gains than that of the KLT [61].

Table 3.1 shows the coding gain of the MLT (using different windows: KBD, Chebyshev-derived, sine, rectangular[4] and designed window), DCT and DFT. As we can see on the Table the difference between the coding gain using different smooth windows is not remarkable but there is a big drop in the coding gain using a rectangular window. For all signals

[4]A rectangular window with length $2M$ is one over the middle M points and zero for the rest.

the coding gain of the MLT using a smooth window is greater than that of the DCT.

Table 3.1 Coding gain in dB of an MLT using different windows, DCT and DFT for female speech, male speech and classical guitar.

Transform	Female speech	Male speech	Classical guitar
MLT, KBD window	18.76	14.70	14.00
MLT, Chebyshev_derived window	18.79	14.71	14.01
MLT, Designed window	18.75	14.63	14.00
MLT, sine window	18.48	14.58	14.07
MLT, rectangular window	10.87	10.57	12.81
DCT	15.10	13.30	14.00
DFT	11.15	10.62	11.04

3.4 Multiresolution Filterbanks

Since audio signals are analyzed by the hearing system on a critical band scale, a nonuniform filterbank with frequency division nearly matched to the critical bands seems preferable over a uniform filterbank. Various multiresolution structures such as nonuniform filterbanks [75, 76, 77, 78, 79] and wavelets[5] [80, 81, 82, 83, 84, 85] have been proposed for audio coding. However as Ferreira in [86] argues the basic assumption in using a multiresolution filterbank is that the high frequency spectral components have a short duration while the low frequency components have long durations. This assumption does not reflect the reality as there is no evidence to support those assumptions. In fact, for steady state parts of the input signal, we need to have a high frequency resolution at all frequencies to achieve a high coding gain. Moreover, some psychoacoustic evidence suggests that the hearing system resolves the spectral components inside a critical band at higher levels than the inner ear [86]. For instance timbre, which characterises a sound, is related to the relative amplitude of certain spectral components regardless of the critical-band scale.

Since audio signals have time-varying characteristics, there is no optimal transform to decompose the signal. In fact for pseudo-stationary parts, a filterbank with a high frequency resolution is needed whereas for transients a multiresolution decomposition would be preferable. For most audio signals, a short term stationarity assumption is valid except

[5]Wavelets are a set of basis functions generated by shifting (in time) and scaling a single prototype function.

for a small fraction of the duration of the audio signal [86]. Therefore, a uniform filterbank with a high frequency resolution is a better choice for the decomposition of an audio signal. However the temporal resolution of the filterbank should be increased when high energy attacks are detected. This requires using a set of filterbanks to be chosen for different situations. We will discuss this issue later.

Despite all the work on wavelet-based audio coding, it seems that a transform coder (using a uniform filterbank) would deliver better quality at low rates [87]. Johnston believes that a high temporal resolution at high frequencies is not needed all the time in order to achieve high quality [87]. Malvar in [61] states that for speech we need a high frequency resolution not only at low frequencies but also at middle frequencies in order to resolve the formant structures. That is the reason the performance of wavelet-based speech coders at low rates is not satisfactory [61]. Another problem with wavelet-based coders is a large algorithmic delay compared to transform-based coders [84].

3.4.1 Adaptive Filterbanks

One of the desirable characteristics of a filterbank is to have a high temporal resolution. As a matter of fact, for high energy transient parts of the input signal, it is desired to localize a short burst of quantization noise to prevent it from spreading over a long period of time.

Some works have been published on adaptive filterbanks to handle this problem [88, 89, 90, 91]. In some coding schemes the temporal resolution of the filterbank is increased by switching to a short window [92, 15, 72, 93].

In a window switching scheme, a suitable window is selected from a set of windows to generate the filterbank. The switching criterion is based on the energy [35, 82] or perceptual entropy [4]. As an alternative to window switching schemes, Herre and Johnston [94] use Temporal Noise Shaping (*TNS*) to continuously adapt the temporal and frequency resolution of the filterbank.

3.4.2 Perfect Reconstruction Conditions in a Window Switching Scheme

In order to handle the attacks, a short window used to generate the filterbank. We have to make sure that the perfect reconstruction property of the overall system (in the absence of quantization) is preserved. A start window is used to switch from a long window to a short window and stop window is used to switch back. The start window is defined as follows

$$h_{\text{start}}(n) = \begin{cases} h_{\text{long}}(n), & 0 \le n \le M-1 \\ 1, & M \le n \le M + \frac{M}{3} - 1 \\ h_{\text{short}}(n-M), & M + \frac{M}{3} \le n \le M + \frac{2M}{3} - 1 \\ 0, & M + \frac{2M}{3} \le n \le 2M - 1. \end{cases} \tag{3.64}$$

Fig. 3.8 shows a transition from a long window to a short window through a start window.

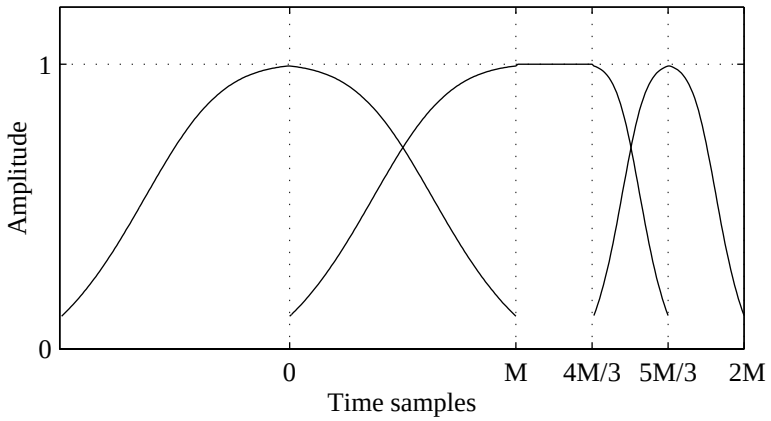


Fig. 3.8 A transition from a long window to a short window via a start window.

Based on Eq. 3.43 and Eq. 3.44, the output of the synthesis filter bank is given by

$$\begin{aligned} y(n) = &\frac{1}{M} h_{\text{start}}(n) \sum_{m=0}^{N-1} h_{\text{start}}(m) x(m) \sum_{k=0}^{M-1} \cos\Big((m-n)(k+\frac{1}{2})\frac{\pi}{M}\Big) \\ &+ \frac{1}{M} h_{\text{start}}(n) \sum_{m=0}^{N-1} h_{\text{start}}(m) x(m) \sum_{k=0}^{M-1} \cos\Big((m+n+M+1)(k+\frac{1}{2})\frac{\pi}{M}\Big). \end{aligned} \tag{3.65}$$

Note that we use h_{start} for both the analysis and synthesis filterbanks.

We find different segments of $y(n)$ as follows. For $0 \le n \le M-1$, the output becomes

$$y(n) = h^2{}_{\text{long}}(n)x(n) - h_{\text{long}}(n)h_{\text{long}}(M-1-n)x(M-1-n), \qquad n = 0, ..., M-1. \tag{3.66}$$

We have used the fact that $h_{\text{long}}(n) = h_{\text{start}}(n)$ for $n = 0, ..., M-1$. As we have seen before, $y(n)$ is a linear combination of the input signal and its time reversed version. In constructing the output signal, the second term of Eq. 3.67 will be cancelled by the time

reversed term of the output of the synthesis filterbank due to the previous block of data. Remember that in a lapped transform, the first half of the current output of the synthesis filterbank contains the same terms as the second half of the synthesis filterbank output with the difference that the time reversed term has different signs. Therefore after adding the successive outputs of the synthesis filterbank, those terms cancel each other which means a perfect construction of the output signal.

For $M \leq n \leq M + \frac{M}{3} - 1$, $y(n)$ equals zero except when $n = m$. For this range of time samples, $h_{\text{start}}(n) = 1$. Therefore the output signal is given by

$$y(n) = x(n). \tag{3.67}$$

For $M + \frac{M}{3} \leq n \leq M + \frac{2M}{3} - 1$, it is given by

$$y(n) = h^2{}_{\text{start}}(n)x(n) - h_{\text{start}}(n)h_{\text{start}}(3M - 1 - n)x(3M - 1 - n). \tag{3.68}$$

Since for this range of time samples $h_{\text{start}}(n) = h_{\text{short}}(n - M)$ (that is the second half of the short window), we get

$$y(n) = h^2{}_{\text{short}}(n - M)x(n) - h_{\text{short}}(n - M)h_{\text{short}}(2M - 1 - n)x(3M - 1 - n). \tag{3.69}$$

We realize that the above equation is a linear combination of the input signal and the time reversed version of the input signal in that specific range of n. When we construct the output signal, the time reversed term will be cancelled by the time reversed term (with an opposite sign) due to the next (short) frame. Therefore the perfect reconstruction property of the system for this segment is also achieved. For the last segment, $n = M + \frac{2M}{3} \leq n \leq 2M - 1$, since $h(n) = 0$, the output of the synthesis filterbank is zero and the output signal is constructed by the overlapping parts of two successive outputs of the synthesis filterbank due to two consecutive short frames. We can easily show that the perfect reconstruction conditions are satisfied over a transition from a short window.

Chapter 4

Audio Compression Structures

This chapter is organized in two parts. In the first part of this chapter, we briefly describe different quantization techniques. An overview of some widely used audio coders and the MPEG audio standards will be presented in the second part of this chapter.

4.1 Quantization

In the quantization block, the spectral components are represented with a given number of bits. The goal is to achieve the best possible quality of the reconstructed signal after quantization. In the process of quantization some information is lost, meaning that that is a *lossy compression* method. However, in audio coding new terms *perceptually lossless coding* or *transparent coding* have been used in the literature. A lossy audio coding scheme can be perceptually lossless if the human ear cannot distinguish between the original and compressed signal.

In some compression systems, a *lossless compression* step may follow the quantization block in order to further reduce the data rate. In lossless compression schemes (also known as noiseless or entropy coding), the original data can be perfectly reconstructed. In order to reduce data rate, the more probable symbols are coded into short binary words and vice-versa [95, 3]. This way the average data rate is reduced. This is fundamentally a variable rate scheme. Conversion to a fixed rate requires sufficient buffering to get a reduced average rate. A number of lossless coding schemes have been used in audio coding such as Huffman codes, run-length codes and arithmetic codes. A typical compression ratio for lossless coding of audio is 2:1.

In the following we will describe two major quantization schemes which are used in audio coding, i.e., *Scalar Quantization (SQ)* and *Vector Quantization (VQ)*.

4.1.1 Scalar Quantization

A scalar quantizer operates on individual values. It divides the range of the input values into N intervals (*cells* or *Voronoi* regions). Each cell is represented by a single value (decision level). It takes a single value as the input and selects the best match to that value from a predetermined set of values (*codebook*). The process of scalar quantization can be modelled as a nonlinear operation in which a range of input values is represented by a single value from the codebook.

In transform coding systems using scalar quantization, the transform coefficients are quantized independently by a set of scalar quantizers and then transmitted to the receiver. If the input waveform is strongly correlated, high energy compactness (into a few coefficients) is obtained after the transform and a significant coding gain over PCM may be achieved by using optimal bit allocation to the scalar quantizers [28]. Scalar quantizers are divided into different classes which we briefly discuss as follows.

Uniform Quantization

In this scheme, all the cells have the same size. The codewords are equally spaced and lie at the middle of the cells. The distance between two successive decision levels (*step size*) is defined as

$$s_q = \frac{x_{\max} - x_{\min}}{N}, \tag{4.1}$$

where s_q is the step size, $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the input and N is the number of quantization levels. This quantization scheme is matched to uniform probability distribution functions.

Nonuniform Scalar Quantization

For inputs with a nonuniform probability distribution a quantizer with unequally-spaced decision levels reduces the MSE for a fixed number of step sizes. In general, for an arbitrary probability density function, the decision levels and cells are found by minimizing the total

distortion given by [3]

$$D = \sum_{i=1}^{N} \int_{R_i} d(x, Q_i(x)) p_{\mathbf{x}}(x) dx, \tag{4.2}$$

where $p_{\mathbf{x}}$ is the probability density function of the input values, $Q_i(.)$ is the ith quantization level, R_i denotes the ith partition (cell) and $d(.,.)$ is the distance (distortion) measure. In most cases there is no closed solution to this optimization problem. Instead some other iterative algorithms such as the Lloyd algorithm [96] are used to design the quantizer.

Some popular schemes of nonuniform quantization are μ-law and A-law methods which are used to quantize speech signals [74]. As an example, the following power law nonuniform quantization scheme has been used in MPEG-1 Layer 3 and MPEG-2 Advanced Audio Coding (AAC) [52, 55]

$$\hat{X}(i) = \text{nint}\left(\left(\frac{|X(i)|}{s_q}\right)^{0.75} - 0.0946\right), \tag{4.3}$$

where $X(i)$ and $\hat{X}(i)$ are the i-th transform coefficient and its quantized value, nint(.) denotes the nearest integer value and s_q is the quantizer step size. This quantizer roughly quantizes big values compared to finer quantization of small values.

4.1.2 Vector Quantization (VQ)

A vector quantizer operates on a set of values and gives out an index to the vector in a lookup table (codebook) which gives the least distortion based on some error criterion. According to Shannon for a fixed number of bits, coding longer blocks of data results in a lower average distortion [3, 97]. This better performance comes from the fact that VQ exploits any correlation among the vector components. Vector Quantization shows a performance advantage over scalar quantization at rates below 1 bit per sample [98]. The disadvantage of VQ methods is the amount of memory required to store the codebooks. Additionally, computation power is needed to search for the best codeword from a large codebook.

The complexity of vector quantization can be reduced by using different schemes such as Gain/Shape separation, multistage and split VQ. In a Gain/Shape approach, the normalized input vector (*shape vector*) is quantized using a vector quantizer and the gain is encoded separately. This technique is widely used and allows for using VQ at a reasonable

complexity.

In a split VQ approach, large vectors are broken into smaller ones and then a VQ system is designed for each subvector. This way the complexity is reduced at the expense of possibly a higher distortion. Split VQ techniques are the most efficient scheme (in the sense of distortion-rate) if used with an adaptive bit allocation scheme in which the available bits are allocated to each subvector based on the local statistics. The adaptive VQ coding gain demonstrates a significant advantage of VQ over scalar quantization in transform coding [99, 100, 101]. In our proposed coder, we use an adaptive VQ scheme along with a perceptually based bit allocation strategy.

Another way to reduce the complexity is to use a multistage VQ structure. In that method the input vector is fed to the first VQ and then the difference between the input vector and the selected codeword is used as the input to the second VQ. The quantized version of the input vector will be the sum of the codewords selected from the first and second codebooks. If there are more than two VQ stages, at each following stage the residual vector (difference between the original and quantized vector) is quantized using a VQ and the selected codeword will be added to the quantized version of the original vector. This way at each stage we obtain a finer quantization of the original vector. In this method like the split-VQ scheme, we usually sacrifice the performance (to some extent) to reduce the complexity.

VQ Design

A vector quantization system consists of a few components, i.e., a lookup table (codebook) to represent the statistics of the vector source, a distortion measure, and a centroid computation procedure. In the following, we briefly discuss those components.

Designing the VQ codebooks (lookup tables), which are used to encode the input signal, is a major part of the (off-line) computational effort. To create a codebook, a large set of vectors with characteristics similar to the source is used to create (train or populate) a codebook. The size of the training set should be large enough to closely represent the input source. The number of training vectors to the number of the codewords should be at least 10 times and more preferably 50 times [98] the number of the codevectors (codewords) in the codebook.

A distortion measure (distance measure or quantization rule) is needed to train the

codebook and to select the best match from the codebook to any input vector. The commonly used distortion measure is the mean squared error (L_2 norm). Other distortion measures are used such as the L_1 norm, likelihood and cepstral distance measures. In all these distortion measures, the error is zero only if the input vector is equal to a codeword from the codebook. However in this thesis we introduce a perceptually based distortion criterion which measures the distance between an input vector and the codewords in a perceptual domain. It takes into consideration only that part of the quantization error which is perceptible to the ear. In this case the perceptual error can be zero for vectors which are not identical.

In creating the codebook of N codewords, the training vectors are clustered into N groups (cells) using the distortion measure. Then a centroid computation algorithm finds the vector which represents the vectors in each cell of the training set.

Iterative methods can be used to design a vector quantization system. As an example we describe the widely used the Generalized Lloyd Algorithm (GLA) [3] as follows.

Generalized Lloyd Algorithm (GLA)

The GLA uses a large set of the sample vectors of the input source and delivers a codebook with the desired size. It is an iterative method which starts with an initial codebook and refines the codebook until the final codebook is obtained.

First the training vectors are clustered around different codewords based on the distortion measure in which a partition (cell) is defined as

$$R_i = \{\mathbf{x} | \forall j; d(\mathbf{x}, \mathbf{c}_i) \leq d(\mathbf{x}, \mathbf{c}_j)\}, \tag{4.4}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the codewords representing the ith and jth cells respectively. In the next step the *centroid* for each cell is found as follows

$$\mathbf{c}_i = \arg\min_{\mathbf{c}} \sum_{\mathbf{x}_i \in R_i} d(\mathbf{x}_i, \mathbf{c}), \tag{4.5}$$

where $i = 1, ..., N$ and N is the number of partitions (cells). This iterative procedure continues until the average distortion (or the change in the average distortion) falls below a certain threshold.

Since the design of a VQ system is a multidimensional optimization problem, there is

a possibility that the codewords obtained may not be globally optimal [3]. Therefore, the initial codebook can have a great impact on the final codebook. Many methods have been proposed to mitigate this problem [3, 102]. One of the widely used methods is the LBG procedure [102] which starts with creating a codebook with only one codeword. Then the first codeword is split into two codewords to create the initial codebook to generate the second codebook. The iterative GLA method is used to find the final codebook at each step. This splitting and training continue until the final codebook is obtained.

Perceptually Trained VQ

In perceptual audio coding, we should take into consideration the limited capability of the hearing system to resolve different sounds. This leads us to define a perceptually-based distortion measure which counts only the audible part of the quantization noise. We incorporate the masking threshold in the distortion measure used while training the codebooks and selecting the best codewords.

We use a modified version of the LBG algorithm [102] with the following perceptually-based distortion measure based on the audible noise energy to design the codebooks [33]. The same error criterion is used to select the best codewords in encoding the input vectors.

For an input vector of spectral components X and the jth codeword $\chi^{(j)}$, the distortion defined by

$$\mathbf{d}(k) \triangleq |X(k) - \chi^{(j)}(k)|^2 - \mathbf{m}(k), \tag{4.6}$$

where $\mathbf{m}$ is the vector of masking thresholds corresponding to X. The energy of the audible noise is calculated by

$$D(X, \chi^{(j)}) = \sum_{k=1}^{K} \max(\mathbf{d}(k), 0), \tag{4.7}$$

where K is the dimension of X. The centroid of each Voronoi region is determined by minimizing the energy of the audible noise as follows

$$\chi_{\text{opt}}^{(j)} = \arg\min_{\chi^{(j)}} \sum_{i=1}^{I} D(X^{(i)}, \chi^{(j)}), \tag{4.8}$$

where I is the number of the vectors in region j.

4.2 Frequency Domain Audio Coders

Frequency domain coding is a popular approach to compressing audio data. The great advantage of frequency domain encoders is the ability to shape the quantization noise based on perceptual principles.

Figure 1.2 (repeated as Fig 4.1) shows a general block diagram of perceptual coders working in the frequency domain. The block diagram consists of the following basic blocks.

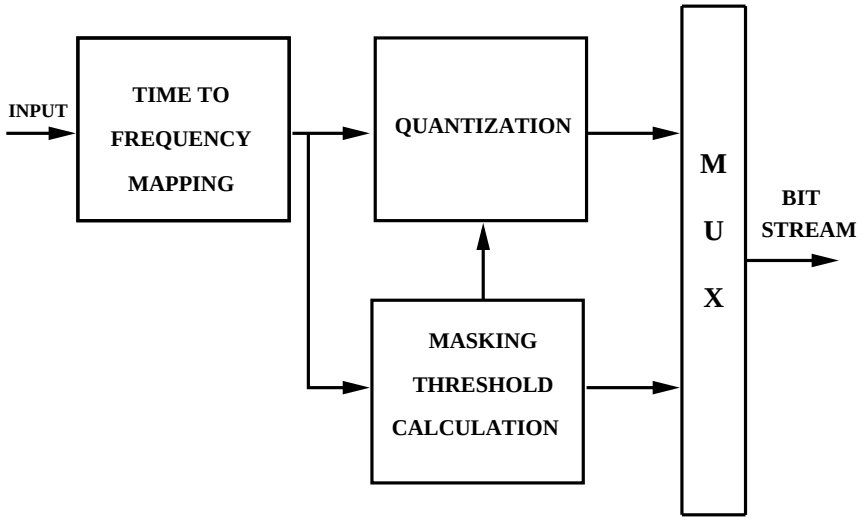


Fig. 4.1 General block diagram of a perceptual coder working in the frequency domain.

- A filterbank or transform is used to decompose the input signal into spectral components.
- The spectrum is used to calculate an estimate of the masking threshold.
- The transform coefficients are quantized and coded using the information about the masking threshold.
- In the last step, the quantized and coded transform coefficients are multiplexed with additional side information to produce a bit stream.

In the following, we briefly describe a number of widely used wideband audio coders which made great contributions to the field of audio coding. Note that almost all popular

audio coders have been designed to code wideband audio data with high quality. On the other hand the coder presented in this dissertation is meant to accommodate narrowband audio signals with acceptable quality. However, the basic structure of our narrowband audio coder is similar in many ways to that of wideband perceptual audio coders.

4.2.1 AT&T Perceptual Audio Coder (PAC)

The AT&T Perceptual Audio Coder (PAC) [103] accommodates monophonic and stereophonic wideband signals (20 Hz to 20 kHz). It also has the ability to handle multi-channel audio signals. The compression ratio is around 8:1 which implies 2 bits per sample. PAC was designed based on the DFT-based PXFM [51] and ASPEC [104] audio coders developed at AT&T. PXFM uses a 2048 point FFT with 1/16 overlap between successive frames of the input signal. The overlap increases the data rate which in return reduces the coding gain. ASPEC uses an MDCT to decompose the input signal. Since the MDCT is a critically sampled filterbank, it does not reduce the coding gain as was the case with PXFM. ASPEC also uses a window switching mechanism to switch a 1024-point window to a 256-point window to reduce pre-echo artifacts. The frequency resolution of ASPEC is half that of PXFM due to using a 1024-point window instead of a 2048-point window. PAC employs a 2048-point window to achieve a good frequency resolution and switches to a 256-point window to reduce the pre-echos. Compared to the previous AT&T audio coders, PAC has a number of new or enhanced features such as composite stereo coding, improved window switching, entropy coding and an improved masking threshold calculation, bit allocation algorithm, and buffer control. Figure 4.2 shows a block diagram of the monophonic version of PAC.

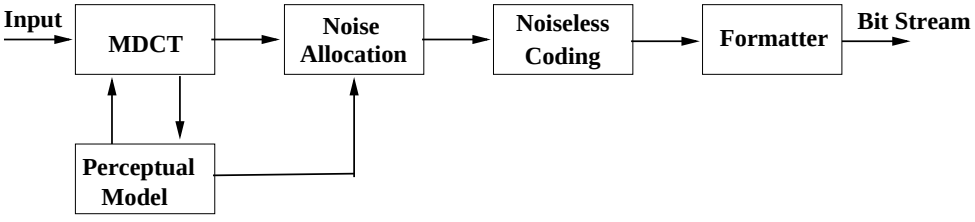


Fig. 4.2 Block diagram of the monophonic PAC encoder [103].

The MDCT filterbank takes in either 2048 or 256 time samples. The perceptual model calculates the masking threshold based on the time-domain signal and the output of the

filterbank. In the noise allocation block, the filterbank outputs are grouped into a small number of samples. Then based on the band masking threshold, a scalar quantizer is selected from a set of 121 quantizers. The noiseless block uses a Huffman codebook from a set of 8 codebooks to entropy code the coefficients in each band. The bit stream formatter generates the bit stream and encodes the whole set of information for transmission or storage. It operates from 32 kbit/s (single channel) up to a 1000 kbit/s (multi channel). Since PAC utilizes noiseless coding, a rate control module is used to adjust the bit rate by raising the global masking threshold for an undermasking situation.

4.2.2 Dolby AC-2 and AC-3 Audio Coders

The AC-2 and AC-3 audio coders [105, 106, 72, 107] were developed by Dolby and made a considerable contribution to the MPEG-2 AAC audio standard. Both AC-2 and AC-3 accommodates 20-kHz bandwidth audio signals. AC-2 operates at data rates of 128–192 kbit/s for monophonic inputs. The main focus of AC-2 is to code independent channels with low complexity and relatively low delay. AC-3 has been designed for single point to multipoint applications and supports 1 to 5 channels. AC-3 supports 32, 44.1 and 48 kHz sample rates and operates at at 32–640 kbit/s (overall bit rate). The AC-2 coding delay is 8-40 msec whereas the AC-3 coding delay is about 100 msec [86]. Figure 4.3 shows a basic block diagram of the Dolby AC-3 encoder. The AC-2/AC-3 coders are based on the

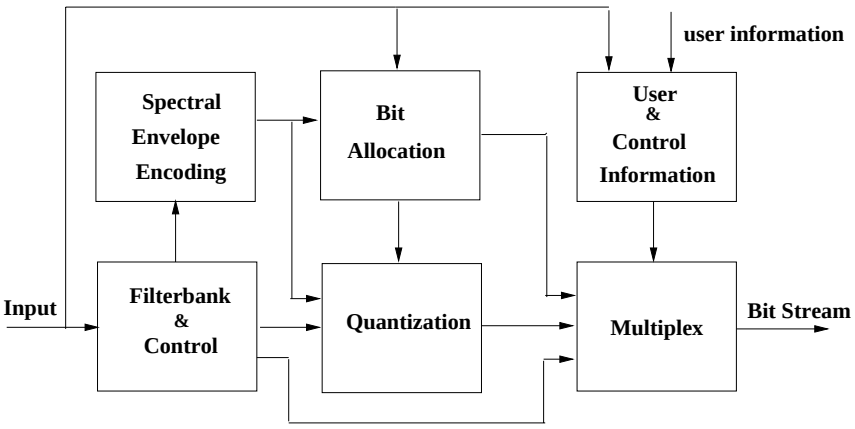


Fig. 4.3 Block diagram of the AC-3 encoder [72].

MDCT. The length of the window is 512 points and for handling the attack transients a window of 256 points is used. The coders use a Kaiser-Bessel-Derived (KBD) window in

order to have good stopband attenuation. For short blocks, only half of the long KBD window (512 points) is used and hence there is no overlap between the short windows [87].

The Noise-to-Mask ratio (NMR) is calculated for each MDCT coefficient. The MDCT coefficients are normalized by the spectral envelope and then scalar quantized. The step size of the scalar quantizer is determined by the corresponding NMR.

The AC-3 audio coder has been chosen as the audio system for the North America high definition television (HDTV) standard and the standard for digital versatile disc (DVD) [87]. AC-3 is also used in cable television and direct broadcast satellite [108].

4.2.3 Sony ATRAC Audio Coder

Adaptive Transform Acoustic Coding (ATRAC) was developed by Sony in the early 1990's for Sony's rewritable minidisc [109, 110]. The merit of this coding paradigm is its relatively simple structure which makes it suitable to be installed in portable low-cost products. The ATRAC encoder takes in a 44.1 kHz stereo audio input and compresses it by a factor of 5, while achieving transparent quality.

This coder uses a hybrid filterbank with a window adapted to the input signal, adaptive bit allocation and scalar quantization to code the input audio. The main difference of this coder from others is its filterbank. The time-to-frequency mapping has been designed by cascading two quadrature mirror filterbanks. The first filterbank splits the input into equal bands (0–11 kHz and 11–22 kHz). The second filterbank divides the lower band into equal bands, i.e., 0–5.5 kHz and 5.5–11 kHz. This time-to-frequency mapping puts more emphasis on the low frequencies which are perceptually more important. The three outputs of the hybrid filterbank are transformed into the frequency domain using three MDCT filterbanks. The MDCT coefficients are divided into groups and quantized using the masking threshold.

4.2.4 NTT Twin-VQ Audio Coder

Transform-domain Weighted Interleaved Vector Quantization (Twin-VQ) audio coder [111, 112, 113] is based on the MDCT. This coding scheme was a candidate for the MPEG-4 audio standard and adopted as one of the tools for MPEG-4 audio at bit rates down to 16 kbit/s. Figure 4.4 shows the structure of the Twin VQ coder. The input signal is transformed into the frequency domain using an MDCT. Window switching is used to reduce pre-echos. The quantization of the MDCT coefficients is done in two steps; first the coefficients are

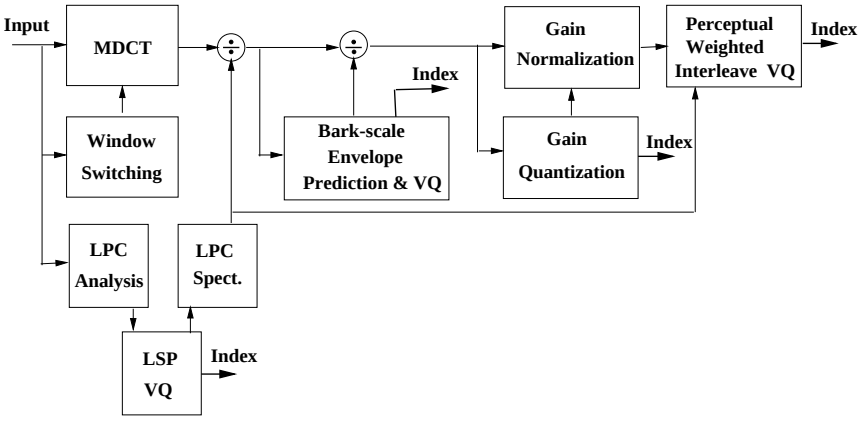


Fig. 4.4 Block diagram of the Twin VQ encoder [114].

flattened by the smooth spectrum which is calculated using quantized LSF coefficients. In the second step, a Bark-scale envelope, which is predicted from the previous frames using a moving average algorithm, is used to farther flatten the transform coefficients. Finally the flattened coefficients are normalized by the corresponding power. The Twin VQ coder employs a weighted interleaved VQ to quantize the normalized MDCT coefficients. In doing so, the processed transform coefficients are interleaved and split into subvectors. Each subvector is quantized by a VQ using an LPC weighted distortion measure. This quantization scheme is robust against channel errors as there is no adaptive bit allocation nor entropy coding used in the coder. An earlier version of the Twin VQ operates at less than 64 kbit/s. The recent version has a new module to extract the pitch from the input signal and operates at 16 and 8 kbit/s [115].

4.3 MPEG Audio Coding Standards

The Moving Picture Experts Group (MPEG), established in 1988, is a working group of the ISO/IEC (International Standards Organization/International Electrotechnical Commission) which produces international standards for compression, decompression, and processing of video and audio [116]. More specifically, MPEG standardizes the syntax of the bit streams and publishes a sample coder description [10].

There are three sets of MPEG audio coding algorithms: MPEG-1 Layer I/II/III, a multichannel extension of MPEG-1 which is referred to as MPEG-2 BC (backward compatible) coders, MPEG-2 AAC (Advanced Audio Coding) and MPEG-4 standard which incorpo-

rates MPEG-2 AAC as well as a CELP coder and a low rate vocoder. Currently some other multimedia standards are under development including MPEG-7 (a content representation standard for information search) and MPEG-21 which will define a multimedia framework to support the delivery of electronic content [10].

The standardization processes by the MPEG are done in different phases which include different Layers. The Layers present a family of coders which differ in complexity and coding efficiency. When going from one Layer to the next one the complexity increases while the maximum compression ratio goes up.

The MPEG standards have been used in many applications such as broadcasting, storage, multimedia and telecommunication, Digital Video Disc (DVD), Cable and Satellite TV, ISDN links, Computer based multimedia, and Internet Radio. In the following we briefly describe the audio part of the MPEG standards.

4.3.1 MPEG-1 Audio Coding Standard

MPEG-1 [15, 116, 117, 55] include 3 Layers, i.e., Layer I/II/III. The three Layers have been defined to be compatible in a hierarchical way, i.e., a decoder designed for a higher Layer is able to decode bit streams produced by a lower Layer encoder. The MPEG-1 audio standard deals with coding of mono or two-channel stereo audio inputs sampled at 32 kHz, 44.1 kHz and 48 kHz. The bit rate ranges from 32–448 kbit/s (Layer I), 32–364 kbit/s (Layer II) and 32–320 kbit/s (Layer III). A block diagram of the MPEG-1 Layer I/II audio standard is shown in Fig. 4.5.

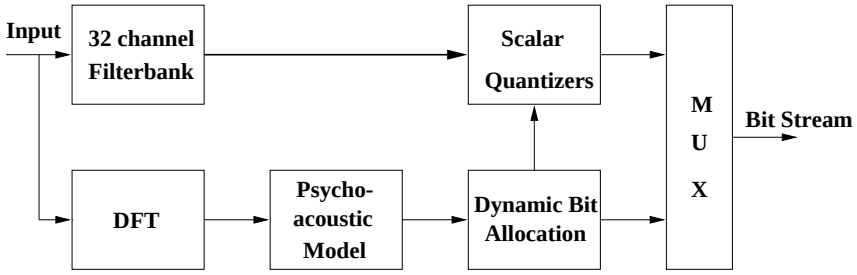


Fig. 4.5 Block diagram of the MPEG-1 Layer I and Layer II audio encoder [118].

The MPEG-1 audio standard is mainly based on three audio coders, i.e., PASC [119], MUSICAM [120] and ASPEC [104]. In MPEG-1 Layers I and II, the input signal is decomposed using a 32-channel filterbank. The masking threshold is calculated using a DFT of 512 points in Layer I or 1024 points in Layers II and III (The MPEG psychoacoustic models have been described in chapter 2.). In Layers I and II, for each subband a set of 12 (Layer I) or 36 (Layer II) consecutive samples are grouped. A scale factor is found for each group as the maximum absolute value of the samples. All samples in each group are normalized by the corresponding scale factor. The normalized coefficients are quantized using nonuniform scalar quantizers (the step size is determined based on the corresponding SMR calculated by the psychoacoustic model). MPEG-1 Layer III (known to the public as *MP3*) is different from the previous Layers in many ways. Figure 4.6 shows a block diagram of the MPEG-1 Layer III encoder. In order to increase the frequency resolution

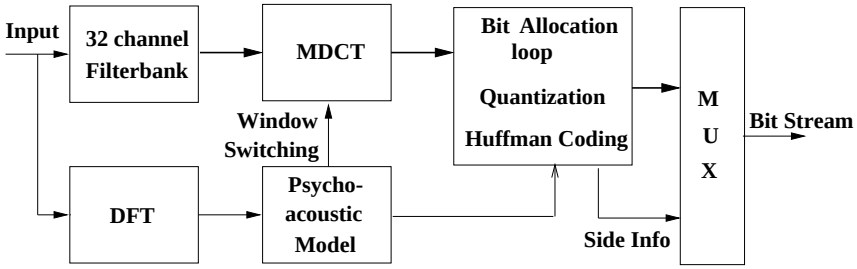


Fig. 4.6 Basic structure of the MPEG-1 Layer III audio encoder [118].

and the coding gain, a hybrid filterbank is used. In doing so, a 12-point or 36-point MDCT is used to decompose each subband signal. The selection of the length of the MDCT is made based on the perceptual entropy of the input signal. Although the hybrid filterbank increases the coding gain, it causes serious leakage between frequency bands. In Layer III, in addition to nonuniform quantizers, entropy coding is also used to reduce the bit rate. If Layer III is used to deliver a fixed bit rate, a control loop is employed to adjust the number of bits assigned in the coding process.

4.3.2 MPEG-2 Audio Coder

MPEG-2 has been designed to meet the demands from the satellite broadcasting and cable television industries. MPEG-2 is currently used in point to point audio links, digital radio links in EUREKA 147 and direct satellite broadcasting [87]. This standard contains two different work items. The first one is the extension to lower sampling frequencies, providing better sound quality at low bit rates (below 64 kbit/s for a mono channel). This version of MPEG-2 accommodates audio signals sampled at 16, 22.05, 24 kHz. The bit rate for this version ranges from 32 to 256 kbit/s (Layer I) and from 8 to 160 kbit/s (Layer II & Layer III) [5].

The second work item deals with multichannel audio. The multichannel version of the MPEG-2 standard (audio part) includes two coding standards, i.e., MPEG-2 (BC) [121] which is backward compatible with the MPEG-1 audio coder and the MPEG-2 AAC (Advanced Audio Coding) which is not backward compatible [16, 52]. Both versions are able to code 5-channel audio inputs plus one low frequency enhancement channel. However AAC provides better audio compression relative to MPEG-2 BC. For 5-channel audio signals, it has been shown in MPEG formal listening tests that MPEG-2 AAC provides slightly better audio quality at 320 kb/s than MPEG-2 BC can provide at 640 kb/s [10]. Since MPEG-2 BC is a multichannel extension of MPEG-1, we only describe the AAC version of MPEG-2 in the following section.

MPEG-2 Advanced Audio Coding (AAC)

MPEG-2 AAC [16, 52] is a state-of-the-art audio coding standard operating at less than 64 kbit/s per channel for multichannel operation and accommodates 1 to 48 channels. AAC outperforms all older audio coders such as AC-3 and PAC. According to Soulodre *et al* [122], for stereo signals, the quality of AAC at 96 kbit/s is comparable to the quality of PAC at 128 kbit/s and AAC at 128 kb/s is significantly better than PAC at 160 kbit/s. The main reason for AAC's superiority is that it uses a filterbank with a finer frequency resolution that enables superior signal compression. Additionally, AAC uses a number of new modules such as Temporal Noise Shaping (TNS) [94] and backward adaptive linear prediction which enhance the coding efficiency. Compared with MPEG-1 Layer III (MP3), AAC is approximately 30% more bit rate efficient due to the improvements implemented by AAC including an improved filter bank, more efficient entropy coding, and better speech

encoding quality [10].

There are three profiles for the AAC standard called the Main Profile, the Low Complexity Profile, and the Scalable Sampling Rate Profile. The Main profile is used when computation power and memory are not constrained. The Low Complexity profile does not have all the processing modules of the main profile and is for the applications in which low complexity is a primary goal. The Scalable Sampling Rate (SSR) profile is meant for scaling the bit rate.

Figure 4.7 shows the structure of MPEG-2 AAC. In the following we briefly describe the processing modules of AAC. The input signal is preprocessed through a pseudo-quadrature

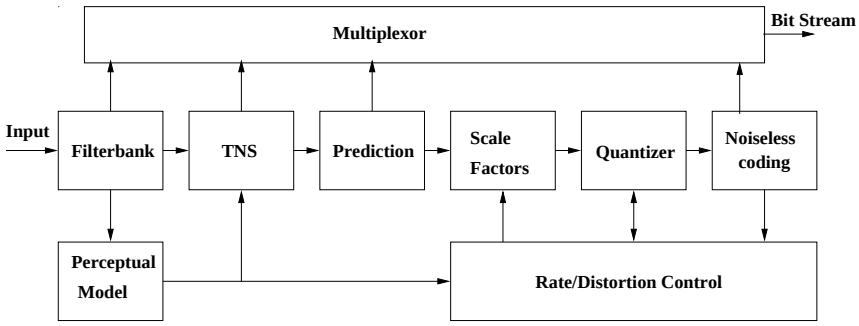


Fig. 4.7 Basic block diagram of the MPEG-2 AAC encoder [52].

mirror filterbank. The gain of each bandpass filter in the filterbank is adjusted to reduce pre-echo artifacts. An MDCT filterbank is used to decompose the input signal. The length of the MDCT is either 2048 points (for a regular window) or 256 points for a short window. Moreover the shape of the window can be switched between a sine window and a KBD window.

A technique called *Temporal Noise Shaping (TNS)* is used in the frequency domain to model the envelope of the input signal in the time domain. This technique is similar to the well known time-domain linear prediction technique. The difference is that the prediction is performed in the frequency domain to approximate the temporal envelope of the input signal. As the linear prediction analysis models the signal spectrum in the frequency domain, the TNS technique, which is done in the frequency domain, models the the envelope of the signal in the time domain. By using this method, instead of quantizing the MDCT coefficients, the difference of the coefficients and their predictions are quantized. This technique is meant to reduce pre-echo artifacts through shaping the noise in the time

domain.

The prediction module uses second-order backward lattice predictors to remove additional redundancy from individual filterbank outputs. The prediction module increases the complexity of the coder and hence is used in the main profile of the AAC. The quantizer module employs nonuniform scalar quantizers to quantize the MDCT coefficients (If the TNS module is activated, a differential scheme is used to quantize the residuals in the frequency domain). The step sizes are determined by the corresponding SMR and the rate/distortion control unit. Note that a psychoacoustic model similar to the model used in the MPEG-1 Layer III is used to calculate the SMR for each frequency band. After quantization of the transform coefficients, the Noiseless coding module applies Huffman coding to vectors of quantized coefficients.

4.3.3 MPEG-4 Audio

The previous MPEG audio standards concentrate on the coding of audio signals with almost transparent quality. The MPEG-4 audio standard has been created to support different applications which range from intelligible speech to high quality multichannel audio [25, 123, 10, 124] .

The MPEG-4 audio standardizes processing modules (tools) for natural and synthetic audio coding at bit rates ranging from 2 kbit/s up to 64 kbit/s. A single coding technique cannot accommodate both speech and audio at all desired bit rates [125]. To achieve the highest audio quality for a wide range of bit rates, three types of codecs have been defined: parametric codecs for mostly narrowband speech samples at 8 kHz at 2–4 kbit/s, CELP codecs for both narrowband and wideband speech at 4–24 kbit/s (up to 24 kbit/s for 8 kHz speech and 14–24 kbit/s for 16 kHz speech), and Time-to-Frequency (T/F) codecs for general audio signals at 6–64 kbit/s per channel. MPEG-4 also defines tools to synthesize sounds based on structured descriptions of audio data (also known as *structured audio coding*) [18, 24]. Moreover, MPEG-4 provides bit rate scalability, complexity scalability and multi-bit rate operation.

Figure 4.8 shows the basic encoder structure of the MPEG-4 audio standard. In the following we briefly describe different parts of the MPEG-4 audio structure and discuss some of its features.

The T/F transform coder uses almost all the modules of MPEG-2 AAC. Additionally,

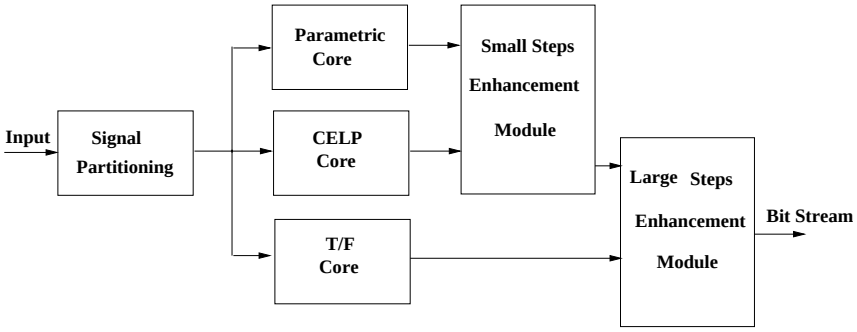


Fig. 4.8 Basic block diagram of the MPEG-4 audio encoder [126].

to increase the compression efficiency, several other modules including the BSAC algorithm (will be discussed in the following paragraph), Twin-VQ quantization [111] for quantization of spectral components at 6–16 kbit/s, perceptual noise substitution in the noise-like regions of the spectrum [127], and long term prediction have been added to the system.

One of the interesting features of MPEG-4 is bit rate scalability, meaning that only a part of the incoming bit stream can be decoded to reconstruct the output signal with lower quality. Bit rate scalability is done in two ways: small step and large step scalability. The bit sliced arithmetic coding (BSAC) algorithm provides small steps scalability. In the BASC algorithm, all MDCT coefficients are quantized in such a way that the quantization noise lies below the masking threshold. The binary representation of four adjacent quantised coefficients are grouped together. Then the bits in the vectors are noiselessly encoded according to their significance, i.e., first the most significant bits (MSB) in each group, up to the least significant bits. To produce a certain bit rate the encoder will use only some of the noiselessly encoded vectors starting from the most significant subvectors. In this algorithm, since there is no preference among the MSB of different groups, contrary to traditional adaptive transform coding, the bandwidth is not reduced. Also the bit rate can be scaled to reduce the bandwidth of the reconstructed signal. For instance, at a bit rate of 16 kbit/s, all MDCT coefficients above 3.5 kHz will be discarded [87]. By using the BSAC algorithm, the decoder can stop anywhere between 16 kbit/s and 64 kbit/s with a 1 kbit/s step size [10].

In large step scalability a base layer bit stream produced by one core coder can be combined with enhancement layer bit streams produced by other core coders to form a higher bit rate. For instance, a base layer bit stream produced by the CELP coder can be

combined with an enhancement bit stream produced by the T/F coder [125].

MPEG-4 provides the Structured Audio (SA) modules to synthesize audio signals at bit-rates from 0.1 to 10 kbit/s [10]. The idea of the structured audio is that a description of the sound is sent to the decoder to produce a similar sound. The description is created using the Structured Audio Orchestra Language (SAOL) and Structured Audio Sample Bank Format (SASBF) [125].

To code speech signals with a natural quality of the compressed signal, a CELP coder is used. The reason to use a CELP coder comes from the fact that there is a big difference between the performance of speech coders and transform coders applied to speech for bit rates below 24 kbit/s [125]. The CELP coder is used for bit rates of 4–24 kbit/s. For bit rates of 2–4 kbit/s the Harmonic Vector eXcitation Coding (HVXC) [128], a sinusoidal narrowband vocoder scheme, is used. MPEG-4 also provides tools for the conversion of a text to speech. The bit rate for this feature spans a range of 200 to 1200 bit/s.

Chapter 5

Overview of the NPAC Encoder

A block diagram of the Narrowband Perceptual Audio Coder (*NPAC*) is shown in Fig. 5.1. The blocks are described in the following sections. We consider monaural audio signals sampled at 8 kHz and bandpass filtered to limit the spectrum to between 50 Hz and 3.6 kHz. An MDCT is used to decompose the input signal into spectral components. The masking threshold is estimated and used in both the adaptive bit allocation and the quantization of the transform coefficients.

5.1 Time-to-Frequency Mapping

A Modified Discrete Cosine Transform (MDCT) [67] is used to transform the audio data. The MDCT provides critical sampling, perfect reconstruction and reduced block edge effects. There is a direct relationship between the MDCT and DFT [35] which implies that the MDCT coefficients represent the frequency content of the input signal. Moreover, FFT-like algorithms can be used to compute the MDCT.

Choice of MDCT window

For the MDCT, windowing is used to select the portion of the input signal to analyze. The length of the window is a compromise between long windows (high coding gain[1]) and short windows (better model transient behaviours and keep coding noise local). Since the characteristics of audio signals vary with time, and since our coder is also intended

[1]Coding gain measures the ability of a transform to concentrate the energy into a few coefficients [74].

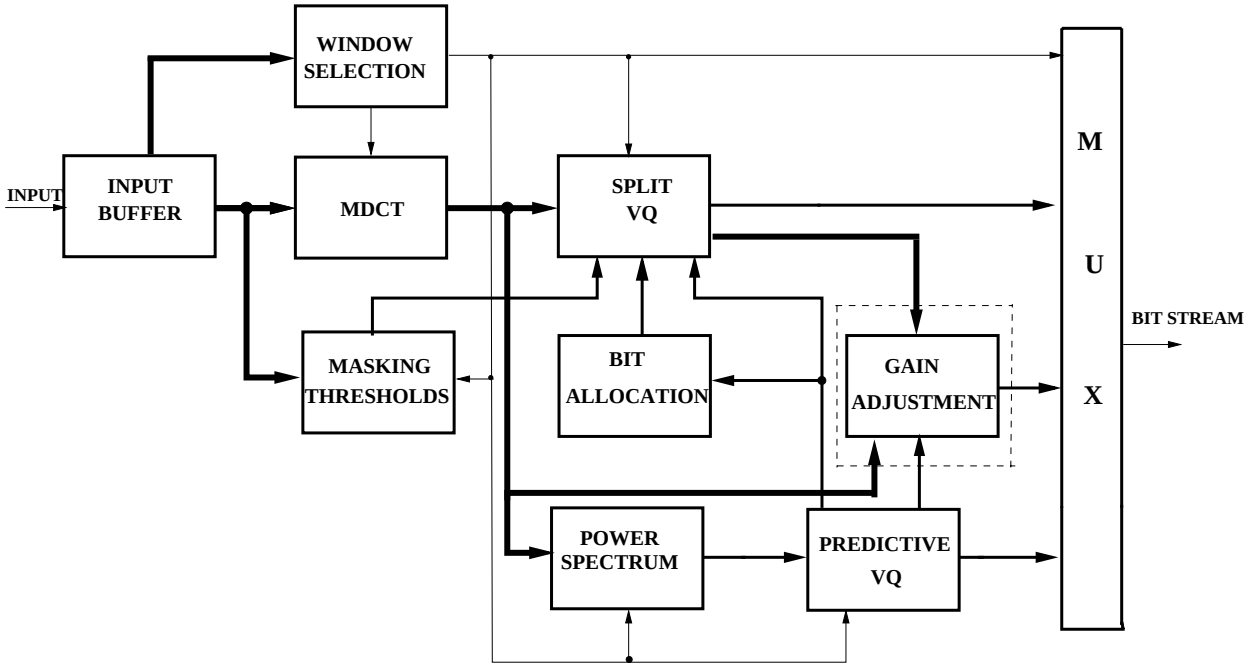


Fig. 5.1 Block diagram of the NPAC coder.

for speech use, we choose a compromise analysis frame length of 30 msec, a period over which speech signals can be considered to be pseudo-stationary. The encoder takes in 240 samples (120 samples from the previous frame and 120 new samples) and uses an MDCT to decompose the block of data. However, sharp transient sounds require a higher temporal resolution. This issue is discussed later.

The shape of the time window used for the MDCT affects the frequency selectivity of the filterbank. We need to trade off resolution in the main lobe versus high attenuation of the sidelobes. A narrow main lobe keeps energy local to the MDCT coefficients and prevents loss of coding gain. The main lobe width should be less than the width of the narrowest critical band (100 Hz). This choice makes it easier to control the perception of the quantization noise and to compute the simultaneous masking thresholds more accurately. On the other hand, the stopband attenuation should be high to reduce spectral leakage.

In the AC-2/AC-3 encoders [72], a KBD window with high stop band attenuation is used. Although this window performs well for many audio signals, it has a poor frequency selectivity that makes it unsuitable for low-pitch harmonic signals. In [16], in order to accommodate a wider range of audio signals, the coder allows for switching between a KBD window and a sine window.

In our coder, we use a single window type. We have designed the time window with a 50 Hz (lowpass prototype) bandwidth. The modulated response has a bandwidth of 100 Hz. Although, for a window of 240 samples, the MDCT coefficients represent steps of 33.3 Hz, the choice of 100 Hz allows us to enhance the stopband rejection of the window response. This window gives an increased coding gain relative to the sine window.

Another approach to reconciling some of these conflicting requirements is to use non-identical windows in the encoder and decoder. In [129], a better frequency response at the analysis stage is obtained at the expense of a less tapered window at the synthesis stage. The latter then adversely affects the transitions between blocks. In our work we use identical analysis and synthesis windows.

Handling Transients

For high energy transient parts of the input signal, it is desired to localize short bursts of quantization noise to prevent them from spreading over a long period of time. We handle this problem by switching to a shorter window when a strong jump in energy

is encountered. As an alternative to switching to short windows at onsets, Herre and Johnston [94] use Temporal Noise Shaping (*TNS*) to continuously adapt the temporal and frequency resolution of the filterbank. The performance of this technique is yet to be fully investigated when employed in an MDCT-based encoder.

Short windows reduce the coding gain and should be avoided when they do not improve the coded signal quality. Since backward temporal masking lasts for about 4 msec while forward temporal masking lasts for about 200 msec, a distinction should be made between rises and falls in the energy of the signal. A simple criterion based on the relative positive change in the energy of the input signal is used. In the time domain, a local estimate is made of the change in signal energy. This is done by splitting the input frame into groups of 3 time samples and calculating the energy of the samples in each interval. The maximum positive change will be found as follows,

$$r = \max(\frac{e_{j+1} - e_j}{e_j}), \tag{5.1}$$

where e_j is the energy of interval j. If r exceeds a threshold value, we switch to a shorter window. Note that in order to maintain perfect reconstruction of the combined analysis and synthesis stages, a start window is used to switch from long to short windows, and a stop window switches back [15].

For the short windows we use a frame length of 10 msec (80 samples). Fig. 5.2 shows the switching of the longer window to a series of shorter windows for a piece of music containing a transient sound.

5.2 Masking

Masking is a property of the hearing system by which a weaker audio signal becomes inaudible in the presence of a louder signal [17]. The masking depends both on the spectral composition of the masker and the signal to be masked as well as their variation with time [32, 40]. In audio coding, the masker is the original input signal and the signal to be masked is the quantization error.

The masking phenomena can be exploited to determine the best assignment for available bits. Bits need only be assigned to the audible spectral components. On the average, more than 50% of the transform coefficients are masked. For the remaining transform coefficients,

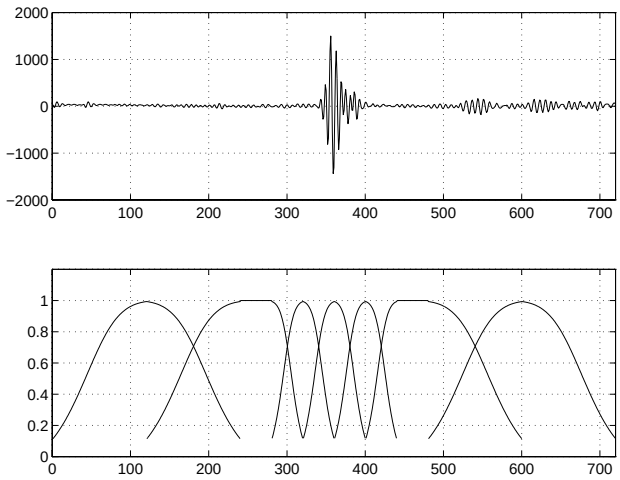


Fig. 5.2 Window switching for a piece of music containing a transient sound.

the step size for the quantizers can be chosen in a way that the quantization noise lies below the masking threshold.

5.2.1 Simultaneous Masking

There are many models for computing the simultaneous masking (masker and maskee present at the same time) threshold [17, 15, 51, 32]. Since the MDCT is employed to decompose the input signal, we use a modified version of the model proposed by Johnston [51] which is based on the work by Zwicker [17] and Schroeder *et al* [32] to calculate the masking threshold corresponding to the MDCT coefficients.

The masking calculation consists of the following steps:

- Calculate the Bark energy spectrum.
- Convolve the Bark energy spectrum with the spreading function to give the excitation curve.
- Subtract an offset (dB) depending on a tonality factor from the excitation curve to give the masking level.

Excitation level

The Bark spectrum is derived from the frequency spectrum with a non-linear transformation of the frequency variable. This gives a measure of the distribution of energies with respect to the critical band numbers. The Bark spectrum is convolved with the spreading function to give an excitation level.

Masking level calculation

The masking threshold is derived from the excitation level by subtracting an offset (in dB) to give the masking level. The offset value depends on whether the signal is tone-like or noise-like.

In contrast to [51] in which a spectral flatness measure is used to identify the nature of the whole frame, we take another approach based on the predictability of the transform coefficients in each critical band. Note that most audio signals have a noise-like structure at high frequencies despite the fact that they may have a strong harmonic structure at low frequencies. Considering this fact, it would be more accurate to identify the nature of the spectrum locally. The tonality factor will be calculated for each critical band using the predicted value of the current subvector [15],

$$\tilde{X}^{(i)} = 2X^{(i-1)} - X^{(i-2)}. \tag{5.2}$$

The relative prediction error is calculated as

$$\delta = \frac{\|X^{(i)} - \tilde{X}^{(i)}\|}{\|X^{(i)}\| + \|\tilde{X}^{(i)}\|}. \tag{5.3}$$

The relative prediction error will be converted to a tonality factor according to [15]

$$a = \min(1, \max(-0.3 - 0.43\log(\delta), 0)). \tag{5.4}$$

The offset value is determined by the tonality factor [51]

$$L_{\text{offs}}(j, a) = a(14.5 + j) + 5.5(1 - a), \tag{5.5}$$

where j is the index of the critical band.

The masking level can then be calculated. However, the masking threshold should be adjusted to take into account the absolute threshold of hearing.[2]

MDCT Masking Threshold

Since the masking threshold is calculated based on the DFT, this masking threshold must be modified for use with the MDCT coefficients. Consider the following relationship between the DFT and MDCT coefficients, [35],

$$C(k) = \sqrt{\frac{2}{M}}\,|S(k)|\cos\Big(\frac{\pi(M+1)(2k+1)}{4M} - \angle S(k)\Big). \tag{5.6}$$

where $S(k)$ is the Fourier transform of the modulated windowed input signal ($2M$ values) and $C(k)$ is an MDCT coefficient. If $m_{\text{DFT}}(k)$ is the masking threshold corresponding to the kth DFT coefficient, then in order to have the same Signal-to-Mask Ratio (SMR) at any coefficient in the DFT and MDCT domain, the following relation should hold:

$$C^2(k)/m_{\text{MDCT}}(k) = |S(k)|^2/m_{\text{DFT}}(k). \tag{5.7}$$

The masking thresholds are then related as follows,

$$m_{\text{MDCT}}(k) = \frac{2}{M} m_{\text{DFT}}(k)\cos^2\Big(\frac{\pi(M+1)(2k+1)}{4M} - \angle S(k)\Big). \tag{5.8}$$

5.2.2 Temporal Masking

Temporal masking occurs when tones occur close in time, but not simultaneously. A signal can be masked by another signal that occurs later (premasking). In addition, a signal can be masked by another signal that ends before the signal begins (postmasking). The duration of premasking is less than 5 msec, whereas that of the postmasking is in the range of 50 to 200 msec [2]. Since incorporating the backward masking of the hearing system into the coder introduces delay with little gain in compression, we neglect that effect and just exploit the forward masking.

We have used the following model which was proposed in [50] as it takes both the effect

[2]To make full use of the absolute threshold of hearing, the reconstructed signal should be played back at the same or lower level than the original.

of the frequency and the level of the masker into account:

$$m_{\mathrm{t}}(f, L) = \alpha + \beta \exp(-f/\gamma), \tag{5.9}$$

where m_{t} is the temporal masking in dB, L is the sound (masker) level of the previous frame in dB, f is the frequency in Hz and α, β, γ are three parameters to be found from experimental data. In [50] three expressions have been fitted to the experimental data for α, β, γ. In this work, we consider the temporal masking if the masker level is more than 30 dB. Based on this assumption and the data given in [50] we have found the following expressions for the above-mentioned parameters:

$$\begin{aligned} \alpha &= 0.001L^2 + 0.2267L + 17.7142, \\ \beta &= -0.0047L^2 + 1.2256L - 24.32548, \\ \gamma &= -0.0002L^4 + 0.0546L^3 - 5.4685L^2 + 234.7411L - 3325.0350. \end{aligned} \tag{5.10}$$

The data reported in [50] indicate the level of the temporal masking at 20 msec after the masker. Although the time interval between successive frames in our coder is 15 msec, and hence the temporal masking level will be underestimated using this formula, we have chosen to use it in order to be conservative. In the coder, we calculate the temporal masking for each critical band. In doing so we assume that all the energy in each critical band is concentrated in the center frequency (except the first band for which we set f to 100 Hz) and the sound level is due to the contribution of all the coefficients in the band. This way, for each frame we calculate the masking threshold at 17 points. If the masking threshold is greater than the sound level in any band, we assume that all the coefficients in that band are masked. If the transform coefficients are not completely masked, the masking threshold will be equally divided among the coefficients.

5.2.3 Calculation of the Combined Masking Threshold

We use a *power-law* rule as follows [50] to combine the temporal and simultaneous masking thresholds

$$m_{\mathrm{net}} = (m_1^p + m_2^p)^{(1/p)}, \tag{5.11}$$

where m_{net} is the net masking threshold due to two masking thresholds m_1 and m_2. According to [50], a value of 0.3 for parameter p is found to be the best match to experimental

data.

5.2.4 Verification of the Masking Models

In order to verify the masking models, the masking thresholds for several audio signals were computed. After replacing the masked coefficients by zeros, there was no perceptual difference between the original and reconstructed signals. If we artificially increase the level of masking to have about 80% of the transform coefficients masked, the quality of the reconstructed signal is still good. This experiment shows that we can concentrate on reproducing the perceptually important spectral components.

5.3 Quantization of the Transform Coefficients

In our coder, we decompose subbands of the transform coefficients into gains and shapes. Then a VQ scheme along with perceptually-based bit allocation is used to quantize the shape vectors. To quantize the gains (scale factors), a predictive/non-predictive scheme is used.

5.3.1 Quantization of the Shape Vectors

One way to accomplish good quantization is to consider the characteristics of the hearing system such as masking phenomena and limited temporal and frequency resolution. Due to the limited number of bits available for coding the transform coefficients, vector quantization is used rather than scalar quantization. We would like to quantize and transmit only unmasked transform coefficients. This approach would require additional bits to identify the masked/unmasked coefficients to reconstruct the audio signal at the receiver. Instead of doing so, we employ a split adaptive VQ scheme to quantize the transform coefficients. The bandwidth division is based on the critical bands. The reason for the perceptually-based band division comes from the fact that the sensitivity of the ear is higher at lower frequencies which implies a higher frequency resolution at lower frequencies. We incorporate the masking threshold while vector quantizing the coefficient without transmitting any information about the masking pattern.

We use a modified version of the LBG algorithm [102] with the perceptually-based distortion measure (defined in Chapter 4) to design the codebooks [33]. The same error

criterion is used to select the best codewords. For a normalized vector X_n and the jth codeword $\chi^{(j)}$

$$\mathbf{d}(k) = |X_n(k) - \chi^{(j)}(k)|^2 - \mathbf{m}_n(k), \tag{5.12}$$

where $\mathbf{m}_n$ is the vector of normalized masking thresholds corresponding to X_n. The normalized energy of the audible noise is calculated by

$$D(X_n, \chi^{(j)}) = \sum_{k=1}^{K} \max(\mathbf{d}(k), 0), \tag{5.13}$$

where K is the dimension of X_n. The centroid of each Voronoi region is determined by minimizing the normalized energy of the audible noise as follows

$$\chi_{\text{opt}}^{(j)} = \arg\min_{\chi^{(j)}} \sum_{i=1}^{I} D(X_n^{(i)}, \chi^{(j)}), \tag{5.14}$$

where I is the number of the vectors in region j.

At very low bit rates, it is not possible to have transparent coding. Since the quantization noise level often goes above the masking threshold, it is appropriate to shape the quantization noise inside each band. Therefore, we may modify the error criterion as follows

$$\begin{aligned} \mathbf{d}_w(k) &= \max\left(\frac{|X_n(k) - \chi^{(j)}(k)|^2 - \mathbf{m}_n(k)}{X_n^2(k) + \mathbf{m}_n(k)}, 0\right), \\ D_w(X_n, \chi^{(j)}) &= \sum_{k=1}^{K} \mathbf{d}_w(k), \end{aligned} \tag{5.15}$$

where D_w is the total weighted quantization noise above the normalized masking threshold. By making this modification, we allow the audible quantization noise to get shaped according to the distribution of energy inside a critical band.

Memory Reduction for Storage of the Codebooks

Vector quantization needs a lot of memory space to store the codebooks. Solutions to this memory problem have been addressed in the literature [130]. In this work, we have dealt with this problem as follows; in the process of training the codebooks, a number of

codebooks with different lengths of power 2 for each critical band are trained (the number of the codebooks depends on the maximum number of bits assigned to the corresponding band). We use these codebooks along with an adaptive bit allocation scheme to assign a variable number of bits to each subvector. We investigated different ways to reduce the memory required to store the codebooks with little loss of quality. In one approach, we find the closest codewords of the largest codebook to the codewords of the second largest codebook (in the mean square sense). Then we order the codewords of the larger codebook to put the selected codewords at the top. We do the same procedure for other codebooks to end up with an embedded codebook for each band. (Note that in each step, in order to find the closest codewords from the largest codebook to the codewords from a codebook with length 2^k we take the first 2^{k+1} codewords.) By doing so, we reduce the required memory by 50% with very little loss of quality.

Another approach that we have taken in the proposed coder is to use the largest codebook to code a long set of training vectors. Then based on the frequency of selection of the codewords, we order the codewords to have the most often selected codewords at the top of the codebook. The resulting codebook shows almost the same performance as when we use separate codebooks to quantize the subvectors. To further reduce the memory, the bands with the same number of coefficients can share the same codebook with little loss of quality.

5.4 Predictive VQ of the Scale Factors

The transform coefficients in each critical band are normalized by the corresponding square root energy which must be transmitted to the receiver as side information.

There exists a high level of similarity among the gain vectors. This similarity is due to the 50% overlap between successive frames which causes the spectra to be highly correlated. This inter-frame correlation can be efficiently exploited by applying a predictive scheme to quantize the scale factors. Shoham in [131] uses the previous few quantized vectors to estimate the current vector. However, processing several past frames makes the prediction scheme more vulnerable to channel errors and also the similarity between the current spectrum with the past ones reduces as we go farther backward. In this work we consider only the previous quantized spectrum to estimate the current frame. In the coder, a predictive/non-predictive VQ scheme is used in the log domain to quantize the scale

factors. Since the level of similarity between successive vectors containing the scale factors are varying according to the nature of the signal, we use the spectral distortion measure to choose the appropriate scheme in the way that the predictive scheme is employed when the root mean-squared difference of the current and previous vectors is less than 6 dB, otherwise the vector of scale factors will be quantized directly. This coding strategy is compatible with the mechanism of the hearing system; in steady parts of the input signal such as voiced speech we need finer quantization of both spectral shapes and gains, whereas for 'unstructured' or noise-like parts more coarse quantization is adequate. This also can be justified through the masking property of the hearing system. As is well known, the masking threshold in the case of tone-masking-noise is lower than that of noise-masking-noise. For that reason, we need finer quantization for pseudo-periodic parts of the input signal. In the predictive scheme, we quantize the vectors containing the scale factors through the following steps (note that all these steps are performed in the log domain.),

- Calculate the mean value of the scale factors

$$\mu_i = \frac{1}{17}\sum_{j=1}^{17} g_j^{(i)}, \tag{5.16}$$

  where $g_j^{(i)}$ is the log gain of band j at time index i.

- Remove the mean value from the scale factors

$$\mathbf{g}_n^{(i)} = \mathbf{g}^{(i)} - \mu_i, \tag{5.17}$$

  where $\mathbf{g}^{(i)}$ is the gain vector.

- Quantize μ_i using a differential quantizer.

- Predict the current normalized vector from the previous normalized vector using the best prediction matrix

$$\begin{aligned} \mathbf{P}_{\text{opt}} &= \arg\min_{\mathbf{P}}(\mathbf{g}_n^{(i)} - \mathbf{P}\hat{\mathbf{g}}_n^{(i-1)}), \\ \tilde{\mathbf{g}}_n^{(i)} &= \mathbf{P}_{\text{opt}}\hat{\mathbf{g}}_n^{(i-1)}, \end{aligned} \tag{5.18}$$

where $\mathbf{P}$ is the prediction matrix, $\hat{\mathbf{g}}_n^{(i-1)}$ is the mean-removed quantized version of the previous vector and $\tilde{\mathbf{g}}_n^{(i)}$ is the prediction of the normalized current vector.

- Form the difference vector

$$\mathbf{d}_g = \mathbf{g}^{(i)} - \hat{\mu}_i - \tilde{\mathbf{g}}_n^{(i)}, \tag{5.19}$$

where $\hat{\mu}_i$ is the quantized mean value of the current vector.

- The difference vector will be quantized using a two stage VQ.

This approach leads to fine quantization of the scale factors in steady state parts of the input signal which is highly desirable for high quality of the coded signal. A total of 37 bits is used to quantize the scale factors. For computational reasons, we limit the size of the codebooks to 2048 codewords. For the predictive scheme, 6 bits for the mean value, 9 bits for the predictor selection and 2×11 bit 2-stage VQ for the difference vectors.

In the nonpredictive scheme, the vector of scale factors is normalized (in the log domain). The normalized vector will be vector quantized using a codebook of 2048 codewords. In the next step the best estimator matrix is selected out of 64 matrices to estimate the current normalized vector based on the observation of the best codeword selected in the first step. Then the difference vector will be formed as it is done in the predictive scheme. Finally the difference vector will be quantized using a codebook of 2048 codewords. Note that 9 bits is spent to quantize the mean value. For a large set of test vectors, the average spectral distortion for steady state frames using the predictive scheme was less than 1.5 dB and for the rest using the non-predictive scheme it was 2.5 dB; the number of quantized vectors with spectral distortion above 4 dB was almost zero.

5.4.1 Design of the Predictor Matrices

The predictor matrices are designed to minimize the average spectral distortion between the normalized gain vectors and the predicted vectors. We take a long training set and will find the predictor matrices using a modified version of the Lloyd algorithm. First we design one predictor matrix for the whole training set and then by perturbing the first matrix and also performing an iterative procedure, we will find new predictor matrices. This procedure will go on until the desired number of predictor matrices are found. For each subset of the training set (corresponding to a predictor matrix), we find the optimal predictor matrix

through the following optimization procedure:

$$\mathbf{P}_{\text{opt}}^{(j)} = \arg\min_{\mathbf{P}^{(j)}} \sum_{i \in R_j} (\mathbf{g}_n^{(i)} - \mathbf{P}^{(j)} \hat{\mathbf{g}}_n^{(i-1)}), \tag{5.20}$$

where R_j contains the time indexes of the vectors belonging to the jth region. Note that, in order to perform the optimization, we need to have the quantized vectors. To overcome this problem, we use the quantized vector obtained through the nonpredictive method and then refine the predictor matrices by repeating the optimization procedure. In each iteration we use the finer quantized value for $\mathbf{g}_n^{(i-1)}$ obtained in the previous iteration.

By using the orthogonality theorem, we find the solution to the optimization problem as follows

$$\mathbf{P}_{\text{opt}}^{(j)} = \mathbf{R}_{01}^{(j)} (\mathbf{R}_{11}^{(j)})^{-1}, \tag{5.21}$$

where $\mathbf{R}_{01}^{(j)}$ is the summation of the cross-correlation matrices of the current and quantized previous vectors in Voronoi region j. $\mathbf{R}_{11}^{(j)}$ is the summation of the autocorrelation matrices of the quantized previous vectors in the same region. We continue the iterations until the required number of predictor matrices are found and the change in the average spectral distortion becomes less than a threshold.

5.4.2 Modification to the Predictor Matrices

By looking at the predictor matrices, we note that the magnitude of the matrix entries decreases as they are farther from the main diagonal. As a matter of fact, each component in the current vector will be predicted mainly by the corresponding and a few adjacent components of the previous quantized vector. We exploit this fact in order to set the far-off diagonal elements of the predictor matrices to zero. By doing so, we reduce the computation load and also the memory for the storage of the predictor matrices. In order to find the predictor matrices, we have to reformulate the optimization procedure. For an example, we assume that the main diagonal and its adjacent diagonals are non-zero and the rest of matrix entries are set to zero. We can easily generalize the following formulation for any number of non-zero diagonals;

$$\tilde{\mathbf{g}}_n^{(i)} = \mathbf{P} \hat{\mathbf{g}}_n^{(i-1)}, \tag{5.22}$$

where $\mathbf{P}$ is the predictor matrix defined

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & & & \\ p_{2,1} & p_{2,2} & \ddots & \mathbf{0} & \\ & \ddots & \ddots & \ddots & \\ & \mathbf{0} & \ddots & \ddots & p_{16,17} \\ & & & p_{17,16} & p_{17,17} \end{bmatrix}. \tag{5.23}$$

Rewrite Eq. 5.22 as

$$\tilde{\mathbf{g}}_n^{(i)} = \mathbf{G}_i \mathbf{c}, \tag{5.24}$$

where

$$\mathbf{G}_i^t = \begin{bmatrix} \hat{g}_{n1}^{(i-1)} & 0 & \dots & 0 & 0 \\ \hat{g}_{n2}^{(i-1)} & 0 & & & \\ 0 & \hat{g}_{n1}^{(i-1)} & & & \\ & \hat{g}_{n2}^{(i-1)} & & \vdots & \\ & \hat{g}_{n3}^{(i-1)} & & & \vdots \\ & 0 & & & \\ \vdots & & & 0 & \\ & & & \hat{g}_{n15}^{(i-1)} & \\ & \vdots & & \hat{g}_{n16}^{(i-1)} & \\ & & & \hat{g}_{n17}^{(i-1)} & 0 \\ & & & 0 & \hat{g}_{n16}^{(i-1)} \\ 0 & 0 & \dots & 0 & \hat{g}_{n17}^{(i-1)} \end{bmatrix}, \tag{5.25}$$

and

$$\mathbf{c} = \begin{bmatrix} p_{1,1} \\ p_{1,2} \\ p_{2,1} \\ p_{2,2} \\ p_{2,3} \\ \vdots \\ p_{16,15} \\ p_{16,16} \\ p_{16,17} \\ p_{17,16} \\ p_{17,17} \end{bmatrix}. \tag{5.26}$$

We have to find $\mathbf{c}$ to minimize the spectral distortion for each subset (Voronoi region) of the training set

$$\mathbf{c}_{\text{opt}}^{(j)} = \arg\min_{\mathbf{c}^{(j)}} \sum_{i \in R_j} (\mathbf{g}_n^{(i)} - \mathbf{G}_i \hat{\mathbf{c}}^{(j)}), \tag{5.27}$$

$\mathbf{c}_{\text{opt}}^{(j)}$ will be the solution to the following linear equations

$$\mathbf{A}\mathbf{c}_{\text{opt}}^{(j)} = \mathbf{y}, \tag{5.28}$$

where

$$\begin{aligned} \mathbf{A} &= \sum_{i \in R_j} \mathbf{G}_i^t \mathbf{G}_i, \\ \mathbf{y} &= \sum_{i \in R_j} \mathbf{G}_i^t \mathbf{g}_n^{(i)}. \end{aligned} \tag{5.29}$$

It is easy to show that $\mathbf{A}$ is a positive definite matrix and therefore we can use the Cholesky method to solve the linear equations. Figure 5.3 shows the average spectral distortion for different predictor matrices as a function of the number of matrices. As it can be observed, there exists a significant gap between the upper curve which corresponds to the diagonal predictor matrix and the other predictors. This is due to the fact that other predictor matrices exploit the lateral correlation among the components of the gain vector. At low rates, the performances of the predictors (except the single diagonal predictor) are almost

the same, but as the number of predictors increases, the performance of the predictor scheme can be enhanced at the cost of a higher computation load and larger memory storage for the predictors.

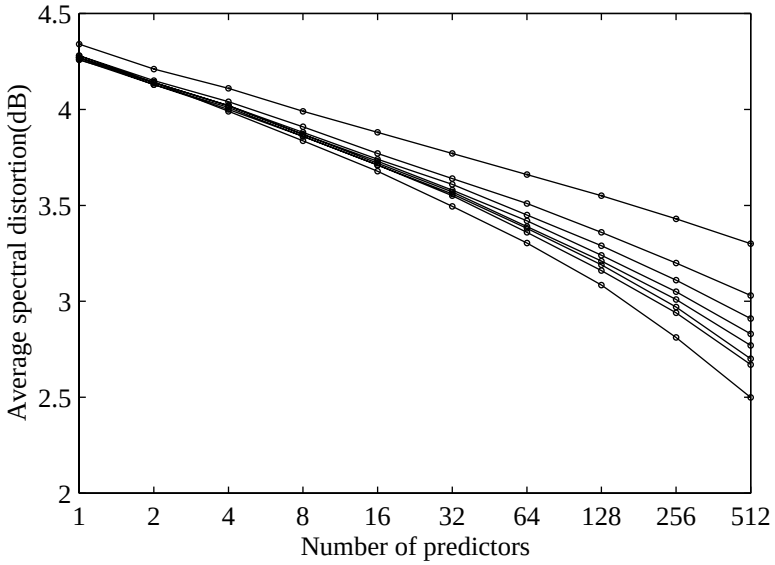


Fig. 5.3 Average spectral distortion versus the number of predictor matrices (from top to bottom) with 1, 3, 5, 7, 9, 11, 13 and 33 non-zero diagonals. The lowest curve corresponds to the predictor matrix with all non-zero diagonals.

In order to lower the computation load and required memory, we have investigated a special case of the above-mentioned procedure in which all the predictor matrix entries on the same diagonal are equal. Viewing this approach from a filtering perspective, we convolve the quantized previous gain vector with a noncausal FIR filter to estimate the current vector. In the following, we examine this approach to obtain the optimization procedure. The current vector is predicted as follows

$$\tilde{\mathbf{g}}_n^{(i)} = \mathbf{v} * \hat{\mathbf{g}}_n^{(i-1)}, \tag{5.30}$$

where $\mathbf{v}$ is the impulse response of the noncausal predictor filter and $*$ denotes convolution. For simplicity, like the previous algorithm, we consider only the predictor filters with three non zero elements. The following formulation can easily be generalized for the predictor

filters with more than three non-zero elements;

$$\begin{aligned} \mathbf{v} &= \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \\ \tilde{\mathbf{g}}_n^{(i)} &= \mathbf{P}\hat{\mathbf{g}}_n^{(i-1)} \end{aligned} \tag{5.31}$$

where

$$\mathbf{P} = \begin{bmatrix} v_2 & v_3 & & & \mathbf{0} \\ v_1 & v_2 & v_3 & & \\ & \ddots & \ddots & \ddots & \\ & & & \ddots & v_3 \\ \mathbf{0} & & & v_1 & v_2 \end{bmatrix}. \tag{5.32}$$

Rewrite Eq. 5.32

$$\begin{aligned} \tilde{\mathbf{g}}_n^{(i)} &= \mathbf{G}_i\mathbf{v}, \\ \mathbf{v}_{\text{opt}}^{(j)} &= \arg\min_{\mathbf{v}^{(j)}} \sum_{i \in R_j} (\mathbf{g}_n^{(i)} - \mathbf{G}_i\hat{\mathbf{v}}^{(j)}), \end{aligned} \tag{5.33}$$

$\mathbf{v}_{\text{opt}}^{(j)}$ will be the solution to the following linear equations

$$\mathbf{A}\mathbf{v}_{\text{opt}}^{(j)} = \mathbf{y}, \tag{5.34}$$

where

$$\begin{aligned} \mathbf{A} &= \sum_{i \in R_j} \mathbf{G}_i^t\mathbf{G}_i, \\ \mathbf{y} &= \sum_{i \in R_j} \mathbf{G}_i^t\mathbf{g}_n^{(i)}. \end{aligned} \tag{5.35}$$

Figure 5.4 shows the average spectral distortion for different predictor filters as a function of the number of predictors. Like the previous approach, there exists a gap between the upper curve which corresponds to the predictor filter with length 1 (single scalar predictor) and the other predictors. At low rates, the performances of the predictors (except the single diagonal predictor) are almost the same, but as the number of predictors increases,

the spectral distortion reaches a saturation value for short filters but for long filters it decreases linearly with increasing number of predictors. Also the rate of decrease in the prediction error becomes smaller as the filter length increases. That is expected as there is not a significant correlation between widely-separated components of the gain vectors. Compared to the first approach, for the same number of predictors, the prediction error is higher in the second approach. This is expected as we assume the same predictor filters to predict all entries of the current gain vector, whereas in the first approach we use different predictor filters to predict different entries of the current gain vectors.

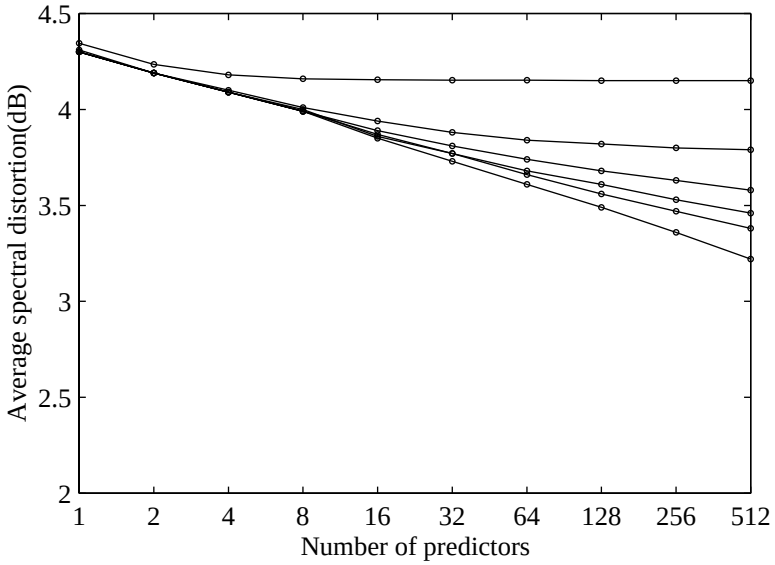


Fig. 5.4 Average spectral distortion versus the number of predictor filters with length (from top to bottom) 1, 3, 5, 7, 9, 17.

5.5 Gain Adjustment

In low rate coding, there are not enough bits to finely quantize the perceptually important coefficients. In this coder we propose the following procedure to reduce the quantization errors by adjusting the gain in each critical band;

$$\rho_{\text{opt}} = \arg\min_{\rho} \sum_{k=1}^{K} \max((X(k) - \rho\hat{X}(k))^2 - \mathbf{m}(k), 0), \tag{5.36}$$

where ρ is the gain adjustment factor, $X, \hat{X}$ are the original and quantized vectors of transform coefficients, $\mathbf{m}$ is the corresponding masking threshold and K is the dimension of the subvector. This optimization procedure gives the optimal ρ to minimize the audible difference between the input and output vectors. To find the optimal adjustment factor we have to do the optimization procedure for each critical band which considerably increases the computation load. Since in low rate coding the quantization noise in most bands is above the masking threshold, we take a suboptimal approach to decrease the computation. First we ignore the masking threshold

$$\rho_{\text{opt}} \approx \arg\min_{\rho} \sum_{k=1}^{K} (X(k) - \rho\hat{X}(k))^2. \tag{5.37}$$

The resulting ρ will be optimal in the squared error sense but suboptimal in a perceptual sense. Note that some critical bands are totally or partially masked and therefore there is no need to lower the quantization noise energy below the masking threshold. In those bands, the adjustment factors are sometimes found to be as large as 1000. To handle this problem and also limit the dynamic range of the adjustment factor we have chosen a range of 0.5 to 2 for this factor. Our experiments have shown that without quantizing the adjustment factors and with the limited range of values there is a significant improvement in the quality of the decoded signal. The quality enhancement is achieved at the cost of a higher rate and a little more computation. The usual trade off between quality and rate manifests itself here. With an overhead of less than 2 kb/s, the adjustment factors (vectors of 17 components with the limited dynamic range) can be finely vector quantized. By using this method, the quality of the decoded audio signals even for speech has been judged good to very good.

Although this block of the encoder adjusts the gains, it cannot be integrated into the gain quantization block. The quantized gains are needed for the bit allocation block whereas the gain adjustment factors are found after performing the bit allocation and shape quantization. However, if the roughly-quantized gains (output of the first stage gain quantization VQ) are used for the bit assignment, this block can be absorbed into the second stage VQ of the gain quantization block. This way the rate can be reduced at the expense of accuracy in the bit allocation.

5.6 Quantization of the Transform Coefficients in Short Frames

When a short window (80 points) is used, a number of changes occur. For a short window, 40 current samples and 40 previous ones are used. The MDCT unit generates only 40 transform coefficients. Although the number of the critical bands remains constant at 17, the distribution of the MDCT coefficients within the bands changes. In this case, some low frequency critical bands have only one coefficient. The 17 critical bands are combined into 7 aggregated bands. This aggregation is performed so that the vector quantization in split VQ can always operate on code vectors of dimension greater than one. Changes in the quantization procedure are required to handle the aggregated bands. A single gain is calculated for each aggregated band. To quantize the spectral shape vectors, the masking threshold is calculated as for the large frames and then used for the corresponding transform coefficients inside an aggregated band.

Since there is little or no similarity between the gain vectors of the consecutive short frames, the gain vectors of dimension 7 are quantized in a nonpredictive manner. In the bit allocation process, because of a rather noise-like nature of the signal in the transient parts, we assume that the masking threshold is 5 dB below the gain in each aggregated band.

5.7 Adaptive Bit Allocation

In traditional transform coders, bit assignment is done based on the distribution of the signal power in the frequency domain aiming at minimizing the total noise power. Since for most audio signals power is concentrated at low frequencies, few bits are assigned to high frequency components. This leads to an output signal which suffers from lowpass effects. In addition to that flaw, the masking phenomena are not fully taken into account which often results in allocating bits to the transform coefficients which are masked.

The aforementioned argument underlines the importance of shaping the noise spectrum based on perceptual principles. Using an adaptive bit assignment based on the perceptual importance of the subbands, the coding noise can be shaped to be less audible than a noise with the same energy without noise shaping. Noise shaping can provide high coding quality without requiring a high (conventional) SNR.

In low rate coding of audio signals, due to the scarcity of bits, unmasked quantization noise (audible noise) is often inevitable. The final goal in low rate coding is to deliver

acceptable quality with no annoying artifacts. This contrasts with the requirement for transparent coding in high rate wideband audio coding. Two different strategies can be considered to shape the audible noise spectrum [132]. In one approach, the quantization noise spectrum is shaped in parallel with the masking threshold curve. This way the audible noise is equally audible in different frequency bands. An alternative approach is to generate a flat noise spectrum above the masking threshold. According to [132, pp. 427–428], these two approaches are different in terms of auditory object formation. In the first approach, the quantization noise has a temporal modulation similar to that of the input signal. Therefore, the input signal and the noise will be perceptually fused to form one auditory object. In the second approach, the noise power is not equally distributed over the frequency range; hence it is audible to various extents at different frequencies. This way, the noise remains perceptually distinct from the input signal.

In our coder bit assignment is done both at the transmitter and the receiver using the quantized gain factors. From the quantized gain factors the masking thresholds are calculated. Note that for each band we need to specify the offset value which is subtracted from the excitation level (in the log domain) in order to obtain the simultaneous masking threshold. The offset value depends on whether the spectrum in each band is tone-like or noise-like. At low bit-rates we cannot afford to code the offset value for each band. However we do distinguish between two cases. In one case the input block of data has a harmonic structure which implies that the spectrum is more tone-like. In the other case the input has a more noise-like spectrum.

In order to distinguish between the two cases, in our implementation we use the same flag which is used in gain quantization to select either the predictive or nonpredictive schemes. When the flag is on, we suppose that the input frame is tone-like. Since for many audio segments, the signal is more tone-like in the low frequency bands than the high frequency bands, we assume higher offset values for the low frequency bands. By doing so, we assign more bits to the low frequency bands to maintain the pitch structure of speech. In each band the distance between the energy and the masking threshold is upper bounded by the offset value (in dB). Hence the maximum number of bits allocated to each band is determined through dividing the corresponding offset value (in dB) by the distortion reduction rate (see the following section). For those frames for which the flag is off, we set the masking threshold for all bands 8 dB below the excitation level. Fig. 5.5 shows the offset values and the maximum number of bits allocated to each transform coefficient in

different frequency bands.

In the case of short frames, since the input signal contains a transient and therefore does not have any harmonic structure, we simply set the masking threshold 6 dB below the spread Bark spectrum.

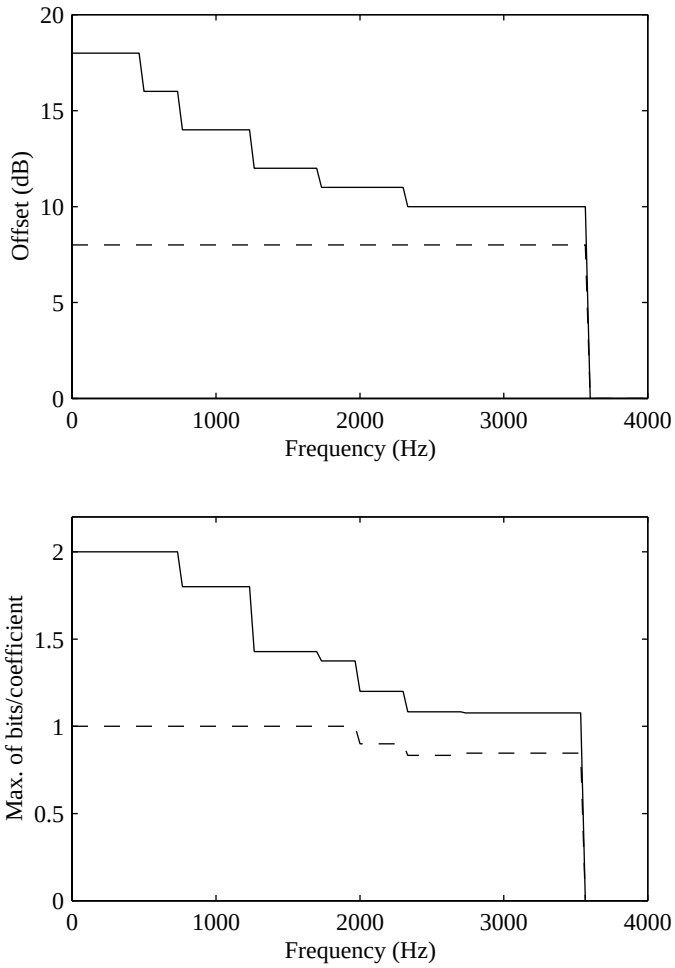


Fig. 5.5 Offset values for calculating the masking threshold (top) and corresponding maximum number of bits per coefficient (bottom) for tone-like frames (solid lines) and noise-like frames (dashed lines).

Critical Band Rate-Distortion Curve

In order to perform bit assignment we need the rate-distortion relationship for each codebook. A large set of vectors is used to measure the average distortion for different numbers

of bits. Although for any rate-distortion curve a "greedy algorithm" can be used to perform the bit assignment[3], we have noticed that the rate-distortion data can be well represented by a line fitted to the experimental data. As an example Fig. 5.6 shows the rate-distortion data for the codebook corresponding to critical band 2 which contains 3 coefficients. The slope of the line which has been fitted to the curve is -2.8 dB/bit. Note that all shape vectors in the test set are normalized and distortion is defined as the average energy of the quantization noise in decibels.

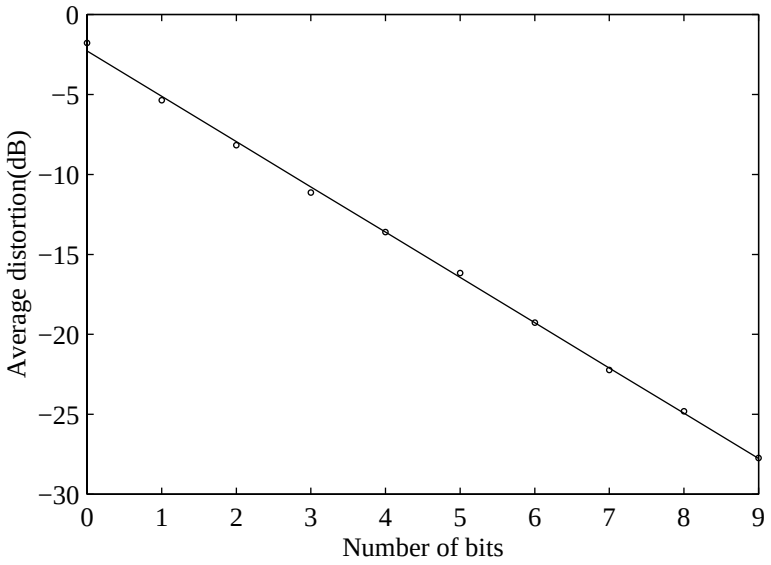


Fig. 5.6 Rate-distortion data for the embedded codebook corresponding to critical band 2 which contains 3 coefficients and its linear approximation.

Table 5.1 shows the slope of the lines fitted to the experimental data for the embedded codebook for each band. The correlation coefficient between the experimental data and the fitted line verifies the accuracy of the linear approximation.

5.7.1 Signal-to-Mask Ratio (SMR)-based Bit Allocation

In this approach bit allocation is performed based on the Signal-to-Mask Ratio (SMR). This way, the resulting noise spectrum will be parallel to the masking threshold curve. Each critical band is considered as a single entity with its corresponding SMR. The SMR is equal to the SNR when the quantization noise is at the threshold of audibility, i.e., when

[3]The distortion must be a convex function of the bit numbers.

Band	Number of coefficients	Slope (dB/bit)	Correlation Coefficient
1	2	4.9	0.998
2	3	2.8	0.999
3	3	2.9	0.999
4	3	2.9	0.999
5	3	2.9	0.999
6	4	2.1	0.999
7	4	2.1	0.999
8	5	1.6	0.998
9	5	1.7	0.998
10	5	1.7	0.999
11	7	1.2	0.999
12	7	1.2	0.998
13	8	1.0	0.997
14	10	0.9	0.998
15	12	0.8	0.998
16	13	0.7	0.999
17	13	0.7	0.999

Table 5.1 Slope of the rate-distortion line and the correlation between the experimental data and the linear approximation for different critical bands

the noise level is at the masking threshold. The SMR for each band is calculated in the following manner

$$\mathrm{SMR}_j = \hat{E}_j - T_j, \tag{5.38}$$

where $\hat{E}_j$ is the quantized log energy in band j, and T_j is the log masking threshold in that band. We assume that the initial distortion (in the log domain) for each band is equal to the corresponding SMR. A "greedy algorithm"[4] using the rate-distortion data can be employed to assign one bit at a time to the band with the largest (updated)*Noise-to-Mask Ratio (NMR)*. After assigning one bit to that band, its NMR on the average decreases by the amount given by the corresponding rate-distortion curve.

As a shortcut, a linear approximation of the rate-distortion data along with the values

[4]Note that the total distortion is a convex function of the bit numbers.

of SMR_j's can be used to allocate bits to each band according to the following formula

$$b_j = \max\left(\frac{\mathrm{SMR}_j b_T}{\lambda_j \sum_{i\in\Omega} (\mathrm{SMR}_i/\lambda_i)}, 0\right), \tag{5.39}$$

where Ω contains the indices of the bands with positive SMR and b_T is the total number of bits available to quantize the shape of the frequency spectrum within the critical bands. The slope of the rate-distortion line, λ_j, indicates the approximate reduction in the Noise-to-Mask Ratio (NMR) for one bit assigned to band j. Note that no bits are assigned to those bands whose SMR is negative. After the first round of bit allocation, the fractional parts of b_j's will be discarded to leave the integer parts. Therefore the total number of bits allocated in the first step will be less than b_T. To allocate the remaining bits, the Noise-to-Mask Ratio (NMR) is approximated for each band taking into account the bits already allocated in the first step,

$$\mathrm{NMR}_j = \hat{E}_j - T_j - \lambda_j b_j. \tag{5.40}$$

After calculating the value of NMR's, one bit at a time is allocated to the band with the largest value of the updated NMR. This process will continue until all remaining bits are allocated.

5.7.2 Energy-based Bit Allocation

In the energy-based approach, bit assignment is performed based on the energy above the masking threshold. The distortion is considered as the audible part of the quantization noise, i.e., the noise above the masking threshold.

The level of audible noise will be relatively higher in the spectral valleys due to the fact that there is less energy above the masking threshold there than in regions corresponding to spectral peaks. We consider two schemes to minimize the audible noise. In the first scheme the maximum of the distortion in the critical bands is minimized. In the second scheme the total audible noise is minimized.

Mini-Max Scheme

The mini-max bit assignment is done through the following optimization procedure

$$\arg\min_{b_j}(\max(D_j(b_j))) \quad \text{subjectto} \quad \sum_{j=1}^{N_b} b_j = b_T, \tag{5.41}$$

where N_b is the number of bands, b_T is the total number of bits available for each frame and D_j is the noise above the masking threshold.

We use a "greedy algorithm" to do the bit assignment. After each bit assigned, the distortion is updated. This way, one bit at a time is assigned to the band with the *largest updated distortion.*

Total Audible Distortion Minimization Scheme

The scheme minimizes the total audible distortion. Therefore the optimization objective function changes to

$$\arg\min_{b_i} \sum_{i=1}^{N_b} D_i \quad \text{subjectto} \quad \sum_{i=1}^{N_b} b_i = b_T. \tag{5.42}$$

According to this approach, one bit at a time goes to the band which results in the *largest reduction* in distortion. This algorithm can be performed using either a greedy or an analytical approach. In the analytical algorithm, the energy above the masking threshold is related to the audible distortion through the following empirical formula

$$D_i = c_i \, \mathcal{E}_i \, 2^{-b_i/\beta_i}, \tag{5.43}$$

where D_i is the energy of the audible noise in band i, $\mathcal{E}_i$ is the energy above the masking threshold, c_i and β_i are constants found from the corresponding rate-distortion curve for the codebook of band i.

The solution to the above is given by

$$b_i = \max\Big(\frac{\beta_i b_T}{\sum_{j=1}^{N_b} \beta_j} + \log_2\big(\frac{\mathcal{E}_i c_i}{\mathcal{E}_{gm}}\big), 0\Big), \tag{5.44}$$

where

$$\mathcal{E}_{gm} = \left(\prod_{i=1}^{N_b} (c_i \mathcal{E}_i)^{\beta_i} \right)^{\left(1/\sum_{i=1}^{N_b} \beta_i \right)} . \tag{5.45}$$

The integer parts of the b_i's are kept and the remaining bits will be distributed one at a time to the band which reduces the total distortion the most.

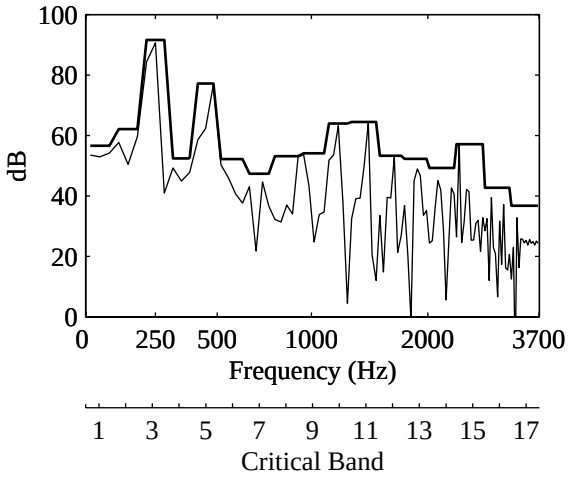
5.7.3 Comparison and Subjective Evaluation of the Bit Assignment Algorithms

Figure 5.7(a) shows the power spectrum and the Bark power spectrum of a frame of voiced speech on the Bark scale. The Bark power spectrum is convolved with the spreading function to obtain the excitation pattern. The excitation and the masking curves are shown in Fig. 5.7(b). As it is seen in Fig. 5.7(b), the offset level, which is subtracted from the excitation pattern, is larger at the low frequency critical bands. Notice that in bands 2, 4, 6 and 7 the energy falls below the masking threshold. The number of bits allocated to different critical bands using the two bit assignment algorithms is shown in Fig. 5.7(c) and Fig. 5.7(d). Comparing the two bit allocation algorithms, we notice that the energy-based algorithm (Fig 5.7(d)) allocates more bits to the bands with a large energy (for instance bands 1, 3, and 5). Both algorithms assign zero bits to the bands whose energy is below or almost below the masking threshold (bands 2, 4, 6, and 7). Note that in this example, since we have made the calculations for a single frame of data, we have ignored temporal masking effects.

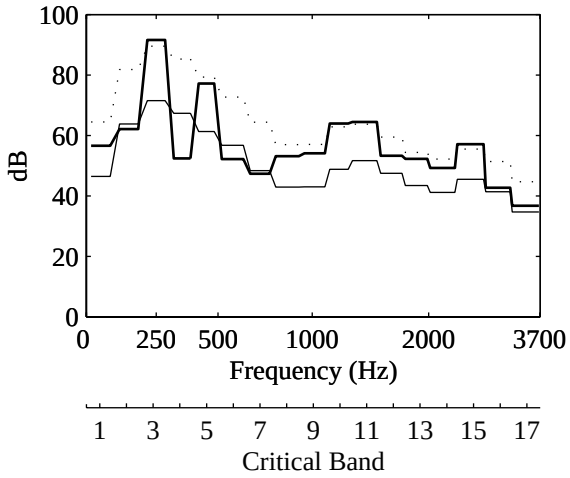
To evaluate the bit assignment algorithms, we performed informal listening tests. We used the perceptual bit assignment schemes, i.e., energy-based approach (the mini-max scheme and the minimization of the total distortion scheme) and the SMR-based algorithm to compress two speech files (male and female) and two pieces of music (soprano and guitar).

In the experiments the proposed narrowband perceptual audio coder (NPAC) was used. Operating at 8 kb/s (120 bits per frame), the coder assigns 81 bits to 17 bands to quantize the spectral shapes. The unquantized adjusted gains[5] were used to de-normalize the quantized shape vectors.

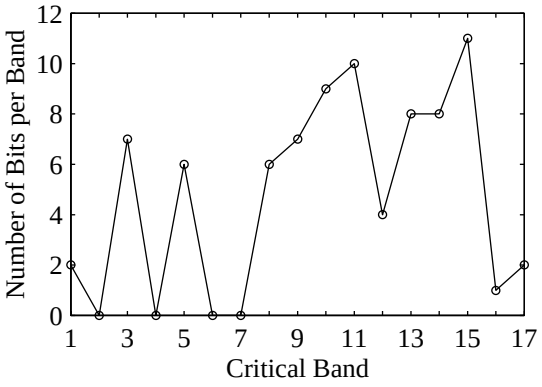
[5]The quantization error is reduced by adjusting the gain in each critical band through the following optimization $\rho_{\text{opt}} \approx \arg\min_\rho \sum_{k=1}^{K} (X(k) - \rho \hat{X}(k))^2$, where ρ is the gain adjustment factor, $X, \hat{X}$ are the original and quantized vectors of transform coefficients and K is the dimension of the subvector.



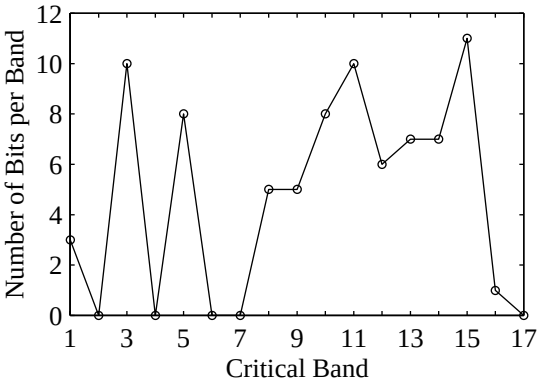
(a) Power spectrum and Bark power spectrum (bold curve).



(b) Bark power spectrum (bold curve), excitation curve (dotted curve) and masking curve (thin curve).



(c) SMR-based bit allocation.



(d) Energy-based (mini-max) bit allocation.

Fig. 5.7 Power spectrum, Bark power spectrum, excitation and masking curves for a frame of voiced speech. The lower plots show the bit allocation using the SMR-based and the Energy-based algorithms.

In the first test, we examined the impact of masking effects on the quality of the decoded signals. In that test, we ignored any masking effect and performed the bit assignment based on the distribution of the signal power. The resulting outputs have a higher SNR compared to the outputs using perceptual bit allocation. However, because a power-based scheme allocates many bits to the low frequency bands and relatively few bits to the high frequency bands, the outputs suffer from incomplete coding of the high frequencies. This result verifies the importance of incorporating the masking effects into any bit assignment algorithm.

The energy-based algorithm which minimizes the total audible distortion resulted in output quality similar to that for the power-based bit allocation. Due to different dimensionality of different critical bands, the distortion reduction rate is higher for the narrower low frequency bands. Moreover for many audio signals, the power is concentrated in the low frequency bands. Therefore, more bits compared to other perceptual schemes are assigned to the low frequency bands. This results in finer quantization of low frequency bands and coarser quantization of the high frequency bands.

The other schemes (the SMR-based and the mini-max) deliver better quality with less high frequency distortion. The results show that both algorithms produce decoded signals which can be distinguished from the original. The SMR-based algorithm causes less high frequency distortion at the expense of a little degradation in the pitch structure. Due to this degradation, the speech segments which are coded using the SMR-based algorithm sound harsher. On the other hand, the decoded audio signals using the energy-based algorithm carry higher levels of high frequency noise which sounds like an echo along with the original signal. Listeners showed a slight preference for the SMR-based allocation scheme over the mini-max scheme.

Therefore, we use the SMR-based bit allocation algorithm in the proposed coder. However, for the future, we believe that the perceptually-optimal bit allocation algorithm for low rate coding should be based on both the distribution of the audible noise and the SMR. This is a compromise between the schemes that might be better than either approach alone.

5.8 Variable Rate Coding

Johnston in [4] proposed a new concept called *perceptual entropy* as the minimum bit rate for transmitting audio signals such that there is no perceivable difference between the original and coded signal. Based on the perceptual entropy criterion, it is possible to use

a lossy compression scheme to code any audio signal without any perceivable distortion at a bit rate equal to its perceptual entropy.

We have conducted an experiment to calculate the number of bits required for each frame of data to achieve transparent quantization of shape vectors. Due to the variation in that number, we can use a source-based variable rate scheme to code audio data in packet-based networks.

To estimate the number of bits needed to achieve transparent coding of the spectral shapes, we use the SMR-based bit allocation algorithm. Table 5.2 shows the instantaneous minimum, the average and the instantaneous maximum bit rates for the shape quantization of the transform coefficients for different audio signals. Note that some frames are temporally masked; therefore no bits are required to code the shapes. We have to add 2.5 kb/s for the gain quantization to the figures in Table 5.2 except for those frames which are totally masked.

McCourt in [133] reports that for a fixed rate coder, a minimum of 11 kb/s is required to perform transparent adaptive vector quantization of the shape vectors. Although the maximum rates shown in Table 5.2 are comparable to the minimum rate reported in [133], the average required rates are much lower than that bit rate. One conclusion from Table 5.2 is that the proposed coder can provide high quality audio for any narrowband input if the maximum number of bits is spent to quantize the shape vectors. Note that around 2.5 kbit/s is also needed to quantize the gains.

Table 5.2 Instantaneous minimum, average and instantaneous maximum rates (kbit/s) for shape quantization.

File	Minimum	Average	Maximum
Female speech	0.0	7.2	11.5
Male speech	0.6	6.9	10.0
Piano	0.0	8.7	11.3
Orchestral	0.9	7.7	10.8

5.9 Performance Evaluation

The proposed coder has been designed to compress any narrowband audio signal. In the coder different processing units have been designed to efficiently reduce the bit rate while

maintaining good audio quality.

The number of bits used to code each frame of the input data is 120 (for a long frame) and 40 (for a short frame), i.e., 1 bit per sample. Almost all of the bits were spent to code the normalized transform coefficients and the gains. Table 5.3 shows the bit allocation for a long and a short frame of data.

Table 5.3 Bit allocation to code a frame of data.

Data	Long Frame	Short Frame
Shape Quantization	81	25
Gain Quantization	37	14
Window Switching Flag	1	1
Gain Quantization Flag	1	0
Total	120	40

We have implemented the proposed coder in the C language. The source code was written for flexible experimentation and not optimized for execution speed. Nevertheless, the coder runs in real time on a computer using a 450 MHz Pentium II processor.

5.9.1 Objective Evaluation

Although there are some perceptually-based measures [134, 60, 48, 135, 136] to evaluate the performance of speech and high rate audio coders, there is still no reliable objective criterion to evaluate the performance of low rate narrowband audio coders. We use a perceptually based criterion, which is the ratio of the energy of the input signal to the energy of the audible noise, for comparing the quality of the coded signal using different coders. This criterion, which we refer to as the Signal-to-Audible-Noise-Ratio (SANR), is calculated as follows:

- Each frame of the original and coded signals is transformed into the spectral components.
- Masking thresholds corresponding to each frame of the original signal are calculated.
- The energy of the audible noise is calculated for each frame.

- Finally the SANR is defined as follows:

$$\text{SANR} = \frac{\sum_{i=1}^{N_f} \| X^{(i)} \|^2}{\sum_{i=1}^{N_f} D^{(i)}} \tag{5.46}$$

where N_f is the number of frames, $X^{(i)}$ is the ith frame of the original signal and $D^{(i)}$ is the energy of the audible noise in the ith frame of the coded signal.

Based on our observations, this criterion is better correlated with subjective ratings than other criteria such as SNR and Segmental SNR. The accuracy of this criterion is strongly dependent on the accuracy of the auditory masking model of the hearing system. Note that for musical inputs there is a high correlation between the subjective quality of the reconstructed signal and the SANR, but for speech inputs the value of the SANR does not necessarily predict the quality for different coders.

5.9.2 Subjective Evaluation

In wideband audio coding, the compression process is transparent for most input material. Nevertheless, the crucial testing involves known difficult-to-code material. If the coder passes this test, it will be transparent for all audio inputs. For low rate narrowband coding, some distortion is inevitable. A wide range of material must be tested to ascertain that the distortion for all inputs is not annoying. In our case, we chose a representation set of material including various types of music, single instrumental music, single and multi-speaker speech, speech with background noise for testing the NPAC encoder.

It is difficult to make valid comparisons with existing coders as, to our knowledge, there is no other low rate coder accommodating both speech and music inputs. However, we have compared the quality of the coded signals using NPAC, the RealAudio[6] music coder operating at 8 kbit/s, the RealAudio speech coder operating at 8.5 kbit/s and the G.729 speech coder [20] operating at 8 kbit/s. The quality of the coded signals were evaluated through informal tests. Eight test audio files including speech, multi-speaker and various

[6]RealAudio is a trademark of RealNetworks, Inc.

music types were presented over headphones to five untrained listeners. Note that none of the test passages was used in training the quantizers of the NPAC encoder.

In the listening test, the compressed signals were distinguishable from the originals. However, the purpose of the test was to know whether the distortions in the output signals were annoying. Due to narrowband nature, we expect the quality at the best of circumstances to be similar to the that of AM broadcast radio.

Compared to the 8 kbit/s RealAudio transform coder, the listeners unanimously believed that the proposed coder delivered significantly better quality for most music passages and never performed worse than the RealAudio music coder. For all speech signals, the proposed coder provided much better quality than the RealAudio music coder.

Compared to the G.729 coder and the 8.5 kbit/s RealAudio speech coder, the listeners preferred the quality of almost all compressed signals using the NPAC coder. The exceptions were for the files containing a single speaker. Even for these cases, the quality was not far below that of the speech coders. Based on our experiments this coder works well as long as there is no strong harmonic structure due to voiced speech. In the case of the pseudo-periodic parts of the input signal, due to the sensitivity of the human ear to small variations of the harmonic structure, some distortion is perceived. However, to our best knowledge, NPAC is the only coder that operates well for a wide variety of narrowband audio data at 8 kbit/s.

In regard to the best expected quality (mentioned above), NPAC met the expectations for almost all test passages. However, some enhancements should be made to NPAC in order to achieve the same quality for single speaker passages as the quality delivered by speech-specific coders such as G.729.

Coding of Speech

The quality of clean voiced speech coded with NPAC is not as good as that of state-of-the-art speech coders such as the G.729 coder. We speculated that it might be caused by the degradation of the pitch structure of voiced speech as NPAC does not explicitly model pitch. In order to verify this hypothesis, we obtained the pitch contour for many speech signals compressed with the G.729 coder and NPAC. The pitch contours were obtained using the pitch estimator algorithm of the G.729 coder. We compared those pitch contours with the pitch contour for the original signals. We observed that the pitch contours of the

signals coded with NPAC were close to the pitch contours for the original signals, even sometimes closer to the originals than those of the signals processed with the G.729 coder. Therefore, we must conclude that, the problem is not entirely due to pitch destruction. Then we hypothesized that the distortion of the spectral envelope might have a role in this problem. This hypothesis was verified since when we replaced the quantized gain factors with the original gain factors, we got better speech quality.

Another source of problems comes from the MDCT. Quantization of the MDCT coefficients causes some uncancelled time aliased components which degrades the speech quality. We also believe that the timbre[7] of speech must be reproduced accurately. Destruction of the timbre produces some distortion even if we keep the pitch structure intact.

One observation made us ponder while we compared the performance of the G.729 coder and NPAC on pieces of single instrumental music with a harmonic structure. We realized that the performance of NPAC is similar to or better than that of the G.729 coder. This observation raised up the question as to why we have different performance of NPAC for single instrumental music and voiced speech. We think that a better performance of the G.729 coder comes from the fact that speech-specific coders such as the G.729 coder are heavily optimized for speech. Moreover, according to many scientists, the human auditory system is highly sensitive to any distortion in speech as different parts of the brain process speech and non-speech stimuli [40]. Some scientists believe that there is a "special mode" for the perception of speech which activates automatically when one listens to speech sounds [40]. This special mechanism requires high accuracy in the compression of speech signals.

[7]Timbre is the attribute of a sound that allows us to differentiate between two sounds of the same pitch, intensity and duration [40].

Chapter 6

Concluding Remarks

The purpose of our research has been to develop a coding structure operating at low bit rates down to 8 kbit/s and delivering moderate audio quality for narrowband audio signals sampled at 8 kHz. To accomplish our goal, the proposed *Narrowband Perceptual Audio Coder (NPAC)* employs a variety of perceptual-based algorithms to remove the perceptually irrelevant parts of the input signal in addition to statistical redundancies. The new algorithms used in the coder include a perceptual error measure in training the VQ codebooks and selecting the best codewords which takes into account the audible parts of the quantization noise, perceptually-based bit allocation algorithms, and an adaptive predictive scheme to vector quantize the scale factors. We have used the Signal-to-Mask Ratio (SMR) measure to find the upper bound of the bit rate for the quantization of the spectral shapes. This upper bound along with the ease of the coder makes it possible to trade off quality versus rate for applications such as data packet based networks . This coder can easily be modified to accommodate a wider range of input signals with different bandwidth and sampling rates.

6.1 Summary of Our Work

In Chapter 1, we expressed the emerging demands for a universal coder capable of accommodating a wide range of narrowband audio data (band-limited to around 4 kHz) at low bit rates down to 8 kbit/s. Specifically, we mentioned some new applications such as broadcasting over Internet, AM broadcasting and satellite communications in which either the available bandwidth is limited or the number of users is large. While state-of-the-art

speech coders provide high quality of speech at 8 kbit/s and below, they perform poorly on non-speech inputs. A gap has existed between the operating bit rates of low rate audio coders and that of speech coders at 8 kbit/s. The challenge is to fill the gap with an appropriate coding structure.

In Chapter 1 the major classes of coding paradigms, i.e., Parametric (source) coding, Hybrid coding and Waveform coding were discussed. Waveform coding in the frequency domain has been chosen as the best alternative for the coding of general audio signals. More specifically, from the two variants of frequency domain coding paradigms, i.e., Subband Coding and Transform Coding, the latter was preferred for reasons which include the existence of fast transforms, higher frequency resolutions and the ease of incorporating masking models into the coder. Finally, the basic structure of perceptual audio coders was presented.

Chapter 2 started with an overview of the physiology of the human ear. The importance of the basilar membrane was pointed out since it decomposes the input signal into its spectral components. Due to the structure of the basilar membrane, it behaves like a nonuniform filterbank (i.e., auditory filterbank). The important concept of critical bands, which approximate the bandwidth of the auditory bandpass filters, was discussed. The auditory masking phenomena were described. The masking phenomena have two main forms, i.e., simultaneous and temporal masking. The physiological basis and psychoacoustical evidence for both were examined. Several widely used masking models were described.

Chapter 3 provided a detailed analysis of lapped transforms. Lapped transforms are a proper choice for transform coders because they perform on overlapping blocks of data which reduces block edge effects. Modulated Lapped Transforms (MLT) or Modified Discrete Cosine Transforms (MDCT) were analyzed. Modulated Lapped Transforms are computationally very efficient as the equivalent filterbank is produced through modulating cosine functions by a prototype low pass time window. The effect of the prototype window on the frequency response of the resulting filterbank was investigated. An optimization procedure to design a good window by trading the width of the transition band versus the stopband attenuation was presented. Lapped Orthogonal Transforms (LOT) in which an identical window is used in the analysis and synthesis stages were compared with Lapped Biorthogonal Transforms (LBT) in which two different windows are used. A new family of windows derived from the Chebyshev polynomial with two tuning parameters was presented. The performance of a number of different windows was investigated using the

coding gain formula [74]. It was found that as long as the window goes smoothly to zero at the boundaries, there is no great difference between the windows. Finally, the issue of adaptive filterbanks was addressed and a window switching method was analysed as a form of adaptive filterbank to reduce pre-echo artifacts in audio coding.

In Chapter 4, we briefly reviewed two main classes of compression schemes widely used in audio coding, i.e., lossy and lossless schemes. In lossless schemes no information is lost during the compression process whereas lossy methods cause some loss of information. However a new terminology called perceptually transparent coding was introduced in which despite some loss of information, no difference between the original and reconstructed signal can be perceived by the ear. Two main lossy schemes, i.e., scalar and vector quantization were described. Nonuniform scalar quantization methods are suitable for a scalar source with a nonuniform probability distribution function. Vector Quantization is a more efficient scheme compared to scalar quantization. It was argued that vector quantization systems provide higher coding gains at the expense of more complexity. The Generalized Lloyd Algorithm (GLA) and the LBG algorithm were briefly reviewed as two iterative methods to design vector quantization systems. A new perceptually based distortion measure was proposed which takes into account the audible part of the quantization noise. That measure was used to design more efficient vector quantization systems for audio coding. In the rest of Chapter 4, some widely used audio coders and the MPEG audio standards including MPEG-1, MPEG2 and MPEG-4 were briefly described.

Chapter 5 introduced the proposed coding structure called Narrowband Perceptual Audio Coder (NPAC). We have described different blocks of the coder along with the related algorithms. An MDCT was used to decompose the input signal into its spectral components. The MDCT coefficients were grouped into 17 subbands to emulate the frequency analysis in the ear. To quantize the transform coefficients, a Gain/Shape approach was taken. The shape vectors were quantized using the perceptually-trained codebooks along with the perceptually-based bit allocation. A number of bit allocation algorithms based on the auditory masking properties were introduced. The relative merits of the algorithms were compared and the SMR-based bit allocation method was preferred over the energy-based bit allocation. To reduce the required memory to store the shape codebooks, a few methods have been suggested and one of them which is related to the source entropy was chosen to design a single embedded codebook for each subband. In the process of quantization, the temporal and simultaneous masking thresholds were used to determine the

acceptable noise level. For simultaneous masking, we have developed a new formulation to obtain the masking thresholds for the MDCT coefficients.

An adaptive predictive/nonpredictive vector quantization scheme has been used to quantize the gains. Different methods to reduce the number of nonzero diagonals of the predictor matrices have been proposed and analyzed. Finally, the upper bound for the bit rate for transparent coding of shape vectors has been found using the SMR-based bit allocation algorithm. That upper bound shows that the information content of the audio data is time-varying and for packet-based networks such as Internet, a coding algorithm with several operating bit rates is appropriate. The subjective performance of NPAC was compared to the low rate RealAudio Coders and the G.729 speech coder.

6.2 Further Enhancements of the NPAC Encoder

Our prime goal has been to compress narrowband audio signals at low rates (1 bit per sample) while achieving acceptable quality. As we have pointed out, we expect to get moderate audio quality contrary to high rate audio coders where transparent coding is the goal. NPAC performs well on most audio signals and outperforms other low rate *audio* coders operating at 8 kbit/s. However, we believe that there is still room for the enhancement of the proposed coder performance. In the following we discuss possible improvements to different modules of the coder.

6.2.1 Quantization of the MDCT Coefficients

We spend 120 bits to encode each block of 240 time samples (120 from the previous frame and 120 new samples). As our bit budget is limited to 8 kbit/s, we had to develop algorithms which were suitable for this constraint. Had the coder been operating at higher bit rates, our approach would have been quite different. For instance, we had to sacrifice the quality for reducing the bit rate in a few modules of NPAC. In order to achieve high coding gains, we have used vector quantization schemes to code different parameters. Vector quantization increases the complexity of the encoder and demands more memory to store the codebooks. The most memory demanding part of the coder is the VQ of the shape vectors. Since we use an adaptive bit allocation algorithm, we have to have different codebooks with different lengths for each subband. However, we noticed that the probabilities of selecting the codewords were widely different. This observation led us to consider entropy coding in

order to reduce the bit rate. However, the drawback of entropy coding is more complexity and a variable bit rate which is not compatible with fixed-rate channels. Instead, we ran a large set of test vectors on the largest codebook for each subband, and then ordered the codewords based on their frequency of selection (the most selected one comes at the top). This way we created a single embedded codebook from which different numbers of codewords are used to encode an input vector. For an example, Figure 6.1 shows the probability of selection of the codewords belonging to a 3 dimensional codebook with 512 codewords designed for the second critical band. Although the length of the codebook is 512 (equivalent to 9 bits), the actual entropy of a large set of 30000 test vectors is about 7 bits. This shows that some codevectors are more frequently selected while most of them rarely selected. We believe that better training of the codebooks with more frequently

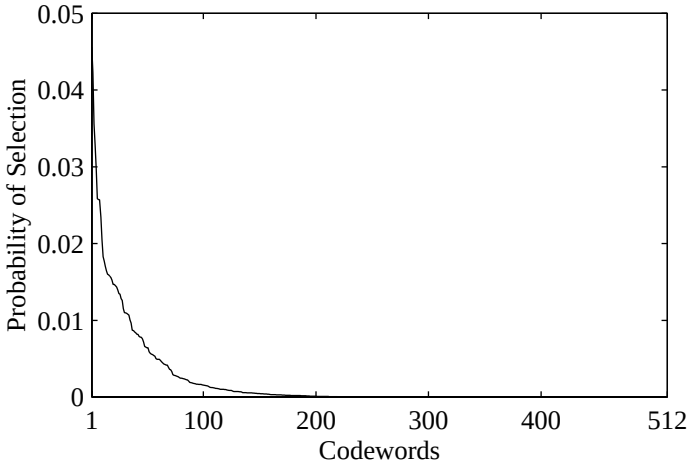


Fig. 6.1 The selection probability of the codewords from a 3 dimensional codebook of 512 codewords.

selected codewords might improve the performance of the shape quantization module.

One last thing that might be exploited is the correlation among the indices of the selected codewords for different subbands. We briefly investigated this issue but did not get improved performance. A more structured vector quantization system might exploit this correlation.

6.2.2 Quantization of the Scale Factors

The scale factors are coded using a predictive/nonpredictive vector quantization scheme. We allocate 37 bits for that purpose, i.e., 30% of the bit budget for a frame of data. We

might replace the existing coding scheme with a more efficient one. We tried a linear prediction analysis to estimate the power spectral density. This way we reduced the required bits to around 20 bits. The resulting scheme causes significant degradation of the compressed signal because we use an MDCT to decompose the input signal and the grouping of the MDCT coefficients is nonuniform. We even performed the IDFT on the MDCT spectrum and then estimated the spectrum using a linear prediction analysis. The result still was unsatisfactory. We believe that more research should be done to reduce the bits needed for this part of the proposed coder.

6.2.3 Masking Threshold

The accuracy of the masking model has a great impact on the performance of the proposed coder as we have incorporated the masking phenomena in different modules of the coder. We have used a modified version of the model proposed by Johnston [51] to estimate the simultaneous masking threshold. Since that model is DFT-based, we have modified the resulting masking thresholds to find the corresponding masking threshold for the MDCT coefficients. We believe that there is room for improving the original model as that model linearly sums the individual masking powers to find the global masking threshold. Since a linear model underestimates the masking threshold, a nonlinear model, e.g., some power law, would better fit the experimental data.

Concerning the temporal masking model, we have fitted polynomials to the experimental data to develop the temporal masking model. A more sophisticated model based on physiological evidence and psychoacoustical data might improve the performance of the coder. A phenomenon called overshoot effect, i.e., a jump in the masking threshold around a high energy attack, needs to be studied to see how important it is to code large jumps. Finally some research can be done on the more accurate model to combine the simultaneous and temporal masking effects.

6.2.4 Bit Allocation Algorithms

An appropriate bit allocation plays a large role in providing good audio quality. We have investigated different strategies to allocate bits. Our conclusion is that an optimal bit allocation algorithm should take into account both the distribution of the energy and the ratio of the signal energy to the masking threshold. The SMR-based algorithm is sensitive

to the masking models. Therefore a more accurate masking model would improve the performance of the SMR-based algorithm. The energy based algorithm usually allocated too many bits to the low frequency bands which results in coarse quantization of high frequency bands. Combining the SMR-based algorithm with the energy-based algorithm could result in better quality of the reconstructed signal.

6.3 Future Research

In the previous section we discussed possible enhancements to the NPAC encoder. In this section, we make some suggestions for future research on more general aspects of low rate audio coding.

- *Scalability:* The proposed coder has been designed to produce a constant bit rate suitable for fixed-rate channels. However, the coder structure is flexible enough to produce variable bit rates. A modified version of the proposed coder to handle input signals with different bandwidth and sampling rates has been developed by S. Plain [137]. For narrowband audio inputs (band-limited to 4 kHz), we just need to change the number of bits spent to quantize the shape vectors. This way we can easily trade quality versus bit rates.

- *Robustness:* For wireless applications, some work needs to be done to evaluate the sensitivity of the bit stream to channel errors. Then an appropriate protection scheme should be added to the coder to make it robust against channel effects. In packet-based networks, isolated erroneous bits are not the main concern; instead some measures should be taken to replace lost packets of data.

- *Objective Evaluation:* Since subjective testing is costly and time consuming, objective methods to evaluate low rate audio coders are appealing. Objective methods for the evaluation of speech and high quality audio have been investigated [134, 60, 48, 135, 136]. However, those methods fail to accurately evaluate the moderate quality provided by low rate audio coders such as NPAC. Although we have modified the traditional objective measure SNR to a new perceptual measure called Signal-to-Audible-Noise Ratio (SANR) to evaluate the performance of our coder, it is not necessarily appropriate for evaluating other low rate coders. One future avenue would be to develop objective measures for the evaluation of low rate audio coders.

- *Speech-specific Mode Coding:* Speech is processed by a specific part of the brain. The human hearing system is very sensitive to any distortion in voiced speech. Some parametric-based modules might be added to the proposed coder to achieve better quality for voiced speech. Those parameters might be the pitch period and the envelope of the harmonics to maintain the pitch and timbre of the voiced input.

- *Object-based Audio Coding:* This new audio coding paradigm seems very promising. It is based on the decomposition of complex audio signals into some audio sources which can be modeled with a few parameters. This way the bit rate can be considerably reduced. However, some issues such as the best way to decompose a complex signal, how to model different audio objects and perceptually-based quantization of the parameters, need to be worked out.

Appendix A

Relation between the DFT and MDCT

The MDCT of a frame of input signal $x(n)$ is defined as [61]

$$C(k) = \sqrt{2/M} \sum_{i=0}^{N-1} x(n)h(n) \cos\Big(\frac{\pi}{M}(n+n_0)(k+0.5)\Big) \tag{A.1}$$

where $h(n)$ is the window function, N is the length of the input frame, $M = N/2$ is the number of transform coefficients in each frame and n_0 is a constant equal to $(M+1)/2$. Write the above formula as

$$C(k) = \sqrt{2/M} \sum_{i=0}^{N-1} \Re\{x(n)h(n) \exp(\frac{-j\pi(n+n_0)(k+0.5)}{M})\} \tag{A.2}$$

$$= \sqrt{2/M}\Re\{\exp\big(j\phi(k)\big)\mathcal{F}\big(s(n)\big)\} \tag{A.3}$$

where $\Re$ denotes the real part and $\mathcal{F}$ denotes the Fourier transform,

$$\phi(k) = \frac{-\pi(N+2)(k+0.5)}{2N} \tag{A.4}$$

$$s(n) = \exp(\frac{-j\pi n}{N})x(n)h(n) \tag{A.5}$$

Finally we get

$$C(k) = \sqrt{2/M}|S(k)|\cos\big(\frac{2\pi n_0(k+0.5)}{N} - \angle S(k)\big) \tag{A.6}$$

Appendix B

A Family of Chebyshev-derived Windows

We modify the magnitude response of a Chebyshev filter to satisfy the perfect reconstruction conditions. Start with the coefficient for a lowpass Chebyshev filter.

$$[\boldsymbol{b} \quad \boldsymbol{a}] = \texttt{cheby1}(N, \alpha_1, \alpha_2); \tag{B.1}$$

where `cheby1` is a MatLab command which generates the coefficients of a Chebyshev filter, N is the length of the window, α_1 and α_2 are two parameters determined by the designer.

$$w = \texttt{abs}(\texttt{freqz}(\boldsymbol{b}, \boldsymbol{a}, N/2)^t); \tag{B.2}$$

where `freqz` is a MatLab command which gives the frequency response of a digital filter.

$$w_n(n) = \frac{w(n)}{w(n)^2 + w(\frac{N}{2} - 1 - n)^2}, \qquad n = 0, ..., N/4 - 1. \tag{B.3}$$

Then we find the window coefficients as follows

$$h_{\text{Cheb}}(n) = w_n(N/4 - 1 - n), \qquad n = 0, ..., N/4 - 1. \tag{B.4}$$

Using the perfect reconstruction constraints, we find the rest of the window coefficients.

$$h_{\text{Cheb}}(n) = (1 - {h_{\text{Cheb}}}^2(N/2 - 1 - n))^{0.5}, \qquad n = N/4, ..., N/2 - 1. \tag{B.5}$$

$$h_{\text{Cheb}}(n) = h_{\text{Cheb}}(N-1-n), \qquad n = N/2, ..., N-1. \tag{B.6}$$

References

[1] A. Gersho, "Advances in Speech and Audio Compression," *Proc. IEEE*, vol. 82, pp. 900–918, June 1994.

[2] K. Pohlmann, *Principles of Digital Audio*. McGraw Hill, 1995.

[3] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[4] J. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (New York), pp. 2524–2527, 1988.

[5] M. Dietz, J. Herre, B. Teichmann, and K. Brandenburg, "Bridging the Gap: Extending MPEG Audio down to 8 kbit/s," in *102nd AES Convention*, (Munich), 1997. Preprint 4508.

[6] M. Dietz, H. Popp, K. Brandenburg, and R. Friedrich, " Audio Compression for Network Transmission ," *J. Audio Eng. Soc.*, vol. 44, pp. 58–72, Jan. 1996.

[7] B. Feiten, R. Schwalbe, and F. Feige, "Dynamically scalable internet audio transmission," in *104th AES Convention*, (Amsterdam), 1998. Preprint 4686.

[8] R. Buchta, S. Meltzer, and O. Kunz, "The WorldStar Sound Format," in *101st AES Convention*, (Los Angeles), 1996. Preprint 4385.

[9] *Digital Broadcasting in A.M. Frequency Bands*. European Union Project 1559 NADIB, 1996. *http://www.sarc.sk/akcie/eureka/data/1559.html*.

[10] *The MPEG Homepage*. *http://drogo.cselt.stet.it/mpeg/*.

[11] F. Rumsey, "Putting Low-Bit-Rate Audio to Work," in *Collected Papers on Digital Audio Bit-Rate Reduction* (N. Gilchrist and C. Grewin, eds.), pp. 155–163, Audio Engineering Society, 1996.

[12] M. Sablatash and T. Cooklev, "Compression of high quality audio signals, including recent methods using wavelet packets," *Digital Signal Processing*, vol. 6, pp. 96–107, Apr. 1996.

[13] G. Davidson, L. Fielder, and M. Anil, "High quality audio transform coding at 128 kbps," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Albuquerque), pp. 1117–1120, 1990.

[14] K. Brandenburg, "OCF - A New Coding Algorithm for High Quality Sound Signals," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Dallas), pp. 141–144, 1987.

[15] K. Brandenburg, G. Stoll, Y. Dehery, J. D. Johnston, L. V. Kerkhof, and E. F. Schroeder, "The ISO/MPEG Audio Codec: A Generic Standard for Coding of High Quality Digital Audio," *J. Audio Eng. Soc.*, vol. 42, pp. 780–791, Oct. 1994.

[16] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 Advanced Audio Coding," in *101st AES Convention*, (Munich), 1996. Preprint 4382.

[17] E. Zwicker and T. Zwicker, "Audio Engineering and Psychoacoustics. Matching Signals to the Final Receiver, the Human Auditory System ," *J. Audio Eng. Soc.*, vol. 39, pp. 115–126, Mar. 1991.

[18] E. D. Scheirer, "The MPEG-4 Structured Audio Standard," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Seattle), pp. 3801–3804, 1998.

[19] A. S. Spanias, "Speech Coding: A Tutorial Review," *Proc. IEEE*, vol. 82, pp. 1541–1582, Oct. 1994.

[20] K. Jarvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, R. Salami, C. Laflamme, and J. P. Adoul, "GSM Enhanced Full Rate Speech Codec," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Munich), pp. 771–774, 1997.

[21] T. Honkanen, J. Vainio, K. Jarvinen, P. Haavisto, R. Salami, and C. L. J. P. Adoul, "Enhanced Full Rate Speech Codec for IS-136 Digital Cellular System," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Munich), pp. 731–734, 1997.

[22] A. DeJaco, W. Gardner, P. Jacobs, and C. Lee, "QCELP:The North American CDMA Digital Cellular Variable Rate Speech Coding Standard," in *Proc. IEEE Workshop on Speech Coding for telecommunications*, (Ste. Adele), pp. 5–6, 1993.

[23] N. Spencer, "An Overview of Digital Telephony Standards," in *Proc. IEE Colloquium on the Design of Digital Cellular Handsets*, pp. 1–7, Mar. 1998.

[24] H. Purnhagen, B. Edler, and C. Ferekidis, "Object-Based Analysis/Synthesis Audio Coder for Very Low Bit Rates," in *104th AES Convention*, (Amsterdam), 1998. Preprint 4747.

[25] B. Edler, "Very Low Bit Rate Audio Coding Development," in *Proc. AES 14th International Conference*, (Seattle, WA), 1997. *http://www.tnt.uni-hannover.de/project/coding/audio/asac/aes_iao.html*.

[26] S. Fururi and M. Sondhi, *Advances in Speech Signal Processing*. Marcel Dekker, Inc., 1992.

[27] P. Mermelestein, "G.722, a New CCITT Coding Standard for Digital Transmission of Wideband Audio Signals," *IEEE Commun. Mag.*, vol. 26, pp. 8–15, Jan. 1988.

[28] R. M. Gray, *Fundamental of Data Compression*. Invited talk at the International Conference on Information, Communications, and Signal Processing, Singapore, Sept. 1997.

[29] R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 25, pp. 299–309, Aug. 1977.

[30] R. Zelinski and P. Noll, "Approaches to Adaptive Transform Speech Coding at Low Bit Rates," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 27, pp. 89–95, Feb. 1979.

[31] J. M. Tribolet and R. E. Crochiere, "Frequency Domain Coding of Speech," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-27, pp. 512–530, Oct. 1979.

[32] M. Schroeder, B. Atal, and J. Hall, "Optimizing Speech Coders by Exploiting Masking Properties of the Human Ear," *J. Acoust. Soc. Am.*, vol. 66, pp. 1647–1652, June 1979.

[33] H. Najafzadeh-Azghandi and P. Kabal, "Perceptual Coding of Narrowband Audio Signals at 8 kb/s," in *Proc. IEEE Workshop on Speech Coding*, (Pocono Manor, Penn.), pp. 109–110, 1997.

[34] P. Kabal and H. Najafzadeh, *Perceptual Audio Coding*. US and Canada patent applications, filed Sept. 1998.

[35] H. Najafzadeh-Azghandi and P. Kabal, "Improving Perceptual Coding of Narrowband Audio Signals at Low Rates," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Phoenix, Arizona), pp. 913–916, 1999.

[36] *Compton's Encyclopedia*. Proteus Enterprises Inc, 1999.

[37] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*. Springer-Verlag, 1999.

[38] *Physics and Psychoacoustic of Music*. The Australian Centre for the Arts and Technology. *http://www.anu.edu.au/ITA/ACAT/drw/*.

[39] *Anatomy and Function of the Ear*. Department of Otolaryngology at the University of Washington. *http://depts.washington.edu/otoweb/ear_anatomy.html*.

[40] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Academic Press, fourth ed., 1997.

[41] S. Cool, "Anatomy of the Ear," 1997. *http://www.anatomy.uq.edu.au/gmc/tutorials/ear*.

[42] E. B. Goldstein, *Sensation and Perception*. Wadsworth, 1989.

[43] J. B. Allen and S. T. Neely, "Micromechanical Models of the Cochlea," *Physics Today*, vol. 45, pp. 40–47, July 1992.

[44] B. C. J. Moore, "Masking in the Human Auditory System," in *Collected Papers on Digital Audio Bit-Rate Reduction* (N. Gilchrist and C. Grewin, eds.), pp. 9–19, Audio Engineering Society, 1996.

[45] E. Terhardt, G. Stoll, and M. Seewann, "Algorithm for Extraction of Pitch and Pitch Salience from Complex Tonal Signals," *J. Acoust. Soc. Am.*, vol. 71, pp. 679–688, Mar. 1982.

[46] X. Durot and J. B. Rault, "A New Noise Injection Model for Audio Compression Algorithm," in *101st AES Convention*, (Los Angeles), 1996. Preprint 4374.

[47] W. Deutsch and A. Noll, "The Perception of Audio Signals Reduced by Overmasking to the Most Prominent Spectral Amplitudes (Peaks)," in *92nd AES Convention*, (Vienna), 1992. Preprint 3331.

[48] J. G. Beerends and J. A. Stemerdink, " A Perceptual Speech-Quality Measure Based on a Psychoacoustic Sound Representation," *J. Audio Eng. Soc.*, vol. 42, pp. 115–123, Mar. 1994.

[49] T. Thiede and E. Kabot, " A New Perceptual Quality Measure for Bit Rate Reduced Audio," in *100th AES Convention*, (Copenhagen), 1996. Preprint 4280.

[50] G. A. Soulodre, *Adaptive Methods for Removing Camera Noise from Film Soundtracks*. PhD thesis, McGill University, Montreal, Canada, 1998.

[51] J. D. Johnston, "Transform Coding of Audio Signals Using the Perceptual Noise Criteria," *IEEE J. Selected Areas in Comm.*, vol. 6, pp. 314–323, Feb. 1988.

[52] International Standard ISO/IEC DIS 13818-7, *Generic coding of moving pictures and associated audio information (Part7)- Advanced Audio Coding (AAC)*, 1966.

[53] R. Veldhuis, "Bit Rates in Audio Source Coding," *IEEE J. Selected Areas in Comm.*, vol. 10, pp. 86–96, Jan. 1992.

[54] M. Lynch, E. Ambikairajah, and A. Davis, "Comparison of Auditory Masking Models for Speech Coding," in *Eurospeech*, (Rhodes, Greece), 1997.

[55] International Standard ISO/IEC JTC1/SC29/WG 11, *Coding of moving pictures and associated audio- Audio*, 1993.

[56] Y. Huang and T. Chiueh, "A New Forward Masking Model and its Application to Perceptual Audio Coding," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Phoenix, Arizona), pp. 665–668, 1999.

[57] A. Harma, "Psychoacoustic Temporal Masking Effects with Artificial and Real Signals," in *Hearing Seminar*, (Espoo, Finland), 1999.

[58] T. Sporer, U. Gbur, J. Herre, and R. Kapust, "Evaluating a Measurement System," *J. Audio Eng. Soc.*, vol. 43, pp. 353–362, May 1995.

[59] B. Novorita, "Incorporation of temporal masking effects into bark spectrum distortion measure," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Phoenix, Arizona), pp. 665–668, 1999.

[60] R. J. Beaton, J. G. Beerends, M. Keyhl, and W. Treurniet, "Objective Perceptual Measurement of Audio Quality," in *Collected Papers on Digital Audio Bit-Rate Reduction* (N. Gilchrist and C. Grewin, eds.), pp. 126–152, Audio Engineering Society, 1996.

[61] H. Malvar, *Signal Processing with Lapped Transforms*. Artech House, 1992.

[62] H. S. Malvar, "Lapped Transform for Efficient Transform/Subband Coding," *IEEE Trans. Acoustics, Speech, Signal Processing*, pp. 969–978, June 1990.

[63] H. S. Malvar and D. H. Staelin, "The LOT: Transform Coding Without Blocking Effects," *IEEE Trans. Acoustics, Speech, Signal Processing*, pp. 553–559, Apr. 1989.

[64] K. Nayebi, T. P. Barnwell, and M. Smith, "Time-Domain Filter Bank Analysis: A New Design Theory," *IEEE Trans. Signal Processing*, pp. 1412–1429, June 1992.

[65] M. Vetterli and D. L. Gall, "Perfect Reconstruction FIR Filter Banks: Some Properties and Factorizations," *IEEE Trans. Acoustics, Speech, Signal Processing*, pp. 1057–1071, July 1989.

[66] P. P. Vaidyanathan, "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *IEEE Proceedings*, pp. 56–93, Jan. 1990.

[67] J. P. Princen and A. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time-Domain Aliasing Cancellation," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 34, pp. 1153–1161, Oct. 1986.

[68] J. Princen, A. Johnson, and A. Bradley, "Subband/Transform Coding using Filter Bank Designs Based ON Time Domain Aliasing Cancellation ," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Dallas, TX), pp. 2161–2164, 1987.

[69] S. C. Chan, "The Generalized Lapped Transform (GLT) for Subband Coding Applications," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 1508–1511, 1995.

[70] H. S. Malvar, "Lapped Biorthogonal Transforms for Transform Coding with Reduced Blocking and Ringing Artifacts," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Munich), pp. 2421–2424, 1997.

[71] T. Ramstad and J. Tanem, "Cosine-Modulated Analysis-Synthesis Filterbank with Critical Sampling and Perfect Reconstruction," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Toronto), pp. 1789–1792, 1991.

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

[73] T. Lookabaugh and M. Perkins, "Application of the Princen-Bradley filtebank to speech and image compression," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, pp. 1914–1926, Nov. 1990.

[74] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice Hall, 1984.

[75] P. Monta and S. Cheung, "Low Rate Audio Coder with Hierarchical Filterbanks and Lattice Vector Quantization," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (South Adelaide, Australia), pp. 209–212.

[76] J. Princen, "The Design of Nonuniform Modulated Filterbanks," *IEEE Trans. Signal Processing*, vol. 43, pp. 2550–2560, Nov. 1995.

[77] S. Wada, " Design of Nonuniform Division Multirate FIR Filter Banks," *IEEE Trans. Cir. and Sys.*, vol. 42, pp. 115–121, Feb. 1995.

[78] K. Nayebi, T. P. Barnwell, and M. Smith, "The Design of Perfect Reconstruction Nonuniform Band Filter Banks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Toronto), pp. 1781–1784, 1991.

[79] J. Li, T. Q. Neguyen, and S. Tantaratana, "A Simple Design Method for Nonuniform Multirate Filter Banks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 1015–1019, 1995.

[80] R. L. Queiroz and K. R. Rao, "Time-Varying Lapped Transforms and Wavelet Packets," *IEEE Trans. Speech and Audio Processing*, pp. 3293–3305, Dec. 1993.

[81] B. Carnero and A. Drygajlo, "Perceptual Speech Coding and Enhancement Using Frame-Synchronized Fast Wavelet Packet Transform Algorithms," *IEEE Trans. Signal Processing*, vol. 47, pp. 1622–1635, June 1999.

[82] D. Sinha and A. Tewfik, "Low Bit Rate Transparent Audio Compression using Adapted Wavelets," *IEEE Trans. Signal Processing*, vol. 41, pp. 3463–3479, Dec. 1993.

[83] P. Philippe, F. M. de Saint Martin, and M. Lever, "Wavelet Packet Filterbanks for Low Time Delay Audio Coding," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 310–322, May 1999.

[84] P. E. Kudumakis and M. B. Sandler, "On the Performance of Wavelets for Low Bit Rate Coding of Audio Signals," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 3087–3090, 1995.

[85] M. Purat and P. Noll, "Audio Coding With Dynamic Wavelet Packet Decomposition Based on Frequency-Varying Modulated Lapped Transforms," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Atlanta), pp. 1021–1024, 1996.

[86] A. J. S. Ferreira, *Spectral Coding and Post-Processing of High Quality Audio.* PhD thesis, University of Porto, Portugal, 1998.

[87] S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing.* PhD thesis, Stanford University, 1998.

[88] J. Princen and J. Johnston, "Audio Coding with Signal Adaptive Filterbanks," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 3071–3074, 1995.

[89] I. Sodagar, K. Nayebi, T. Barnwell, and M. Smith, "Time-Varying Analysis-Synthesis Systems Based on Filter Banks nad Post Filtering," *IEEE Trans. Signal Processing*, vol. 43, pp. 2512–2524, Nov. 1995.

[90] D. Sinha and J. Johnston, "Audio Compression at Low Bit Rates using a Signal Adaptive Switched Filterbank," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Atlanta, GA), pp. 1053–1056, 1996.

[91] K. Nayebi, T. P. Barnwell, and M. Smith, "Analysis-Synthesis System with Time-Varying Filter Bank Structures," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Francisco), pp. 617–620, 1992.

[92] M. Iwadare, A. Sugiyama, F. Hazu, A. Hirano, and T. Nishitani, " A 128 kb/s Hi-Fi Audio CODEC Based on Adaptive Transform Coding with Adaptive Block Size MDCT ," *IEEE J. Selected Areas in Comm.*, vol. 10, pp. 138–144, Jan. 1992.

[93] A. Sugiyama, F. Hazu, M. Iwadare, and T. Nishitani, "Adaptive Transform Coding with Adaptive Block Size(ATC-ABS)," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Albuquerque), pp. 1093–1096, 1990.

[94] J. Herre and J. D. Johnston, "Enhancing the performance of perceptual audio coders by using temporal noise shaping TNS," in *101st AES Convention*, (Los Angeles), 1996. Preprint 4384.

[95] J. Storer, *Data Compression*. Computer Science Press, 1988.

[96] S. P. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, Mar. 1982.

[97] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Oct. 1998.

[98] J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proc. IEEE*, vol. 73, pp. 34–69, Nov. 1985.

[99] V. Cuperman, "On Adaptive Vector Transform Quantization for Speech Coding," *IEEE Trans. Communications*, vol. 37, pp. 261–267, Mar. 1989.

[100] A. Gersho and Y. Shoham, " Hierarchical Vector Quantization of Speech with Dynamic Codebook Allocation ," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Diego), pp. 10.9.1–10.9.4, 1984.

[101] A. Gersho, T. Ramstad, and I. Versvik, "Fully Vector Quantized Subband Coding with Adaptive Codebook Allocation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (San Diego), pp. 10.7.1–10.7.4, 1984.

[102] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Communications*, vol. 28, pp. 84–95, Jan. 1980.

[103] J. D. Johnston, D. Sinha, S. Dorward, and S. R. Quackenbush, "AT&T Perceptual Audio Coding (PAC)," in *Collected Papers on Digital Audio Bit-Rate Reduction* (N. Gilchrist and C. Grewin, eds.), pp. 73–82, Audio Engineering Society, 1996.

[104] K. Brandenburg, J. Herre, J. D. Johnston, Y. Mahieux, and E. F. Schroeder, "ASPEC. Adaptive Spectral Perceptual Entropy Coding of High Quality Music Signals ," in *90th AES Convention*, (New York), 1990. Preprint 3011.

[105] M. Bosi and G. Davidson, "High Quality Low Rate Audio Transform Coding for Transmission and Multimedia Applications," in *93rd AES Convention*, (San Francisco), 1992. Preprint 3365.

[106] T. Scott and M. Bosi, "Use of Low Bit-Rate Coding for High Quality Audio over Telephone Lines," in *93rd AES Convention*, (San Francisco), 1992. Preprint 3362.

[107] L. D. Fielder and D. P. Robinson, "AC-2 and AC-3: The Technology and its Application," in *5th AES Regional Convention*, (Australia), 1995. Preprint 4022.

[108] T. Painter and A. Spanias, "Review of Algorithms for Perceptual Coding of Digital Audio Signals ," in *Proceedings of the 13th International Conference on Digital Signal Processing*, (Santorini, Greece), pp. 179–208.

[109] K. Tsutsui, H. Suzuki, O. Shimoyoshi, M. Sonohara, K. Akagiri, and R. M. Heddle, "ATRAC: Adaptive Transform Acoustic Coding for MiniDisc," in *Collected Papers on Digital Audio Bit-Rate Reduction* (N. Gilchrist and C. Grewin, eds.), pp. 95–101, Audio Engineering Society, 1996.

[110] K. Tsutsui, H. Suzuki, O. Shimoyoshi, M. Sonohara, K. Akagiri, and R. M. Heddle, "ATRAC: Adaptive Transform Acoustic Coding for MiniDisc," in *93rd AES Convention*, (San Francisco), 1992. Preprint 3456.

[111] N. Iwakami, T. Moria, and S. Miki, " High-Quality Audio-Coding at less than 64 kbit/s by using Transform-Domain Weighted Interleave Vector Quantization (TwinVQ) ," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 3095–3098, 1995.

[112] T. Moriya, N. Iwakami, K. Ikeda, and S. Miki, " Extension and Complexity Reduction of TwinVQ Audio Coder," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Atlanta, GA), pp. 1029–1032, 1996.

[113] N. Iwakami and T. Moriya, "Transform Doamin Weighted Interleave Vector Quantization (Twin-VQ)," in *101st AES Convention*, (Los Angeles), 1996. Preprint 4377.

[114] A. Jin, T. Moriya, T. Norimatsu, M., and T. Ishikawa, "Scalable Audio Coder Based on Quantizer Units of MDCT Coefficients," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Phoenix, AZ), Mar. 1999.

[115] T. Moriya, N. Iwakami, A. Yin, K. Ikeda, and S. S. Miki, "Design of Transform Coder for both Speech and Audio Signals at 1 bit/sample," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Munich), pp. 1371–1374, 1997.

[116] D. Pan, "A Tutorial on MPEG/Audio Compression ," *IEEE Multimedia*, vol. 2, no. 2, pp. 60–74, 1995.

[117] S. Shlien, "Guide to MPEG-1 Standard," *IEEE Trans. Broadcasting*, vol. 40, pp. 206–218, Dec. 1994.

[118] P. Noll, "Wideband speech and audio coding," *IEEE Communications Magazine*, vol. 31, pp. 34–44, Nov. 1993.

[119] G. C. P. Lokhoff, "DCC–Digital Compact Cassette," *IEEE Transactions on Consumer Electronics*, vol. 37, pp. 702–706, Aug. 1991.

[120] Y. F. Dehery, M. Lever, and P. Urcun, "A MUSICAM Source Codec for Digital Audio Broadcasting and Storage," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Toronto), pp. 3605–3608, 1991.

[121] K. Brandenburg, "Audio Coding for TV and Multimedia," in *Proceedings of the IEE International Broadcasting Convention.*

[122] G. Soulodre, T. Grusec, M. Lavoie, and L. Thibault, "Subjective Evaluation of State-of-the-Art 2-Channel Audio Codecs ," *J. Audio Eng. Soc.*, vol. 46, pp. 164–177, Mar. 1998.

[123] B. Edler, "Current Status of the MPEG-4 Audio Verification Model Development," in *101st AES Convention*, (Los Angeles), 1996. Preprint 4376.

[124] S. Quackenbush, "Coding of Natural Audio in MPEG-4," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Seattle), pp. 3797–3800, 1998.

[125] B. Edler, "Speech Coding in MPEG-4," *International Journal of Speech Technology*, vol. 2, pp. 289–303, May 1999.

[126] J. D. Johnston, S. Quackenbush, G. Davidson, K. Brandenburg, and J. Herre, "MPEG Audio Coding," in *Wavelet, Subband and Block Transforms in Communications and Multimedia* (A. N. Akansu and M. J. Medley, eds.), pp. 207–253, Kluwer Academic Publishers, 1999.

[127] D. Schulz, "Improving Audio Codecs by Noise Substitution," *J. Acoust. Soc. Am.*, vol. 44, pp. 593–598, July 1996.

[128] M. Nishiguchi and J. Matsumoto, "Harmonic and Noise Coding of LPC Residuals with Classified Vector Quantization," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Detroit), pp. 484–487, 1995.

[129] G. Smart and A. Bradley, "Filterbank Design Based on Time Domain Aliasing Cancellation with Nonidentical Windows," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (South Adelaide, Australia), pp. 185–188, 1994.

[130] W. Y. Chan and A. Gersho, "Constrained-storage Quantization of Multiple Vector Sources by Codebook Sharing," *IEEE Trans. Communications*, vol. 39, pp. 11–13, Jan. 1991.

[131] Y. Shoham, "Vector Predictive Quantization of the Spectral Parameters for Low Rate Speech Coding," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Dallas), pp. 51.2.1–51.2.4, 1987.

[132] R. Veldhuis and A. Kohlrausch, "Waveform Coding and Auditory Masking," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), pp. 427–428, Elsevier, 1995.

[133] P. M. McCourt, "Critical Band Quantisation Analysis for Masked Distortion Speech Coding," in *IEEE DSP Workshop*, (Norway), 1996.

[134] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality*. Prentice Hall, 1988.

[135] *Objective Quality Measurement of Telephone-band (300-3400 Hz) Speech Codecs* . ITU-T Recommendation P.861, Feb. 1998.

[136] S. Wang, A. Sekey, and A. Gersho, "Auditory Distortion Measure for Speech Coding," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, (Toronto), pp. 493–496, 1991.

[137] S. Plain, "Bit Rate Scalabilty in Audio Coding," Master's thesis, McGill University, Montreal, Canada, 2000.