Design of a scalable parametric audio coder

Design of a Scalable Parametric Audio Coder

PROEFONTWERP

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. R.A. van Santen, voor een commissie aangewezen door het College voor Promoties in het openbaar te verdedigen op dinsdag 6 januari 2004 om 16.00 uur

door

Francois Philippus Myburg

geboren te Wolmaransstad, Zuid-Afrika

De documentatie van het proefontwerp is goedgekeurd door de promotoren:

prof.dr.ir. M.L.J. Hautus
en
prof.dr. A.A.C.M. Kalker

Reproduction: Universiteitsdrukkerij Technische Universiteit Eindhoven
This work was sponsored by Koninklijke Philips Electronics N.V.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Myburg, Francois P.

Design of a scalable parametric audio coder / by Francois P. Myburg. -
Eindhoven : Technische Universiteit Eindhoven, 2004.
Proefontwerp. - ISBN 90-386-0762-8
NUR 919
Subject headings : data compression / digital audio engineering / numerical optimisation
2000 Mathematics Subject Classification : 68P30, 65K05

aan my vrou

Summary

Compression, or coding, of digital audio allows the transmission and storage of sound on channels with limited bandwidth and devices with restricted storage capacity. Traditional audio-coding paradigms strive to preserve the waveform of the audio signal, and are referred to as waveform coders. Waveform coders obtain a bit rate reduction by utilising models of the human perception of sound. In the parametric audio-coding paradigm, both a parametric signal model and a perceptual model are used. The utilisation of a parametric signal model allows these coders to achieve even lower bit rates than waveform coders, while maintaining high audio quality. The parameters of the signal model are related to the decomposition of an audio signal into a number of components, such as sinusoids and noise.

An important functionality of any audio coder used in a dynamic environment, is adaptivity. In this work, we describe the design, implementation, and validation of a modular, adaptive parametric audio coder. Our coder is adaptive in the sense that it supports the production of a scalable bit stream. This allows our coder to operate at a number of bit rates without having to re-encode the audio material for each of these bit rates. A scalable bit stream contains a number of layers, each consuming a specified bit rate. Layers may be removed from the scalable bit stream, thus lowering the bit rate while the decoder is still able to decode the scaled bit stream. An audio coder that produces a scalable bit stream allows a number of people, connected at different bit rates to a network, or using storage devices with varying storage capacity, to decode and enjoy audio content.

We utilise a flexible, parametric signal model that allows the capture of dynamic signal behaviour. In addition, we develop optimisation techniques that are able to obtain accurate estimates of the signal-model parameters. High audio quality is obtained by realizing a well-tuned trade-off between sinusoids and noise for each bit rate at which our scalable coder operates. Furthermore, results obtained in listening tests show that the audio quality provided by our scalable coder is comparable to the quality delivered by state-of-the-art, non-scalable audio coders. This is remarkable, since a scalable coder is generally less efficient than a non-scalable version would be. The research and the software developed are also of importance for existing parametric audio coders. The modules designed in this project, such as the optimisation- and noise adaptation modules, may be usable in other parametric audio coders.

Contents

List of Acronyms

acronym	explanation	first used on page
AAC	Advanced Audio Coding	5
ACF	Auto-Correlation Function	58
AF	Ambiguity Function	47
ARMA	Auto-Regressive Moving Average	31
BLS	Base Layer Synthesis	106
BRS	Bit-Rate Scalability	106
CAP	Collective Amplitude Parameters	50
CD	Compact Disk	1
CELP	Code Excited Linear Prediction	4
CQS	Continuous Quality Scale	140
CRC	Communications Research Centre	140
DEMUX	Bit-Stream De-Multiplexer	37
DFT	Discrete Fourier Transform	20
DPE	Detector and Pitch Estimator	49
DVD	Digital Versatile Disk	2
EBU	European Broadcasting Union	13
ERB	Equivalent Rectangular Bandwidth	26
FFT	Fast Fourier Transform	25
FIR	Finite Impulse Response	88
GSM	Groupe Spéciale Mobile	2
HAF	High-order AF	47
HCA	Harmonic Complex Analysis	49
HC_SC	Harmonic-Complex Sub-Component	116
HILN	Harmonic and Individual Lines plus Noise	5
HPE	Harmonic Parameter Estimation	49
HVXC	Harmonic Vector Excitation Coding	5
IFE	Initial Frequency Estimation	49
ISA	Individual Sinusoidal Analysis	49
IS_SC	Individual-Sinusoid Sub-Component	116
ITU	International Telecommunication Union	13

ITU-R	ITU Radiocommunication	13
LAR	Log-Area Ratio	87
LPC	Linear Predictive Coding	9
LPF	Low-Pass Filter	58
LSF	Line Spectral Frequency	87
MBE	Multi-Band Excitation	20
MPEG	Moving Picture Experts Group	4
MP3	MPEG-1 Audio Layer 3	5
MUSHRA	Multi Stimulus Test with Hidden Reference and Anchor	139
MUX	Bit-Stream Multiplexer	36
NA	Noise Adaptation	120
N_C	Noise Component	116
ND	Noise Decoder	89
NEC	Noise Entropy Coding	86
OLA	Overlap-Add	70
PCQ	Prediction Coefficient Quantisation	86
PEAQ	Perceptual Evaluation of Audio Quality	14
PHAF	Product HAF	47
PLP	Laguerre-based Pure Linear Prediction	66
PP	Peak Picking	52
PPC	Philips Parametric Coder	6
QoS	Quality of Service	10
SA	Sinusoidal Analysis	65
SACD	Super Audio CD	2
SC	Sinusoidal Coder	71
SD	Sinusoidal Decoder	89
SEC	Sinusoidal Entropy Coding	73
SL	Sinusoidal Linking	72
SMR	Signal-to-Mask Ratio	30
SPE	Sinusoidal Parameter Estimation	49
SQ	Sinusoidal Quantisation	72
SQAM	Sound Quality Assessment Material	13
STFT	Short-Time Fourier Transform	45
TEM	Temporal Envelope Measurement	68
TEQ	Temporal Envelope Quantisation	86
TES	Temporal-Envelope Shaper	90
TwinVQ	Transform-domain Weighted Interleave Vector Quantisation	5
WD	Wigner Distribution	47
WNG	White-Noise Generator	90

Chapter 1

General Introduction

1.1 Audio and the need for coding

Music brings pleasure to the lives of many people. Sound reproduction technologies allow people to enjoy music wherever and whenever they want. Huge improvements in the techniques used to record and reproduce sound have narrowed the gap between the "live" listening experience and the "reproduced" listening experience. The most notable advance in this field in recent history is the development of technologies, such as that of the compact disk (CD), to store audio in digital format. As a result, digital audio, and in particular CD audio, has become the widely-accepted reference for high quality audio. However, preference is still expressed for the quality of analogue vinyl records above that of CDs by some audiophiles. This illustrates the subjectivity with which audio quality is perceived. This subjectivity is not restricted to the reproduction technology only, it extends to the quality of performance and recording as well.

To obtain an understanding of the human perception of sound, scientists have studied the human auditory system extensively. The mechanical components located on the periphery of our auditory system (the outer, middle, and inner ear) are well understood and have been modelled successfully. The outer ear consists of the external part of the ear, or pinna, and the external ear canal. Its function is to channel sound waves to the ear drum, located at the transition between the outer and middle ear. The ossicles in the middle ear, comprising the hammer, anvil, and stirrup, pick up and amplify vibrations of the ear drum. The transfer function of the outer and middle ear is related to the threshold-in-quiet, which specifies the level of a pure tone that is just audible in a noiseless environment. The spiral-shaped cochlea and semicircular canals, located in the inner ear, are filled with fluids. The vibrations from the middle ear are transmitted through the fluids and cause the coiled basilar membrane, located inside the cochlea, to vibrate. The sound transduction aspects of the inner ear can be modelled by a filterbank consisting of a set of bandpass filters, where the filter bandwidth increases with increasing frequency. The transfer function of the outer and middle ear, and the filterbank describing the basilar membrane, provide a model

of the functioning of the outer, middle, and inner ear, which is sufficiently accurate for many applications.

Vibrations of the basilar membrane are picked up by hair-like extensions of the sensory cells located on the membrane. The electric potential of these cells varies as a function of the incoming sound. As a consequence, electric pulses are caused in the auditory nerve which forms a part of the neural pathway towards the cortex. Attempts at understanding and modelling the way in which the human brain interprets sound, represented by electrical impulses, have been less successful than attempts at understanding and modelling the mechanical components located on the periphery of our auditory system.

Psycho-acoustic or perception models of the human auditory system are used to predict the detection of certain auditory stimuli, to provide an indication of the human capability to discriminate between different auditory stimuli, to locate the direction from which a sound is coming, et cetera.

The digital format of sound, as utilised on CD, reflects psycho-acoustical principles. The sampling frequency of 44.1 kHz reflects the psycho-acoustical principle that an auditory stimulus having a frequency higher than 20 kHz does not lead to a hearing sensation. The quantisation noise introduced by 16 bit sample accuracy, at most $\approx$96 dB below the signal level, is mostly inaudible. The noise introduced by old analogue recording equipment generally has a higher level than that introduced by modern digital recording. This is observed when listening to a CD where the original analogue recording has been converted to digital format. Furthermore, stereophonic representation of audio, also catered for by analogue sound formats like vinyl records, allows the localisation of sounds, leading to a more realistic reproduction.

The excellent quality of CD audio comes at the expense of a high bit rate of 1.4 Mbits per second (Mbits/s). While this bit rate is accommodated successfully by high-end storage devices, like the CD, DVD, and hard disk, the capabilities of modern communication channels, like GSM and the Internet, and smaller storage devices, like Flash memory audio players, are inadequate when confronted with this high bit rate. To enable transmission over communication channels and storage on smaller devices, this bit rate has to be reduced while maintaining high audio quality.

1.2 Audio coding

Compression, or coding, of digital sound aims at reducing the bit rate while maintaining high audio quality. In general, two types of coding can be identified, lossless and lossy coding.

In a lossless coding scheme, the decoded audio data is an exact copy of the original audio data. Lossless coding of sound has found its way into consumer products like Super Audio CD (SACD) [1]. However, due to the nature of sound, lossless

coding yields an approximate compression ratio of only two.

In many scenarios, a higher compression ratio is required. This calls for the application of a lossy coding scheme, where the decoded audio data is a distorted version of the original audio data. By exploiting properties of the human auditory system, the effect of distortions on the perceived audio quality can be minimised. Thus, there exists a trade-off between bit rate (compression ratio) and perceptual distortion (audio quality). This fundamental trade-off plays a central role not only in audio coding, but in all areas of lossy data compression.

From this trade-off, the following question arises: what is the smallest perceptual distortion attainable at a given bit rate for a certain data source? (Or, conversely, what is the lowest bit rate at which a given perceptual distortion can be attained for a certain data source?) In rate-distortion theory, an answer is provided by the rate-distortion characteristic, a performance bound which divides the rate-distortion plane between achievable and non-achievable regions. For example, at a target bit rate R_{T} and corresponding perceptual distortion D_{P}, the rate-distortion pair $(R_{\mathrm{T}}, D_{\mathrm{P}})$ is said to be achievable if $D_{\mathrm{P}} \geq D(R_{\mathrm{T}})$, where $D(R)$ denotes the rate-distortion characteristic at bit rate R[1]. However, if $D_{\mathrm{P}} < D(R_{\mathrm{T}})$, the rate-distortion pair $(R_{\mathrm{T}}, D_{\mathrm{P}})$ is not achievable. An illustration of the rate-distortion characteristic, and a number of achievable and non-achievable rate-distortion pairs is provided in Figure 1.1. Three rate-distortion pairs at the target bit rate R_{T} are considered in this figure. The pair

[1]The rate-distortion characteristic $D(R)$ is the smallest possible distortion at which the pair $(R, D(R))$ is achievable for a given data source. In literature, $D(R)$ is usually referred to as the distortion-rate characteristic, see e.g. [2]. However, this term sounds somewhat awkward in the context of this text. Since we will not make use of $R(D)$, which is usually referred to in literature as the rate-distortion characteristic, we can refer to $D(R)$ as the rate-distortion characteristic.

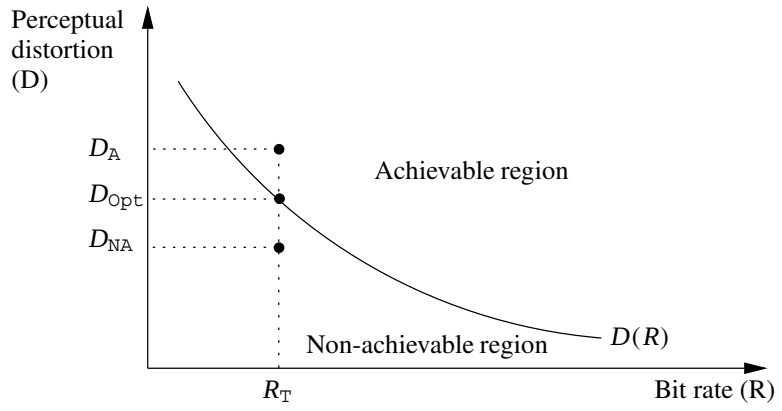


Figure 1.1: *The rate-distortion characteristic $D(R)$ forms a boundary between the achievable and non-achievable regions in the rate-distortion plane.*

(R_{T}, D_{A}) is achievable, though not optimal, while the pair (R_{T}, D_{Opt}) is achievable and optimal. Finally, the pair (R_{T}, D_{NA}) is not achievable. Note that the perceptual distortion decreases with increasing bit rate. In other words, by increasing the bit rate, a higher audio quality can be achieved; or, lowering the bit rate results in a degradation in audio quality.

There are two obstacles that have to be overcome in order to derive the rate-distortion characteristic for general real-world audio data. The *first* obstacle is the subjectivity of audio quality. Subjectivity creates a considerable barrier in the definition of a reliable measure of perceptual distortion, since the quality delivered by a lossy coding scheme will eventually be judged by a human listener. The *second* obstacle is that in order to calculate a reliable rate-distortion characteristic for a specific data source, audio in our case, an accurate model for the source is required. Due to the variability encountered in audio material, it is hard to define such a source model.

A more pragmatic approach is to derive an operational rate-distortion characteristic. This is achieved by applying a real coding strategy to a given audio data set for a number of bit rates, and by measuring the resulting perceptual distortion. In this approach, the coding parameters are optimised to obtain the highest possible audio quality for each bit rate considered, The operational rate-distortion characteristic thus obtained is very useful in understanding the possibilities and limitations of lossy audio coding.

We will only consider lossy audio coding in this thesis, so we can drop "lossy" and continue with the term "audio coding." The main challenge facing the designer of an audio coder is to obtain the highest possible audio quality for each bit rate at which the coder is designed to operate. Many different audio coders have been developed. In the following section, we will consider a number of these approaches.

1.3 Coding techniques and coding standards

Coding of digital sound has a long history. Generally, coding techniques have focused on either speech or general audio. Speech coding has a longer history than audio coding as such, dating back to the work of Homer Dudley in the 1930s [3, 4]. The G.721 [5] and FS1015 [6] speech coding standards, dating from the 1980s, are examples of early industry-wide standards. Speech coding is applied in most voice-based communication systems used today. The division between speech and general audio coding is still apparent in more recent coding standards like MPEG-4 (Moving Pictures Expert Group) [7], as discussed in the following.

A very successful speech-coding paradigm is CELP (Code Excited Linear Prediction) coding [8], where a high audio quality is achieved at low bit rates by utilising a speech-production model in combination with a psycho-acoustic model. CELP coding is applied in the range 4 – 24 kbits per second (kbits/s). To obtain even lower

bit rates, parametric speech coders are utilised. Here, *parametric* refers to the representation of a speech signal by model parameters. The HVXC (Harmonic Vector eXcitation Coding) parametric speech coder operates in the range 2 – 4 kbits/s [8].

In MPEG-4, support for the coding of general audio is provided by the AAC (Advanced Audio Coding) [9, 10] transform-coding technique. In a transform coder, perceptual irrelevance as well as statistical redundancy in an audio signal is identified and exploited in the frequency-domain representation of the signal obtained through a discrete-time unitary transform. One of the most popular transform-coding techniques of general audio is MPEG-1 Audio Layer 3, better known as MP3 [9]. With a high-quality MP3 encoder, one can code stereophonic audio at a bit rate of 128 kbits/s while maintaining "near-CD" quality, a reduction in bit rate by a factor of ten when compared to CD audio. Today, MP3 audio files, coded at 128 kbits/s stereo, are the de-facto compressed audio-file format used on the Internet. AAC is the successor of MP3 and achieves a higher audio quality than MP3 (operating at 128 kbits/s stereo) at a reduced bit rate of 96 kbits/s stereo [7]. Transform coding techniques belong to the wider class of waveform coding. In waveform coding, an attempt is made to preserve the waveform of the original audio signal. Waveform audio coders achieve a bit rate reduction by relying heavily on perception models. Masking is the vital property of perception utilised, and is based mostly on models of the outer, middle, and inner ear. Masking models are utilised to determine when one sound masks another. A masked sound is undetectable, even to a well-trained and sensitive listener. There are two types of masking, *spectral* (or simultaneous) masking and *temporal* (or non-simultaneous) masking. Of the two, spectral masking is utilised the most. By exploiting spectral masking, the audio signal is distorted by a lossy coding method, lowering the bit rate. As long as the distortions introduced remain below the masked threshold, their presence will not degrade the audio quality.

Waveform coders, like MP3, are designed to operate at multiple bit rates. To obtain low bit rates, bandwidth limitation is usually applied to ensure a clean reproduction at lower frequencies. A drawback of waveform coding is the apparent rapid degradation in audio quality at bit rates below approximately 40 kbits/s. Parametric audio coders utilise a signal model in combination with a perception model, and are able to obtain a high audio quality at bit-rates lower than 40 kbits/s. The parametric representation allows independent pitch and time-scale modification by the decoder in a straight-forward manner. In contrast to waveform coders, parametric audio coders do not necessarily strive to code the waveform of the audio signal, and very little, if any, reduction in bandwidth is applied. HILN (Harmonic Individual Line and Noise) is the first parametric audio coder accepted within the MPEG-4 standard. HILN operates at a rate of either 6 or 16 kbits/s mono [11]. Although poor, the audio quality produced by HILN at these bit rates is comparable to the quality of current state-of-the-art waveform coding techniques, TwinVQ at 6 kbits/s mono and

AAC at 16 kbits/s mono [11]. More recently, the parametric audio coder developed by Philips Research in Eindhoven [12] has been submitted in reaction to the MPEG call-for-proposals made in 2001 [13]. This coder, which we will refer to as PPC (Philips Parametric Coder), operates at a bit rate of 24 kbits/s mono, and produces a higher audio quality than AAC at 24 kbits/s mono [12]. This result illustrates the potential of parametric audio coding.

A notable drawback of HILN (and other parametric audio coders) is that an increase in the bit rate does not lead to a proportional increase in audio quality. The bit-rate versus quality of parametric audio coding is compared to waveform audio coding in Figure 1.2. This figure illustrates that while parametric coders have an advantage in the range of low bit rates, waveform coders outperform parametric coders at high bit rates. This observation was the motivation for combining HILN with a waveform coder at higher bit rates, resulting in a so-called hybrid audio coder [14]. Furthermore, parametric audio coders are not able to code speech signals with the same quality as dedicated speech coders at comparable bit rates. To improve the quality of coded speech material, HILN is combined with a parametric speech coder as well [14].

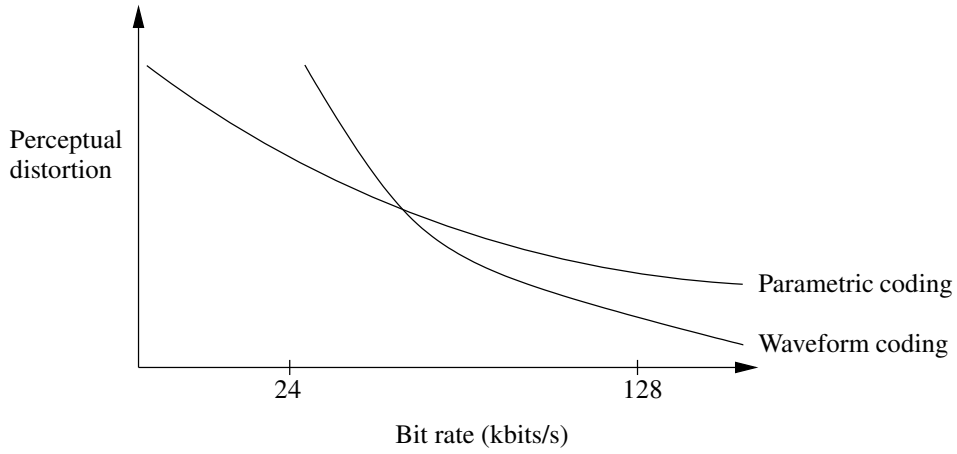


Figure 1.2: *Impression of the bit rate versus perceptual distortion attained by waveform and parametric audio coders.*

1.4 Parametric audio coding

As we noted in the previous section, a parametric audio coder is based on a parametric signal model. In such a model, the audio signal is described by a set of parameters. These parameters are related to the decomposition of an audio signal into the follow-

ing three signal components:

Sinusoids The sinusoidal component models the tonal, quasi-stationary elements, also called partials, of an audio signal with sine-like functions, called sinusoids. Partials are identified as spectral peaks in the amplitude spectrum of the audio signal, see Figure 1.3 for an illustration. Sinusoids are usually parameterised by amplitude, frequency, and phase. Sinusoids are harmonically related when their frequency parameters are integer multiples of a common, or fundamental, frequency. Voiced speech and string instrumental music are examples of tonal sounds modelled by sinusoids.

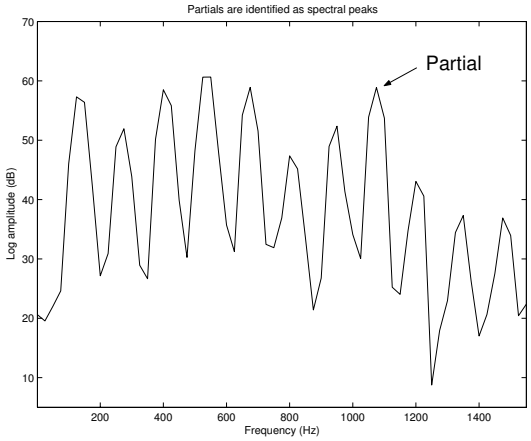


Figure 1.3: *Partials are identified as spectral peaks in the amplitude spectrum.*

Transients The transient component models non-stationary events that take place during a very brief period of time. The temporal envelope of a transient usually exhibits either a burst of energy, with a fast attack and exponential decay, or a sudden increase in signal power level. The onset of a transient has to be encoded with high precision in order to match the temporal resolution of the human auditory system. The sound produced by a castanet is a good example of a sound containing transients.

Residual The residual, being the difference between the original and the sum of transients and sinusoids, is the remaining component. The residual is modelled by synthetic noise with a suitable temporal and spectral envelope.

Of the three signal components, the sinusoidal component is the most dominant in terms of its contribution to audio quality and bit rate. Sinusoids are detected and

extracted from an audio signal on a per-frame basis by the sinusoidal coder. Several strategies have been applied to identify and remove sinusoids from an audio signal. Two popular approaches can be identified. The *first* approach is that of matching pursuits [15], which is a greedy iterative algorithm that selects the sinusoid that best fits the audio signal in each iteration. This method of matching pursuits has been extended with a psycho-acoustical distortion measure, and the extended method is called psychoacoustic-adaptive matching pursuits [16]. The *second* approach is based on identifying and determining the parameters of all sinusoids simultaneously [17].

After the detection and extraction of sinusoids from the signal, a significant coding gain is achieved by linking sinusoids in adjacent frames to form sinusoidal trajectories, or tracks, and by applying time-differential coding of the amplitude and frequency parameters of sinusoids on the track, thus exploiting the stationarity of the audio signal, see e.g. [11, 12, 18, 19]. In practice, the coding of phase parameters differentially over time, yields very little, if any, coding gain. For this reason, phase parameters are not coded at low bit rates in most parametric audio coders. The decoder estimates the lacking phase parameters by utilising phase continuation to ensure a smooth waveform of the decoded signal. When phase continuation is applied, the waveform of the decoded signal will not match the waveform of the audio signal, since errors made in the estimation of frequency parameters accumulate in the phase. Phase continuation is the *first* of two operations which causes a parametric audio coder to cease being a waveform coder. The second operation takes place during noise coding, which will be explained below.

The transient component plays an important role during brief periods of time. Therefore, its effect on the audio quality and bit rate is restricted to isolated and infrequent events in an audio signal, and the transient component has the smallest impact on the overall audio quality and average bit rate. Transient coding is enabled when a transient is detected. Several models of the transient component have been proposed in literature. Levine applied a transform coder to code the audio signal during a transient [18] while Verma exploited the duality between time and frequency to model transients with sinusoids [19]. Ali applied a wavelet analysis to encode transients [20]. The approach taken in PPC and HILN is to determine the temporal envelope of a transient, and to estimate a number of sinusoids [12, 21].

The requirement to obtain low bit rates prohibits the coding of the waveform details of the residual component. Instead, a noise coder is applied in the encoder to capture the most essential characteristics of this signal, such as its spectral and temporal envelopes. The decoder then models the residual with synthetic noise exhibiting the same spectral and temporal envelopes. Modelling the residual with noise is the *second* operation during which information about the signal waveform is discarded. As a result, the residual component is often referred to as the noise component in literature, see e.g. [11, 12, 18, 19]. This terminology reflects one of the core, though

questionable, assumptions made in parametric coding of audio, namely that the residual is stochastic.

The spectral-temporal resolution afforded by the noise model should match the resolution of the human auditory system. Filterbank implementations, see e.g. [22], and parametric modelling of the residual, such as LPC (Linear Predictive Coding) [11, 23] and ARMA (Auto-Regressive Moving Average) modelling [24], are popular approaches. The contribution of the noise signal to the audio quality of the complete decoded signal is second only to the sinusoidal signal, and a well-tuned balance between tones (or sinusoids) and noise in the decoded signal is crucial in obtaining a signal with high quality. At higher bit rates, the waveform details of the residual component should be captured by a waveform coder to obtain a high audio quality, resulting in a hybrid coder, see e.g. [14].

Several approaches have been adopted in literature concerning the order in which the sinusoidal and transient coders are applied. In all approaches, the noise coder is applied to the residual resulting from the sinusoidal and transient coders. In the coder of Levine, transform coding was utilised to code the complete signal when a transient occurred, while during non-transient intervals, the sinusoidal and noise coders were applied in cascade [18]. In HILN, a pre-analysis step is performed to detect transients and determine the amplitude envelope of the audio signal in the frame [21]. The presence of a transient is signalled to the sinusoidal coder, which then analyses the audio signal over a short frame. In the coder of Ali, the sinusoidal, transient, and noise coders are applied in cascade [20]. After sinusoidal coding, the sinusoidal component is subtracted from the audio signal, resulting in a first residual, which is then used as input to the transient coder. The transient component is subtracted from the first residual, resulting in a second residual, which is then coded by the noise coder. In contrast to the coder of Ali, PPC and the coder of Verma apply sinusoidal coding after transient coding, followed by noise coding [12, 25]. The reasoning behind this approach is that sinusoids are suitable functions for modelling the tonal, quasi stationary aspects of an audio signal. The presence of transients disturbs the stationarity of the audio signal, thus complicating the task of the sinusoidal coder. By removing transients from the audio signal prior to sinusoidal coding, this problem is avoided.

Summarising, tones (or sinusoids) and noise are the two dominant elements in the parametric signal model. These elements correspond to well-known aspects of both auditory perception and the physical sources of natural audio signals.

1.5 Scalable coding

An audio codec consists of an encoder and a decoder. The encoder processes an audio signal, and produces a bit stream with a bit rate (substantially) lower than the bit rate of the unprocessed audio signal. This bit stream is stored or transmitted. The

decoder interprets the bit stream and produces a decoded signal which approximates the original audio signal. In practice, external factors may pose restrictions to the encoder, bit stream, or decoder. Therefore, an audio coder operating in a dynamic environment has to adapt to a wide range of external conditions, while maintaining acceptable to high audio quality. Furthermore, the degradation in audio quality, as a result of deteriorating operating conditions, should be graceful. An example of a dynamic environment is the Internet, where no QoS (Quality of Service) is guaranteed, packet losses occur, usually in a burst-like fashion, and where the connection speed of users to the network varies. The adaptivity of an audio coder comes in several forms, and depends on the environment in which it is used. In Section 1.5.1, we consider bit-rate scalability, while bit-rate control is described in Section 1.5.2. Finally, encoder-decoder complexity scalability is considered in Section 1.5.3.

1.5.1 Bit-rate scalability

Bit-rate scalability, also called embedded coding, allows the removal of specified parts, called layers, from the bit stream. Upon reception of the scaled bit stream, the decoder should be able to interpret the remaining layers in order to generate a meaningful decoded signal. Removal of layers can take place before or during transmission, or before decoding. Bit-rate scalability is a useful functionality when the encoder is serving multiple clients with heterogeneous bit rates. Bandwidth scalability is a special case of bit-rate scalability, where layers represent bands of the spectrum [7].

The information contained in the layers is related if the layers are mutually dependent. A dependent-layer structure is usually organised into a base layer and a number of refinement (or enhancement) layers; examples include [7, 25, 26, 27, 28]. Layers may also be mutually independent, establishing a so-called multiple description of the audio signal [29].

In a dependent-layer structure, two types of bit-rate scalability are applied in practice:

1. *Large-step scalability.* Layers contribute large portions to the bit rate, usually in the order of 10 kbits/s [7, 25, 26, 27].

2. *Fine-grain scalability.* Layers contribute small portions to the bit rate, usually in the order of 1 kbits/s [7, 25, 28].

Layers in a large-step implementation may contain different types of information, e.g., different signal components, whereas layers in a fine-grain implementation usually contain strongly related information, e.g., successive refinement.

The layered structure induces additional overhead (or side information) in each layer, thus reducing the coding efficiency when compared to a non-scalable bit stream.

However, the most prominent loss of coding efficiency incurred by a layered structure is due to the number of bit rates at which the audio coder must operate. An audio coder tuned to deliver the highest possible audio quality at a particular bit rate will outperform a bit-rate scalable version of the same coder that has to obtain the highest possible audio quality for a number of bit rates. This resulting loss in audio quality at a particular bit rate is referred to as *scalability loss*. Another way of quantifying the reduced efficiency of a bit-rate scalable coder is *rate loss*, which is the difference between the bit rate required by a scalable coder to obtain a particular quality and the bit rate required by a non-scalable coder to obtain the same quality [30, 31]. Scalability loss and rate loss are considered more closely with the help of Figure 1.4. At a target bit rate R_{T}, the perceptual distortion achieved by the scalable coder is denoted by D_{S}, while the perceptual distortion delivered by the non-scalable coder is denoted by D_{NS}. The scalability loss at the target bit rate is then $\Delta D = D_{\mathrm{S}} - D_{\mathrm{NS}}$, while the rate loss at perceptual distortion D_{S} is $\Delta R = R_{\mathrm{T}} - R_{\mathrm{NS}}$.

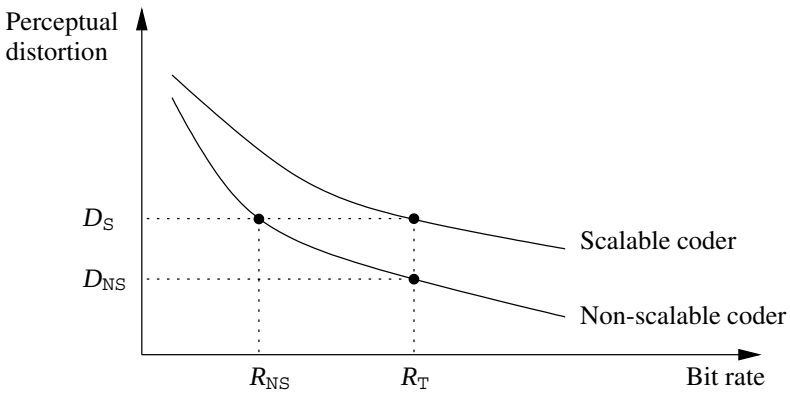


Figure 1.4: *Illustration of the scalability loss and rate loss suffered by a scalable coder in comparison to a non-scalable coder.*

Given the losses incurred by bit-rate scalable coders, the logical question is how these can be minimised. This issue has received attention from the field of classical rate-distortion theory under the umbrella of successive refinement of information. Successive refinement refers to the hierarchical coding of data from a source where a first approximation is obtained by using a few bits of information, and where closer approximations are derived as more information about the data is received. In a successive refinement scheme without rate loss, the goal is to achieve the rate-distortion bound of non-scalable coding at each bit rate. Equitz and Cover have shown that a rate-distortion problem is successively refinable without rate loss if and only if the solutions of the individual rate-distortion problems can be written as a Markov chain [32]. They went on to identify a number of data sources, with corresponding

distortion measures, for which successive refinement without rate loss is possible. However, the complexity of the decoding scheme makes a real implementation unpractical. To mitigate this complexity barrier, Tuncel and Rose considered additive successive refinement, and they derived conditions under which a data source is additively refinable without rate loss [33]. While illustrating the potential of scalable coding, we point out that the results reported in [32, 33] hold only in the limit of large dimensions and for a limited number of data sources and distortion measures. In a real audio coder, an audio signal is analysed on a per-frame basis. This implies that scalable audio coders will always be less efficient than their non-scalable counterparts. In an effort to understand the quality losses incurred in scalable audio and speech coders, Ramprashad considered the effect of popular signal transformations, as utilised in many coders, on the efficiency of a layered description [30]. By considering both synthetic and real-world signals, the expected scalability and rate losses were quantified for a number of transformations.

Bit-rate scalability is an important theme in the MPEG-4 standard [7]: in Version 1 of the standard, coding strategies are combined to create a scalable bit stream. For example, CELP coding can provide the base layer and AAC the refinement layers. Individual coding strategies offer bit-rate scalability also. In particular, AAC supports large-step scalability. In Version 2 of the standard, fine-grain scalability for AAC was introduced by utilising Bit-Sliced Arithmetic Coding (BSAC) [7]. Verification tests have shown that this fine-grain scalability tool performs well over a wide range of bit rates. HILN can operate in a stand-alone bit-rate scalable mode with almost no scalability loss [27], or can be combined with HVXC to cover a wider range of bit rates and signals [14, 34].

Apart from HILN, the parametric audio coder developed by Verma also offers the bit-rate scalability functionality [25]. Possibilities for making PPC bit-rate scalable were investigated by Myburg [35].

1.5.2 Bit-rate control

Bit-rate control allows the encoder to code audio at a specified short-time target bit rate. The target bit rate may vary over time. Bit-rate control is a useful functionality when the encoder is aware of the momentary channel capacity between sender and receiver. A basic form of bit-rate control is usually employed to code audio at a fixed bit rate. Possibilities for introducing bit-rate control to PPC were investigated by Myburg [35].

1.5.3 Encoder-decoder complexity scalability

Encoder complexity scalability allows the use of different encoding algorithms with varying complexity to create valid bit streams, while decoder complexity scalability

allows the use of different decoding algorithms with varying complexity [7]. Alternatively, decoder complexity scalability can be achieved by utilising only a part of the bit-stream in the decoding process. In this sense, bit-rate scalability can be employed to achieve decoder complexity scalability. In general, lower complexity will lead to lower audio quality. Complexity scalability is a useful functionality when the encoder or decoder is to be used on platforms with varying computation power or memory capacity.

1.6 Benchmarking

The main criterion in the comparison of audio coders is the audio quality obtained over a range of bit rates. In comparing audio coders, the largest obstacle to overcome is the subjectivity of perceived audio quality. Subjectivity makes it hard, though not impossible, to obtain a reliable measure of audio quality. Besides, a number of external factors, like sound reproduction equipment and listening environment, influence an audio quality measurement.

Formal listening tests, specified in ITU-R (International Telecommunication Union Radiocommunication) Recommendations [36, 37, 38], are widely used to measure audio quality. The primary goals of a listening test are acquiring measurements of the audio quality delivered by a coder relative to uncoded and coded audio material, and comparing coders.

The main recommendations of ITU-R are summarised in the following. The audio material used in a listening test comprises a number of excerpts, where each excerpt has a duration of approximately 10 seconds, and not more than 20 seconds, in order to avoid fatiguing individuals and to reduce the total duration of the listening test. The selection of audio material should be such that the artefacts introduced by the codec(s) under examination are revealed. Audio material may be selected from the EBU-SQAM (European Broadcasting Union - Sound Quality Assessment Material) CD [39]. Uncoded excerpts are always used as a reference in listening tests. Quality scores are given to each coded excerpt by a panel of individuals experienced in the critical assessment of audio quality. Each individual undergoes training prior to the actual test to become familiar with the audio material, typical artefacts, and the test environment. A number of subjective quality-scales exists. The scale used depends on whether one coder is compared to the reference material or whether two coders are mutually compared. Appropriate procedures for analysis of subjective data are specified. Conducting these tests is expensive, time consuming, and often impractical. To conduct cheaper and more practical informal listening tests, usually only a number of the recommendations is honoured.

Having a reliable objective measure of audio quality would avoid time consuming listening tests, thus simplifying matters substantially. Though proposals for objective

measurement of audio quality exist, the reliability of objective measurements of low and intermediate audio quality is questioned [40].

Objective measurement techniques are based on the properties of human perception, and were applied to speech coders first [41, 42]. Later, these techniques were extended to wide-band audio coders [43]. At present, the standardised state-of-the-art method, called PEAQ (Perceptual Evaluation of Audio Quality) [44, 45], is able to provide objective audio-quality measurements exhibiting a high correlation to the subjective measurements given by a panel of experienced individuals, for high-quality encoded audio material [46, 47].

1.7 Project description

The implementation of PPC developed by Philips Research in Eindhoven is optimised for 24 kbits/s mono. To improve the audio quality at this bit-rate and to make PPC more flexible, Philips Research defined a three-year design project in conjunction with Technische Universiteit Eindhoven. This three-year project has been a follow-up of the eight-month final project of the designers programme Mathematics for Industry at Technische Universiteit Eindhoven, completed in 2000 [35].

The objective is to design and implement a software prototype bit-rate scalable parametric audio coder based on PPC. The performance of the prototype has to be evaluated by conducting listening tests. In addition to the prototype itself, the software has to be accompanied by a user's manual, and the design of the prototype has to be documented.

The prototype should satisfy at least the following two requirements.

1. The signal analysis method applied in PPC has to be adapted and improved to obtain a higher audio quality for both speech and audio signals.

2. The coder must be bit-rate scalable. A high audio quality has to be obtained at high bit rates, and a graceful degradation in quality, with decreasing bit rate, has to be achieved. The target range of supported bit rates is 10 to 40 kbits/s.

1.8 Description of the prototype and summary of results

The requirement to code both speech and audio signals with a high quality over the specified range of bit rates is addressed by utilising both harmonically related and non-harmonically related sinusoids. The strategy applied in HILN and other parametric audio coders is similar in this sense. Flexible models of harmonically related and non-harmonically related sinusoids are defined. The model of non-harmonically related sinusoids allows temporal non-stationarity in both amplitude and frequency. In addition to temporal non-stationarity, the model of harmonically related sinusoids

contains a stretching parameter to compensate for the fact that the frequencies of harmonics produced by stiff-stringed musical instruments, like the piano, are not integer multiples of the fundamental frequency [48].

Obtaining accurate estimates of sinusoidal parameters is crucial, for two reasons. The first reason stems from psycho-acoustical considerations. The accuracy of these parameter estimates should be high enough to ensure that estimation errors are below the threshold of detection. The second reason is that the deterministic character of the signal should be captured completely by the sinusoidal component, and not be present in the residual component. To obtain accurate estimates of the sinusoidal parameters, iterative optimisation methods, based on the Levenberg-Marquardt method, are utilised. The performance of these methods is illustrated by considering both synthetic and real-world signals.

By utilising Laguerre-based linear prediction, the spectral envelope of the residual is modelled with a resolution that matches the spectral resolution of the human auditory system. Also, the rate at which the short-time energy in the residual is measured, is matched to the temporal resolution of the human auditory system.

The bit rate can be lowered by applying a selective transmission of sinusoidal amplitude and frequency parameters on a track. The lacking parameters are estimated by interpolation in the decoder. A decrease in bit rate achieved in this manner leads to a graceful degradation in audio quality [35]. For speech signals, where temporal non-stationarity is common, interpolation can lead to a degradation in audio quality that is larger than acceptable. The utilisation of harmonically related sinusoids is advantageous in this regard, since it allows the modelling of voiced speech by harmonically related sinusoids. This provides the opportunity to apply a selective transmission of amplitude and frequency parameters while limiting the resulting degradation in audio quality.

Identifying a number of sinusoids as being harmonically related also affords the opportunity to lower the bit rate, since only the amplitude parameters, fundamental frequency, and number of sinusoids are required.

In the proposed approach to achieving bit-rate scalability, the sinusoidal parameters are distributed over the base and refinement layers by utilising a rate-distortion optimisation mechanism that minimises the perceptual distortion associated with a layer, relative to the target bit rate. We choose to code the residual corresponding to the sinusoidal component in the base layer, and the noise parameters thus obtained are placed in the base layer. Therefore, the spectral-temporal aspects of the audio signal described by sinusoids in the refinement layers are incorporated in the noise parameters also. Thus, the part of the audio signal represented by sinusoids in the refinement layers is doubly described. The lowest bit rate at which a reasonable audio quality is achieved in our approach, is 16 kbits/s.

When only the base layer is received by the decoder, the sinusoidal and noise pa-

rameters provide a complementary description of the complete audio signal. When the base layer and one or more refinement layers are received by the decoder, the noise component is adapted by the decoder to match the sinusoidal component. Adaptation is based on a band-rejection filter.

The audio quality delivered by the bit-rate scalable codec was evaluated by conducting a subjective listening test. Results from the listening test indicate that the degradation in audio quality, as layers are stripped from the bit stream, is graceful. Furthermore, the bit-rate scalable codec delivers an audio quality equal to that delivered by PPC at similar bit rates.

A stand-alone software prototype bit-rate scalable encoder and decoder, based on the modular design presented in this thesis, is implemented in the MATLAB programming language [49]. The user's manual for this software is contained in an internal Philips Research report [50].

1.9 Contributions of this work

The contributions of this thesis can be classified into two categories: sinusoidal analysis, summarised in Section 1.9.1, and bit-rate scalability, summarised in Section 1.9.2. Furthermore, a stand-alone software prototype bit-rate scalable encoder and decoder, based on the modular design presented in this thesis, is implemented. The merits of our software prototype are underlined by the results obtained in listening tests.

1.9.1 Sinusoidal analysis

We utilise the polynomial amplitude and phase model of non-harmonically related sinusoids put forward by George and Smith in [51]. For harmonically related sinusoids, we propose a flexible model that incorporates both temporal non-stationarity and a stretching effect. Furthermore, we develop low-cost algorithms which are able to provide accurate estimates of the sinusoidal parameters.

1.9.2 Bit-rate scalability

We propose an encoder which makes a dual description, in terms of sinusoids and noise, of a part of the audio signal, and a sinusoidal description of the remaining part. We define the syntax of a bit stream which supports the bit-rate scalability functionality. Depending on the number of layers contained in the bit stream received by the decoder, the corresponding sinusoidal and noise components may not be well matched. We put forward a decoder which exploits the dual description, made in the encoder and contained in the (scaled) bit stream, to match the decoded noise

component to the decoded sinusoidal component. A suitable mechanism is proposed by which the noise component is matched to the sinusoidal component.

1.10 Outline

The deterministic and stochastic signal representation utilised in parametric audio coding is considered in Chapter 2. Identifying the deterministic and stochastic signal components in an audio signal is a difficult problem. The resulting ambiguity in the parametric signal model is exploited to provide the framework for a bit-rate scalable coder. This chapter lays the foundation for the remaining chapters.

Chapter 3 provides a concise definition of the deterministic signal component, sinusoids, and the stochastic signal component, noise. It describes methods for analysis that are used to obtain estimates of the parameters that describe these components. Linking, quantisation, and coding of these parameters is briefly discussed, and we pay attention to the decoding process. In this chapter, we develop an object-based parametric audio coder. The performance of the various algorithms is demonstrated by considering synthetic and real-world signals.

Scaling the output of the audio coder to obtain a scalable bit stream is the main theme of Chapter 4. We pay special attention to the perceptual relevance of sinusoids and describe the rate-distortion optimisation method.

In Chapter 5, we concentrate on the results obtained from the listening test, and in Chapter 6, on main conclusions and recommendations.

Chapter 2

Deterministic versus Stochastic Coding

2.1 Introduction

The basic sound elements utilised by parametric audio coders are sinusoids (deterministic) and noise (stochastic). In this chapter, we will consider the challenges, consequences, and possibilities arising from the decomposition of sound into these two elements. This chapter is build up as follows.

Section 2.2 considers the classification of tonal signal components. A classification based on the properties of the audio signal alone is unsatisfactory. Instead, a combination with psycho-acoustic principles is needed. However, it remains difficult to classify all tonal parts of an audio signal. Apart from the problem of classification, it is questionable whether tones and noise are sufficient to produce a high-quality representation of audio. It seems that there is more to audio than only tones and noise. This section lays the foundation for Chapter 3, where models of the sinusoidal and noise component are presented, methods are described to obtain the model parameters, and a perceptual weighting is assigned to model parameters.

Section 2.3 provides a summary of existing bit-rate scalable parametric audio coders. It is argued that the approach taken in these coders suffers from a number of drawbacks. We present a high-level design of a new bit-rate scalable parametric audio coder that does not suffer from these drawbacks. The argumentation used in that section, together with the high-level design, form the basis for the more detailed design of a bit-rate scalable parametric audio coder given in Chapter 4

2.2 Trade-off considerations

Several deterministic plus stochastic signal models have been utilised in the coding of speech and audio signals. We will mention only a few examples of such models here. An early deterministic model, based on sine-wave models, is the phase vocoder proposed by Flanagan and Golden [52]. Many parametric audio coders are based on the deterministic model for speech coding proposed by McAulay and Quatieri [17]. Their deterministic model is characterised by sinusoidal amplitude, frequency, and

phase parameters, and they proposed linking sinusoids in adjacent frames to form tracks. In addition to sine-wave modelling of speech, noise modelling of individual harmonics, classified as being stochastic, was applied by Griffin and Lim in their Multi-Band Excitation (MBE) vocoder [53]. A stochastic model was added to the deterministic model of McAulay and Quatieri and applied to the coding of general audio signals by Serra [54].

It is well known that an audio signal $s[n]$ can be decomposed into a sum of complex sinusoids by means of the Discrete Fourier Transform (DFT). However, experience has shown that a deterministic decomposition is not efficient from a bit-rate versus perceptual-distortion point-of-view, in the sense that the audio quality increases gradually with the increasing bit rate when only sinusoids are used to approximate the audio signal. The reason is that both tones and noise are fundamental entities in the field of auditory perception. The sound produced by a flute, for example, contains these two elements, a set of harmonically related partials and the hissing (or noisy) sound produced by the flow of air. Speech is another example, where a distinction is made between voiced (tonal) and un-voiced speech (noise). Removing either one of these elements will give the impression of an un-natural sound in both examples. This pleads for the inclusion of a noise signal component. Figure 2.1 illustrates that decomposing an audio signal in terms of sinusoids plus noise yields a higher audio quality than a sinusoidal decomposition, especially at low bit rates. These observations form the basis for the utilisation of a deterministic plus stochastic model to approximate an audio signal at low bit rates. According to the deterministic plus stochastic model, a discrete-time audio signal $s[n]$ is closely approximated by

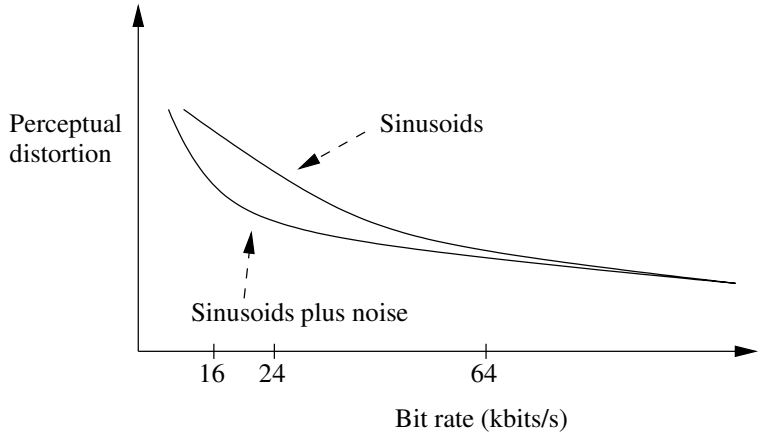


Figure 2.1: *An impression of the bit rate versus perceptual distortion characteristic of the deterministic decomposition and of the deterministic plus stochastic decomposition.*

sinusoids and noise in the sense that

$$\psi(s[n]) \approx \psi(s_{\texttt{sinusoid}}[n] + s_{\texttt{noise}}[n]). \tag{2.1}$$

In this expression, ψ maps an audio signal to a perceptual space, $s_{\texttt{sinusoid}}[n]$ is the sinusoidal component and $s_{\texttt{noise}}[n]$ the noise component. The optimal decomposition of $s[n]$ into sinusoids and noise minimises the perceptual distortion at the target bit rate.

Instead of utilising the constant-amplitude and constant-frequency sinusoids afforded by the DFT, the sinusoidal component is modelled more efficiently by an expression of the form

$$s_{\texttt{sinusoid}}[n] = \sum_{k}^{N_{\texttt{c}}[n]} a_k[n] \cos \theta_k[n], \tag{2.2}$$

where dynamic signal behaviour is taken into account. It is assumed that both the amplitude a_k and frequency $\dot{\theta}_k$ of partial k vary slowly over time. Therefore, the audio signal is analysed on a per-frame (or segment) basis instead of on a per-sample basis. In most cases, both the amplitude and frequency parameters of all sinusoids are assumed constant in a frame. The amplitude is then denoted by a_k, and the argument of the cosine is denoted by a linear polynomial $\theta_{k,1} + \theta_{k,2}n$, where $\theta_{k,1}$ is the phase and $\theta_{k,2}$ the frequency. A coding gain is achieved by applying differential coding of the sinusoidal amplitude and frequency parameters. Tracking and inter-frame differential coding (also called time-differential coding) of sinusoidal parameters on a track was first proposed by McAulay and Quatieri [17], and is applied in a number of sinusoidal coders, see e.g. [21, 12, 19, 18, 54]. In this approach, related sinusoids in adjacent frames are linked to form sinusoidal trajectories, or tracks, before time-differential coding of the sinusoidal parameters is applied. When a sinusoid in a frame can not be linked to a sinusoid in the previous frame, a track is *started*. When a sinusoid can be linked to one in the previous frame, the track is *continued*, and when a sinusoid can not be linked to one in the following frame, the track is *ended*. In Figure 2.2, a time-frequency plot, illustrating the start, continuation, and end of a number of tracks, is given. A further reduction in bit rate is obtained by not encoding the phase parameters, and by applying phase continuation in the decoder to ensure a smooth waveform over frame boundaries. Besides inter-frame differential coding, intra-frame differential coding of sinusoidal parameters has also been applied [55].

As mentioned in Section 1.4, the sinusoids are determined first, and the residual is modelled by synthetic noise. Figure 2.3 presents an illustration of this sequential coding process. The sinusoidal coder is applied to the audio signal $s[n]$ and estimates the parameters of those sinusoids that will result in the smallest perceptual distortion at the target bit rate. The sinusoidal component $\hat{s}_{\texttt{sinusoid}}[n]$ is generated from the

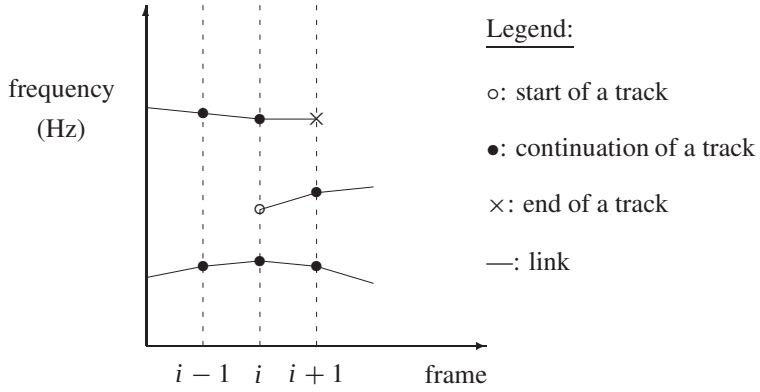


Figure 2.2: *Linked individual sinusoids form tracks.*

estimated parameters by the sinusoidal decoder, and is an estimate of the optimal sinusoidal component $s_{\texttt{sinusoid}}[n]$ at the target bit rate.

The assumption made in practice is that the residual

$$s_{\texttt{residual}}[n] = s[n] - \hat{s}_{\texttt{sinusoid}}[n] \tag{2.3}$$

sounds like noise. The noise coder is applied to the residual to capture its temporal and spectral envelopes while ignoring the fine-structure of its waveform. The noise component, generated by the noise decoder from the estimated noise parameters, is denoted by $\hat{s}_{\texttt{noise}}[n]$, and will sound like noise.

The audio signal is then represented by $\hat{s}_{\texttt{sinusoid}}[n] + \hat{s}_{\texttt{noise}}[n]$ at the target bit rate. In general, this representation will be of lower quality than the optimal

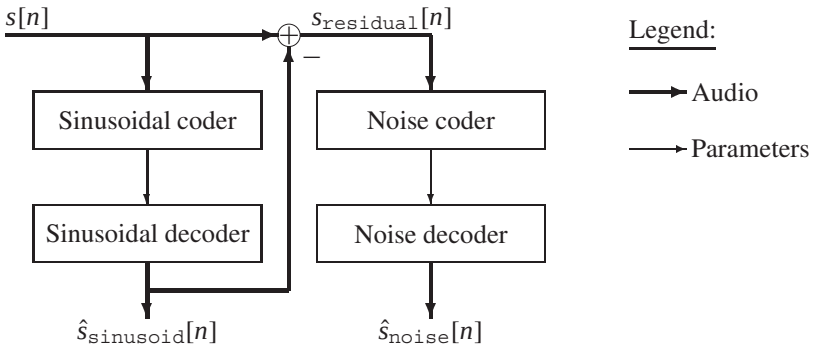


Figure 2.3: *Serial processing applied in parametric audio coders. Sinusoids are extracted first, and* $\hat{s}_{\texttt{sinusoid}}[n]$ *includes the transients in the interest of simplicity. The residual* $s_{\texttt{residual}}[n]$ *is coded by a noise coder, resulting in the noise signal* $\hat{s}_{\texttt{noise}}[n]$.

representation $s_{\texttt{sinusoid}}[n] + s_{\texttt{noise}}[n]$ at the target bit rate. The loss in quality is depicted in Figure 2.4.

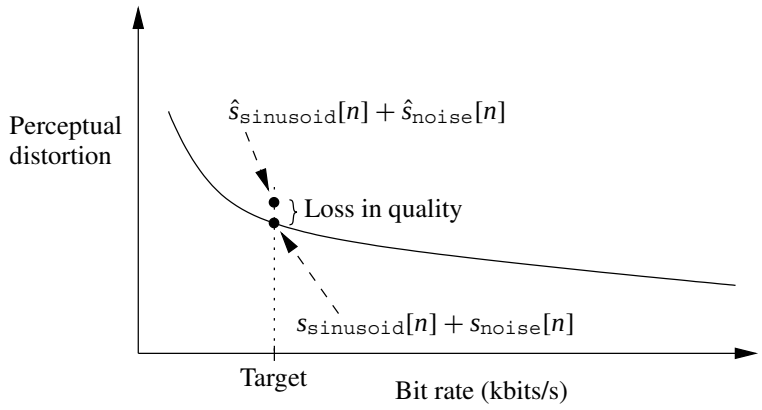


Figure 2.4: *The realised decomposition $\hat{s}_{sinusoid}[n] + \hat{s}_{noise}[n]$ is of lower quality than the optimal decomposition $s_{sinusoid}[n] + s_{noise}[n]$, located on the rate-distortion characteristic at the target bit rate. The loss in quality is indicated.*

Clearly, errors made in the extraction of sinusoids will influence the noise coder. Quoting Serra from [23, p. 79]: "To restrict the residual to be a stochastic signal simplifies enormously the residual signal, but it implies that the deterministic component has to account for whatever is not stochastic. . . . Therefore in the current system, the extraction of the deterministic part is more critical than before."

The main steps taken in parametric audio coding are considered from a different perspective in Figure 2.5. It is reasonable to assume that tones and noise form only part of any audio signal. In reality, there is a region of ambiguity, indicated by a tonal - noise transition region in part (a) of this figure. The parametric model assumes that a clear boundary, whose location is determined by the target bit rate, exists, and that any audio signal is closely approximated by sinusoids and noise. In the sinusoidal coding process, the position of the model boundary is estimated on the basis of the given target bit rate, resulting in an estimate of the sinusoidal component $\hat{s}_{\texttt{sinusoid}}[n]$ and the noise component $\hat{s}_{\texttt{noise}}$, see part (b) of the figure. The tonal - noise transition region and model boundary illustrated in this figure will prove useful in pointing out the drawbacks of current bit-rate scalable parametric audio coders and in highlighting the advantages of our proposal for bit-rate scalability, see Section 2.3.2.

Four factors influence the definition of the estimated sinusoidal component. First and foremost is the identification of sinusoids in $s[n]$ and determination of their perceptual relevance. The sinusoidal model itself and the method used to obtain the model parameters form the second factor. Quantisation of the sinusoidal parameters is the third factor, and phase continuation the fourth.

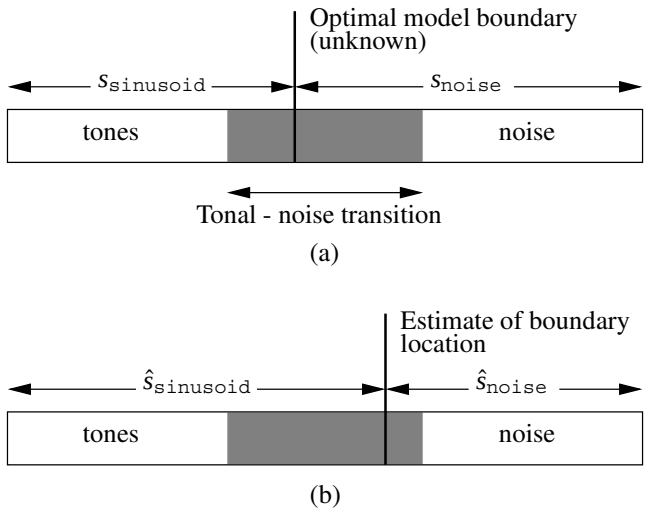


Figure 2.5: *The main steps taken in parametric audio coding. (a) In reality, an audio signal contains tones, noise, and a mixture of both, indicated by a tonal-noise transition region. The parametric signal-model assumes that any audio signal can be closely approximated by sinusoids and noise. The optimal boundary between sinusoidal and noise signal components is determined by the target bit rate. (b) In the sinusoidal coding process, the model boundary is estimated, given the target bit rate. This results in an estimate of the sinusoidal component* $\hat{s}_{sinusoid}$ *and the noise component* $\hat{s}_{noise}$.

In Section 2.2.1, the first factor is considered in more detail; the remaining factors are briefly considered next. The model of the sinusoidal component and the method used to estimate the model parameters play a crucial role in the ability of the sinusoidal coder to remove partials completely from $s[n]$. Therefore, the sinusoidal model has to be able to capture dynamic signal behaviour, such as temporal non-stationarity. A method to obtain accurate estimates of the model parameters is very desirable. The sinusoidal amplitude and frequency parameters are quantised according to the just notice-able differences in amplitude and frequency, based on psycho-acoustical results [56]. Phase parameters are not transmitted at low bit rates, and the decoder applies phase continuation to ensure a smooth waveform over frame boundaries.

Section 2.2.2 examines the main considerations in choosing a stochastic model of the residual component.

2.2.1 Sinusoidal coding

To obtain the tonal part of a frame, local peaks are identified in the amplitude spectrum of the (windowed) frame obtained by applying the DFT. The DFT bin at which

the amplitude spectrum peaks provides an estimate of the frequency, the value of the amplitude spectrum at the bin an estimate of the amplitude, and the phase spectrum at the bin an estimate of the phase.

Two factors influence the spectral definition of a peak: frame duration and temporal non-stationarity of the signal in the frame. On the one hand, the ability to discriminate between two frequencies, or the spectral resolution, is inversely proportional to the frame duration. In other words, spectral resolution is inversely proportional to temporal resolution. Temporal non-stationarity of partials in the frame, on the other hand, will degrade their spectral image in the sense that spectral peaks become broader as temporal non-stationarity increases. Therefore, to detect spectral peaks, one has to find a balance between a sufficiently high spectral resolution and limiting temporal non-stationarity. Applying a window with a very good side-lobe structure to the frame will reduce the interference among partials at the expense of broader spectral peaks, thus reducing the effective spectral resolution. The Hann and Hamming windows [57] are popular choices in audio coding.

Figure 2.6 illustrates the trade-off between the desirable spectral resolution and limiting temporal non-stationarity by considering a voiced male-speech fragment. Three frames, all centred around time $t = 0$ and windowed by the rectangular window, are depicted in parts (a) – (c) of this figure. The frame durations are 100 ms, 40 ms, and 10 ms, respectively. The amplitude spectra, depicted on a logarithmic scale in parts (d) – (f) of this figure, are obtained from the DFT applied to the Hann-windowed frames. Part (d) is the spectral counterpart of (a). Observe that the low-frequency (harmonic) partials, below 1.2 kHz, are well defined, while the higher partials are poorly defined. This is due to the large degree of temporal non-stationarity of the signal, clearly visible in part (a). Therefore, temporal non-stationarity in a frame can severely degrade its spectral image. Part (e) is the spectral counterpart of (b). Even though the signal is non-stationary in part (b) too, the clearly discernable spectral peaks illustrate the good trade-off between spectral resolution and temporal non-stationarity obtained. Part (f) is the spectral counterpart of (c). The spectral resolution is clearly too low to identify individual peaks.

Zero padding does not increase the spectral resolution; it results in the interpolation of the frequency spectrum, providing a more smoothly-appearing spectrum, and therewith more accurate estimates of sinusoidal frequencies from peak locations. For specific sequence lengths, the DFT can be calculated by the computationally efficient FFT algorithm. A suitable sequence length can be obtained by zero padding any sequence, allowing efficient calculation of its spectrum.

From a perceptual point-of-view, the following can be said about the spectral resolution of the human auditory system. The periphery of the human auditory system can be modelled by a passive filterbank with filters having increasing bandwidth towards higher centre frequencies. An audio signal is band-pass filtered, and the

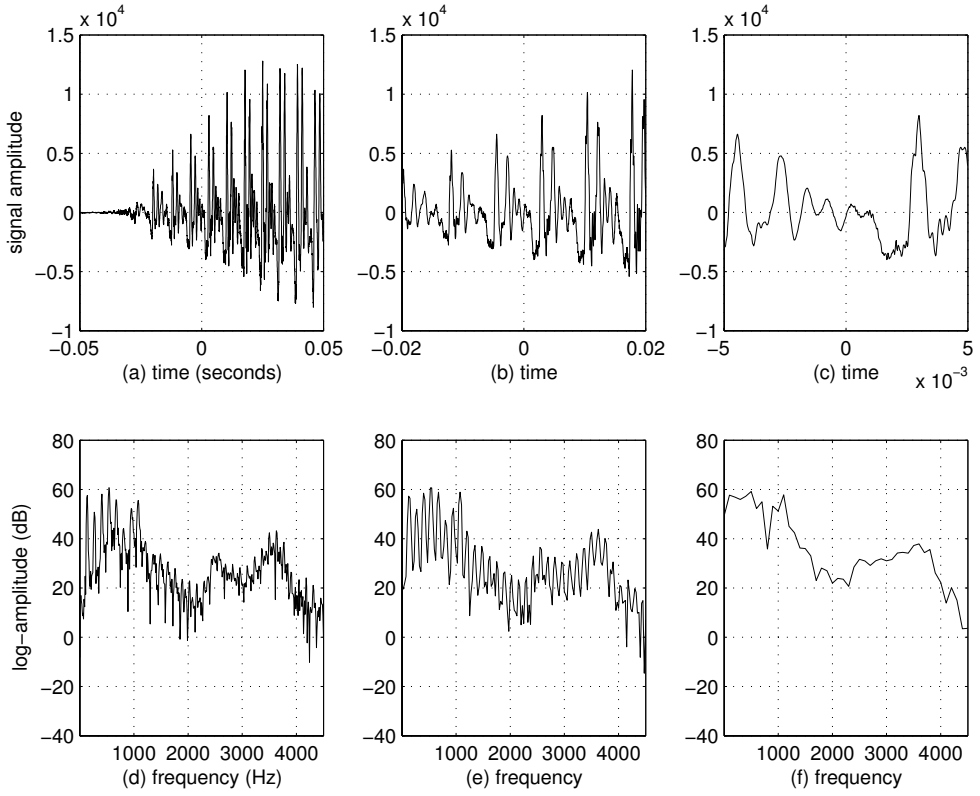


Figure 2.6: *The trade-off between spectral resolution and temporal non-stationarity. The time signals in parts (a) – (c) are windowed with the rectangular window. The Hann window is used in the calculation of the corresponding spectra, illustrated in parts (d) – (f).*

band-pass filtered signals are further processed by the remaining part of the auditory system. The auditory-filter bandwidths are approximated by the Bark (after Barkhausen) [56] and ERB (Equivalent Rectangular Bandwidth) scales [58]. Critical bandwidth scales give an indication of the bandwidth over which the auditory system integrates signal power. Related to the critical bandwidth is the human ability to discriminate frequencies.

As in the case of spectral analysis, the temporal resolution of the human auditory system is inversely proportional to its spectral resolution. This suggests the need for analysis on multiple time scales, or multi-resolution analysis, matching the temporal-spectral resolution of the auditory system. In practice, longer frames for low frequencies and shorter frames for higher frequencies are used in parametric audio coders [12, 18, 19, 22], resembling the time-frequency resolution trade-off

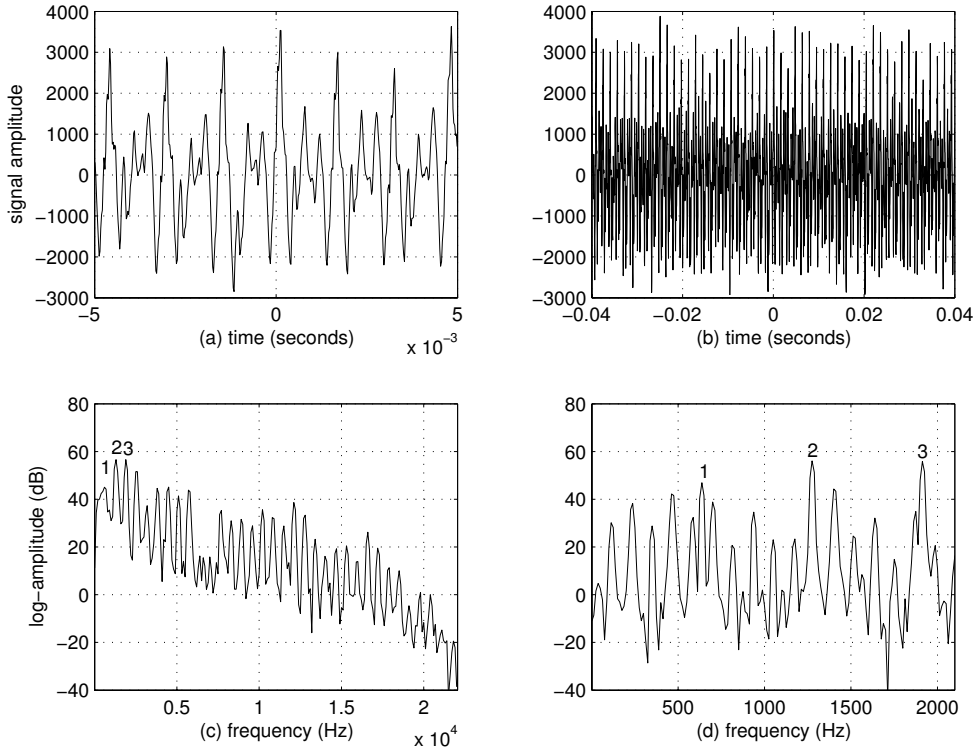


Figure 2.7: *The advantage of multi-resolution analysis. A frame, centred around time $t = 0$, with 10 ms duration, is illustrated in part (a). Its log-amplitude spectrum is given in part (c). A longer frame with 80 ms duration, also centred around time $t = 0$, is illustrated in part (b), and its log-amplitude spectrum is given in part (d). The first three harmonics are indicated in parts (c) and (d).*

achieved by wavelet transforms. In most cases, however, only a few time scales, two or three, are utilised in the interest of maintaining low computational complexity.

From a signal point-of-view, multi-resolution analysis can be advantageous too, as illustrated in Figure 2.7 where a bagpipes fragment is considered. Two frames, both centred around $t = 0$ and windowed by the rectangular window, are depicted in parts (a) and (b) of this figure. The frame durations are 10 ms and 80 ms, respectively. The amplitude spectra, depicted on a logarithmic scale in parts (c) and (d) of this figure, are obtained from applying the DFT to the Hann-windowed frames. Part (c) is the spectral counterpart of (a). The harmonic partials, with a fundamental frequency of $\approx$640 Hz, are clearly visible. Closer inspection of the spectrum, however, reveals a poorly defined first harmonic. Part (d) is the spectral counterpart of (b), where only

the 0 – 2 kHz frequency band is shown. The spectrum reveals clear low-frequency partials that are not apparent from part (c).

Applying a fixed segmentation in the analysis of an audio signal does not take local signal behaviour into account. The segmentation length is usually a compromise between average signal behaviour and bit-rate constraints, and simplifies the analysis process. Adaptive segmentation, based on local signal behaviour, can lead to more efficient coding: on the one hand, stationary parts of the audio signal can be modelled with larger frames, while, on the other hand, frame boundaries can be adjusted according to transient-onset positions, avoiding pre-echo artefacts in the synthesis. This improved performance comes at the cost of additional computational complexity. Adaptive segmentation in the context of a rate-distortion optimal sinusoidal coder was considered by Heusdens and van de Par [59].

Concerning the analysis of temporal non-stationary harmonic signals, like voiced speech, we make two remarks. The first remark is that it is useful to apply pitch-synchronous analysis, as proposed by McAulay and Quatieri [60, Chapter 4] and Serra [23]. A frame spanning a few (say three or four) periods of the fundamental frequency provides sufficient spectral resolution to discriminate between the harmonic partials, while limiting the potential effect of temporal non-stationarity. The second remark is that temporal non-stationarity in the form of a linear frequency chirp can be removed from the signal by transforming, or warping, the time axis [61]. However, when the audio signal contains a mixture of both harmonically-related partials and non harmonically-related partials, time-warping may not be appropriate, since the frequency behaviour of the non harmonically-related partials will be altered.

In its most basic form, a spectral peak is defined as a local maximum in the amplitude spectrum of a windowed frame. However, not all peaks are equally well-defined, as is apparent from Figures 2.6 and 2.7. In order to determine whether a spectral peak does indeed represent a partial, some definition must be formulated. Popular definitions are based either on the spectral image or on the temporal predictability over time of a signal component. A spectral definition dictates the level of the peak in comparison to the level of adjacent DFT bins on a dB scale [54]. This definition fails when a partial is non-stationary. The temporal predictability of a signal component takes adjacent frames into consideration by extrapolating the parameters from adjacent frames to the frame under consideration, and comparing them to the estimated parameters. For example, we denote the estimated phase and frequency parameters of a spectral peak in frame i by $\hat{\theta}_1^{(i)}$ and $\hat{\theta}_2^{(i)}$ respectively, and the number of samples between the centra of two successive frames by N_{s}. The phase and frequency parameters in adjacent frames are denoted in a similar manner. The predicted phase from frame $i-1$ at the centre of frame i is given by

$$\hat{\theta}_{1,\mathtt{pred1}}^{(i)} = \hat{\theta}_1^{(i-1)} + \hat{\theta}_2^{(i-1)} N_{\mathrm{s}},$$

and the predicted phase from frame $i+1$ by

$$\hat{\theta}_{1,\mathtt{pred2}}^{(i)} = \hat{\theta}_{1}^{(i+1)} - \hat{\theta}_{2}^{(i+1)} N_{\mathtt{s}}.$$

The spectral peak in frame i is considered to be a partial when both

$$|\hat{\theta}_{1}^{(i)} - \hat{\theta}_{1,\mathtt{pred1}}^{(i)}| \qquad \text{and} \qquad |\hat{\theta}_{1}^{(i)} - \hat{\theta}_{1,\mathtt{pred2}}^{(i)}|$$

are small enough. This definition relies on accurate estimates of the model parameters.

In low bit rate coding applications based on parametric audio coding, the restriction posed on the bit rate translates into a restriction on the number of sinusoids that can be transmitted to the decoder. To obtain a decoded signal with high audio quality, the selection of sinusoids to transmit should be such that the perceptual distortion introduced in the decoded signal as a result of not having all sinusoids at hand, should be as small as possible. To estimate the perceptual distortion of a partial, the masked threshold is utilised. The standard assumption made in audio coding is that the masked threshold quantifies the distortion, introduced by the lossy coding method, that can be masked by the input signal. As long as the distortions introduced remain below the masked threshold, they are considered to be inaudible and the decoded signal is classified as being a perceptually lossless version of the original. This assumption, as we will argue in the following, is not compatible to recent findings within the field of psycho-acoustics. For a detailed description of masking models applied in audio coding, we refer to [62] and [63].

We now present a summary of the main determinants of the masked threshold and its underlying assumptions. The masked threshold is determined on a frame basis by analysing the windowed frame in the frequency domain. The masked threshold depends on the tonal and noise elements (or maskers) contained in the audio signal. This follows from the fact that tones are less effective maskers than noise [64, 65]. Early identification techniques of tonal and noise elements were based on the spectral flatness of a critical band [66], while more recent techniques consider the predictability of individual frequency components over time [62]. Once the complete frequency band is divided between tonal and noise maskers, the masking properties of each masker, quantified by a spreading function, are determined. Spreading functions are utilised to reflect the band-pass characteristic of the auditory filters. The amount of masking depends on the character of the masker (tonal or noisy) and the bulk of the spreading function is restricted to the critical band centred around the masker. The band-pass characteristic is further approximated by extending the spreading function to surrounding critical bands. The total masked threshold is then determined by power addition of the separate spreading functions of all signal components [62]. The assumption made in defining a spreading function in this way is that the detectability of a distortion component is determined mainly by the auditory filter spectrally

centred around this distortion component. However, this is a reasonable assumption only when the distortion is narrow-band. Detecting multiple simultaneous narrow-band distortions, possibly spanning many critical bands, is more likely than detecting an individual distortion, even if all distortions are masked [67, 68]. This finding suggests that the human auditory system is capable of integrating acoustical information over a range of auditory filters, thereby increasing the detectability of distortions. Note that this finding has direct implications for audio coding, where distortions are usually introduced across a wide range of frequencies. In addition to the human ability to integrate across frequencies, the human auditory system is also capable of integrating acoustical information over time, up to a certain maximum time. This temporal integration time is approximately 300 ms [69]. As a result, the detectability of a distortion increases with increasing duration. Recently, a new psycho-acoustical masking model, which incorporates the integration of acoustical information over time and frequency, was proposed by van de Par et al. [70]. In contrast to traditional masking models, this masking model does not require that tonal and noise maskers be identified in the audio signal. This masking model provides a measure for the detectability of small distortions, and can be used to obtain a masking curve, thus making it suitable for audio-coding applications.

In parametric audio coding, the relation between a spectral peak and the masked threshold, also referred to as the SMR (signal-to-mask ratio), is usually taken as a measure of the perceptual relevance of the corresponding partial. Perceptual relevance, when defined in this way, is the perceptual distortion introduced when the corresponding sinusoid is not included in the decoded signal [70]. The higher the SMR, the more relevant the partial is considered to be and the more likely that the corresponding sinusoid will be transmitted. Furthermore, spectral peaks lying below the masked threshold are assumed to be masked, and the corresponding sinusoids are usually not transmitted. However, from the argumentation given above, we note that the perceptual relevance of a tone is not defined by its SMR restricted to a single (short) frame only. The total duration of a tone, or a track in the case of parametric audio coding, influences its perceptual relevance. Furthermore, removal of a number of masked sinusoids in a frame may lead to a perceptible distortion.

The factors considered in this section form the basis of the sinusoidal analysis of an audio signal and the perceptual categorisation of sinusoids thus obtained, described in Chapter 3.

2.2.2 Noise coding

Models of the residual component capture the temporal and spectral envelopes of $s_{\texttt{residual}}[n]$ only, while ignoring the fine-structure of its waveform. The temporal-spectral resolution afforded by the model of the residual component should fit the parameters describing the human perception of broad-band noise.

The main observation in this regard is that the human auditory system is sensitive to the short-time total energy in a critical band only, and not to the distribution of energy within the critical band. Therefore, if $s_{\texttt{residual}}[n]$ is indeed broad-band noise, a perceptually indistinguishable replica can be generated by band-pass filtering white noise (where the filters represent critical bands) and correcting the short-time energy in each critical band. A noise model based on the spectral properties of auditory perception was proposed by Goodwin [22]. Other popular models of the residual component are based on LPC (Linear Predictive Coding) [23, 11] or ARMA (Auto-Regressive Moving Average) [24] modelling. The main drawback of LPC and ARMA models is that the underlying spectral resolution is not matched to that of the human auditory system. Frequency warping in combination with LPC, termed warped LPC [71], does allow transformation (or warping) of the frequency axis according to psycho-acoustic principles, and has been applied to audio coding [72, 73].

There seems to be no concensus regarding a quantitative description of the auditory temporal resolution as a function of frequency. Studies do suggest, however, that the temporal resolution at low frequencies (4 ms at 1 kHz) is lower than the temporal resolution at higher frequencies (1 ms at 4 kHz) [74].

Our model of the residual is based on these considerations, and is discussed in Chapter 3.

2.3 Bit-rate scalable parametric audio coding

The application of the parametric signal model to bit-rate scalability is considered in this section. A summary of existing bit-rate scalable parametric audio coders is given in Section 2.3.1. In Section 2.3.2, the main drawbacks of these scalable coders are discussed and a scenario in which these drawbacks are overcome is sketched. These drawbacks are overcome by an alternative design of a bit-rate scalable parametric audio coder, presented in Section 2.3.3.

2.3.1 Existing scalable parametric coders

Three bit-rate scalable parametric audio coding schemes have been presented in literature. Two are based on HILN ([26] and [27]) and the other on the parametric coder developed by Verma [25]. Sequential processing of the audio signal, as illustrated in Figure 2.3, is applied in all three coders, and partitioning of the sinusoidal component into layers is the common strategy. In particular, sinusoidal tracks are categorised according to their perceptual relevance, where the most relevant tracks are placed in the base layer, while less relevant tracks are distributed over refinement layers. There are a number of differences among these implementations though; these are summarised in the following.

First, we consider the bit-rate scalable implementation of HILN described in [27]. The sinusoidal component in HILN contains two sub-components: a harmonic complex and individual sinusoids. The harmonic complex and noise component, if at hand, are always placed in the base layer. The perceptually most-relevant sinusoids are placed in the base layer too, while the less-relevant sinusoids are placed in the (only) refinement layer. Sinusoids in the same layer may be linked over frame boundaries. Furthermore, sinusoids in the base layer, in a frame, may be linked to sinusoids in the refinement layer, in the next frame, but linking in the opposite direction is not allowed. The layers are therefore dependent. No sinusoidal phase parameters are encoded, and the decoder utilises phase continuation to avoid discontinuities at frame boundaries. The base layer requires 6 kbits/s, and the refinement layer 10 kbits/s, amounting to a maximum bit rate of 16 kbits/s. Subjective evaluations carried out during MPEG standardisation testing revealed that the scalability loss of the bit-rate scalable HILN coder is insignificant in comparison to the non-scalable HILN [27].

Second, we consider the bit-rate scalable coder of Verma. No harmonic complex is utilised, and sinusoidal trajectories are categorised in so-called quality layers, where all tracks belonging to a specific quality layer have a signal-to-mask ratio within specified limits. Notable differences with HILN are that the noise component is not coded in the base layer, and sinusoidal phase parameters are coded in higher layers. Furthermore, the scalable bit-stream contains eight layers in total, where the base layer requires 6 kbits/s, and all layers 80 kbits/s. As in the case of HILN, the layers in this coder are inter-dependent too. The contents of the bit-streams are compared in Figure 2.8.

We observe that the highest bit rate obtained by the coder of Verma, namely 80 kbits/s, is much higher than the bit rate at which most other parametric audio coders operate.

2.3.2 Drawbacks of existing scalable parametric coders

From the perspective of Figure 2.5, the consequences of first applying serial processing to define the sinusoidal and noise components, and then generating a layered bit-stream, are illustrated in Figure 2.9 (a). In this illustration, the bit-stream contains at least three layers. The base layer contains the parameters of a number of sinusoids and no parameters of the noise component. The base and first refinement layers combined contain more sinusoidal parameters than the base layer alone would, and all noise parameters. All layers combined contain the complete set of sinusoidal and noise parameters.

When all layers are available to the decoder, part (a).I of this figure, the sinusoidal and noise parameters describe the complete signal, and the sinusoidal and noise components are well matched.

When the base and first refinement layers are available to the decoder, part (a).II

Legend:

BL: Base layer
RL #: Refinement layer
HC: Harmonic complex
IS: Individual sinusoids
QL #: IS Quality layer
(N)P: (No) Phase
N: Noise component
T: Transient component

HILN

Rate	Contents
16	HC, N, IS QL 1&2, NP
6	HC, N, IS QL 1, NP

Verma

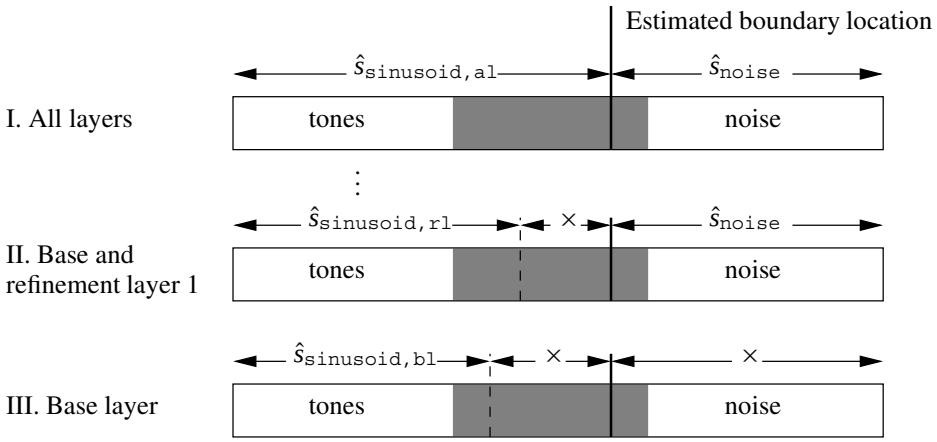
	Rate	Contents
RL 7	80	IS QL 1–5, P, N, T
RL 6	45	IS QL 1–4, P, N, T
RL 5	32	IS QL 1–3, P, N, T
RL 4	20	IS QL 1–3, NP, N, T
RL 3	16	IS QL 1&2, NP, N, T
RL 2	12	IS QL 1&2, NP, N
RL 1	10	IS QL 1&2, NP
BL	6	IS QL 1, NP

Figure 2.8: *The bit-stream contents of HILN and the coder of Verma compared. The scalable bit-stream produced by HILN contains only two layers. A harmonic complex is utilised and no phase parameters are transmitted. The scalable bit-stream produced by the coder of Verma contains more layers and spans a larger range of bit rates. The phase parameters of sinusoids are transmitted in refinement layers five to seven, and no harmonic complex is utilised. The bit rates given in this illustration are cumulative rates.*

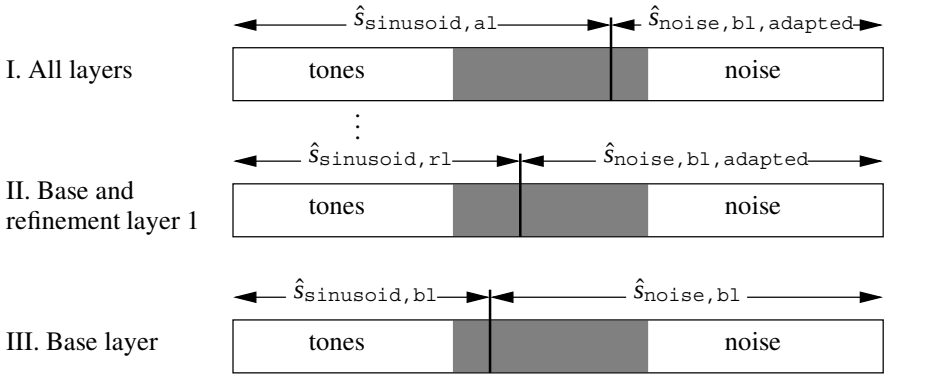
of this figure, a number of sinusoidal parameters are lacking. The resulting audio often sounds un-natural since spectral holes occur when missing sinusoids are not compensated for by noise. Therefore, the sinusoidal ($\hat{s}_{\texttt{sinusoid,rl}}[n]$) and noise components ($\hat{s}_{\texttt{noise}}[n]$) are not well matched in this case. That part of the audio signal not represented by sinusoids is indicated by a $\times$ in this part of the figure.

When only the base layer is available to the decoder, part (a).III of this figure, no noise parameters are available. Sinusoids alone, and in particular sinusoids with continued-phase parameters, rarely produce audio of reasonable quality. Those parts of the audio signal not represented by sinusoids or noise are indicated by $\times$ in this part of the figure.

A more desirable situation is depicted in part (b) of Figure 2.9. Again, the bit-stream contains at least three layers. The difference in comparison to part (a) is that the model boundary is located at a different position in each layer, and the sinusoidal and noise components are well matched in each layer. In this approach, the tonal-noise transition region, as illustrated in Figure 2.5 (a), is exploited to obtain a well-tuned trade-off between sinusoids and noise. The size of the sinusoidal component is determined by bit-rate constraints, and the noise component, which is relatively



(a) Standard approach to layer definition



(b) Desirable approach to layer definition

Figure 2.9: *Representation of an audio signal in a layered bit-stream. (a) In the standard approach to defining layers, the sinusoidal and noise components do not match, since parts of the audio signal remain un-coded when not all layers are available. (b) The desirable approach to defining layers aims to obtain a noise component in each layer that is well matched to the sinusoidal component contained in that particular layer. Thus, in this approach, the complete audio signal is always coded.*

cheap in terms of bit rate, provides a stochastic model of the rest. Keep in mind that if the target bit rate of the base layer is too low, the number of sinusoids admitted to the base layer may not be a sufficient representation of the tonal content of the audio signal.

The loss in quality incurred in the standard approach, described above, is illus-

trated in Figure 2.10 by utilising the rate-distortion characteristics of the sinusoidal and sinusoidal plus noise decompositions (refer to Figure 2.1). We emphasise that the sinusoidal and sinusoidal plus noise rate-distortion characteristics given in this figure are illustrations, and not based on measurement data. When all layers are available in both approaches (at bit rate R_{I}), the sinusoidal and noise components are well matched in both approaches, and the audio quality obtained lies in the sinusoidal plus noise rate-distortion characteristic. When only the base layer is available (at bit rate R_{III}), the sinusoidal signal representation (Figure 2.9 (a).III) suffers a quality loss, denoted by ΔD_{III}, in comparison to the sinusoidal plus noise signal representation (Figure 2.9 (b).III). When the base and first refinement layers are available (at bit rate R_{II}), the audio quality obtained in the standard approach (Figure 2.9 (a).II) lies somewhere in between the rate-distortion characteristic of the sinusoidal and sinusoidal plus noise signal representations, since the sinusoidal and noise signal components are not well matched. A higher audio quality is obtained at this bit rate by the desired approach (Figure 2.9 (b).II), since the sinusoidal and noise signal components are well matched in this approach. The loss in quality suffered by the standard approach in comparison to the desired approach is denoted by ΔD_{II}.

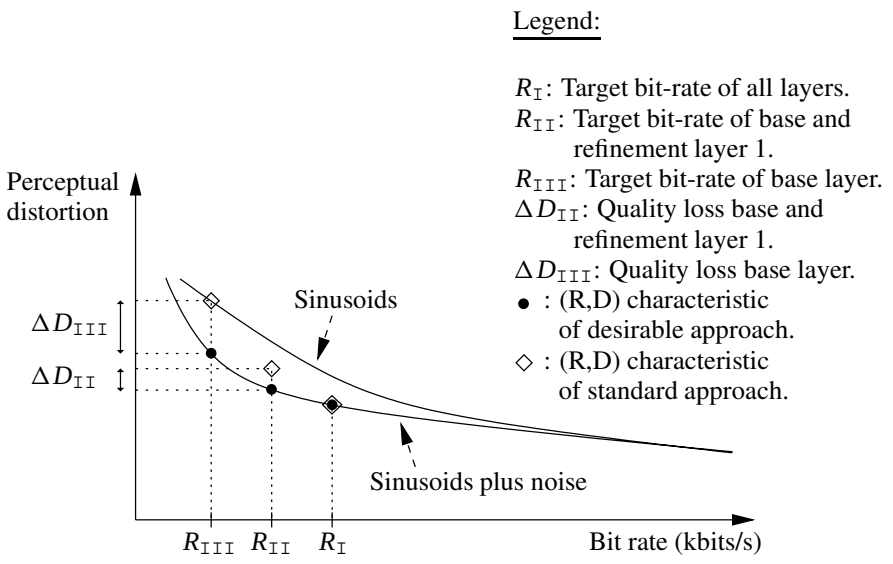


Figure 2.10: *Illustration of the bit rate versus perceptual distortion characteristic of the two approaches to bit-rate scalability illustrated in Figure 2.9.*

2.3.3 General aspects of the proposed bit-rate scalability design

In this section, we propose a suitable mechanism for realising a low-cost well-tuned trade-off between sinusoids and noise in each layer. Adapting the noise component to the sinusoidal component in each layer can be achieved in two ways.

First, the residual corresponding to the sinusoidal component in a particular layer can be coded by the noise coder, for each layer. Unless there is a large amount of redundancy among the resulting noise components, this approach will be inefficient from a bit rate point-of-view.

Second, and more interesting, is to define one noise component, corresponding to the sinusoidal component in the base layer, and to attenuate this component in the decoder if more layers are received. Therefore, the encoding process assumes a parallel structure, as illustrated in Figure 2.11.

The audio signal $s[n]$ is coded by the sinusoidal coder. The sinusoidal parameters are distributed over the base and refinement layers, with the most important param-

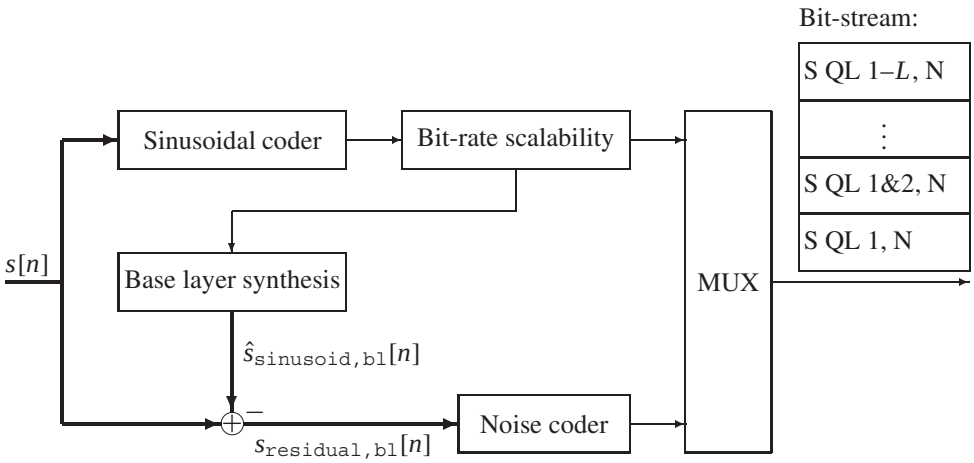


Figure 2.11: *Diagram of the bit-rate scalable encoder. Thick vectors represent signal paths, while thin vectors represent parameter paths. The audio signal $s[n]$ is coded by the sinusoidal coder. The sinusoidal parameters are grouped into layers by the bit-rate scalability functionality. The sinusoidal parameters corresponding to the base layer are synthesised, resulting in the sinusoidal signal $\hat{s}_{\texttt{sinusoid,bl}}[n]$. This signal is subtracted from the input audio signal $s[n]$, resulting in $s_{\texttt{residual,bl}}[n]$, which is coded by the noise coder. The sinusoidal and noise parameters are finally multiplexed (MUX) into a scalable bit-stream, containing a number of layers.*

eters placed in the base layer. The sinusoids contained in the base layer are synthesised, resulting in $\hat{s}_{\texttt{sinusoid,bl}}[n]$, and subtracted from the audio signal, resulting in $s_{\texttt{residual,bl}}[n]$, which is then coded by the noise coder. The noise component is coded in the base layer. Those spectral-temporal aspects of the audio signal described by sinusoids in the refinement layers are described by the noise parameters too. Thus, a dual description of a part of the audio signal is made.

In the bit-stream received by the decoder, a number of layers may be lacking. The sinusoidal decoder obtains the sinusoidal component $\hat{s}_{\texttt{sinusoid}}[n]$ from the sinusoidal parameters contained in the bit-stream. The noise component $\hat{s}_{\texttt{noise,bl}}[n]$ is obtained by the noise decoder. The decoder then adapts the noise component to the sinusoidal component, resulting in $\hat{s}_{\texttt{noise,bl,adapted}}[n]$, see Figure 2.12. Adaptation of the noise component is not a straight-forward process, since a deterministic signal is compared to a stochastic signal. The decoded signal is the sum of the sinusoidal and

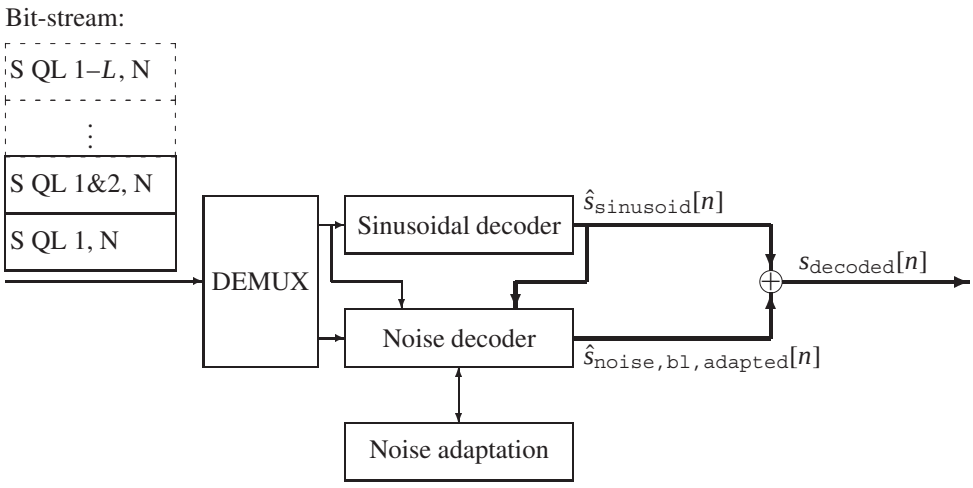


Figure 2.12: *Diagram of the bit-rate scalable decoder. Thick vectors represent signal paths, while thin vectors represent parameter paths. The bit-stream received by the decoder may not contain all layers. Those layers contained in the bit-stream are de-multiplexed (DEMUX) and the sinusoidal parameters are passed on to the sinusoidal decoder, while the noise parameters are passed on to the noise decoder. The noise component is adapted to match the decoded sinusoidal component* $\hat{s}_{sinusoid}[n]$*. The noise component may be adapted by utilising either the sinusoidal parameters or* $\hat{s}_{sinusoid}[n]$*. The adapted noise component, denoted by* $\hat{s}_{noise,bl,adapted}[n]$*, is added to the sinusoidal component to form the decoded signal* $s_{decoded}[n]$*.*

(adapted) noise components,

$$s_{\texttt{decoded}}[n] = \hat{s}_{\texttt{sinusoid}}[n] + \hat{s}_{\texttt{noise,bl,adapted}}[n]. \tag{2.4}$$

Bit-rate scalability, based on scaling the sinusoidal component in the encoder and appropriately adapting the noise component in the decoder, is considered in Chapter 4.

2.4 Summary

In this chapter, we considered the decomposition of an audio signal into sinusoids and noise. Representing an audio signal by sinusoids alone is not efficient from a bit rate versus audio quality point-of-view. A sinusoidal plus noise signal representation is more efficient at low bit rates.

To compensate for dynamic signal behaviour, an audio signal is analysed on a per-frame basis to obtain the parameters representing the sinusoidal and noise signal components.

The most important issues arising in sinusoidal coding were considered. In particular, it was argued that temporal non-stationarity of the audio signal in a frame complicates the detection of partials from the amplitude spectrum. The advantages of multi-resolution analysis were illustrated. Furthermore, the need for a model of the sinusoidal component which is capable of capturing dynamic signal behaviour was formulated.

Psycho-acoustic models play an important role in audio coding. The main determinants and shortcomings of the masked threshold obtained from popular masking models were described. The shortcomings of these masking models are overcome by a new masking model, which takes both integration of acoustical information over time and frequency into account. The consequences of the ability of the human auditory system to integrate acoustical information over time and frequency for parametric audio coding were briefly considered.

An overview of existing bit-rate scalable parametric audio coders was given. The main drawback of these scalable coders is that they are not able to maintain a well-tuned trade-off between sinusoids and noise in all layers comprising the scalable bit-stream. The main contribution of this chapter is the proposal of a new codec structure to achieve a well-tuned trade-off between sinusoids and noise in all layers. In our proposal, the structure of both the encoder and decoder differs from that of existing bit-rate scalable parametric audio coders.

2.5 Conclusion

This chapter has laid the foundation upon which the remainder of the thesis will be based. In particular, two core issues mentioned in this chapter will be considered in

detail in the remainder.

The first issue is the need for a flexible model of the sinusoidal component and an algorithm to obtain accurate estimates of the model parameters. Chapter 3 considers this issue, and provides a modular design of the analysis process.

The second issue is the need for a well-tuned balance between tones and noise in a bit-rate scalable parametric audio coder. Current bit-rate scalable parametric audio coders are not capable of delivering a well-tuned balance between tones and noise for all layers. We have proposed a new codec structure to achieve a well-tuned balance between tones and noise. A bit-rate scalable codec, based on this codec structure, is developed in Chapter 4.

Chapter 3

Analysis, Coding, and Decoding

3.1 Introduction

The purpose of this chapter is to describe the analysis of an audio signal by the encoder, and the synthesis of the decoded audio signal by the decoder. The notation used as well as a number of definitions are given in Section 3.2. The encoder consists of a sinusoidal and a noise coder. The sinusoidal coder is discussed in Section 3.3, and the noise coder in Section 3.4. Design specifications for both the sinusoidal and noise coders are given in Section 3.5. The decoder is discussed in Section 3.6. Results of the sinusoidal coder are discussed in Section 3.7 by considering both synthetic and real-world signals.

3.2 Notation and definitions

The discrete counterpart $s[n]$ of a continuous-time audio signal $s(t)$ is obtained by sampling $s(t)$ at a rate of f_s Hz. The analysis process is frame-based and therefore, we restrict ourselves to describing the analysis of a single frame. For convenience, we translate the frame such that it is centred around sample $n = 0$. As a consequence, the frame contains an odd number of samples N_s, and the total frame duration is

$$T = \frac{N_s}{f_s} \text{ seconds.}$$

The frame is denoted in vector notation by

$$\mathbf{s} = \left[\; s[-\tfrac{N_s-1}{2}] \quad \ldots \quad s[0] \quad \ldots \quad s[\tfrac{N_s-1}{2}] \right]^{\mathrm{T}}, \tag{3.1}$$

where $^{\mathrm{T}}$ denotes transposition. The model parameters are obtained by analysing a weighted, or windowed, version of the frame. A discrete window with positive elements is denoted in vector notation by

$$\mathbf{w} = \left[\; w[-\tfrac{N_s-1}{2}] \quad \ldots \quad w[0] \quad \ldots \quad w[\tfrac{N_s-1}{2}] \right]^{\mathrm{T}}, \tag{3.2}$$

and the windowed signal is denoted as

$$\mathbf{s}_{\mathbf{w}} = \mathrm{diag}(\mathbf{w})\,\mathbf{s}, \tag{3.3}$$

where $\mathrm{diag}(\mathbf{w})$ is the diagonal matrix with the elements of $\mathbf{w}$ on the leading diagonal and zeros elsewhere.

The audio signal $s[n]$ is approximated by a signal $s_{\mathtt{model}}[n]$ which is completely determined by a set of model parameters. The aim of the analysis process is to find the optimal set of model parameters which minimises the cost function

$$c = \|\mathbf{s} - \mathbf{s}_{\mathtt{model}}\|_{\mathbf{w}}^2. \tag{3.4}$$

The weighted norm of $\mathbf{x} \in \mathbb{C}$ is

$$\|\mathbf{x}\|_{\mathbf{w}} \triangleq \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbf{w}}} \in \mathbb{R},$$

and is based on the weighted Euclidean inner-product

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{w}} \triangleq \mathbf{y}^{\mathrm{H}} \mathrm{diag}(\mathbf{w})\,\mathbf{x} \in \mathbb{C}, \tag{3.5}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{C}^{N_{\mathrm{s}}}$ and $\mathbf{y}^{\mathrm{H}}$ is the Hermitian of $\mathbf{y}$. The spectrum of $\mathbf{s}$ is obtained by applying the DFT operation

$$(\mathrm{DFT}\,\mathbf{s})[m] \triangleq \sum_n s[n] W_{N_{\mathrm{s}}}^{mn}, \tag{3.6}$$

where $m \in \{-\frac{N_{\mathrm{s}}-1}{2}, \ldots, \frac{N_{\mathrm{s}}-1}{2}\}$. The Fourier matrix is related to $W_{N_{\mathrm{s}}} = \exp\{-j B_{\mathrm{s}}\}$, where B_{s} is the DFT bin-size

$$B_{\mathrm{s}} = \frac{2\pi}{N_{\mathrm{s}}} \text{ radians.}$$

3.3 Sinusoidal analysis

The sinusoidal component is divided into a harmonic complex sub-component and an individual sinusoids sub-component. A harmonic sub-component is utilised for two reasons. The first reason is that harmonic sounds occur naturally, sometimes in isolation, and sometimes in combination with other sounds. The second reason is that the frequency parameters of a harmonic complex can be represented by only two parameters in the simplest case: the fundamental frequency and the number of harmonics. This is an efficient representation, especially if the number of harmonics is high. Individual sinusoids are used to model partials that are not part of the harmonic complex. This decomposition of the sinusoidal component has been adopted in several parametric audio coders, see e.g. [75, 23, 76]. In PPC, only individual sinusoids are utilised [12].

To improve readability of the sinusoidal models and the analysis methods, we consider the complex-valued counterpart of a real-valued signal, obtained by applying the Discrete Hilbert Transform [77, pp. 59–62].

Ideally, harmonic frequencies are integer multiples of the fundamental frequency θ_{f}. The simplest model of a harmonic complex, where the amplitude and frequency of the harmonic partials are constant for the duration of the frame, is

$$s_{\mathrm{hc}}[n] = \sum_{k=1}^{N_{\mathrm{h}}} a_k e^{j(\theta_{k,1} + k\theta_{\mathrm{f}} n)}. \tag{3.7}$$

In this expression, N_{h} denotes the number of harmonics, $a_k \in \mathbb{R}^+$ the constant amplitude of the kth harmonic and $\theta_{k,1} \in [-\pi, \pi)$ its phase. Similarly, the basic constant-amplitude and constant-frequency model of individual sinusoids is

$$s_{\mathrm{is}}[n] = \sum_{k=1}^{N_{\mathrm{c}}} a_k e^{j(\theta_{k,1} + \theta_{k,2} n)}, \tag{3.8}$$

where N_{c} denotes the number of sinusoids and $\theta_{k,2}$ the frequency of sinusoid k.

Even though these basic models are often utilised in parametric audio coders, they are inadequate for a wide range of audio signals, as we will argue in the following section. Extensions of these models and current methods to estimate their parameters are reviewed in Section 3.3.1. We present our model in Section 3.3.2 and our parameter-estimation algorithm in Section 3.3.3.

3.3.1 Existing sinusoidal models and estimation techniques

Harmonic complex

The basic harmonic signal-model given in (3.7) is insufficient in practice, for two reasons.

The *first* reason is the temporal non-stationarity of certain harmonic signals. The non-stationary nature of voiced speech, in particular, has led to harmonic models incorporating dynamic signal behaviour in a frame. For instance, the time warper proposed by Sluijter and Janssen is based on a linear instantaneous-frequency $k(\theta_{\mathrm{f}} + 2\theta_{\mathrm{c}} n)$ of partial k, where θ_{c} is the frequency chirp and n is discrete time [61]. Non-stationary amplitude behaviour is common too, refer to Figure 2.6 on page 26 for an illustration. Amplitude variation can be modelled by a low-order polynomial. The pitch (or fundamental frequency) of voiced speech is a very important parameter in most speech-coding paradigms. As a result, pitch estimation has received considerable attention in literature, see e.g. [78, 79, 60, 80]. In addition to an estimate of the fundamental frequency, an estimate of the frequency chirp in voiced speech can be obtained by applying time warping [61].

The *second* reason is that for stiff-stringed musical instruments, like the piano, the integer relation between the frequency of harmonic partial k and the fundamental frequency θ_{f}, namely $k\theta_{\mathrm{f}}$, is a reasonable approximation for the first few harmonic partials at most. Stiff strings result in raised frequencies of the harmonic partials. This effect is called stretching or inharmonicity, and is most prominent in the higher harmonics. For a stiff string, the frequency of the kth harmonic is predicted as

$$k\theta_{\mathrm{f}}\sqrt{1 + Bk^2}, \tag{3.9}$$

where the inharmonicity parameter

$$B = \frac{\pi^3 d^4 E}{64 T L^2} > 0 \tag{3.10}$$

depends on the string attributes: diameter d, Young's elasticity modulus of the string material E, tension T, and length L [48]. A number of techniques have been described in literature to estimate both B and θ_{f}. Lattard measured these parameters by means of manual estimation of partial frequencies, aided by a spectral analyser, and application of a variant of (3.9) [81]. Galembo and Askenfelt estimated these parameters by utilising inharmonic comb filters and applying a full search over pre-specified ranges of θ_{f} and B [82]. Later, Galembo and Askenfelt applied pitch detectors based on the cepstrum and harmonic product spectrum, as used in speech coders, to estimate these parameters [83].

The only parametric coder found in literature that incorporates inharmonicity in its model of the harmonic complex is HILN [75], where the frequency of the kth harmonic is predicted as [84, Annex A]

$$k\theta_{\mathrm{f}}(1 + \tilde{B}k). \tag{3.11}$$

An initial, coarse, estimate of the fundamental frequency is derived from the cepstrum of the audio signal. Thereafter, the frequency parameters of all harmonics, spanning the complete frequency band and derived from the fundamental frequency, are refined by performing a regression-based high-accuracy frequency-estimation technique [21]. These refined frequency parameters are then utilised to refine the initial estimate of the fundamental frequency and to estimate $\tilde{B}$ by minimising the total error between the refined frequency parameters and those calculated according to $k\theta_{\mathrm{f}}(1 + \tilde{B}k)$ [84, Annex A].

Individual sinusoids

Like the basic harmonic signal-model, the model of individual sinusoids, see (3.8), is inadequate when the partials are non-stationary. To account for temporal non-stationarity, this model has to be extended. Two kinds of temporal non-stationarity

can be identified: non-stationarity in amplitude and in frequency. Non-stationary amplitude behaviour has been considered in the modelling of transients. It was observed that transients can be modelled well with damped sinusoids, where the amplitude parameters exhibit exponential variation $a_k[n] = a_k e^{\gamma_k n}$, with $\gamma_k < 0$ [22, 85]. Such exponential amplitude variation was combined with signal-dependent segmentation to improve transient modelling in a parametric audio coder [86, 87]. Alternatively, modelling non-stationary amplitude behaviour of partials by polynomials was proposed by George and Smith [51]. Non-stationary frequency behaviour is usually modelled by phase polynomials. George and Smith considered a quadratic phase polynomial of the form $\theta_k[n] = \theta_{k,1} + \theta_{k,2}n + \theta_{k,3}n^2$ sufficient for modelling non-stationary partials [51].

Obtaining optimal values of the frequency $\theta_{k,2}$ and chirp $\theta_{k,3}$ is a difficult problem in general, since these parameters are contained non-linearly in the cost function (3.4). Several time-frequency distributions have been utilised to obtain estimates of the frequency and chirp parameters. Among them, the short-time Fourier transform (STFT) [88] is the most popular. Although being computationally efficient, the spectro-temporal resolution afforded by the STFT is not very flexible. Furthermore, when the STFT is applied to a frame taken from an audio signal, the frame is decomposed into a set of *stationary* basis functions. If the audio signal is *non-stationary* in the frame, the decomposition provided by the STFT will become distorted, as we have shown in Section 2.2.1. However, as long as the signal exhibits modest non-stationary in the frame, the decomposition provided by the STFT is still useful. A straight-forward method for obtaining (initial) estimates of the frequency parameters is to identify peaks in the amplitude spectrum of $\mathbf{s_w}$ (also called peak-picking), as described in Section 2.2.1. The accuracy of the frequency estimates thus obtained is determined by the DFT bin-size. More accurate estimates are usually desirable from a psycho-acoustical point-of-view, and a number of methods have been applied in parametric audio coding to improve the accuracy of initial frequency estimates obtained by peak-picking. Such analysis methods include:

Zero padding and interpolation. Zero-padding the frame and interpolating the spectrum in the vicinity of peaks is a simple and fast way of obtaining more accurate frequency estimates [23].

Signal derivatives. Signal derivatives were utilised by Desainte-Catherine and Marchand to obtain accurate frequency-parameter estimates [89].

Phase distortion analysis. Local distortions in the phase spectrum around peak locations contain information about the temporal behaviour of the corresponding partial. This information has been exploited by Masri to estimate a linear frequency chirp and an exponential amplitude [90, 76].

Constrained optimisation. Hamdy and Tewfik formulated a constrained (non-linear) optimisation problem with cost function similar to the cost function given in (3.4) [91] . The sinusoidal frequency parameters are restricted to ensure that only the DFT bin in which the peak was identified is considered:

$$\theta_{2,\texttt{initial}} - \frac{B_{\texttt{s}}}{2} < \theta_{2,\texttt{improved}} < \theta_{2,\texttt{initial}} + \frac{B_{\texttt{s}}}{2},$$

where $B_{\texttt{s}}$ is the DFT bin-size, $\theta_{2,\texttt{initial}}$ the initial estimate of the frequency, and $\theta_{2,\texttt{improved}}$ the improved estimate of the frequency. The amplitudes are restricted in a similar manner. The solution of this problem yields accurate estimates of the frequency and amplitude parameters.

Gauss-Newton optimisation. Depalle and Hélie utilised the Gauss-Newton method to obtain accurate amplitude and frequency estimates [92]. Their signal model was a sum of sinusoids with constant amplitude and constant frequency. The first-order approximation of the spectrum of the weighted signal model around the model-parameter estimates was used in their method. Due to the sensitivity of their method to the window shape, they designed windows with no side-lobes. In the iterative process, sinusoids were merged when they became too close in frequency.

Newton's method. Vos et al. utilised a gradient search based on Newton's method to improve initial frequency estimates [93]. This method was extended by Heusdens and Vos to include exponential amplitude behaviour [86].

Linear regression on the phase data. Tretter has shown that frequency estimation is equivalent to linear regression on the phase data [94]. A disadvantage of his method is that the phase must be unwrapped before regression can be applied. Kay avoided phase unwrapping by considering the differentiated phase data [95]. Kay's technique has been extended and applied in parametric audio coding to improve frequency estimates and obtain chirp estimates by Edler et al. [21].

Furthermore, two main approaches to obtaining frequency and amplitude parameters of partials in a frame can be identified. The *first* approach is called matching pursuits [15], and is a greedy iterative algorithm that selects the sinusoid (denoted by $\mathbf{g}_{\texttt{opt}}$ and parameterised by an amplitude, phase, and frequency) that best fits the audio signal $\mathbf{s}$, in the sense that $|\langle \mathbf{s}, \mathbf{g}_{\texttt{opt}} \rangle_{\mathbf{w}}|$ is a maximum for all possible sinusoids $\mathbf{g}$, in each iteration. The sinusoid is subtracted from the signal to obtain a residual signal which will be considered in the following iteration. Matching pursuits has been extended to include a psycho-acoustical distortion measure, and the resulting method is referred to as psychoacoustic adaptive matching pursuits [16]. An advantage of

matching pursuits is that spectral peaks due to sidelobes will not be extracted. An important disadvantage is that errors made in the extraction of a sinusoid will deteriorate the accuracy of sinusoidal parameters estimated in subsequent iterations. The *second* approach is based on identifying all spectral peaks simultaneously [17]. The frequency parameters are obtained from the peak locations first. Subsequently, the amplitude and phase parameters can be obtained from amplitude and phase spectra, respectively, or by solving a set of linear least-squares equations [51]. An important advantage of this approach is that no error propagation takes place. A disadvantage is that spectral peaks, due to sidelobes, may be selected in the process of identifying spectral peaks.

A more flexible spectral-temporal resolution for a single-partial signal with a linear frequency chirp is obtained by applying the Wigner distribution (WD) [88]. The major drawback of the WD is large cross-terms stemming from the case where multiple partials are present in the signal. To remedy this situation, a weighted version of the WD is often used, leading to the Cohen class of time-frequency distributions [88]. Parametric estimation techniques that exploit the explicit polynomial-phase structure of a signal model usually utilise the Fourier spectrum of the WD or Cohen distributions. The resulting spectra are referred to as the ambiguity function (AF) or high-order ambiguity function (HAF) [96], respectively. However, the (H)AF suffers from the presence of cross-terms when more than one complex-exponential is present in the signal. To suppress cross-terms, the product high-order ambiguity function (PHAF) was proposed [97]. Multi-partial polynomial-phase signals with a wide dynamic range of amplitudes remain problematic, however, even when using the PHAF. An algorithm that is able to deal with a large dynamic range of amplitude parameters, and which may therefore be suitable for application in parametric audio coding, was proposed by Ikram and Zhou [98]. This algorithm estimates the number of partials and the phase-polynomial order of each partial. This is done iteratively, where the order of the most prominent component in the signal is estimated, and the component subsequently removed from the signal, whereupon the process is repeated. Like other matching pursuit algorithms, the error made in the extraction of one partial influences the extraction of the following ones [98]. Error propagation is undesirable, especially if the number of components in the signal is large. For this reason, we will not consider this algorithm further.

3.3.2 Proposed model of the sinusoidal component

Our model of the harmonic complex, which integrates both factors mentioned in Section 3.3.1, namely temporal non-stationarity and inharmonicity, is not utilised in any other parametric audio coder. Temporal non-stationarity and inharmonicity of the harmonic partials are modelled by defining the phase function ξ_k of component k

as

$$\xi_k(\theta, B) \triangleq k\theta\sqrt{1 + Bk^2}. \tag{3.12}$$

Note that the case $B = 0$ reduces this expression to the usual true-harmonic relation, which makes it suitable for harmonic signals where no stretching effect is present. The expression (3.12) is derived from the basic wave equation for the ideal string [48]. As a result, the partial frequencies measured for real stiff strings abide closely by this expression [48, 81, 99], while the model utilised in HILN (3.11) is less accurate.

Polynomial amplitude behaviour can be measured efficiently by solving a set of linear equations once the frequency and chirp parameters are estimated. A linear polynomial is considered sufficient to model amplitude variation in a frame.

Therefore, the complete model of the harmonic complex is

$$s_{\mathtt{hc}}[n] = \sum_{k=1}^{N_{\mathtt{h}}} (a_{k,1} + a_{k,2}n) e^{j\,(\theta_{k,1} + \xi_k(\theta_{\mathtt{f}} + \theta_{\mathtt{c}} n, B) n)}, \tag{3.13}$$

where $\theta_{k,1} \in [-\pi, \pi)$ is the phase, $a_{k,1} \in \mathbb{R}^+$ the constant amplitude, and $a_{k,2} \in \mathbb{R}$ the amplitude sweep of harmonic k. The amplitude sweep is restricted such that

$$a_{k,1} + a_{k,2}n > 0 \tag{3.14}$$

for all n. The contribution of the chirp to the instantaneous frequency in a frame is usually small. It is insightful to express the contribution of the chirp to the instantaneous frequency as a percentage

$$\theta_{\mathtt{c},\%} = \frac{|\theta_{\mathtt{c}}|}{\theta_{\mathtt{f}}} 100 N_{\mathtt{s}}.$$

The frequency chirp is restricted by $\theta_{\mathtt{c},\%} \leq \theta_{\mathtt{c},\mathtt{max}\%}$. Note that this yields an indirect restriction on the frame length $N_{\mathtt{s}}$. Furthermore, the inharmonicity parameter can only attain a limited range of values. For instance, for the piano, B lies in the range $0 < B \lessapprox 0.015$ [100]. In the interest of simplicity, we utilise only one harmonic complex in the analysis.

Similarly, our model of the individual-sinusoids sub-component contains both polynomial amplitude and polynomial frequency variation in the frame:

$$s_{\mathtt{is}}[n] \triangleq \sum_{k=1}^{N_{\mathtt{c}}} (a_{k,1} + a_{k,2}n) e^{j\,(\theta_{k,1} + (\theta_{k,2} + \theta_{k,3} n) n)}. \tag{3.15}$$

This model resembles the proposal of George and Smith [51]. Again, the contribution of the chirp to the instantaneous frequency is expressed as a percentage

$$\theta_{k,3,\%} = \frac{|\theta_{k,3}|}{\theta_{k,2}} 100 N_{\mathtt{s}},$$

and is restricted by $\theta_{k,3,\%} \leq \theta_{3,\max\%}$.

The sinusoidal component $s_{\mathtt{sinusoid}}[n]$ is then given by

$$s_{\mathtt{sinusoid}}[n] = s_{\mathtt{hc}}[n] + s_{\mathtt{is}}[n]. \tag{3.16}$$

3.3.3 Parameter analysis of the audio signal

Preliminaries of the analysis

The analysis process is split into two parts: per frame, parameters describing the non-constant polynomial phase ($\theta_{\mathtt{f}}$, $\theta_{\mathtt{c}}$, B, $\theta_{k,2}$, and $\theta_{k,3}$) and parameters describing the amplitude and constant phase ($a_{k,1}$, $a_{k,2}$, and $\theta_{k,1}$) are estimated. The estimation of $\theta_{\mathtt{f}}$, $\theta_{\mathtt{c}}$, B, $\theta_{k,2}$, and $\theta_{k,3}$ is further divided into two steps: detection and parameter improvement. Detecting the presence of a harmonic complex is based on identifying periodicity in $s[n]$, and is carried out by the Detector and Pitch Estimator (DPE) module, which provides an initial estimate $\hat{\theta}_{\mathtt{f}}$ of the fundamental frequency. Detection of individual sinusoids is based on identifying spectral peaks, and is carried out by the Initial Frequency Estimation (IFE) module. The IFE module provides initial estimates of the individual-sinusoid frequency parameters $\hat{\theta}_{k,2}$ and distortion parameters D_k.

The purpose of the Harmonic Parameter Estimation (HPE) module is to minimise the cost function

$$c_{\mathtt{hc}} = \|\mathbf{s} - \mathbf{s}_{\mathtt{hc}}\|_{\mathbf{w}}^2$$

by refining the harmonic polynomial-phase parameter estimates $\hat{\theta}_{\mathtt{f}}$, $\hat{\theta}_{\mathtt{c}}$, and $\hat{B}$. Minimising the cost function $c_{\mathtt{hc}}$ is a difficult problem since these parameters are contained non-linearly in the cost function. We employ an iterative search technique to find the parameters that minimise the cost function. Furthermore, we observe that if the initial estimates of $\theta_{\mathtt{f}}$, $\theta_{\mathtt{c}}$, and B are far from the optimal parameters, an iterative search method may become stuck in a local minimum of the cost function, and the refined parameters may not be globally optimal.

Similarly, the purpose of the Sinusoidal Parameter Estimation (SPE) module is to minimise the cost function

$$c_{\mathtt{is}} = \|\mathbf{s} - \mathbf{s}_{\mathtt{is}}\|_{\mathbf{w}}^2$$

by refining the individual-sinusoids polynomial-phase parameter estimates $\hat{\theta}_{k,2}$ and $\hat{\theta}_{k,3}$. The SPE module utilises an iterative search technique to minimise the cost function $c_{\mathtt{is}}$. In PPC, no iterative search technique is utilised to refine initial frequency parameter estimates of individual sinusoids.

The detection module (DPE) of the harmonic complex and the improvement module (HPE) of the harmonic-model parameters are contained in the Harmonic Complex Analysis (HCA) module; similarly, the Individual Sinusoidal Analysis (ISA) module contains the IFE and SPE modules, see Figure 3.1. After detection and parameter

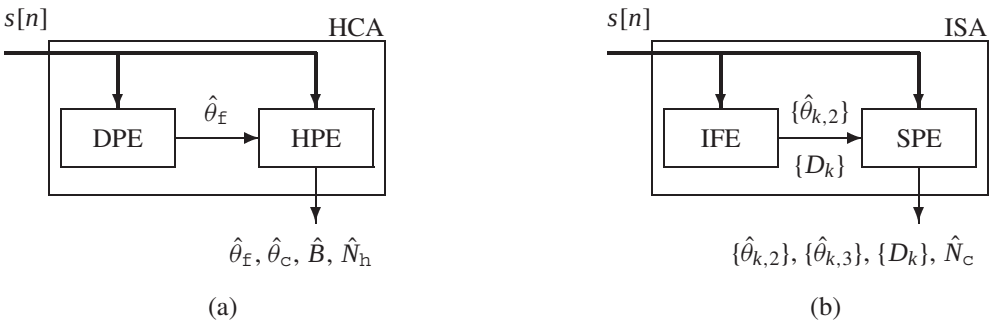


Figure 3.1: *The analysis modules utilised to obtain the individual-sinusoids and harmonic-complex model parameters. (a) The Harmonic Complex Analysis (HCA) module comprises the Detector and Pitch Estimator (DPE), and Harmonic Parameter Estimation (HPE) modules. The DPE module analyses the audio signal $s[n]$ to determine whether a pitch is present. The estimate of the pitch, or fundamental frequency, is denoted by $\hat{\theta}_f$. The HPE module refines $\hat{\theta}_f$ and determines $\hat{\theta}_c$, $\hat{B}$, and $\hat{N}_h$ by analysing the audio signal. (b) The Individual Sinusoidal Analysis (ISA) module comprises the Initial Frequency Estimation (IFE) and Sinusoidal Parameter Estimation (SPE) modules. The IFE module detects and selects individual sinusoids in the audio signal $s[n]$, and is based on peak-picking. The frequency estimates are denoted by $\{\hat{\theta}_{k,2}\}$, and the corresponding distortion parameters by $\{D_k\}$. The SPE module refines $\{\hat{\theta}_{k,2}\}$ and determines $\{\hat{\theta}_{k,3}\}$ by analysing the audio signal. The number of individual sinusoids is denoted by $\hat{N}_c$.*

improvement, the Collective Amplitude Parameters (CAP) module, see Figure 3.2, estimates the optimal polynomial-amplitude ($\hat{a}_{k,1}$ and $\hat{a}_{k,2}$) and constant-phase parameters ($\hat{\theta}_{k,1}$) that minimise the cost function

$$c_{\texttt{sinusoid}} = \|\mathbf{s} - \mathbf{s}_{\texttt{sinusoid}}\|^2_{\mathbf{w}},$$

given the improved polynomial-phase parameters of the harmonic complex and individual sinusoids. In contrast to the polynomial-phase parameters, the polynomial-amplitude and constant phase parameters are contained linearly in the cost function $c_{\texttt{sinusoid}}$, and the optimal parameters can be directly estimated.

Overview of the proposed analysis

Detecting the pitch of voiced speech has received widespread attention in the field of speech coding, and many successful strategies exist, see e.g. [78, 79, 60, 80]. Our design of the DPE module is based on ideas gathered from existing pitch detection strategies. Detection of individual sinusoids is based on spectral peak-picking, peak validation, and estimating the perceptual relevance of the peak.

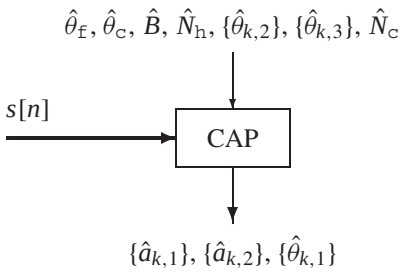


Figure 3.2: *The Collective Amplitude Parameters (CAP) module determines the polynomial-amplitude, $\hat{a}_{k,1}$ and $\hat{a}_{k,2}$, and constant-phase parameters $\hat{\theta}_{k,1}$, given the higher-order polynomial-phase parameters of the harmonic complex and individual sinusoids from the audio signal $s[n]$.*

The initial parameter estimates provided by the DPE and IFE modules are close to the optimal values in most cases. A suitable method for fine-tuning these initial estimates is the classical Gauss-Newton method. As we have mentioned in Section 3.3.1, Newton and Gauss-Newton optimisation techniques have been applied in parametric audio coding to refine constant frequency and (exponential) amplitude parameters of individual sinusoids. Recall that our model of individual sinusoids contains an additional frequency chirp, which requires the extension of existing techniques. Furthermore, no references were found in literature in which Newton-like optimisation techniques were utilised to refine estimates of the fundamental frequency, frequency chirp, and inharmonicity parameters of the harmonic complex. Levenberg-Marquardt optimisation is an extension of the Gauss-Newton method, with the added advantage that it is a descent method, and convergence is thus guaranteed [101, 102]. More details concerning Levenberg-Marquardt optimisation, including improvements on the proposal made by Marquardt, can be found in the textbook by Dennis and Schnabel [103]. In Appendix A, we provide a short overview of the Gauss-Newton and Levenberg-Marquardt optimisation methods. The parameter-improvement modules (HPE and SPE) are based on Levenberg-Marquardt optimisation.

The main ideas behind these modules are described in the following. We start by describing the analysis of individual sinusoids, after which the analysis of the harmonic complex is described. Next, we describe how the amplitude and constant-phase parameters are determined. We conclude by proposing a suitable strategy for reducing the computational complexity of the analysis. Details concerning the analysis are given in the design specifications in Section 3.5.

The IFE module

The purpose of the IFE module is to provide rough estimates of the sinusoidal frequencies and the corresponding perceptual relevance of each sinusoid. Partials are detected by identifying peaks in the amplitude spectrum of the weighted audio signal $\mathbf{s_w}$ (3.3). The amplitude spectra corresponding to a number of time scales are considered. The IFE module is illustrated in Figure 3.3. In this figure, the set $\hat{\Theta}_{\texttt{initial}} = \{\hat{\theta}_{k,2,\texttt{initial}}\}$ contains the initial estimates of the sinusoidal frequencies, obtained by applying peak picking (PP) to $|(\mathrm{DFT}\,\mathbf{s_w})[m]|$. The PEAK module selects those sinusoids with frequencies $\{\hat{\theta}_{k,2}\}$ from $\hat{\Theta}_{\texttt{initial}}$ which satisfy some criterion. The PEAK module is described in the following section. The distortion D_k associated with sinusoid k is determined by the MASK module, which is an implementation of the psycho-acoustical masking model described in [70]. A value of $D_k = 1$ implies a distortion at the threshold of detectability. Values larger than one imply detectability, and values smaller than one imply that the distortion is masked, and therefore not detectable. Sinusoids with a distortion less than $D_{\texttt{min}}$ are removed from $\{\hat{\theta}_{k,2}\}$.

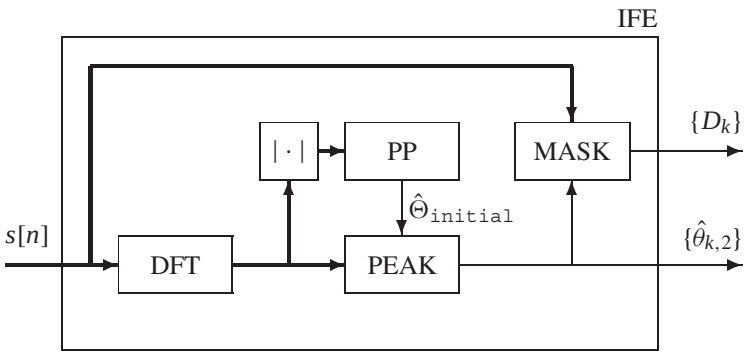


Figure 3.3: *Block diagram of the Initial Frequency Estimation (IFE) module. The spectrum of the audio signal $s[n]$ is calculated by the DFT module. Peak picking (PP) is applied to the amplitude spectrum of $s[n]$ to obtain a set of initial frequency estimates $\hat{\Theta}_{initial}$. The PEAK module selects those frequencies $\hat{\theta}_{k,2}$ from $\hat{\Theta}_{initial}$ that satisfy a certain definition. The distortion D_k associated with a sinusoid with frequency $\hat{\theta}_{k,2}$ is determined by the MASK module.*

The PEAK module

Temporal non-stationarity of partials in the frame will result in broader spectral peaks, as was shown in Section 2.2.1. One of the main challenges in parametric audio coding is finding a reliable criterion for the identification of spectral peaks, even when partials are non-stationary. The purpose of the PEAK module is to formulate such a criterion. The output of this module is the subset of frequencies $\{\hat{\theta}_{k,2}\}$ from the complete set of frequencies $\hat{\Theta}_{\texttt{initial}}$ which satisfy the criterion. The technique described in this section to identify sinusoids resembles the technique proposed by Thomson [104].

From the window $w[n]$ used to window the audio signal $s[n]$, see (3.2), the following amplitude-modulated windows are derived:

$$p_1[n] = \frac{w[n]}{\|w[n]\|} \tag{3.17}$$

$$p_2[n] = \frac{nw[n]}{\|nw[n]\|} \tag{3.18}$$

$$p_3[n] = \frac{n^2 w[n]}{\|n^2 w[n]\|} \tag{3.19}$$

$$\vdots$$

$$p_K[n] = \frac{n^K w[n]}{\|n^K w[n]\|}. \tag{3.20}$$

The amplitude-modulated windows $p_1[n], \ldots, p_K[n]$ are used to capture non-stationary signal behaviour. We observe that the effective spectral bandwidth, or width of the mainlobe(s), of an amplitude-modulated window $p_k[n]$ increases with increasing k.

We denote the set of indices on the DFT grid, corresponding to the set of frequencies $\hat{\Theta}_{\texttt{initial}}$, by $M = \{m_k\}$. Our method is based on matching a linear combination of the amplitude-modulated windows to the weighted audio signal $\mathbf{s_w}$ in the frequency domain around each estimate $m = m_k$. To this end, we isolate the spectral peak located at bin m_k by defining $\mathbf{y}_{\texttt{translated},k}$ to be equal to $(\mathrm{DFT}\,\mathbf{s_w})[m]$ in the interval $[m_k - M_{\mathrm{bw}}, m_k + M_{\mathrm{bw}}]$. Therefore,

$$\mathbf{y}_{\texttt{translated},k} = \left[\, (\mathrm{DFT}\,\mathbf{s_w})[m_k - M_{\mathrm{bw}}] \quad \ldots \quad (\mathrm{DFT}\,\mathbf{s_w})[m_k] \quad \ldots \quad (\mathrm{DFT}\,\mathbf{s_w})[m_k + M_{\mathrm{bw}}] \right]^{\mathrm{T}}.$$

The bandwidth $2M_{\mathrm{bw}} + 1$ around m_k, covered by $\mathbf{y}_{\texttt{translated},k}$, should be matched to the effective bandwidth of the amplitude-modulated windows given above.

In a similar manner, the amplitude-modulated windows are transformed to the

frequency domain and we consider their spectra in the interval $[-M_{\mathtt{bw}}, M_{\mathtt{bw}}]$:

$$\begin{aligned}
\mathbf{x}_1 &= \begin{bmatrix} (\mathrm{DFT}\,\mathbf{p}_1)[-M_{\mathtt{bw}}] & \dots & (\mathrm{DFT}\,\mathbf{p}_1)[0] & \dots & (\mathrm{DFT}\,\mathbf{p}_1)[M_{\mathtt{bw}}] \end{bmatrix}^{\mathrm{T}} \\
\mathbf{x}_2 &= \begin{bmatrix} (\mathrm{DFT}\,\mathbf{p}_2)[-M_{\mathtt{bw}}] & \dots & (\mathrm{DFT}\,\mathbf{p}_2)[0] & \dots & (\mathrm{DFT}\,\mathbf{p}_2)[M_{\mathtt{bw}}] \end{bmatrix}^{\mathrm{T}} \\
\mathbf{x}_3 &= \begin{bmatrix} (\mathrm{DFT}\,\mathbf{p}_3)[-M_{\mathtt{bw}}] & \dots & (\mathrm{DFT}\,\mathbf{p}_3)[0] & \dots & (\mathrm{DFT}\,\mathbf{p}_3)[M_{\mathtt{bw}}] \end{bmatrix}^{\mathrm{T}} \\
&\vdots \\
\mathbf{x}_K &= \begin{bmatrix} (\mathrm{DFT}\,\mathbf{p}_K)[-M_{\mathtt{bw}}] & \dots & (\mathrm{DFT}\,\mathbf{p}_K)[0] & \dots & (\mathrm{DFT}\,\mathbf{p}_K)[M_{\mathtt{bw}}] \end{bmatrix}^{\mathrm{T}}.
\end{aligned}$$

We define $\mathbf{y}_{\mathtt{model}}$ to be a linear combination of the patterns $\mathbf{x}_1, \dots, \mathbf{x}_K$. Therefore,

$$\mathbf{y}_{\mathtt{model}} \triangleq \sum_{l=1}^{K} c_l \mathbf{x}_l.$$

Our aim now is to determine how well $\mathbf{y}_{\mathtt{model}}$ can be fitted to the spectral peak captured in $\mathbf{y}_{\mathtt{translated},k}$. If the partial is stationary, we expect that $\mathbf{y}_{\mathtt{translated},k}$ will be closely approximated by the pattern $c_1\mathbf{x}_1$. If the partial is non-stationary, more patterns will be required. The optimal coefficients $\hat{c}_l$ are found by minimising

$$\|\mathbf{y}_{\mathtt{translated},k} - \mathbf{y}_{\mathtt{model}}\|^2_{\mathbf{w}_{\mathtt{freq}}}, \tag{3.21}$$

where $\mathbf{w}_{\mathtt{freq}} \in \mathbb{R}^{2M_{\mathtt{bw}}+1}$ is a weighting function in the frequency domain. The minimisation of (3.21) is a standard least-squares problem where the solution is obtained by solving the normal equations resulting from this expression. The optimal coefficients $\hat{c}_l$ thus obtained are used to generate

$$\hat{\mathbf{y}}_{\mathtt{model}} = \sum_{l=1}^{K} \hat{c}_l \mathbf{x}_l.$$

The fraction

$$\tau_k = \frac{\|\mathbf{y} - \hat{\mathbf{y}}_{\mathtt{model}}\|^2_{\mathbf{w}_{\mathtt{freq}}}}{\|\mathbf{y}\|^2_{\mathbf{w}_{\mathtt{freq}}}} \tag{3.22}$$

is an indication of how well the spectral peak can be modelled by a Kth order amplitude polynomial $\hat{\mathbf{y}}_{\mathtt{model}}$. If $\tau_k < \tau_{\mathtt{min}}$, then component k is identified as being a "true" spectral peak. Suitable values for K, $M_{\mathtt{bw}}$, and $\tau_{\mathtt{min}}$, as well as the spectral window $\mathbf{w}_{\mathtt{freq}}$ will be given in the design specifications in Section 3.5.

The SPE module

The initial frequency estimates $\{\hat{\theta}_{k,2}\}$ and distortion $\{D_k\}$ are provided by the IFE module. The purpose of the SPE module is to refine each estimate $\hat{\theta}_{k,2}$ and determine

the frequency chirp $\hat{\theta}_{k,3}$. If the frequency $\hat{\theta}_{k,2}$ of sinusoid k in the current frame corresponds to the frequency of a sinusoid in the previous frame, the initial estimate of the frequency chirp can be taken equal to that of the corresponding sinusoid in the previous frame. Otherwise, $\hat{\theta}_{k,3}$ can be taken zero.

As we have mentioned before, the Levenberg-Marquardt optimisation method, summarised in Appendix A, is utilised to improve the estimates $\hat{\theta}_{k,2}$ and $\hat{\theta}_{k,3}$.

We denote the model parameters in vector notation by

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_{1,2} & \theta_{1,3} & \dots & \theta_{\hat{N}_\mathrm{c},2} & \theta_{\hat{N}_\mathrm{c},3} \end{bmatrix}^\mathrm{T}. \tag{3.23}$$

At the start of an iteration, the model of the individual sinusoids is

$$\begin{aligned} s_{\mathtt{is}}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1; n) &= \sum_{k=1}^{\hat{N}_\mathrm{c}} \hat{a}_{k,1} e^{j(\hat{\theta}_{k,1} + \hat{\theta}_{k,2} + \hat{\theta}_{k,3} n) n} \\ &= \sum_{k=1}^{\hat{N}_\mathrm{c}} \hat{A}_{k,1} e^{j(\hat{\theta}_{k,2} + \hat{\theta}_{k,3} n) n}. \end{aligned} \tag{3.24}$$

The optimal complex-valued amplitudes

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} \hat{A}_{1,1} & \hat{A}_{2,1} & \dots & \hat{A}_{\hat{N}_\mathrm{c},1} \end{bmatrix}^\mathrm{T},$$

with $\hat{A}_{k,1} = \hat{a}_{k,1} \exp\{j\hat{\theta}_{k,1}\}$, are found by solving the standard least-squares problem

$$\begin{aligned} &\min_{\mathbf{A}_1} \|\mathbf{s} - \mathbf{s}_{\mathtt{is}}(\hat{\boldsymbol{\theta}}; \mathbf{A}_1)\|_\mathbf{w}^2 \\ \Longleftrightarrow &G\hat{\mathbf{A}}_1 = \mathbf{P}_{\mathbf{A}_1}, \end{aligned} \tag{3.25}$$

where $G \in \mathbb{C}^{\hat{N}_\mathrm{c} \times \hat{N}_\mathrm{c}}$ is the Gram matrix. The linearised version of $\mathbf{s}_{\mathtt{is}}$ around the model-parameter estimates is

$$\begin{aligned} s_{\mathtt{is,lin}}(\boldsymbol{\theta}; \hat{\mathbf{A}}_1) &= s_{\mathtt{is}}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1) + J^\mathrm{T}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1) \cdot \boldsymbol{\Delta}_{\boldsymbol{\theta}} \\ &= s_{\mathtt{is}}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1) + \sum_{k=1}^{\hat{N}_\mathrm{c}} (\Delta_{\theta_{k,2},\mathrm{n}} \mathbf{p}_{\theta_{k,2},\mathrm{n}} + \Delta_{\theta_{k,3},\mathrm{n}} \mathbf{p}_{\theta_{k,3},\mathrm{n}}), \end{aligned} \tag{3.26}$$

where the patterns

$$\begin{aligned} p_{\theta_{k,2}}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1; n) &= j\hat{A}_{k,1} n e^{j(\hat{\theta}_{k,2} + \hat{\theta}_{k,3} n) n} \\ p_{\theta_{k,3}}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1; n) &= j\hat{A}_{k,1} n^2 e^{j(\hat{\theta}_{k,2} + \hat{\theta}_{k,3} n) n} \end{aligned} \tag{3.27}$$

are normalised

$$\mathbf{p}_{\theta_{k,2},\mathrm{n}} = \frac{\mathbf{p}_{\theta_{k,2}}}{\|\mathbf{p}_{\theta_{k,2}}\|_\mathbf{w}} \qquad \mathbf{p}_{\theta_{k,3},\mathrm{n}} = \frac{\mathbf{p}_{\theta_{k,3}}}{\|\mathbf{p}_{\theta_{k,3}}\|_\mathbf{w}}. \tag{3.28}$$

The Jacobian is

$$J(\boldsymbol{\theta}_{\mathrm{n}}) = \begin{bmatrix} \mathbf{p}_{\theta_{1,2},\mathrm{n}} & \mathbf{p}_{\theta_{1,3},\mathrm{n}} & \cdots & \mathbf{p}_{\theta_{\hat{N}_{\mathrm{c}},2},\mathrm{n}} & \mathbf{p}_{\theta_{\hat{N}_{\mathrm{c}},3},\mathrm{n}} \end{bmatrix},$$

and the vector containing the estimation errors is

$$\boldsymbol{\Delta}_{\boldsymbol{\theta}_{\mathrm{n}}} = \begin{bmatrix} \Delta_{\theta_{1,2},\mathrm{n}} & \Delta_{\theta_{1,3},\mathrm{n}} & \cdots & \Delta_{\theta_{\hat{N}_{\mathrm{c}},2},\mathrm{n}} & \Delta_{\theta_{\hat{N}_{\mathrm{c}},3},\mathrm{n}} \end{bmatrix}^{\mathrm{T}}.$$

The optimal $\hat{\Delta}_{\theta_{k,2},\mathrm{n}}$s and $\hat{\Delta}_{\theta_{k,3},\mathrm{n}}$s are found by solving

$$\begin{aligned} &\min_{\{\Delta_{\theta_{k,2},\mathrm{n}}, \Delta_{\theta_{k,3},\mathrm{n}}\}} \|\mathbf{s} - \mathbf{s}_{\mathtt{is,lin}}(\boldsymbol{\theta}; \hat{\mathbf{A}}_1)\|_{\mathbf{w}}^2 \\ \Longleftrightarrow\; & H\hat{\boldsymbol{\Delta}}_{\boldsymbol{\theta}_{\mathrm{n}}} = \mathbf{P}_{\boldsymbol{\Delta}_{\boldsymbol{\theta}}\mathrm{n}}, \end{aligned} \tag{3.29}$$

where $H \in \mathbb{C}^{2\hat{N}_{\mathrm{c}} \times 2\hat{N}_{\mathrm{c}}}$ is the system matrix. Regularisation of the system matrix is applied as required by the Levenberg-Marquardt method, please refer to Appendix A. The estimation errors are obtained in the following way

$$\hat{\Delta}_{\theta_{k,2}} = \frac{\hat{\Delta}_{\theta_{k,2},\mathrm{n}}}{\|\mathbf{p}_{\theta_{k,2}}\|_{\mathbf{w}}} \qquad \hat{\Delta}_{\theta_{k,3}} = \frac{\hat{\Delta}_{\theta_{k,3},\mathrm{n}}}{\|\mathbf{p}_{\theta_{k,3}}\|_{\mathbf{w}}},$$

after which the model-parameter estimates are improved

$$\hat{\theta}_{k,2} := \hat{\theta}_{k,2} + \hat{\Delta}_{\theta_{k,2}} \qquad \hat{\theta}_{k,3} := \hat{\theta}_{k,3} + \hat{\Delta}_{\theta_{k,3}},$$

and the iterative process is repeated.

If, for a specific sinusoid k, the following condition is satisfied

$$\frac{|\hat{\theta}_{k,3}|}{\hat{\theta}_{k,2}} 100 N_{\mathrm{s}} \geq \theta_{3,\mathtt{max\%}}, \tag{3.30}$$

this sinusoid is removed and the number of sinusoids is decreased $\hat{N}_{\mathrm{c}} := \hat{N}_{\mathrm{c}} - 1$. We consider this occurrence to be an indication that the phase of the underlying partial can not be modelled by a quadratic polynomial.

Two factors were found that improve the conditioning of the system matrix in each iteration [105]:

1. Using the normalised vectors $\mathbf{p}_{\theta_{k,2},\mathrm{n}}$ and $\mathbf{p}_{\theta_{k,3},\mathrm{n}}$ instead of their non-normalised counterparts $\mathbf{p}_{\theta_{k,2}}$ and $\mathbf{p}_{\theta_{k,3}}$ significantly lowers the condition number of the system matrix H.

2. By ensuring that the partials are well-spaced in frequency: $\min_{k,l} |\hat{\theta}_{k,2} - \hat{\theta}_{l,2}| \geq 2B_{\mathrm{s}}$, where B_{s} denotes the DFT bin size, results in a low condition number of the system matrix.

One interesting phenomenon observed when applying this algorithm to real-world signals is that sinusoidal frequencies can converge to the same value. This usually happens when the underlying non-stationary partial exhibits more than one spectral peak. Given this observation, two sinusoids k and l are merged when $|\hat{\theta}_{k,2} - \hat{\theta}_{l,2}| < \delta_{\texttt{min}}$, and the number of individual sinusoids is decreased $\hat{N}_{\texttt{c}} := \hat{N}_{\texttt{c}} - 1$. More details concerning the SPE module are given in the design specifications in Section 3.5.

The DPE module

Pitch detection has received widespread attention in literature, in particular in the field of speech coding. Our aim here is to design a simple pitch detector which is able to provide an initial estimate of the fundamental frequency. The optimisation carried out in the HPE module will refine the estimate of the fundamental frequency and determine the inharmonicity and chirp parameters. The DPE module strives to achieve the following before the fundamental frequency is estimated:

1. limiting temporal non-stationarity,
2. removing formant fundamental-frequency interaction, and
3. limiting inharmonicity of harmonic partials.

Once these effects in the audio signal are reduced, the fundamental frequency is estimated by utilising the auto-correlation function, which is well-understood and has been applied in numerous pitch detectors [78]. A convincing pitch-related peak in the auto-correlation function is considered sufficient evidence of the presence of a harmonic complex. We note that the presence of additional signal components may make a harmonic complex difficult to detect using this approach. Therefore, a harmonic complex will only be detected if it is dominant.

The following effects will degrade the pitch-related peak in the auto-correlation function, and are therefore taken into account in the design of the pitch detector:

Temporal non-stationarity of $s[n]$ in the form of a frequency chirp or an amplitude sweep. We weaken this effect by carrying out pitch detection on multiple time scales (with related frequency bands) where the number of periods of the fundamental frequency in each time scale is bounded. Therefore, a time scale used for the detection of low pitches will have a longer duration than a time scale used for higher pitches.

Interaction between a formant and the fundamental-frequency is a problem that has received much attention in the field of speech coding, see e.g. [80]. Formants are resonances of the vocal tract and can amplify single partials while attenuating others. In the worst case, the first formant coincides with the second harmonic partial (with frequency $2\theta_{\texttt{f}}$). Applying LPC to the audio signal

$s[n]$ results in a spectrally flat prediction residual $r[n]$ if the prediction coefficients are determined by minimising $\|r[n]\|^2$ [106]. In the prediction residual, formants and the fundamental frequency no longer interfere. Furthermore, spectral smoothing of the prediction coefficients can be applied to avoid sharp resonances in the transfer function of the LPC synthesis filter [107].

Inharmonicity of harmonic partials. The effect of inharmonicity on the harmonic frequencies is most prominent in higher harmonics. Therefore, the prediction residual is low-pass filtered (LPF) to limit inharmonicity. For a given time scale, the low-pass filter cut-off frequency is chosen such that (at least) a few harmonic partials remain in the low-pass filtered residual. The reason for this is that not all harmonics are equally well-defined. In some cases, complete harmonic partials may be missing.

After these operations have been carried out, the normalised auto-correlation function (ACF) is determined, and the highest peak after the first zero crossing is the cue for determining whether the signal is periodic or not. If the peak lies in the lag-range stipulated by the time scale, and the value of the auto-correlation function at this peak is above a certain threshold, the DPE module indicates that a pitch was found and gives as output the estimate $\hat{\theta}_{\mathtt{f}}$. Figure 3.4 illustrates the processing steps taken in the pitch detector.

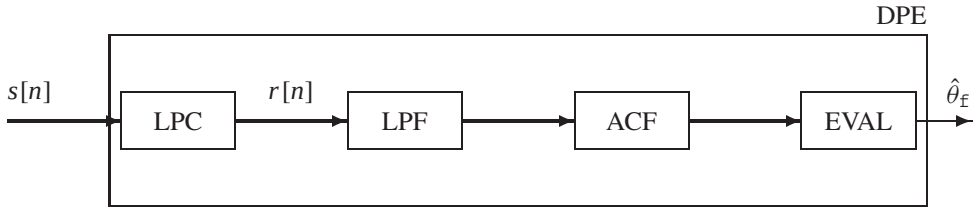


Figure 3.4: *Block diagram of the Detector and Pitch Estimator (DPE) module. The audio signal $s[n]$ is passed through a whitening filter (LPC) first, resulting in a spectrally flat residual $r[n]$. The residual is low-pass filtered (LPF) and the auto-correlation function (ACF) is determined. The highest peak in the auto-correlation function after the first zero crossing is evaluated to determine whether the signal is periodic. If periodicity is detected, the estimate of the fundamental frequency is given as output.*

The HPE module

The initial estimate of the fundamental frequency is provided by the DPE module. The purpose of the HPE module is to refine the estimate of the fundamental fre-

quency, and to estimate the frequency chirp θ_c, inharmonicity B, and the number of harmonics N_h. Initial estimates of θ_c and B can be taken zero if no additional knowledge, e.g. from the previous frame, is at hand.

A special consideration is the influence of inharmonicity on the estimation of N_h. If an accurate estimates, $\hat{B}$ and $\hat{\theta}_\mathrm{f}$, of the inharmonicity and fundamental frequency are at hand, N_h can be estimated by matching the predicted frequencies $\xi_1(\hat{\theta}_\mathrm{f}, \hat{B}), \xi_2(\hat{\theta}_\mathrm{f}, \hat{B}), \ldots, \xi_{N_{\mathrm{h,max}}}(\hat{\theta}_\mathrm{f}, \hat{B})$ to the set of frequencies obtained by identifying spectral peaks. However, if no a priori knowledge about the inharmonicity coefficient is at hand, a search over B and θ_f has to be carried out to estimate the parameters N_h, B, and θ_f. This approach was proposed in [84, Annex A] to estimate the inharmonicity and improve the estimate of the fundamental frequency, given a fixed estimate of the number of harmonics.

In contrast, we combine the Levenberg-Marquardt optimisation with a simple strategy to conduct a directed search to estimate the parameters of the harmonic complex. Given the initial estimates of inharmonicity and the fundamental frequency, N_h is estimated by matching the predicted frequency parameters, derived from the initial estimates, to the frequency parameters obtained by identifying spectral peaks. Given the estimate of N_h thus obtained, the optimisation process is then executed until convergence is achieved, after which the improved estimates of θ_f and B are used to update the estimate of N_h in the same manner as described above. This process is repeated until the harmonic complex can not be extended further.

We denote the model parameters in vector form by $\boldsymbol{\theta} = \begin{bmatrix}\theta_\mathrm{f} & \theta_\mathrm{c} & B\end{bmatrix}^\mathrm{T}$. At the start of an iteration, the model of the harmonic complex is

$$\begin{aligned} s_\mathrm{hc}(\hat{\boldsymbol{\theta}}; \hat{\mathbf{A}}_1; n) &= \sum_{k=1}^{\hat{N}_\mathrm{h}} \hat{a}_{k,1} e^{j(\hat{\theta}_{k,1} + \xi_k(\hat{\theta}_\mathrm{f} + \hat{\theta}_\mathrm{c} n, \hat{B})n)} \\ &= \sum_{k=1}^{\hat{N}_\mathrm{h}} \hat{A}_{k,1} e^{j\xi_k(\hat{\theta}_\mathrm{f} + \hat{\theta}_\mathrm{c} n, \hat{B})n}. \end{aligned} \tag{3.31}$$

The optimal complex-valued amplitudes, contained in

$$\hat{\mathbf{A}}_1 = \begin{bmatrix}\hat{A}_{1,1} & \hat{A}_{2,1} & \ldots & \hat{A}_{\hat{N}_\mathrm{h},1}\end{bmatrix}^\mathrm{T}$$

with $\hat{A}_{k,1} = \hat{a}_{k,1} \exp\{j\hat{\theta}_{k,1}\}$, are found by solving the standard least-squares problem

$$\begin{aligned} &\min_{\mathbf{A}_1} \|\mathbf{s} - \mathbf{s}_\mathrm{hc}(\hat{\boldsymbol{\theta}}; \mathbf{A}_1)\|_\mathbf{w}^2 \\ \Longleftrightarrow &G\hat{\mathbf{A}}_1 = \mathbf{P}_{\mathbf{A}_1}, \end{aligned} \tag{3.32}$$

where $G \in \mathbb{C}^{\hat{N}_\mathrm{h} \times \hat{N}_\mathrm{h}}$ is the Gram matrix. The linearised version of $\mathbf{s}_\mathrm{hc}$ around the

model-parameter estimates is

$$\begin{aligned}\mathbf{s}_{\mathrm{hc,\,lin}}(\boldsymbol{\theta};\hat{\mathbf{A}}_1) &= s_{\mathrm{hc}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1) + J^{\mathrm{T}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1)\cdot\boldsymbol{\Delta}_{\boldsymbol{\theta}} \\ &= \mathbf{s}_{\mathrm{hc}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1) + \left(\frac{\partial \mathbf{s}_{\mathrm{hc}}}{\partial \theta_{\mathrm{f}}}\right)(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1)\Delta_{\theta_{\mathrm{f}}} + \\ &\quad \left(\frac{\partial \mathbf{s}_{\mathrm{hc}}}{\partial \theta_{\mathrm{c}}}\right)(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1)\Delta_{\theta_{\mathrm{c}}} + \left(\frac{\partial \mathbf{s}_{\mathrm{hc}}}{\partial B}\right)(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1)\Delta_B \\ &= \mathbf{s}_{\mathrm{hc}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1) + \Delta_{\theta_{\mathrm{f}},\mathrm{n}}\mathbf{p}_{\theta_{\mathrm{f}},\mathrm{n}} + \Delta_{\theta_{\mathrm{c}},\mathrm{n}}\mathbf{p}_{\theta_{\mathrm{c}},\mathrm{n}} + \Delta_{B,\mathrm{n}}\mathbf{p}_{B,\mathrm{n}},\end{aligned} \tag{3.33}$$

where

$$\begin{aligned}p_{\theta_{\mathrm{f}}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1;n) &= \sum_{k=1}^{\hat{N}_{\mathrm{h}}} j\,\hat{A}_{k,1}\xi_k(1,\hat{B})ne^{j\xi_k(\hat{\theta}_{\mathrm{f}}+\hat{\theta}_{\mathrm{c}}n,\hat{B})n}, \\ p_{\theta_{\mathrm{c}}}(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1;n) &= \sum_{k=1}^{\hat{N}_{\mathrm{h}}} j\,\hat{A}_{k,1}\xi_k(1,\hat{B})n^2e^{j\xi_k(\hat{\theta}_{\mathrm{f}}+\hat{\theta}_{\mathrm{c}}n,\hat{B})n}, \\ p_B(\hat{\boldsymbol{\theta}};\hat{\mathbf{A}}_1;n) &= \sum_{k=1}^{\hat{N}_{\mathrm{h}}} j\,\hat{A}_{k,1}\frac{k^2}{2(1+\hat{B}k^2)}\xi_k(\hat{\theta}_{\mathrm{f}}+\hat{\theta}_{\mathrm{c}}n,\hat{B})ne^{j\xi_k(\hat{\theta}_{\mathrm{f}}+\hat{\theta}_{\mathrm{c}}n,\hat{B})n},\end{aligned} \tag{3.34}$$

and where the normalised vectors are

$$\mathbf{p}_{\theta_{\mathrm{f}},\mathrm{n}} = \frac{\mathbf{p}_{\theta_{\mathrm{f}}}}{\|\mathbf{p}_{\theta_{\mathrm{f}}}\|_{\mathbf{w}}} \qquad \mathbf{p}_{\theta_{\mathrm{c}},\mathrm{n}} = \frac{\mathbf{p}_{\theta_{\mathrm{c}}}}{\|\mathbf{p}_{\theta_{\mathrm{c}}}\|_{\mathbf{w}}} \qquad \mathbf{p}_{B,\mathrm{n}} = \frac{\mathbf{p}_B}{\|\mathbf{p}_B\|_{\mathbf{w}}}.$$

The Jacobian is

$$J(\boldsymbol{\theta}_{\mathrm{n}}) = \begin{bmatrix}\mathbf{p}_{\theta_{\mathrm{f}},\mathrm{n}} & \mathbf{p}_{\theta_{\mathrm{c}},\mathrm{n}} & \mathbf{p}_{B,\mathrm{n}}\end{bmatrix},$$

and the vector containing the normalised estimation errors is

$$\boldsymbol{\Delta}_{\boldsymbol{\theta}_{\mathrm{n}}} = \begin{bmatrix}\Delta_{\theta_{\mathrm{f}},\mathrm{n}} & \Delta_{\theta_{\mathrm{c}},\mathrm{n}} & \Delta_{B,\mathrm{n}}\end{bmatrix}^{\mathrm{T}}.$$

The optimal $\hat{\Delta}_{\theta_{\mathrm{f}},\mathrm{n}}$, $\hat{\Delta}_{\theta_{\mathrm{c}},\mathrm{n}}$, and $\hat{\Delta}_{B,\mathrm{n}}$ are found by solving

$$\begin{aligned}&\min_{\{\Delta_{\theta_{\mathrm{f}},\mathrm{n}},\Delta_{\theta_{\mathrm{c}},\mathrm{n}},\Delta_{B,\mathrm{n}}\}} \|\mathbf{s}-\mathbf{s}_{\mathrm{hc,\,lin}}(\boldsymbol{\theta};\hat{\mathbf{A}}_1)\|_{\mathbf{w}}^2 \\ \Longleftrightarrow\; &H\hat{\boldsymbol{\Delta}}_{\boldsymbol{\theta}_{\mathrm{n}}} = \mathbf{P}_{\boldsymbol{\Delta}\boldsymbol{\theta}\mathrm{n}},\end{aligned} \tag{3.35}$$

where $H \in \mathbb{C}^{3\times 3}$ is the system matrix. Regularisation of the system matrix is applied as required by the Levenberg-Marquardt method, please refer to Appendix A. We note that normalisation of the regression vectors $\mathbf{p}_{\theta_{\mathrm{f}}}$, $\mathbf{p}_{\theta_{\mathrm{c}}}$, and $\mathbf{p}_B$ results in a system matrix with a lower condition number than when these vectors are not normalised. The estimation errors are obtained as follows

$$\hat{\Delta}_{\theta_{\mathrm{f}}} = \frac{\hat{\Delta}_{\theta_{\mathrm{f}},\mathrm{n}}}{\|\mathbf{p}_{\theta_{\mathrm{f}}}\|_{\mathbf{w}}} \qquad \hat{\Delta}_{\theta_{\mathrm{c}}} = \frac{\hat{\Delta}_{\theta_{\mathrm{c}},\mathrm{n}}}{\|\mathbf{p}_{\theta_{\mathrm{c}}}\|_{\mathbf{w}}} \qquad \hat{\Delta}_B = \frac{\hat{\Delta}_{B,\mathrm{n}}}{\|\mathbf{p}_B\|_{\mathbf{w}}},$$

and the estimates are updated

$$\begin{aligned}\hat{\theta}_{\mathtt{f}} &:= \hat{\theta}_{\mathtt{f}} + \hat{\Delta}_{\theta_{\mathtt{f}}} \\ \hat{B} &:= \hat{B} + \hat{\Delta}_{B} \\ \hat{\theta}_{\mathtt{c}} &:= \hat{\theta}_{\mathtt{c}} + \hat{\Delta}_{\theta_{\mathtt{c}}}.\end{aligned} \tag{3.36}$$

If necessary, $\hat{\theta}_{\mathtt{c}}$ and $\hat{B}$ are altered to ensure that

$$\begin{aligned}\frac{|\hat{\theta}_{\mathtt{c}}|}{\hat{\theta}_{\mathtt{f}}} 100 N_{\mathtt{s}} &\leq \theta_{\mathtt{c,\ max\%}} \\ B_{\mathtt{min}} \leq \hat{B} &\leq B_{\mathtt{max}}.\end{aligned}$$

After the iterative process converges, the improved estimates $\hat{\theta}_{\mathtt{f}}$ and $\hat{B}$ are used to update the estimate of $\hat{N}_{\mathtt{h}}$. This process is described in detail in the design specifications, see Section 3.5.

The CAP module

The purpose of the CAP module is to determine the polynomial amplitude and constant phase parameters of all harmonics and individual sinusoids. To describe how the amplitude parameters are obtained, we express $s_{\mathtt{sinusoid}}[n]$ as

$$\begin{aligned}s_{\mathtt{sinusoid}}(\hat{\mathbf{x}}; \mathbf{A}; n) &= \sum_{k=1}^{N_{\mathtt{h}}+N_{\mathtt{c}}} (a_{k,1} + a_{k,2} n) \cos(\theta_{k,1} + f(\hat{x}_{k,2}, \hat{x}_{k,3} n) n) \\ &= \sum_{k=1}^{N_{\mathtt{h}}+N_{\mathtt{c}}} \frac{a_{k,1} + a_{k,2} n}{2} \left(e^{j(\theta_{k,1} + f(\hat{x}_{k,2}, \hat{x}_{k,3} n) n)} + e^{-j(\theta_{k,1} + f(\hat{x}_{k,2}, \hat{x}_{k,3} n) n)} \right)\end{aligned} \tag{3.37}$$

where the non-constant polynomial-phase coefficients are contained in

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_{1,2} & \hat{x}_{1,3} & \dots & \hat{x}_{N_{\mathtt{h}}+N_{\mathtt{c}},2} & \hat{x}_{N_{\mathtt{h}}+N_{\mathtt{c}},3} \end{bmatrix}^{\mathrm{T}} \tag{3.38}$$

and where

$$f(\hat{x}_{k,2}, \hat{x}_{k,3} n) = \hat{x}_{k,2} + \hat{x}_{k,3} n \tag{3.39}$$

with

$$\hat{x}_{k,2} = \begin{cases} \xi_k(\hat{\theta}_{\mathtt{f}}, \hat{B}) & \text{if } 1 \leq k \leq N_{\mathtt{h}} \\ \hat{\theta}_{k,2} & \text{if } N_{\mathtt{h}} < k \leq N_{\mathtt{h}} + N_{\mathtt{c}} \end{cases} \tag{3.40}$$

$$\hat{x}_{k,3} = \begin{cases} \xi_k(\hat{\theta}_{\mathtt{c}}, \hat{B}) & \text{if } 1 \leq k \leq N_{\mathtt{h}} \\ \hat{\theta}_{k,3} & \text{if } N_{\mathtt{h}} < k \leq N_{\mathtt{h}} + N_{\mathtt{c}}. \end{cases} \tag{3.41}$$

The optimal amplitude and constant phase parameters are found by solving the linear least-squares problem

$$\min_{\mathbf{A}} \|\mathbf{s} - \mathbf{s}_{\texttt{sinusoid}}(\hat{\mathbf{x}}; \mathbf{A})\|_{\mathbf{W}}^2, \tag{3.42}$$

where

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \theta_{1,1} & \dots & a_{N_\text{h}+N_\text{c},1} & a_{N_\text{h}+N_\text{c},2} & \theta_{N_\text{h}+N_\text{c},1} \end{bmatrix}^\text{T},$$

and where the optimal $\hat{\mathbf{A}}$ contains the desired parameters. This approach was proposed by George and Smith [51]. The amplitude sweep $\hat{a}_{k,2}$ is restricted by

$$\hat{a}_{k,1} + \hat{a}_{k,2} n > 0 \tag{3.43}$$

to ensure that the instantaneous amplitude is always positive for the duration of the frame. Finally, the real-valued approximation of the sinusoidal component is given by

$$\hat{s}_{\texttt{sinusoid}}[n] = \sum_{k=1}^{N_\text{h}+N_\text{c}} (\hat{a}_{k,1} + \hat{a}_{k,2} n) \cos(\hat{\theta}_{k,1} + f(\hat{x}_{k,2}, \hat{x}_{k,3} n) n). \tag{3.44}$$

3.3.4 Reducing the computational complexity

In the analysis of individual sinusoids (the SPE module), determination of the amplitude parameters in (3.25) can be an obstacle when the number of sinusoids $\hat{N}_\text{c}$ is large. The largest effort, however, is required by regularising and solving the normal equations in (3.29) when $\hat{N}_\text{c}$ is large. The largest portion of the computational effort required in the analysis of the harmonic complex (in the HPE module) is the determination of the amplitude parameters in (3.32). The system matrix in (3.35) is a three-by-three matrix, and solving the corresponding normal equations requires little effort.

We reduce the computational effort required by the HPE and SPE modules in the following way. Firstly, we observe that the off-diagonal elements of the Gram matrices in (3.25) and (3.32) are small in comparison to the diagonal elements when the number of samples N_s in the frame is sufficiently large. The first-order approximation of the inverse of the Gram matrix is utilised to obtain an estimate of the optimal amplitude parameters:

$$\begin{aligned} G\hat{\mathbf{A}}_1 &= \mathbf{P}_{\mathbf{A}_1} \\ \implies \Lambda(I + \Lambda^{-1}\delta G)\hat{\mathbf{A}}_1 &= \mathbf{P}_{\mathbf{A}_1} \\ \implies \hat{\mathbf{A}}_1 &\approx \Lambda^{-1}\mathbf{P}_{\mathbf{A}_1} - \Lambda^{-1}\delta G \Lambda^{-1}\mathbf{P}_{\mathbf{A}_1}, \end{aligned}$$

where G is written as $G = \Lambda + \delta G$, and where Λ contains the leading diagonal of G and zeros elsewhere, and δG contains the off-diagonal elements of G and zeros on

the leading diagonal. Here we have used the power-series expansion of the inverse

$$(I + \Lambda^{-1}\delta G)^{-1} = \sum_{k=0}^{\infty}(-1)^k(\Lambda^{-1}\delta G)^k.$$

Secondly, we observe that, similar to the Gram matrices, the off-diagonal elements of the system matrix in (3.29) are small in comparison to its diagonal elements when the number of samples in the frame, $N_{\mathtt{s}}$, is large enough to ensure that spectral peaks are well-separated in the frequency domain. Given this observation, we apply the optimisation to one sinusoid at a time. The advantage of this approach is that a unique regularisation, which we denote by λ_k, is applied to the system matrix, denoted by H_k, corresponding to each sinusoid k. Given $\hat{A}_{k,1}$, the system matrix $H_k \in \mathbb{C}^{2\times 2}$, following from

$$\min_{\{\Delta_{\theta_{k,2},\mathtt{n}}, \Delta_{\theta_{k,3},\mathtt{n}}\}} \|\mathbf{s} - \mathbf{s}_{\mathtt{is,lin}}([\theta_{k,2} \quad \theta_{k,3}]^{\mathrm{T}}; \hat{A}_{k,1})\|^2_{\mathbf{w}},$$

is the identity matrix

$$H_k = J([\hat{\theta}_{k,2} \quad \hat{\theta}_{k,3}]^{\mathrm{T}})^{\mathrm{H}} \cdot \mathrm{diag}(\mathbf{w}) \cdot J([\hat{\theta}_{k,2} \quad \hat{\theta}_{k,3}]^{\mathrm{T}}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where the Jacobian is

$$J([\hat{\theta}_{k,2} \quad \hat{\theta}_{k,3}]^{\mathrm{T}}) = \left[\mathbf{p}_{\theta_{k,2},\mathtt{n}} \quad \mathbf{p}_{\theta_{k,3},\mathtt{n}}\right],$$

and where

$$\langle \mathbf{p}_{\theta_{k,2},\mathtt{n}}, \mathbf{p}_{\theta_{k,3},\mathtt{n}} \rangle_{\mathbf{w}} = \langle \mathbf{p}_{\theta_{k,3},\mathtt{n}}, \mathbf{p}_{\theta_{k,2},\mathtt{n}} \rangle_{\mathbf{w}} = 0$$

and

$$\|\mathbf{p}_{\theta_{k,2},\mathtt{n}}\|^2_{\mathbf{w}} = \|\mathbf{p}_{\theta_{k,3},\mathtt{n}}\|^2_{\mathbf{w}} = 1.$$

The parameter-estimation errors are then given by

$$\begin{bmatrix} \Delta_{\theta_{k,2},\mathtt{n}} \\ \Delta_{\theta_{k,3},\mathtt{n}} \end{bmatrix} = \frac{1}{1+\lambda_k} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot J([\hat{\theta}_{k,2} \quad \hat{\theta}_{k,3}]^{\mathrm{T}})^{\mathrm{H}} \cdot \mathrm{diag}(\mathbf{w}) \cdot (\mathbf{s} - \mathbf{s}_{\mathtt{is}}([\hat{\theta}_{k,2} \quad \hat{\theta}_{k,3}]^{\mathrm{T}}, \hat{A}_{k,1})).$$

3.3.5 Multi-resolution analysis

Multi-resolution analysis of individual sinusoids and pitch-synchronous analysis of harmonic sounds have a number of advantages, as was argued in Section 2.2.1. The time scales used in multi-resolution analysis are defined beforehand since no a priori knowledge about the audio signal is assumed, while the time scale used in pitch-synchronous analysis depends on the estimate of the fundamental frequency. An

attractive alternative to multi-resolution analysis is variable-length analysis (or adaptive segmentation) [108, 109, 110]. Adaptive segmentation, in combination with the distribution of sinusoids over the segments, has been applied to parametric audio coding in the context of a rate-distortion optimal framework [59]. The main advantage of adaptive segmentation is the possibility to choose a segment size that suits local signal behaviour. In particular, periods of stationarity in the audio signal should be identified, and the frame boundaries should be set accordingly. The main disadvantage of adaptive segmentation is the high computational complexity required to obtain the optimal segment size and distribution of sinusoids over the segments. Observe that, in contrast to multi-resolution analysis, pitch-synchronous analysis does adapt to local signal behaviour. Given this observation and the relative simplicity of multi-resolution analysis in comparison to adaptive segmentation, we choose to utilise multi-resolution and pitch-synchronous analysis.

An important consideration in multi-resolution analysis is the relation between the various time scales and the update-rate of sinusoidal-model parameters. When these two factors are *related*, the model parameters obtained on a particular time scale are coded at a rate related to the time-scale duration; that is, low-frequency sinusoidal parameters are updated less frequent than high-frequency parameters. The main advantage in this approach is that the parameters are not updated at a rate higher than necessary. Main disadvantages are increased complexity of the encoder and bit-stream: parameters obtained from each time scale have to be dealt with separately, which makes the relation among time scales more complicated (e.g. inter-frame linking across time scales). When these two factors are *unrelated*, the parameter update-rate is decoupled from the various time scales utilised in the analysis. The main advantage of this approach is that the parameter update-rate becomes a free parameter. This is a significant advantage, since it was observed that an increase in the parameter update-rate can lead to an improvement in audio quality [12]. Simplicity of the encoder and bit-stream are additional advantages. The main disadvantage is a higher bit-rate resulting from the more frequent update of model parameters. However, the higher bit-rate is partly compensated for by the increased redundancy of model parameters which is exploited by entropy coding. The simplicity of the coder and bit-stream, and the possibility to increase the audio quality are the two reasons for choosing a fixed update-rate of sinusoidal-model parameters.

Multiple time scales are utilised to analyse $s[n]$ to obtain the parameters of $s_{\mathtt{hc}}[n]$ and $s_{\mathtt{is}}[n]$. As mentioned before, the ISA module utilises a number of fixed time-scales, while the HCA module utilises pitch-synchronous analysis. At this point, we are faced with the choice of the order in which the ISA and HCA modules are applied. There are two possibilities: applying them in *cascade* (where a residual is generated after the first module is applied) or in *parallel*. When ISA and HCA are applied in *cascade*, the HCA module has to be applied first, since applying the ISA module first

will result in a residual from which harmonics have been removed. The definition of a residual is another consideration when serial processing is applied: if the pitch-synchronous frame has a shorter duration than the longest individual-sinusoids time scale, a suitable residual for the ISA module can only be obtained by extrapolation. This approach may yield a reasonable result only when the difference in frame duration is minor. Furthermore, errors made in the extraction of the harmonic complex will contaminate the residual, degrading the estimation of the individual-sinusoid parameters. When ISA and HCA are applied in *parallel*, these obstacles are avoided. The complication in this choice is that the HCA and ISA modules must communicate in some way to avoid the duplicate modelling of harmonic partials. This problem of communication is solved when the HCA module is applied to the audio signal $s[n]$ first, after which the ISA module is applied to $s[n]$ also, with the harmonic-model parameters as additional input. The ISA module then models all partials in $s[n]$ except the harmonic partials. Given these considerations, we choose to apply ISA and HCA in parallel, where the HCA module is executed before the ISA module. After the HCA and ISA modules have been applied, the amplitude parameters are obtained by applying the CAP module. The HCA, ISA, and CAP modules are encapsulated in the Sinusoidal Analysis (SA) module, illustrated in Figure 3.5.

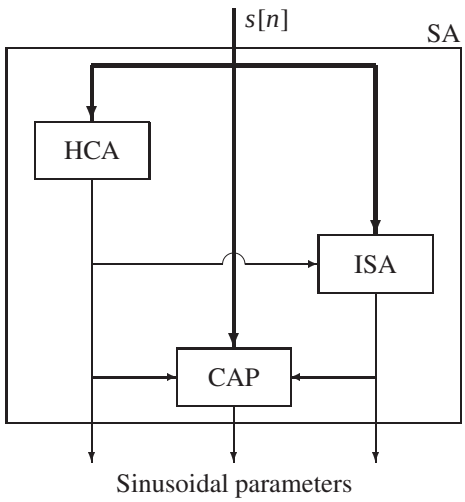


Figure 3.5: *Processing of the audio signal $s[n]$ in parallel. The parameters of the harmonic complex, if present in $s[n]$, are estimated by the Harmonic Complex Analysis (HCA) module. The parameters of individual sinusoids are estimated by the Individual Sinusoidal Analysis (ISA) module, where duplicate modelling of harmonic partials is avoided. The amplitude parameters are determined by the Collective Amplitude Parameters (CAP) module.*

3.4 Noise analysis

After the sinusoidal component $\hat{s}_{\texttt{sinusoid}}[n]$ (3.44) is determined, the spectral and temporal envelopes of the residual

$$s_{\texttt{residual}}[n] = s[n] - \hat{s}_{\texttt{sinusoid}}[n] \tag{3.45}$$

are modelled by a noise coder. A brief overview of existing approaches to noise coding was given in Section 2.2.2. Warped LPC is a particularly attractive technique, for two reasons. The first reason is that it allows a frequency-dependent spectral resolution which can be matched to the spectral resolution of the human auditory system. The second reason is that Warped LPC is computationally efficient, and encoding the prediction and gain parameters is relatively cheap. Laguerre-based Pure Linear Prediction (PLP[1]) is an alternative to Warped LPC, where the structure of the prediction filter in the analysis and synthesis schemes are identical [113]. Furthermore, the prediction residual is spectrally flat when the prediction coefficients are calculated using data input windowing. For these reasons, we utilise Laguerre-based Pure Linear Prediction to model the spectral envelope of $s_{\texttt{residual}}[n]$. The temporal envelope of the prediction residual is modelled by its short-time energy. Therefore, our model of the residual closely resembles the approach taken in PPC [73].

3.4.1 Spectral and temporal model of the residual

The Laguerre filters utilised in Laguerre-based PLP have transfer functions

$$H_k(\lambda; z) \triangleq \frac{\sqrt{1-\lambda^2}}{1-z^{-1}\lambda}\left(\frac{z^{-1}-\lambda}{1-z^{-1}\lambda}\right)^{k-1},$$

where $k = 1, \ldots, O_{\text{p}}$ and the pole $\lambda \in \mathbb{R}$ is restricted by $|\lambda| < 1$ to guarantee stability. A suitable choice for λ yields a warped frequency axis that is a close approximation of the ERB rate scale [114]. The PLP analysis scheme, illustrated in Figure 3.6 (a), is utilised to obtain the spectrally flat prediction residual $r[n]$. The transfer function $F_{\texttt{ana}}(z)$ of the analysis filter is

$$\begin{aligned} F_{\texttt{ana}}(z) &= F_{\texttt{ana}}(\alpha_1, \ldots, \alpha_{O_{\text{p}}}; \lambda; z) \\ &= 1 - z^{-1}\sum_{k=1}^{O_{\text{p}}} \alpha_k H_k(\lambda; z). \end{aligned} \tag{3.46}$$

[1] We note that this acronym is also used by Hermansky et al. for their method called "Perceptually based linear predictive analysis" [111, 112]. However, since their method is not considered in this thesis, we use PLP to indicate Laguerre-based Pure Linear Prediction.

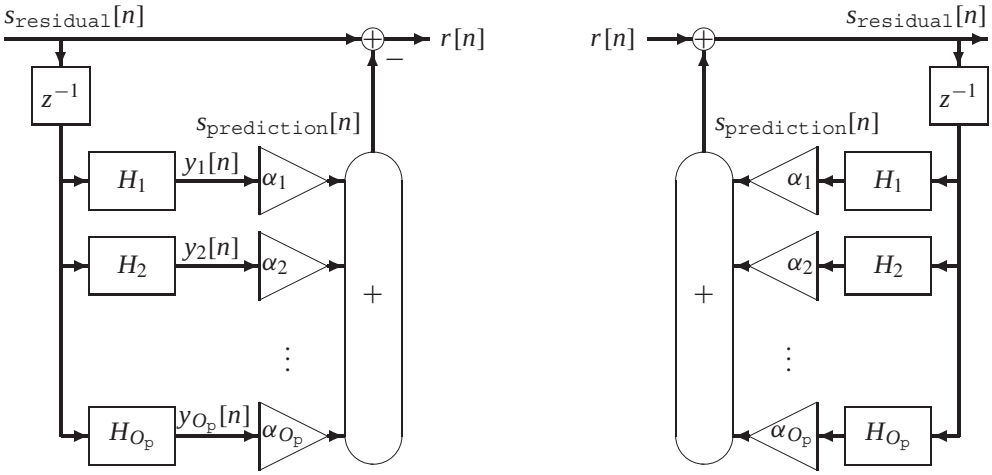


Figure 3.6: *Pure linear prediction (PLP) analysis (a) and synthesis (b) structure. The Laguerre filters H_k are stable and causal. The prediction of the input signal $s_{\texttt{residual}}[n]$ is $s_{\texttt{prediction}}[n]$. The prediction coefficients are α_k, where $k = 1, \ldots, O_p$, and the prediction residual is $r[n]$.*

The decoder, which will be discussed later, utilises the PLP synthesis scheme, illustrated in Figure 3.6 (b). The optimal prediction coefficients are unique and determined by solving the following set of normal equations

$$\begin{aligned}
&\min_{\{\alpha_k\}} \left\| \mathbf{s}_{\texttt{residual}} - \mathbf{s}_{\texttt{prediction}} \right\|_{\mathbf{w}}^2 \\
\Longleftrightarrow \quad & \langle \mathbf{s}_{\texttt{residual}} - \hat{\mathbf{s}}_{\texttt{prediction}}, \mathbf{y}_l \rangle_{\mathbf{w}} = 0 \\
\Longleftrightarrow \quad & \sum_k \langle \mathbf{y}_k, \mathbf{y}_l \rangle_{\mathbf{w}} \, \hat{\alpha}_k = \langle \mathbf{s}_{\texttt{residual}}, \mathbf{y}_l \rangle_{\mathbf{w}} \\
\Longleftrightarrow \quad & G\,\hat{\boldsymbol{\alpha}} = \mathbf{P}, \qquad (3.47)
\end{aligned}$$

where $\hat{\mathbf{s}}_{\texttt{prediction}} = \sum_k \hat{\alpha}_k \mathbf{y}_k$, and where the $\mathbf{y}_k$s are defined in Figure 3.6 (a). It was proven in [115] that the prediction residual

$$r[n] = s_{\texttt{residual}}[n] - \hat{s}_{\texttt{prediction}}[n]$$

is a spectrally flattened version of the input $s_{\texttt{residual}}[n]$. Prediction coefficients are calculated on a per-frame basis. Before the prediction residual is generated, the prediction coefficients are quantised, and the quantised prediction coefficients are used to derive the prediction residual $r[n]$. The advantage of this approach is that the PLP module in the encoder and its counterpart in the decoder will use the same prediction

coefficients. The disadvantage is that the characteristics (i.e., spectral flatness) of the prediction residual obtained in this way will be influenced by the quantisation of the prediction coefficients.

The temporal envelope of the prediction residual is modelled by its short-time energy,

$$E = \|\mathbf{r}\|^2_{\mathbf{w}_{\texttt{short}}}, \tag{3.48}$$

and is determined by the Temporal Envelope Measurement (TEM) module. The rate at which the temporal envelope is measured should be matched to the temporal resolution of the human auditory system. Therefore, several measurements of the short-time energy are made within a frame.

The noise coder will transmit only the prediction coefficients and short-time energy measurements to the decoder. The prediction residual $r[n]$ will not be considered further.

3.5 Design specifications

In this section, design specifications are provided for the sinusoidal coder in Section 3.5.1 and the noise coder in Section 3.5.2. Before the encoder and decoder are considered, the test material, definition of the time axis, parameter update-rate, and windows used are specified.

The set of audio files used in testing this prototype is the set provided in the MPEG call-for-proposals made in 2001 [13], see Table 3.1. The test material is re-

Test item	Description
es01	Vocal (Suzan Vega)
es02	German speech
es03	English speech
si01	Harpsichord
si02	Castanets
si03	Pitch pipe
sm01	Bagpipes
sm02	Glockenspiel
sm03	Plucked strings
sc01	Trumpet solo and orchestra
sc02	Orchestral piece
sc03	Contemporary pop music

Table 3.1: *Test material (in WAVE format) used in testing the prototype. The es∗ excerpts are speech signals, the si∗ excerpts are single instruments with only one note sounding at a time, the sm∗ excerpts are simple sound mixtures, and the sc∗ excerpts are complex sound mixtures [13].*

sampled from 48 kHz to a frequency of $f_s = 44.1$ kHz while maintaining the sample accuracy of 16 bits, and the excerpts are in WAVE format. An audio signal in WAVE format has values lying in the range $[-1, 1]$.

In order to describe the analysis process of the signal in frame i, we first define the time axis. We denote the update-rate of sinusoidal-model parameters by

$$f_{\text{UR}} = \frac{1}{t_{\text{UR}}} \text{ Hz.} \tag{3.49}$$

In frame i, the audio signal is analysed around time instant $(i-1)t_{\text{UR}}$ to estimate the sinusoidal-model parameters; here we assume that the first frame, $i = 1$, is centred around time $t = 0$. Assuming that time instant $t = 0$ corresponds to sample $n = 1$, the sample corresponding to an arbitrary time t is $tf_s + 1$. To ensure that the time instant $(i-1)t_{\text{UR}}$ corresponds to a sample for all i, we require that $t_{\text{UR}} f_s \in \mathbb{N}$. The number of samples corresponding to t_{UR} is $N_{s,\text{UR}} = t_{\text{UR}} f_s$. Frame i spans samples $(i-1.5)N_{s,\text{UR}}+1$ to $(i-0.5)N_{s,\text{UR}}+1$. If $N_{s,\text{UR}}$ is even, then the frame boundary will correspond to a sample, while if $N_{s,\text{UR}}$ is odd, the frame boundary will fall between two samples, see Figure 3.7. Note that frame i contains an odd number of samples

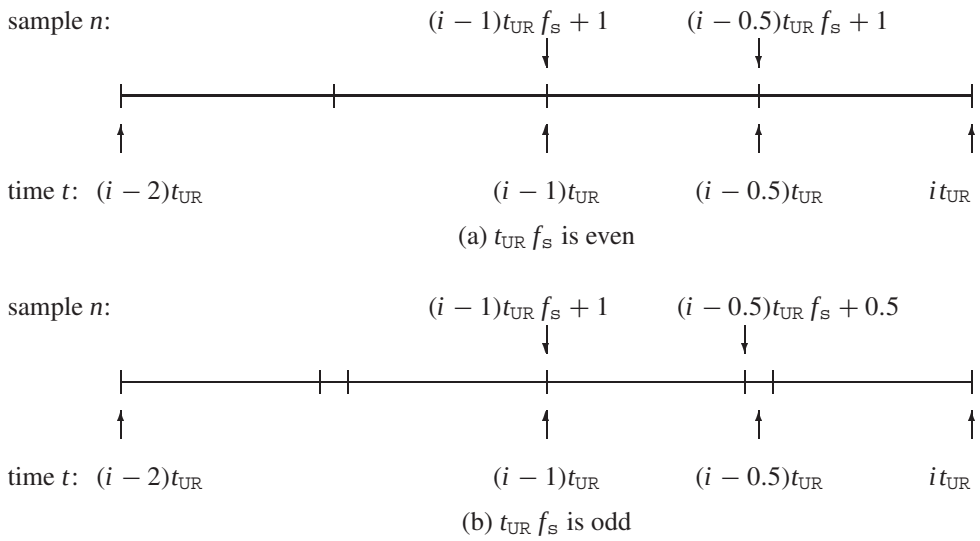


Figure 3.7: *Definition of the time axis. (a) $t_{UR} f_s$ is even. The boundary between frames i and $i+1$, located at $(i-0.5)t_{UR}$ seconds, corresponds to sample $(i-0.5)N_{s,UR}+1$, where $N_{s,UR} = t_{UR} f_s$. Note that this sample belongs to frame $i+1$ also. The frame contains $N_s = N_{s,UR}+1$ samples, an odd number. (b) $t_{UR} f_s$ is odd. The frame boundary falls between two samples in this case. The last sample in frame i is $(i-0.5)N_{s,UR}+0.5$. The frame contains $N_s = t_{UR} f_s = N_{s,UR}$ samples.*

in both cases. However, when $N_{\mathtt{s},\mathtt{UR}}$ is even, the sample at the boundary between frames i and $i+1$, located at the time instant $(i-0.5)t_{\mathtt{UR}}$ seconds and corresponding to sample $(i-0.5)N_{\mathtt{s},\mathtt{UR}}+1$, will belong to both frames. Furthermore, in this case, the frame contains $N_{\mathtt{s}} = N_{\mathtt{s},\mathtt{UR}}+1$ samples, which do not correspond to the duration $t_{\mathtt{UR}}$. This scenario is likely to cause confusion, and is therefore excluded. Hence the restriction that $N_{\mathtt{s},\mathtt{UR}} = t_{\mathtt{UR}} f_{\mathtt{s}}$ be odd.

The parameter update-rate plays an important role in the audio quality obtained by the coder. For parametric audio coders, $t_{\mathtt{UR}}$ varies between 8 ms [12] and 32 ms [75, 25]. The non-stationary character of speech signals, in particular, requires the utilisation of a high update-rate. Informal experiments carried out with updating the model parameters every 10 ms indicated that speech sounded metallic, a common artefact in parametric audio coders. Updating parameters every 8 ms alleviated the metallic artefact to some degree. For this reason, parameters are updated every

$$t_{\mathtt{UR}} = \frac{N_{\mathtt{s},\mathtt{UR}}}{f_{\mathtt{s}}} = \frac{355}{44100} \approx 8 \text{ ms}. \tag{3.50}$$

The synthesis is based on overlap-add (OLA), and an amplitude-complementary window is required as a result. The Hann window $\mathbf{w} \in \mathbb{R}^{2N_{\mathtt{s},\mathtt{UR}}+1}$, with elements

$$w[n] = \begin{cases} \frac{1}{2}\left[1+\cos(\pi \frac{n}{N_{\mathtt{s},\mathtt{UR}}})\right] & \text{if } -N_{\mathtt{s},\mathtt{UR}} \leq n \leq N_{\mathtt{s},\mathtt{UR}} \\ 0 & \text{else,} \end{cases} \tag{3.51}$$

is used in both the analysis and synthesis.

The ERB scale plays an important role in the quantisation of sinusoidal parameters. The ERB scale maps a frequency θ (in radians) to $e(\theta)$ (in ERB) by

$$e(\theta) \triangleq 21.4 \log_{10}\left(4.37\frac{44.1}{2\pi}\theta + 1\right), \tag{3.52}$$

given the sampling frequency $f_{\mathtt{s}} = 44.1$ kHz. The width of a critical band $e_{\mathtt{BW}}(\theta)$ (in radians) at a frequency θ is given by

$$e_{\mathtt{BW}}(\theta) = 24.7\left(4.37\frac{44.1}{2\pi}\theta + 1\right). \tag{3.53}$$

The ERB scale is illustrated in Figure 3.8.

Zero padding a sequence to a length being the nearest power of two is applied in all cases where it would facilitate the use of the FFT algorithm to compute the DFT of a (windowed) frame.

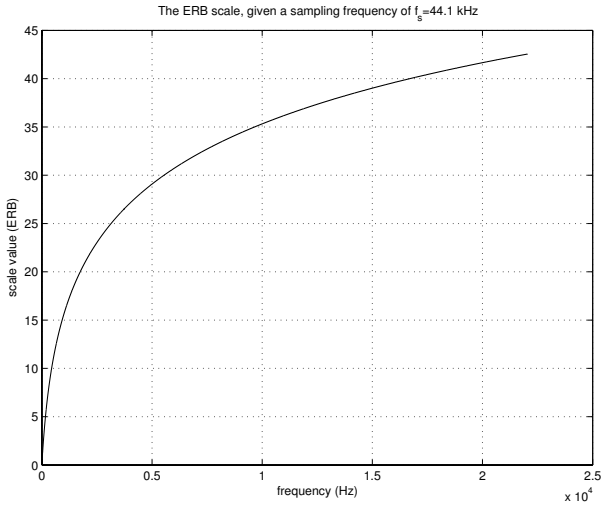


Figure 3.8: *The ERB scale.*

3.5.1 Sinusoidal coder

All the modules comprising the sinusoidal coder (SC) are illustrated in Figure 3.9.

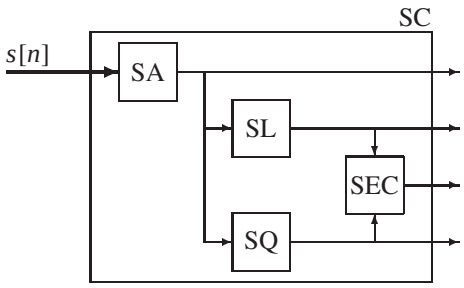


Figure 3.9: *In the Sinusoidal Coder (SC), the audio signal $s[n]$ is analysed by the Sinusoidal Analysis (SA) module to obtain the sinusoidal parameters. Individual sinusoids in adjacent frames are linked by the Sinusoidal Linking (SL) module. Furthermore, the SL module links harmonic complexes in adjacent frames. The sinusoidal parameters are quantised by the Sinusoidal Quantisation (SQ) module. The quantisation indices and linking information is passed to the Sinusoidal Entropy Coding (SEC) module, which applies entropy coding to remove redundancy and determines the cost of coding each sinusoidal-model parameter.*

Module	Sub-module	Parameter	Value
SA	DPE	$\theta_{\mathtt{f,min}} f_\mathtt{s}/(2\pi)$	25 Hz
		$\theta_{\mathtt{f,max}} f_\mathtt{s}/(2\pi)$	1800 Hz
	HPE	$\theta_{\mathtt{c,max\%}}$	10%
		$N_{\mathtt{h,min}}$	2
		$N_{\mathtt{h,max}}$	150
		$B_{\mathtt{min}}$	0
		$B_{\mathtt{max}}$	0.01
	IFE	$\theta_{2,\mathtt{min}} f_\mathtt{s}/(2\pi)$	25 Hz
		$\theta_{2,\mathtt{max}} f_\mathtt{s}/(2\pi)$	16000 Hz
		$D_{\mathtt{min}}$	0.25
	SPE	$\theta_{3,\mathtt{max\%}}$	10%
		$\Delta_{\mathtt{min}}$	$5 \cdot 10^{-4}$
		$N_{\mathtt{c,max}}$	150
	CAP	$a_{1,\mathtt{dB,min}}$	-100 dB
		$a_{1,\mathtt{dB,max}}$	0 dB
		$a_{2,\mathtt{max\%}}$	100%
SL	HCL	$e_{\theta_{\mathtt{f,link}}}$	0.6 ERB
	ISL	$e_{\theta_{2,\mathtt{link}}}$	0.3 ERB
SQ	HCQ	$N_{\theta_\mathtt{f}}$	12 bits
		$N_{\theta_\mathtt{c}}$	5 bits
		N_B	12 bits
	ISQ	N_{θ_2}	9 bits
		N_{θ_3}	5 bits
	CAQ	N_{a_1}	6 bits
		N_{a_2}	4 bits
		N_{θ_1}	5 bits

Table 3.2: *Summary of the parameter choices for the sinusoidal coder.*

The main parameters in the sinusoidal coder and values chosen are summarised in Table 3.2.

The following sections discuss the sinusoidal coder in more detail. We start by describing the modules comprising the Sinusoidal Analysis (SA) module. These are the PEAK, DPE and HPE (concerning the harmonic complex), IFE and SPE (concerning individual sinusoids), and CAP modules. The Sinusoidal Linking (SL) module is then described, followed by an explanation of the Sinusoidal Quantisation (SQ) module. Finally, attention is paid to the entropy coding applied in the Sinusoidal Entropy

Coding (SEC) module.

The PEAK module

The number of amplitude-modulated windows K, the spectral bandwidth $M_{\mathtt{bw}}$, and spectral window $\mathbf{w}_{\mathtt{freq}}$ utilised by the PEAK module need to be specified. Suitable thresholds for peak classification, $\tau_{\mathtt{min}}$, are chosen for each module utilising the PEAK module.

The number of amplitude-modulated windows, K, utilised depends on the degree of temporal non-stationarity of a partial in the frame. In this sense, the value chosen for K should be dependent on the frame duration. Another point-of-view is that the peaks selected by the PEAK module are used as input to the individual-sinusoids parameter-refinement module SPE. Recall that the SPE module utilises the first-order approximation of a quadratic phase-polynomial in n, refer to Equation (3.26) in the description of the SPE module. This first-order approximation yields a quadratic amplitude-polynomial in n, refer to Equation (3.27). The estimates of the frequency and frequency-chirp parameters are improved by utilising this quadratic amplitude-polynomial. A value of $K = 3$ would be a suitable choice to determine whether a peak can be well-modelled by a quadratic amplitude-polynomial. For this reason, a value of $K = 3$ is chosen.

The bandwidth $M_{\mathtt{bw}}$ is determined by the effective bandwidth (width of the mainlobe) of $(\mathrm{DFT}\,\mathbf{p}_1)[m]$, $(\mathrm{DFT}\,\mathbf{p}_2)[m]$, and $(\mathrm{DFT}\,\mathbf{p}_3)[m]$, see Figure 3.10. (The patterns $p_1[n]$, $p_2[n]$, and $p_3[n]$ were defined in Equations (3.17), (3.18), and (3.19), respectively.) From this figure, it is clear that a value of $M_{\mathtt{bw}} = 3$, corresponding to the bin-size of the unpadded sequence, covers the mainlobes of all three windows. When zero-padding is applied, $M_{\mathtt{bw}}$ should be adapted correspondingly:

$$M_{\mathtt{bw,padded}} = M_{\mathtt{bw}} \left\lceil \frac{N_{\mathtt{s}} + N_{\mathtt{zeros}}}{N_{\mathtt{s}}} \right\rceil, \tag{3.54}$$

where $N_{\mathtt{zeros}}$ denotes the number of zeros padded.

We utilise the Hann spectral window $\mathbf{w}_{\mathtt{freq}} \in \mathbb{R}^{2M_{\mathtt{bw,padded}}+1}$, refer to (3.21).

The DPE module

Before the LPC, LPF, ACF and MAX sub-modules inside the DPE module (Figure 3.4) are described, the time scales are chosen.

We start by defining the range of allowable fundamental frequencies. The lower end of this range should reflect the lower bound of the lowest tone that can be produced by most musical instruments. The higher end of this range should be chosen such that utilising a harmonic complex is advantageous from a coding point-of-view.

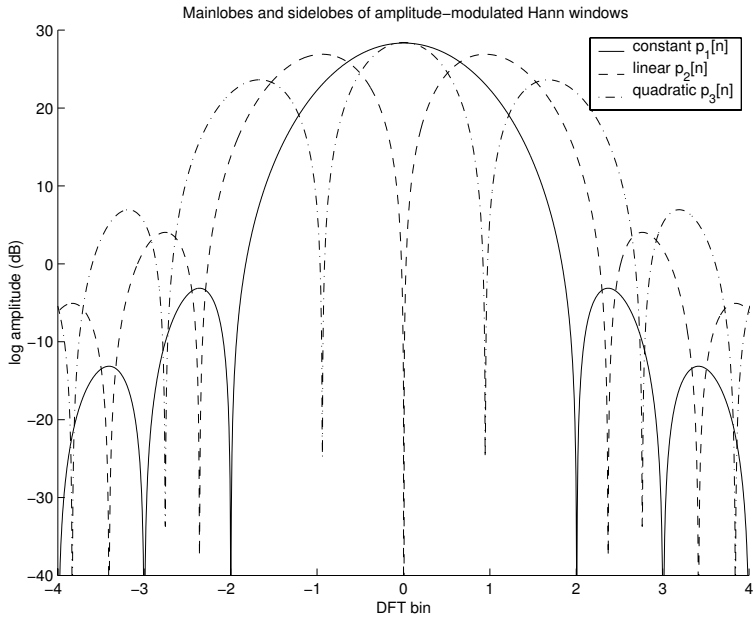


Figure 3.10: *Mainlobes of amplitude-modulated Hann windows.*

We propose the range

$$\left[\theta_{\mathtt{f,min}}\frac{f_{\mathtt{s}}}{2\pi}, \theta_{\mathtt{f,max}}\frac{f_{\mathtt{s}}}{2\pi}\right] = [25, 1800] \text{ Hz}, \tag{3.55}$$

where the higher end of this range ensures that at least ten harmonic components can be present, given a sampling frequency of $f_{\mathtt{s}} = 44.1$ kHz. We comment that no fundamental frequency outside this range was found in the test material. In fact, this range was found to be conservative, as we will show in the results given in Section 3.7.

The time scales are based on the following considerations. In order to obtain a well-defined estimate of the fundamental frequency from the auto-correlation function, experimentation revealed that at least eight periods of the fundamental frequency are required. To limit the effect of temporal non-stationarity, no more than twelve periods of the fundamental should be present in the time scale. This defines the frequency range $[\, f_{\mathtt{low}},\ f_{\mathtt{high}}]$ (in Hz) and frame duration $t_{\mathtt{d}}$ (in s) associated with each time scale, see columns two and three of Table 3.3. In the interest of robustness, it is necessary to allow overlapping frequency bands in order to ensure that a fundamental frequency located (at or near the) border between two frequency bands is not missed. A consequence of this is that a harmonic complex may be detected in two time scales.

Time scale	$t_{\mathtt{d}}$ ms	$[f_{\mathtt{low}}, f_{\mathtt{high}}]$ Hz	$f_{\mathtt{cut}}$ Hz	$r_{\mathtt{ds}}$
1	320	[25, 37.5]	300	49
2	220	[35, 52.5]	420	35
3	160	[50, 75]	600	24
4	120	[70, 105]	840	17
5	80	[95, 142.5]	1140	12
6	60	[135, 202.5]	1620	9
7	40	[195, 292.5]	2340	6
8	30	[280, 420]	3360	4
9	20	[400, 600]	4800	3
10	14	[580, 870]	6960	2
11	10	[850, 1275]	10200	1
12	6	[1200, 1800]	14400	1

Table 3.3: *The window duration t_d is given in ms. The frequency band associated with each time scale is $[f_{low}, f_{high}]$ Hz. The cut-off frequency of the low-pass filter is $f_{cut} = 8 f_{high}$ Hz. The decimation rate is r_{ds}, assuming a sampling frequency of $f_s = 44.1$ kHz.*

The DPE module is applied only if the windowed frame $\mathbf{s_w}$ corresponding to a particular time scale is a non-zero sequence. An estimate of the fundamental frequency is found as follows.

LPC A relatively low LPC prediction order, $O_{\mathrm{p}} = 8$, gives satisfactory results in practice. Spectral smoothing of the optimal prediction coefficients $\hat{\alpha}_k$ is applied in the form $\hat{\alpha}_k := \gamma^k \hat{\alpha}_k$ [116], where $k = 1, \ldots, O_{\mathrm{p}}$ and $\gamma = 0.99$.

LPF Under a given time scale, the low-pass filter has a cut-off frequency $f_{\mathtt{cut}} = 8 f_{\mathtt{high}}$, ensuring the presence of at least eight harmonics and at most twelve harmonics when no inharmonicity is present; that is, when $B = 0$. The eighth-order Butterworth low-pass filter is used. After low-pass filtering, decimation (or down-sampling) is applied to better reflect the bandwidth of the signal and reduce the computational load of computing the auto-correlation function. To avoid aliasing, the down-sampling rate $r_{\mathtt{ds}}$ is chosen as

$$r_{\mathtt{ds}} = \left\lfloor \frac{f_{\mathtt{s}}}{3 f_{\mathtt{cut}}} \right\rfloor < \frac{f_{\mathtt{s}}}{2 f_{\mathtt{cut}}},$$

where the last term in this expression denotes the critical down-sampling rate. The magnitude of the frequency response of the filter is more than -25 dB down at $1.5 f_{\mathtt{cut}}$, justifying the choice of $r_{\mathtt{ds}}$. The resulting sampling frequency is

$$f_{\mathtt{ds}} = \frac{f_{\mathtt{s}}}{r_{\mathtt{ds}}} > 2 f_{\mathtt{cut}}.$$

A summary of $f_{\texttt{high}}$ and $r_{\texttt{ds}}$ are given in columns four and five of Table 3.3.

ACF The normalised auto-correlation function $r_{\mathbf{s_w}}[\tilde{n}]$ is calculated as

$$r_{\mathbf{s_w}}[\tilde{n}] = \frac{\sum_n r_{\texttt{ds}}[n+\tilde{n}] w_{\texttt{ds}}[n+\tilde{n}] r_{\texttt{ds}}[n]}{\sum_n |r_{\texttt{ds}}[n]|^2 w_{\texttt{ds}}[n]},$$

where $r_{\texttt{ds}}[n]$ is the down-sampled version of $r[n]$, after LPC, and $w_{\texttt{ds}}[n]$ is the corresponding Hann window.

MAX We denote the position of the largest peak in $r_{\mathbf{s_w}}[\tilde{n}]$, after the first zero-crossing, by $\tilde{n}_{\texttt{max}}$. If

$$\frac{f_{\texttt{ds}}}{\tilde{n}_{\texttt{max}}} \in [\, f_{\texttt{low}},\ f_{\texttt{high}}]$$

and $r_{\mathbf{s_w}}[\tilde{n}_{\texttt{max}}] > r_{\texttt{threshold}}$, where the threshold is chosen as $r_{\texttt{threshold}} = 0.8$, the initial estimate of the fundamental frequency under the current time scale is

$$\hat{\theta}_{\texttt{f}} = \frac{f_{\texttt{ds}}}{f_{\texttt{s}}} \frac{2\pi}{\tilde{n}_{\texttt{max}}}, \tag{3.56}$$

re-normalised to $f_{\texttt{s}}$.

After this process is repeated for time scales one to twelve, and one or more possible fundamental frequencies are found, the estimate $\hat{\theta}_{\texttt{f}}$ with the largest $r_{\mathbf{s_w}}[\tilde{n}_{\texttt{max}}]$ is chosen as the estimate provided by the DPE module.

The computational load is substantially reduced if a harmonic complex is present in the previous frame. In this case, only one time scale, matched to the estimate of the fundamental frequency $\hat{\theta}_{\texttt{f,pf}}$ in the previous frame, is considered. The current frame contains

$$N_{\texttt{s}} = 10 \left\lceil \frac{2\pi}{\hat{\theta}_{\texttt{f,pf}} + 2\hat{\theta}_{\texttt{c,pf}} N_{\texttt{s,UR}}} \right\rceil$$

samples, corresponding to ten periods of the predicted fundamental frequency at the centre of the current frame. After LPC is applied to the frame, the prediction residual is low-pass filtered by the eighth-order Butterworth filter with cut-off frequency $\theta_{\texttt{cut}} = 8(\hat{\theta}_{\texttt{f,pf}} + 2\hat{\theta}_{\texttt{c,pf}} N_{\texttt{s,UR}})$, and the low-pass filtered signal is decimated at a rate

$$r_{\texttt{ds}} = \left\lfloor \frac{2\pi}{3\theta_{\texttt{cut}}} \right\rfloor.$$

If the new estimate of the fundamental frequency, obtained from the normalised auto-correlation function,

$$\hat{\theta}_{\texttt{f,cf}} = \frac{f_{\texttt{ds}}}{f_{\texttt{s}}} \frac{2\pi}{\tilde{n}_{\texttt{max}}},$$

where $f_{\mathtt{ds}} = f_{\mathtt{s}}/r_{\mathtt{ds}}$, adheres to

$$|e(\hat{\theta}_{\mathtt{f},\mathtt{pf}} + 2\hat{\theta}_{\mathtt{c},\mathtt{pf}} N_{\mathtt{s},\mathtt{UR}}) - e(\hat{\theta}_{\mathtt{f},\mathtt{cf}})| \leq 0.7 \text{ ERB}$$

and $r_{\mathbf{s_w}}[\tilde{n}_{\mathtt{max}}] > 0.8$, then $\hat{\theta}_{\mathtt{f},\mathtt{cf}}$ is the estimate of the fundamental frequency in the current frame. The threshold 0.7 ERB is slightly higher than the threshold 0.6 ERB for linking. If these conditions are not satisfied, the analysis procedure, as described above, is carried out over all time scales.

The HPE module

The pitch-synchronous frame duration corresponds to six periods of the initial estimate of the fundamental frequency

$$N_{\mathtt{s}} = 6 \left\lceil \frac{2\pi}{\hat{\theta}_{\mathtt{f}}} \right\rceil . \tag{3.57}$$

This frame has a briefer duration than the frame in which the initial estimate of the fundamental frequency was determined. The reason for this is that six periods of the fundamental frequency provide sufficient spectral resolution to discriminate harmonic partials. To reduce computational complexity, the signal is low-pass filtered and decimated. The low-pass filter cutoff-frequency is determined by the maximum number of harmonics present in a signal. Due to the high number of harmonics in the si01 (Harpsichord) excerpt, the maximum number of harmonics is set at $N_{\mathtt{h},\mathtt{max}} = 150$. The low-pass cutoff-frequency is chosen as $\theta_{\mathtt{cut}} = \min\{1.25 N_{\mathtt{h},\mathtt{max}} \hat{\theta}_{\mathtt{f}}, \pi\}$, and the eight-order Butterworth low-pass filter is applied. The low-pass filtered signal is decimated at a rate

$$r_{\mathtt{ds}} = \max\left\{ \left\lfloor \frac{\pi}{3\theta_{\mathtt{cut}}} \right\rfloor , 1 \right\} . \tag{3.58}$$

The magnitude of the frequency response is more than -25 dB down at $1.5\theta_{\mathtt{cut}}$ when $1.5\theta_{\mathtt{cut}} < \pi$.

The initial number of harmonics $\hat{N}_{\mathtt{h}}^{(1)}$ is determined by applying the PEAK module to the first twenty harmonic frequencies

$$\left\{ \xi_1(\hat{\theta}_{\mathtt{f}}, 0), \ldots, \xi_{20}(\hat{\theta}_{\mathtt{f}}, 0) \right\} ,$$

where we take the initial estimate of B as zero. The function ξ_k was defined in (3.12). A suitable peak identification criterion, as required by the PEAK module, was determined after experimentation to be $\tau_{\mathtt{min}} = 0.022$. The output of the PEAK module is used to construct the binary vector $\mathbf{P}_{\mathtt{mask}}^{(1)} \in \mathbb{N}^{20}$. A value of $P_{\mathtt{mask}}[k] = 1$ indicates that the k-th harmonic, with frequency $\xi_k(\hat{\theta}_{\mathtt{f}}, 0)$, corresponds to a spectral peak, while a value of $P_{\mathtt{mask}}[k] = 0$ indicates that the harmonic does not correspond to a spectral

peak. The initial number of harmonics $\hat{N}_{\mathrm{h}}^{(1)}$ then corresponds to the index preceding the first zero in $\mathbf{P}_{\mathtt{mask}}^{(1)}$. Therefore, if no inharmonicity is present in the audio signal, the HPE module may start with a high $\hat{N}_{\mathrm{h}}^{(1)}$, while if inharmonicity is present, the HPE module will select the first harmonic peaks that correspond to a frequency grid with no inharmonicity. If $\hat{N}_{\mathrm{h}}^{(1)} < N_{\mathrm{h},\mathtt{min}} = 2$, no inharmonicity can be estimated, and the HPE module indicates that no harmonic complex is present. Given $\hat{N}_{\mathrm{h}}^{(1)}$, the optimisation process is applied until convergence is achieved.

Next, the harmonic complex is extended by determining $\hat{N}_{\mathrm{h}}^{(2)}$. This is done by applying the PEAK module to the set of frequencies

$$\left\{\xi_1(\hat{\theta}_{\mathrm{f}}, \hat{B}), \ldots, \xi_{\hat{N}_{\mathrm{h}}^{(1)}+20}(\hat{\theta}_{\mathrm{f}}, \hat{B})\right\}.$$

Again, the output of the PEAK module is used to construct a binary vector, denoted by $\mathbf{P}_{\mathtt{mask}}^{(2)}$. To ensure that the process does not get stuck at a non-peak, $\hat{N}_{\mathrm{h}}^{(2)}$ corresponds to the index preceding the first zero after index $\hat{N}_{\mathrm{h}}^{(1)} + 1$ in $\mathbf{P}_{\mathtt{mask}}^{(2)}$. Therefore, non-peaks may be modelled by harmonics. Given $\hat{N}_{\mathrm{h}}^{(2)}$, the optimisation process is applied until convergence is achieved, and the number of harmonics is extended in a similar way.

The process ends after iteration i when five non-peaks are modelled by the harmonic complex; that is, when $\mathbf{P}_{\mathtt{mask}}^{(i+1)}$ contains more than five zeros in the indices 1 to $\hat{N}_{\mathrm{h}}^{(i+1)}$.

The frequency chirp is limited by $\theta_{\mathrm{c},\mathtt{max\%}} = 10\%$, and the inharmonicity parameter is restricted to lie between $B_{\mathtt{min}} = 0$ and $B_{\mathtt{max}} = 0.01$.

The IFE module

The IFE module utilises a number of time scales to detect spectral peaks. In finding a suitable frame length N_{s}, one has to consider the trade-off between the ability to discriminate between frequencies and limiting temporal non-stationarity. First, however, the complete frequency range is chosen. The frequency range should reflect the human ability to detect auditory stimuli while average signal behaviour should be considered too. It is more or less accepted that auditory stimuli outside the range [20, 20000] Hz do not lead to a hearing sensation. In the test material, however, no perceptually relevant partials below 25 Hz were found, while very few partials have a frequency higher than 16 kHz. For this reason, a slightly reduced frequency range

$$\left[\theta_{2,\mathtt{min}}\frac{f_{\mathrm{s}}}{2\pi}, \theta_{2,\mathtt{max}}\frac{f_{\mathrm{s}}}{2\pi}\right] = [25, 16000]\ \mathrm{Hz} \tag{3.59}$$

is chosen. The IFE module is applied only if the windowed frame $\mathbf{s_w}$ corresponding to a particular time scale is a non-zero sequence. The four time scales and the

Time scale	$t_{\mathtt{d}}$ ms	$[f_{\mathtt{low}}, f_{\mathtt{high}}]$ Hz	$\tau_{\mathtt{min}}$
1	160	[25, 750]	0.015
2	88	[45, 1500]	0.027
3	44	[90, 8000]	0.05
4	20	[200, 16000]	0.01

Table 3.4: *The time-scale duration t_d is given in ms. The corresponding frequency band covers $[f_{low}, f_{high}]$ Hz. The threshold for peak detection is τ_{min}.*

associated frequency bands given in Table 3.4 were determined empirically after experimentation.

We notice the large overlap among the frequency bands. A consequence of the overlapping frequency bands is that a partial may be detected on multiple time scales. When this happens, the PEAK module is utilised to select the candidate from the set of duplicates with the smallest modelling error τ_k, refer to Equation (3.22).

Sinusoids too close to harmonics are considered to be harmonics, and are removed from the individual-sinusoids sub-component. Due to the limited frequency resolution afforded by the DFT, and temporal non-stationarity of partials, the initial frequency estimates of duplicate partials will seldom be identical. Therefore, suitable thresholds have to be specified. Since the harmonic complex is a model with a limited degree of freedom, the distance in frequency between sinusoids detected here and harmonics should be larger than the distance in frequency between duplicate sinusoids before removal. Sinusoids are considered duplicates of the same partial when they are closer than 0.24 ERB in frequency, and sinusoids closer than 0.33 ERB to harmonics are considered to be harmonics. These values were determined empirically. The thresholds for peak identification, $\tau_{\mathtt{min}}$, were determined empirically for each time scale, and are given in column four of Table 3.4.

The distortion provided by the MASK module is dependent on the frame duration $t_{\mathtt{d}}$, and is rescaled to match the parameter update-rate $t_{\mathtt{UR}}$ in the following way:

$$D_k := \frac{t_{\mathtt{UR}}}{t_{\mathtt{d}}} D_k. \tag{3.60}$$

After this re-scaling, a distortion $D_k = 1$ lies at the threshold of detection in a frame with duration $t_{\mathtt{UR}}$, while a distortion $D_k > 1$ is detectable, and a distortion $D_k < 1$ not. Sinusoids with a distortion $D_k < D_{\mathtt{min}}$, where $D_{\mathtt{min}} = 0.25$, are not considered. It is known that the human auditory system integrates acoustical information over time. Therefore, if sinusoids on a track do indeed represent the same partial, the human auditory system will integrate these D-values and a track may become detectable even if the D-values of all partials on the track are smaller than one. $D_{\mathtt{min}}$ is chosen such that a track with average length, and with sinusoids having D-values

equal to $D_{\mathtt{min}}$, will have an integrated D-value at the threshold of detection. The integration of the D-values is described in more detail in Chapter 4, where the bit-rate scalability functionality is described.

The SPE module

The SPE module refines the parameters of an individual sinusoid in the same time scale under which the IFE module detected the sinusoid. In contrast to the IFE module, the SPE module applies low-pass filtering and down-sampling of the windowed audio signal to alleviate the computational burden. Again, the eight-order Butterworth low-pass filter is applied, with cut-off frequency $f_{\mathtt{cut}} = 1.25\, f_{\mathtt{high}}$ Hz, see Table 3.4 for the values of $f_{\mathtt{high}}$. The down-sampling rate is chosen as

$$r_{\mathtt{ds}} = \left\lfloor \frac{f_{\mathtt{s}}}{3\, f_{\mathtt{cut}}} \right\rfloor < \frac{f_{\mathtt{s}}}{2\, f_{\mathtt{cut}}},$$

and the resulting sampling frequency is

$$f_{\mathtt{ds}} = \frac{f_{\mathtt{s}}}{r_{\mathtt{ds}}} > 2\, f_{\mathtt{cut}}.$$

The parameters of sinusoids belonging to the same time scale are improved simultaneously, where time scale 1 is considered first, etc. Even though an effort was made in the IFE module to remove duplicates of the same partial, it is possible that the frequency estimates of sinusoids are adapted in the SPE module such that it is necessary to check for duplicates again. Therefore, during each iteration, sinusoids in the current time scale are compared to sinusoids in the

1. current time scale,
2. harmonic complex (if present), and
3. previous time scales.

When the frequency parameters of sinusoids in the same time scale are converging to the same value, they are merged when they are closer than 0.165 ERB in frequency to ensure that the condition number of the system matrix is small enough. The threshold of 0.33 ERB to harmonic partials, specified in the IFE module, is maintained here. Finally, sinusoids are removed when they are closer than 0.225 ERB to sinusoids from the previous time scales. This threshold is slightly lower than the 0.24 ERB used in the IFE module. As in the IFE module, these thresholds were determined empirically.

At most 10 iterations are carried out per time scale, and the iterations are ended sooner if

$$\max_k \frac{|\hat{\Delta}^{(i)}_{x_{k,2}}| + |\hat{\Delta}^{(i)}_{x_{k,3}}|\frac{N_\mathtt{s}}{2}}{\hat{x}^{(i)}_{k,2} + |\hat{x}^{(i)}_{k,3}|\frac{N_\mathtt{s}}{2}} \leq \Delta_{\mathtt{min}} = 5 \cdot 10^{-4}.$$

The frequency chirp is limited by $\theta_{3,\mathtt{max\%}} = 10\%$. A sufficient higher bound on the number of individual sinusoids is specified as $N_{\mathtt{c},\mathtt{max}} = 150$.

The CAP module

Only one time scale, containing $N_\mathtt{s} = 2N_{\mathtt{s},\mathtt{UR}} + 1$ samples, is used to determine the amplitude parameters. Therefore, amplitude parameters in adjacent frames are determined with 50% overlap, see Figure 3.11. The constant amplitudes $a_{k,1}$, when converted to dB, are restricted to lie in the range $[a_{1,\mathtt{dB},\mathtt{min}}, a_{1,\mathtt{dB},\mathtt{max}}]$ dB. In choosing this range, the following considerations are taken into account. An audio signal $s[n]$ is related to Sound Pressure Level (SPL) through the conversion formula

$$10\log_{10}(P) + G \quad \text{dB SPL}, \tag{3.61}$$

where

$$P = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=-N}^{N} |s[n]|^2 \tag{3.62}$$

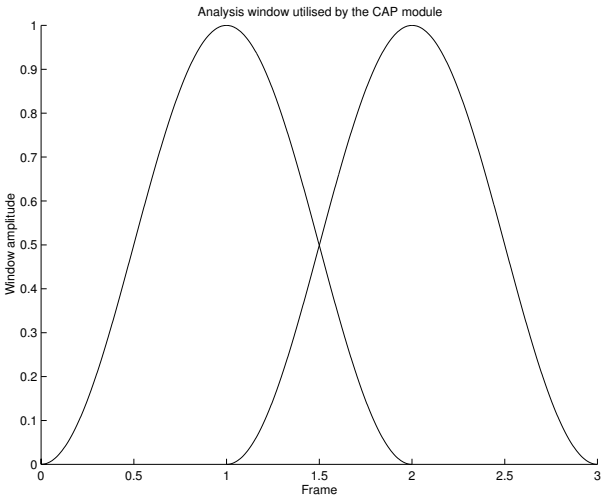


Figure 3.11: *Amplitude-complementary Hann window $w[n]$ with 50% overlap utilised in the CAP module.*

is the power of $s[n]$ and G is a correction factor which depends on the sound reproduction level. Since the sound reproduction level, or volume, will vary, no definite value for G exists. If one assumes that the volume is set such that the power of a full-amplitude sinusoid (with $a_{1,\mathtt{max}} = 1$) yields a level of 96 dB SPL (well below the threshold of pain $\approx$130 dB SPL [56, page 1]), then

$$\begin{aligned} 10\log_{10}\left(\frac{|a_{1,\mathtt{max}}|^2}{2}\right) + G = 96 \quad \text{dB SPL} \\ \Longrightarrow G = 99 \quad \text{dB} \end{aligned} \tag{3.63}$$

and $a_{1,\mathtt{dB},\mathtt{max}} = 20\log_{10} a_{1,\mathtt{max}} = 0$ dB. The absolute threshold of hearing (or threshold in quiet) corresponds to -5 dB SPL at 4 kHz [63]. Therefore, the minimum amplitude is

$$\begin{aligned} 10\log_{10}\left(\frac{|a_{1,\mathtt{min}}|^2}{2}\right) + G = -5 \quad \text{dB SPL} \\ \Longrightarrow a_{1,\mathtt{dB},\mathtt{min}} = -101 \quad \text{dB}. \end{aligned} \tag{3.64}$$

Given these considerations, we choose

$$[a_{1,\mathtt{dB},\mathtt{min}}, a_{1,\mathtt{dB},\mathtt{max}}] = [-100, 0] \quad \text{dB}. \tag{3.65}$$

Sinusoids with amplitudes smaller than $a_{1,\mathtt{dB},\mathtt{min}}$ are removed. The amplitude sweeps $a_{k,2}$ are limited by $a_{2,\mathtt{max\%}} = 100\%$ to avoid a negative amplitude.

Linking

Individual sinusoids in adjacent frames should be linked to form a track when they represent the same partial. Differential encoding of the amplitude and frequency parameters of linked sinusoids results in a significant coding gain if the underlying partial has a stationary character. Similarly, two harmonic complexes in adjacent frames should be linked if they represent the same harmonic sound. In the interest of simplicity, an individual sinusoid is not linked to a harmonic in the harmonic complex, nor vice versa.

To determine whether individual sinusoids in adjacent frames represent the same partial, a definition of similarity has to be formulated. Given the observation that there is a close relation between the ERB scale and the human ability to discriminate frequencies, frequencies of individual sinusoids in adjacent frames are matched on an ERB scale to determine whether they should be linked. This approach is taken in a number of parametric audio coders [11, 12]. Therefore, two individual sinusoids in adjacent frames ($i-1$ and i) are linked if their corresponding instantaneous frequencies at the frame boundary adhere to the following constraint

$$\left| e(\hat{\theta}_{k,2}^{(i-1)} + \hat{\theta}_{k,3}^{(i-1)} N_{\mathtt{s},\mathtt{UR}}) - e(\hat{\theta}_{l,2}^{(i)} - \hat{\theta}_{l,3}^{(i)} N_{\mathtt{s},\mathtt{UR}}) \right| \leq e_{\theta_{2,\mathtt{link}}}, \tag{3.66}$$

where $e_{\theta_{2,\mathtt{link}}} = 0.3$ ERB and where the frequency chirp is taken into account. This value for $e_{\theta_{2,\mathtt{link}}}$ ensures that the instantaneous frequencies at the frame boundary are well matched. The linking procedure has an iterative character. Initially, the set of linking candidates contains all sinusoids from frames $i-1$ and i. At each iteration step, the sinusoids in frame $i-1$ and frame i that are the closest on the ERB scale, and for which (3.66) holds, are linked and are removed from the list of linking candidates, after which the next iteration is performed. The iterations end when there are no sinusoids satisfying (3.66), or when the list of linking candidates is empty. When a sinusoid in frame i can not be linked to a sinusoid in frame $i-1$, a track is started up in frame i. When a sinusoid in frame i can be linked to one in frame $i-1$, the track is continued from frame $i-1$ to the current frame. Sinusoidal tracks in frame $i-1$ that can not be linked to sinusoids in frame i end in frame $i-1$.

Similarly, the harmonic complex in frame i is linked to the harmonic complex in frame $i-1$ if

$$\left| e(\hat{\theta}_{\mathtt{f}}^{(i-1)} + \hat{\theta}_{\mathtt{c}}^{(i-1)} N_{\mathtt{s},\mathtt{UR}}) - e(\hat{\theta}_{\mathtt{f}}^{(i)} - \hat{\theta}_{\mathtt{c}}^{(i)} N_{\mathtt{s},\mathtt{UR}}) \right| \leq e_{\theta_{\mathtt{f},\mathtt{link}}}. \tag{3.67}$$

The linking criterion $e_{\theta_{\mathtt{f},\mathtt{link}}}$ used here can not be as strict as $e_{\theta_{2,\mathtt{link}}}$ used for linking individual sinusoids, since the fundamental frequency and frequency chirp are derived from all harmonic partials. Our choice $e_{\theta_{\mathtt{f},\mathtt{link}}} = 0.6$ ERB was found to result in linked harmonic complexes spanning many frames in most cases. Similar to tracks, a distinction is made between starting and linked harmonics in a harmonic complex. Linking is performed by the Sinusoidal Linking (SL) module illustrated in Figure 3.9.

We note that other parametric audio coders additionally employ amplitude and phase matching to determine whether individual sinusoids should be linked or not. The merits of additional amplitude and phase matching in the prototype described here, should be investigated.

Quantisation

The resolution with which amplitude and frequency parameters are quantised should be matched to the just-noticeable differences in frequency and amplitude, known from psycho-acoustics.

Detecting changes in frequency depends on the duration of the tone. As a rule, the just-noticeable difference in frequency decreases with increasing tone duration: $\approx$0.2 Bark at a 10 ms duration and $\approx$0.01 Bark at a 500 ms duration [56, page 117]. This corresponds to $\approx$0.3 ERB and $\approx$0.02 ERB, respectively.

The fundamental frequency is uniformly quantised on an ERB grid ranging from $e(\theta_{\mathtt{f},\mathtt{min}})$ to $e(\theta_{\mathtt{f},\mathtt{max}})$ by $N_{\theta_{\mathtt{f}}} = 12$ bits, which results in the very fine quantisation step-size 0.0048 ERB. This small step-size is required to keep the resulting step-size of higher harmonics small enough to avoid artefacts. Since the inharmonicity

parameter B has a direct influence on the frequencies in the harmonic complex, $N_B = 12$ bits are used to uniformly quantise B on the grid ranging from $B_{\mathtt{min}}$ to $B_{\mathtt{max}}$. The resulting step size is approximately $2.5 \cdot 10^{-6}$.

The frequency $\hat{\theta}_2$ of an individual sinusoid is uniformly quantised on a grid ranging from $e(\theta_{2,\mathtt{min}})$ to $e(\theta_{2,\mathtt{max}})$ by $N_{\theta_2} = 9$ bits. The resulting quantisation step-size is 0.076 ERB. This quantisation error corresponds to the just-noticeable difference in frequency for a tone with duration around 50 ms [56, page 117].

The frequency chirp $\hat{\theta}_{\mathtt{c}}$ is transformed to

$$\hat{\theta}_{\mathtt{c},\%} = 100 \frac{\hat{\theta}_{\mathtt{c}}}{\hat{\theta}_{\mathtt{f},\mathtt{q}}} N_{\mathtt{s},\mathtt{UR}} \tag{3.68}$$

and is subsequently uniformly quantised on a grid ranging from $-\theta_{\mathtt{c},\mathtt{max}\%}$ to $\theta_{\mathtt{c},\mathtt{max}\%}$ by $N_{\theta_{\mathtt{c}}} = 5$ bits. One quantisation level is removed to include $\theta_{\mathtt{c},\%} = 0$ on the grid, and the resulting step size is 0.67%. Similarly, the frequency chirp $\hat{\theta}_3$ of an individual sinusoid is transformed to

$$\hat{\theta}_{3,\%} = 100 \frac{\hat{\theta}_3}{\hat{\theta}_{2,\mathtt{q}}} N_{\mathtt{s},\mathtt{UR}} \tag{3.69}$$

and uniformly quantised on a grid ranging from $-\theta_{3,\mathtt{max}\%}$ to $\theta_{3,\mathtt{max}\%}$ by $N_{\theta_3} = 5$ bits, with one level removed to include $\theta_{3,\%} = 0$ on the grid, and the resulting step size is 0.67%.

The constant amplitudes $\hat{a}_1$ are transformed to values in dB by

$$\hat{a}_{1,\mathtt{dB}} = 20 \log_{10}(\hat{a}_1).$$

The transformed amplitudes are uniformly quantised on a grid ranging from $a_{1,\mathtt{dB},\mathtt{min}}$ to $a_{1,\mathtt{dB},\mathtt{max}}$ by $N_{a_1} = 6$ bits. This results in a quantisation step size of 1.59 dB which is a reasonable approximation of the just-noticeable difference in level [56, pp. 160 – 162]. The amplitude sweep $\hat{a}_2$ is transformed to

$$\hat{a}_{2,\%} = 100 \frac{\hat{a}_2}{\hat{a}_{1,\mathtt{q}}} \frac{N_{\mathtt{s},\mathtt{UR}}}{2} \tag{3.70}$$

and uniformly quantised on a grid ranging from $-a_{2,\mathtt{max}\%}$ to $a_{2,\mathtt{max}\%}$ by $N_{a_2} = 4$ bits, where one quantisation level is removed to include $a_{2,\%} = 0$ on the grid. The resulting quantisation step size is 14.29%. Finally, the constant phase $\hat{\theta}_1$ is uniformly quantised on a grid ranging from $-\pi$ to π by $N_{\theta_1} = 5$ bits.

Quantisation is applied by the Sinusoidal Quantisation (SQ) module, see Figure 3.9. Our quantisation strategy resembles the approaches taken in other parametric audio coders [11, 12].

Coding the model parameters in the bit-stream

At the start of a harmonic complex, all quantisation indices are absolutely encoded. When a harmonic complex is linked, a distinction is made between starting and linked harmonics, similar to the distinction made between starting and linked tracks. For the starting harmonics, the constant-amplitude quantisation indices are absolutely encoded, while for the linked harmonics, these indices are time-differentially encoded. The quantisation indices corresponding to the amplitude-sweep $\hat{a}_{k,2}$, phase $\hat{\theta}_{k,1}$, and frequency-chirp $\hat{\theta}_{\mathrm{c}}$ parameters are always absolutely encoded. Furthermore, when a harmonic complex is linked, the fundamental frequency $\hat{\theta}_{\mathrm{f}}$, and inharmonicity parameter $\hat{B}$ are time-differentially encoded.

Similarly, at the start of a track, the quantisation indices are absolutely encoded, while for linked sinusoids, the quantisation indices corresponding to the constant-amplitude $\hat{a}_{k,1}$ and frequency $\hat{\theta}_{k,2}$ parameters are time-differentially encoded. The quantisation indices corresponding to the amplitude-sweep $\hat{a}_{k,2}$, phase $\hat{\theta}_{k,1}$, and chirp $\hat{\theta}_{k,3}$ parameters are absolutely encoded for linked sinusoids.

Huffman coding is applied to remove redundancy and thus lower the number of bits required to transmit the quantisation indices. The Sinusoidal Entropy Coding (SEC) module, see Figure 3.9, applies entropy coding.

3.5.2 Noise coder

The noise coder was described in Section 3.4. This section provides design specifications for the noise coder. The main parameters in the noise coder and their values chosen are summarised in Table 3.5. All the modules comprising the noise coder (NC) are illustrated in Figure 3.12. The prediction coefficients are determined by the Laguerre-based Pure Linear Prediction (PLP) module. These prediction coefficients are used to model the spectral envelope of the signal $s_{\mathtt{residual}}[n]$. The pre-

Module	Variable	Value
PLP	λ	0.7
	O_{p}	24
	γ	0.98
TEM	$E_{\mathtt{dB,min}}$	-85 dB
	$E_{\mathtt{dB,max}}$	15 dB
PCQ	N_l	6 bits
	$l_{\mathtt{max}}$	8
TEQ	N_E	6 bits

Table 3.5: *Summary of the parameter choices for the PLP, TEM, PCQ, and TEQ modules.*

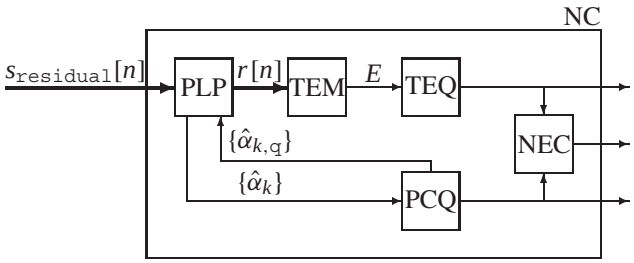


Figure 3.12: *In the Noise Coder (NC) module, the input signal ($s_{\texttt{residual}}[n]$) is predicted by the Laguerre-based Pure Linear Prediction (PLP) module. The prediction residual $r[n]$, being spectrally flat, is passed to the Temporal Envelope Measurement (TEM) module. The temporal envelope E is quantised by the Temporal Envelope Quantisation (TEQ) module. The prediction coefficients $\hat{\alpha}_k$ obtained by the PLP module are quantised by the Prediction Coefficient Quantisation (PCQ) module, and the quantised prediction coefficients, $\hat{\alpha}_{k,q}$, are fed back to the PLP module, which creates the prediction residual $r[n]$ from these quantised coefficients. The quantised prediction coefficients and quantised short-time energy are input to the Noise Entropy Coding (NEC) module, which applies entropy coding to remove redundancy and determines the cost of coding each noise-model parameter.*

diction coefficients are quantised by the Prediction Coefficient Quantisation (PCQ) module, and the quantised prediction coefficients are fed back and used by the PLP module to generate the prediction residual $r[n]$. The temporal envelope of the prediction residual, denoted by E, is determined by the Temporal Envelope Measurement (TEM) module. These temporal envelope measurements are quantised by the Temporal Envelope Quantisation (TEQ) module. The quantised prediction coefficients and temporal envelope measurements are entropy encoded in the Noise Entropy Coding (NEC) module. These modules are described in the following.

The PLP module

In the interest of simplicity, the prediction coefficients are measured at the same rate as the sinusoidal parameters. The optimal prediction coefficients are obtained from a frame, windowed by the Hann window, containing $N_{\texttt{s}} = 2N_{\texttt{s,UR}} + 1 = 711$ samples, similar to the CAP module. A value of $\lambda = 0.7$ is chosen since it yields a warped frequency axis that is a reasonable approximation of the ERB scale at a sampling frequency of $f_{\texttt{s}} = 44.1$ kHz [114]. The signal $\mathbf{s}_{\texttt{residual}}$ is zero-padded with 200 samples to allow sufficient damping of the Laguerre filters [117]. A relatively low prediction order of $O_{\texttt{p}} = 24$ is chosen to limit the cost of the noise component.

The TEM module

The short-time energy is measured in five sub-frames of length $t_{\mathtt{SUR}}$, where

$$t_{\mathtt{SUR}} = \frac{1}{5} t_{\mathtt{UR}} = \frac{N_{\mathtt{s,SUR}}}{f_{\mathtt{s}}} = \frac{71}{44100} \approx 1.6 \text{ ms}. \tag{3.71}$$

This corresponds reasonably well to the temporal resolution of the human auditory system [74]. Similar to the amplitude parameters and the prediction coefficients, the temporal level is determined with 50% overlap, see Figure 3.13, where the Hann window $w_{\mathtt{short}}[n]$, generated according to (3.51) with $N_{\mathtt{s,SUR}} = 71$ samples, is utilised.

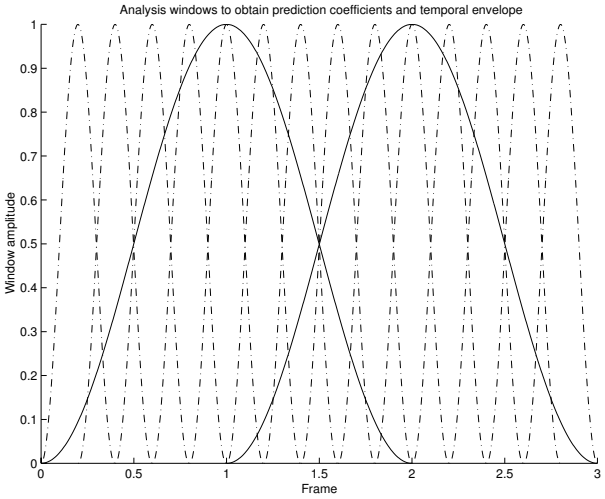


Figure 3.13: *The Hann window $w[n]$ (solid line) containing $N_s = 2N_{s,\,UR} + 1 = 711$ samples is used to obtain the prediction coefficients. The Hann window $w_{\,short}[n]$ (dash-dotted line) containing $N_s = 2N_{s,\,SUR} + 1 = 143$ samples is used to measure the short-time energy.*

Quantisation

Direct quantisation of LPC prediction coefficients may result in an unstable LPC synthesis filter and can produce relatively large spectral distortions. To avoid these problems, a large number of bits are required. Therefore, alternative representations of prediction coefficients are desired for efficient quantisation of these coefficients. Two popular representations are log-area ratios (LARs) and line spectral frequencies (LSFs) [60, Chapter 12]. Prediction coefficients are first mapped to LARs or LSFs, after which quantisation and inter-frame differential encoding are applied. Although

LARs have a slightly lower coding efficiency than LSFs, the LAR representation allows the use of a varying number of prediction coefficients (O_{p}) per frame. Removing LAR coefficients lowers the resolution with which the spectral envelope of the signal $s_{\mathtt{residual}}[n]$ is described. The possibility to easily scale the number of LARs, and thus the resolution with which the residual component is approximated, leads us to the choice of utilising LARs.

Before the prediction coefficients $\hat{\alpha}_k$, associated with a minimum-phase Laguerre-based PLP analysis filter, can be mapped to LARs, they have to be transformed first to prediction coefficients $\hat{d}_k$, corresponding to a minimum-phase FIR filter [118]. The transformed prediction coefficients are spectrally smoothed $\hat{d}_k := \gamma^k \hat{d}_k$ [116], where γ is chosen as $\gamma = 0.98$. The spectrally smoothed $\hat{d}_k$s are mapped to LARs $\hat{\alpha}_{\mathtt{LAR},k}$, and uniformly quantised in the range $[-\alpha_{\mathtt{LAR,max}}, \alpha_{\mathtt{LAR,max}}]$, where $\alpha_{\mathtt{LAR,max}} = 8$, and with $N_l = 6$ bits. We denote the quantised prediction coefficients in the original domain by $\hat{\alpha}_{k,\mathtt{q}}$. Quantisation of the prediction coefficients is carried out by the Prediction Coefficient Quantisation (PCQ) module, see Figure 3.12. The quantised prediction coefficients $\hat{\alpha}_{k,\mathtt{q}}$ are fed back to the PLP module, which generates the prediction residual $r[n]$ from the quantised prediction coefficients.

The temporal envelope measurements E are transformed to a dB scale by $E_{\mathtt{dB}} = 10 \log_{10} E$ and uniformly quantised on a grid ranging from $E_{\mathtt{dB,min}}$ to $E_{\mathtt{dB,max}}$. For the maximum value of the temporal envelope, $E_{\mathtt{max}}$, we consider the full-band spectrally flat random signal $s_{\mathtt{random}}[n]$ with variance $\sigma^2_{s_{\mathtt{random}}} = 1/3$. Passing $s_{\mathtt{random}}[n]$ through the PLP module will result in the residual $r_{\mathtt{random}}[n]$ having the same variance $\sigma^2_{r_{\mathtt{random}}} = 1/3$. Consequently, the expected value of the temporal envelope under the Hann window $w_{\mathtt{short}}[n]$ is

$$\begin{aligned} \mathfrak{E}\{E_{\mathtt{max}}\} &= \mathfrak{E}\left\{\|\mathbf{r}_{\mathtt{random}}\|^2_{\mathbf{w}_{\mathtt{short}}}\right\} \\ &= \mathfrak{E}\{|r_{\mathtt{random}}[n]|^2\} \sum_n |w_{\mathtt{short}}[n]| \\ &= \sigma^2_{r_{\mathtt{random}}} \sum_n |w_{\mathtt{short}}[n]| \\ &= 23.67, \end{aligned} \tag{3.72}$$

where $\mathfrak{E}$ denotes expectation. Therefore, we choose $E_{\mathtt{dB,max}} = 10 \log_{10} 23.67 = 13.74 \approx 15$ dB. It is difficult to say something about the minimum short-time energy. A dynamic range of 100 dB was found to be sufficient for practical purposes. Therefore, $[E_{\mathtt{dB,min}}, E_{\mathtt{dB,max}}] = [-85, 15]$ dB. This range is uniformly quantised with $N_E = 6$ bits, resulting in a quantisation step-size of 1.4 dB. Quantisation of the temporal envelope measurements is carried out by the Temporal Envelope Quantisation (TEQ) module, illustrated in Figure 3.12.

Coding the model parameters in the bit-stream

At the start of an audio excerpt and in refresh frames, the prediction coefficient quantisation indices are absolutely encoded. In the remaining frames, these quantisation indices are time-differentially encoded. Similarly, the temporal envelope quantisation indices are absolutely encoded in the first and refreshment frames, while time-differential encoding is applied in the remainder of the frames. Huffman coding is applied to remove redundancy and thus lower the number of bits required to transmit the quantisation indices. The Noise Entropy Coding (NEC) module, illustrated in Figure 3.12, applies entropy coding.

3.6 Decoder

The decoder consists of the sinusoidal decoder (SD) and noise decoder (ND). The decoding process in both the SD and ND modules is based on overlap-add (OLA), similar to the approach taken in PPC [12].

The sinusoidal decoder generates the sinusoidal component

$$\hat{\mathbf{s}}^{(i)}_{\texttt{sinusoid}} \in \mathbb{R}^{2N_{\texttt{s,UR}}+1}$$

in frame i out of the quantised parameters contained in the bit-stream

$$\hat{s}^{(i)}_{\texttt{sinusoid}}[n] = \sum_{k=1}^{N^{(i)}_{\texttt{h}}+N^{(i)}_{\texttt{c}}} (a^{(i)}_{k,1,\texttt{q}} + a^{(i)}_{k,2,\texttt{q}} n)\cos(\theta^{(i)}_{k,1,\texttt{q}} + f(x^{(i)}_{k,2,\texttt{q}}, x^{(i)}_{k,3,\texttt{q}} n)n),$$

where $n = -N_{\texttt{s,UR}}, \ldots, N_{\texttt{s,UR}}$ and where $f(x_2, x_3 n)$ was defined in (3.39) in Section 3.3.3. The complete decoded sinusoidal component $\hat{s}_{\texttt{sinusoid}}[n]$, after frame i is decoded, is then updated by OLA as

$$\hat{s}_{\texttt{sinusoid}}[n+(i-1)N_{\texttt{s,UR}}+1] := \hat{s}_{\texttt{sinusoid}}[n+(i-1)N_{\texttt{s,UR}}+1] + \hat{s}^{(i)}_{\texttt{sinusoid}}[n]w[n],$$

where $w[n]$ is the Hann window.

When the constant-phase parameters of linked individual sinusoids and linked harmonics are not available, *phase continuation* is applied. The constant-phase $\theta^{(i)}_{k,1}$ of sinusoid k in frame i is estimated by using the constant-phase $\theta^{(i-1)}_{l,1}$, frequency $x^{(i-1)}_{l,2}$ and frequency chirp $x^{(i-1)}_{l,3}$ of sinusoid l in frame $i-1$ by

$$\theta^{(i)}_{k,1} = \theta^{(i-1)}_{l,1} + f(x^{(i-1)}_{l,2}, x^{(i-1)}_{l,3} N_{\texttt{s,UR}})N_{\texttt{s,UR}}. \tag{3.73}$$

The Noise Decoder (ND) is illustrated in Figure 3.14. The noise decoder starts by generating a spectrally flat stochastic signal $r_{\texttt{stochastic}}[n]$, with unit variance. In

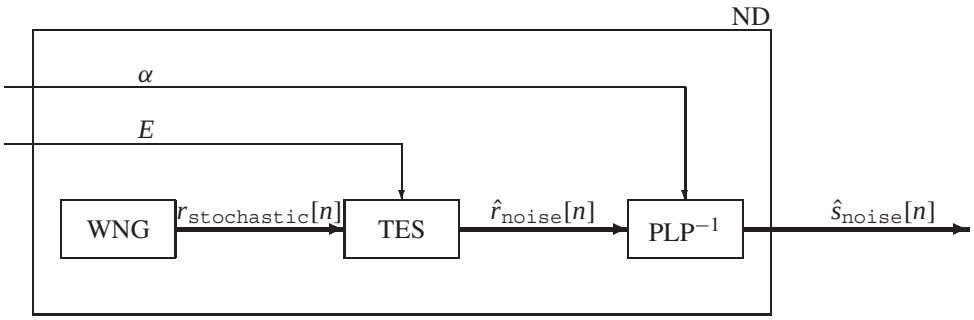


Figure 3.14: *The Noise Decoder (ND) consists of a white-noise generator (WNG), a temporal-envelope shaper (TES) driven by the short-time energy E, and the PLP synthesis scheme (PLP^{-1}), driven by the prediction coefficients α. The spectrally-flat stochastic signal produced by the WNG module is denoted by $r_{\texttt{stochastic}}[n]$. The temporally-shaped stochastic signal is denoted by $\hat{r}_{\texttt{noise}}[n]$, and the spectrally shaped stochastic signal is denoted by $\hat{s}_{\texttt{noise}}[n]$.*

sub-frame j, the signal $r_{\texttt{stochastic}}[n]$ is given the correct temporal envelope, $E_{\texttt{q}}^{(j)}$, to form $\hat{r}_{\texttt{noise}}^{(j)}[n]$:

$$\hat{r}_{\texttt{noise}}^{(j)}[n] = \left(\frac{E_{\texttt{q}}^{(j)}}{\sum_n w_{\texttt{short}}[n]} \right) r_{\texttt{stochastic}}[n + (j-1)N_{\texttt{s,SUR}} + 1], \qquad (3.74)$$

where $n = -N_{\texttt{s,SUR}}, \ldots, N_{\texttt{s,SUR}}$. The complete $\hat{r}_{\texttt{noise}}[n]$ is formed by OLA

$$\hat{r}_{\texttt{noise}}[n+(j-1)N_{\texttt{s,SUR}}+1] = \hat{r}_{\texttt{noise}}[n+(j-1)N_{\texttt{s,SUR}}+1] + \hat{r}_{\texttt{noise}}^{(j)}[n] w_{\texttt{short}}[n].$$

The noise component $\hat{s}_{\texttt{noise}}[n]$ is generated by passing $\hat{r}_{\texttt{noise}}[n]$ through the PLP synthesis module (PLP^{-1}) with prediction coefficients $\{\alpha_{k,\texttt{q}}^{(i)}\}$ updated once every frame. The PLP synthesis scheme is illustrated in Figure 3.6 (b), and the transfer function of the PLP synthesis scheme is given by

$$\begin{aligned} F_{\texttt{synth}}(z) &= \frac{1}{F_{\texttt{ana}}(z)} \\ &= \frac{1}{1 - z^{-1} \sum_{k=1}^{O_{\texttt{p}}} \alpha_{k,\texttt{q}} H_k(\lambda; z)}, \end{aligned} \qquad (3.75)$$

refer to (3.46) and Figure 3.6 (b). The decoded audio signal is then

$$s_{\texttt{decoded}}[n] = \hat{s}_{\texttt{sinusoid}}[n] + \hat{s}_{\texttt{noise}}[n].$$

3.7 Results

In this section, the performance of the sinusoidal analysis is illustrated by considering a number of examples. The harmonic-complex is considered in Section 3.7.1 and the individual-sinusoids in Section 3.7.2. Finally, statistics of the parameters comprising the sinusoidal component are given in Section 3.7.3.

We do not consider the noise component here. The reason is that our goal is to design a bit-rate scalable coder. The diagram of the bit-rate scalable coder is given in Figure 2.11 on page 36. In the bit-rate scalable coder, the noise coder is applied to the residual corresponding to the sinusoidal component contained in the base layer $\hat{s}_{\texttt{sinusoid,bl}}[n]$. Since $\hat{s}_{\texttt{sinusoid,bl}}[n]$ is a considerably scaled-down version of the complete sinusoidal signal $\hat{s}_{\texttt{sinusoid}}[n]$, we expect the statistics of the noise component, matched to $\hat{s}_{\texttt{sinusoid,bl}}[n]$, to be different from the noise component matched to $\hat{s}_{\texttt{sinusoid}}[n]$ in this chapter. The bit-rate scalability functionality is described in Chapter 4, and the noise component will therefore be considered in that chapter.

3.7.1 Harmonic complex

The performance of the Harmonic Complex Analysis (HCA) module, see Figure 3.1 for a block diagram of this module, is illustrated by considering both a synthetic and a real-world signal. The purpose of considering a synthetic signal is to illustrate the ability of the HCA module to obtain the model parameters when the signal is generated according to the signal model. Example 3.7.1 considers a synthetic signal. The ability of the HCA module to find convincing model-parameter estimates from a real-world signal is illustrated in Example 3.7.2. Finally, Example 3.7.3 illustrates a time-frequency plot where evolution of the fundamental frequency is considered over a number of frames taken from a real-world signal.

Example 3.7.1 (Synthetic signal) The non-constant polynomial-phase parameters of the synthetic harmonic signal are given in column two of Table 3.6. The synthetic signal contains both an inharmonicity B and frequency-chirp parameter $\theta_{\texttt{c,\%}}$

Parameter	Value	DPE	HPE
$\theta_{\texttt{f}}$	205	212	204.9
$\theta_{\texttt{c,\%}}$	4		3.9973
B	0.001		0.00100
N_{h}	30		30

Table 3.6: *Model parameters of the synthetic harmonic signal. The initial estimate of the fundamental frequency is provided by the Detector and Pitch Estimator (DPE) module. The refined parameters are obtained by the Harmonic Parameter Estimation (HPE) module.*

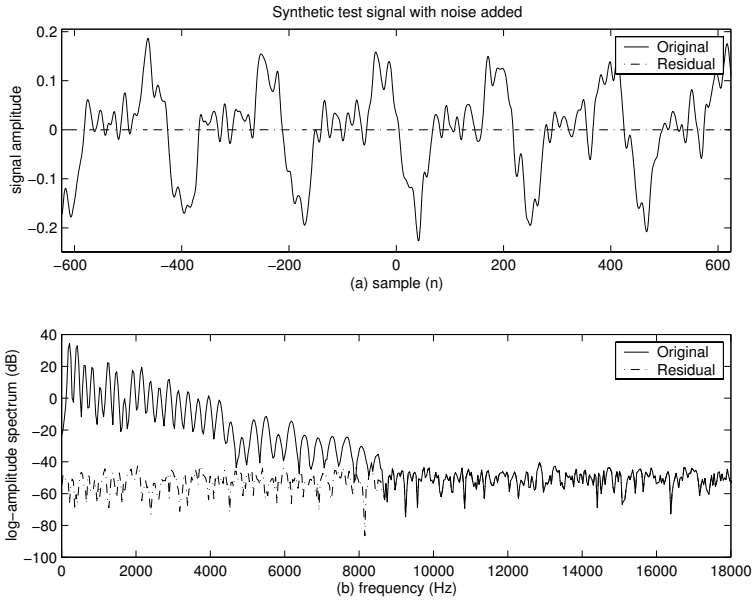


Figure 3.15: *Synthetic input signal. (a) Temporal waveform of the original (solid line) and the residual (dash-dotted line) under the rectangular window. (b) Log-amplitude spectrum of the original (solid line) and the residual (dash-dotted line) under the Hann window.*

of non-negligible magnitude, and has constant amplitude behaviour. Spectrally flat (white) synthetic noise with a signal-to-noise ratio of 50 dB is added to the signal. The waveform of the synthetic signal under the rectangular window is illustrated in Figure 3.15 (a), while the log-amplitude spectrum under the Hann window is illustrated in part (b) of this figure. The amplitude parameters were chosen such that the spectrum has a negative overall slope, which is the case in most real-world signals. Furthermore, the frequency chirp causes the spectral peaks to become broader with increasing frequency, while the inharmonicity causes the difference in frequency between adjacent spectral peaks to increase with increasing frequency.

The Detector and Pitch Estimator (DPE) module provided an initial estimate of the fundamental frequency of 212 Hz, see column three of Table 3.6. This is a reasonable approximation of the true fundamental frequency.

The Harmonic Parameter Estimation (HPE) module used zero as an initial estimate of the frequency chirp and inharmonicity parameters. The improved parameter estimates obtained by the HPE module are given in column four of Table 3.6. The improved parameter estimates are very close to the true parameters used in the generation of the signal. The small deviations from the exact parameters are due to the synthetic noise contained in the signal. Therefore, the HPE module is able to obtain

accurate estimates of the model parameters.

The modelling error under the rectangular window is depicted in Figure 3.15 (a) by the dash-dotted line, and the log-amplitude spectrum of the modelling error under the Hann window is given in part (b) of this figure, again by the dash-dotted line. From the log-amplitude plot, we conclude that the HPE module removed the relevant part of the signal, and the residual approximates the spectrally-flat noise contained in the input signal. The signal-to-noise ratio in the approximation is 55.74 dB.

Example 3.7.2 (Real signal) A frame from the si01 (Harpsichord) excerpt is considered in this example. The waveform of the signal under the rectangular window is illustrated in Figure 3.16 (a) by the solid line, and the log-amplitude spectrum, under the Hann window and up to 11 kHz, is illustrated in part (b), again by the solid line. By visual inspection of part (a), we conclude that the waveform is periodic, and a large number of harmonic partials are distinguishable in part (b).

The DPE module provided an initial estimate of the fundamental frequency of 131 Hz, see column two of Table 3.7. The HPE module used zero as initial estimate

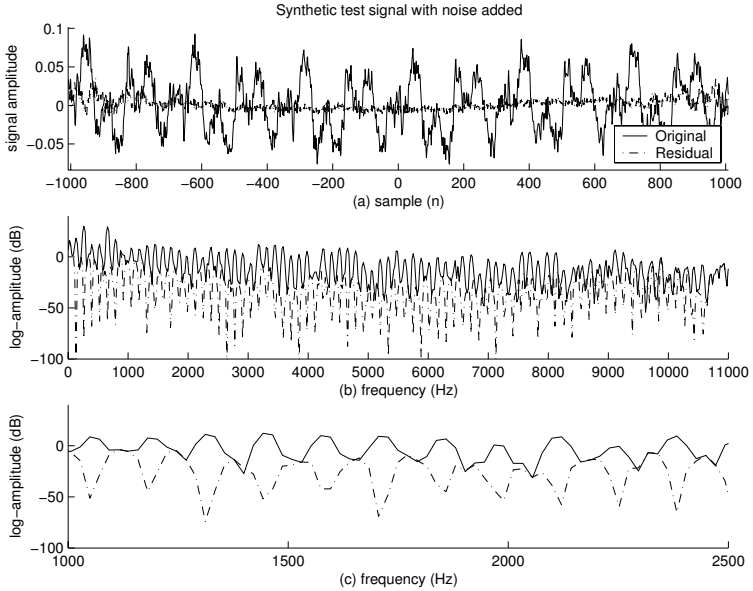


Figure 3.16: *Frame from the si01 (Harpsichord) excerpt. (a) Temporal waveform of the original (solid line) and the residual (dash-dotted line) under the rectangular window. (b) Log-amplitude spectrum of the original (solid line) and the residual (dash-dotted line) under the Hann window. The horizontal axis shows the frequency in Hz. (c) Zoomed-in spectrum in the range 1000 to 2500 Hz.*

Parameter	DPE	HPE
$\theta_{\mathtt{f}}$	131	131.90
$\theta_{\mathtt{c},\mathtt{s}}$		0.0081
B		0.000013
N_{h}		77

Table 3.7: *Model parameter estimates of a frame taken from the si01 (Harpsichord) excerpt. The initial estimate of the fundamental frequency is provided by the Detector and Pitch Estimator (DPE) module. The refined parameters are obtained by the Harmonic Parameter Estimation (HPE) module.*

of the frequency chirp and inharmonicity parameters. The final parameter estimates obtained by the HPE module are given in column three of Table 3.7. The estimate of the fundamental frequency was altered only slightly, while the measured frequency chirp and inharmonicity parameters are almost negligible. Furthermore, a large number of harmonics, namely $\hat{N}_{\mathrm{h}} = 77$, was estimated.

From the polynomial-phase parameter estimates, given in Table 3.7, the optimal parameters of a linear amplitude polynomial are determined. The signal-to-noise ratio in the approximation thus obtained is 16 dB under the Hann window. The modelling error, under the rectangular window, is depicted in Figure 3.16 (a) by the dash-dotted line, while its log-amplitude spectrum, under the Hann window, is given in part (b), again by the dash-dotted line. To consider the behaviour of the modelling error in the frequency domain more closely, we zoom in on the spectrum in part (c). We observe that the model is able to capture signal behaviour in the direct vicinity of the spectral peaks only (note the deep valleys in the residual at those frequencies where spectral peaks in the original occur), while the valleys between peaks in the original are not captured by our model (note that the residual has the same amplitude as the original at these frequencies).

Given the high number of harmonics detected and the signal-to-noise ratio obtained, we conclude that the HCA module was able to provide a satisfactory model of the frame considered in this example.

Example 3.7.3 (Time-frequency plot) We consider a time-frequency plot of the fundamental frequency over a number of frames taken from a voiced utterance in the es02 (male speech) excerpt. Figure 3.17 illustrates the time-frequency plot of the initial estimate of the fundamental frequency obtained from the DPE module. The temporal non-stationary character of speech is visible from the variation in the fundamental frequency over the frames.

We observe that the estimates of the fundamental frequency lie on a discrete grid. The non-uniform distance between adjacent levels on the grid of fundamental

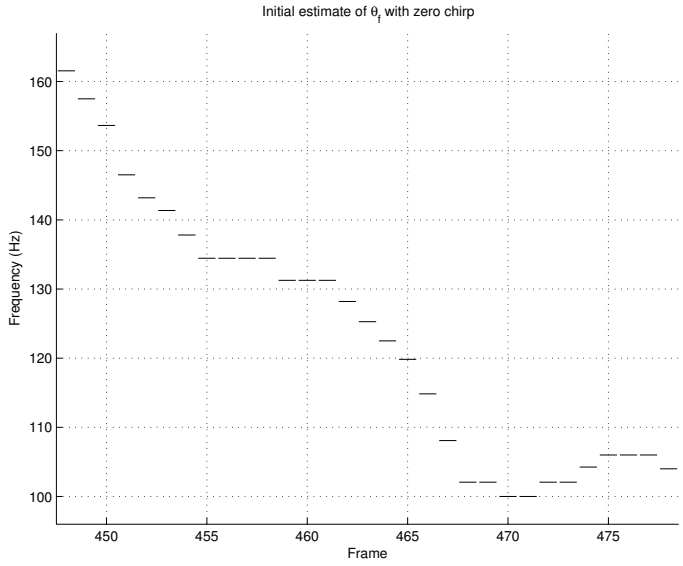


Figure 3.17: *Time-frequency plot of the fundamental frequency taken from the es02 (male speech) excerpt. The output of the DPE module is illustrated here.*

frequencies, afforded by the auto-correlation function, is determined by the sampling frequency (f_s) and the delay ($\tilde{n}$) at which the pitch-related peak is located:

$$\Delta_{\theta_f}(f_s;\tilde{n}) = \frac{f_s}{\tilde{n}} - \frac{f_s}{\tilde{n}+1} = \frac{f_s}{\tilde{n}(\tilde{n}+1)}.$$

As an illustration, we consider the initial estimates of the fundamental frequency in frames 469 and 470. The sampling frequency of time scale 5, on which the harmonic complex was detected, is

$$f_{ds} = \frac{44100}{12} = 3675 \text{ Hz},$$

see Table 3.3. The pitch-related peak in frame 469 is located at a lag of $\tilde{n}_{\max,469} = 36$ samples, and corresponds to an estimated fundamental frequency of 102.1 Hz, while the peak in frame 470 lies at a lag of $\tilde{n}_{\max,470} = 37$ samples, corresponding to an estimated fundamental frequency of 99.3 Hz.

The refined estimates of the fundamental frequency and frequency chirp, obtained from the HPE module, are illustrated in Figure 3.18. The refined estimates are a clear improvement over the initial estimates, since the transition in frequency over frame boundaries is much smoother than the transitions obtained from the initial estimates. From Figures 3.17 and 3.18 we conclude that the HPE module is able to improve the

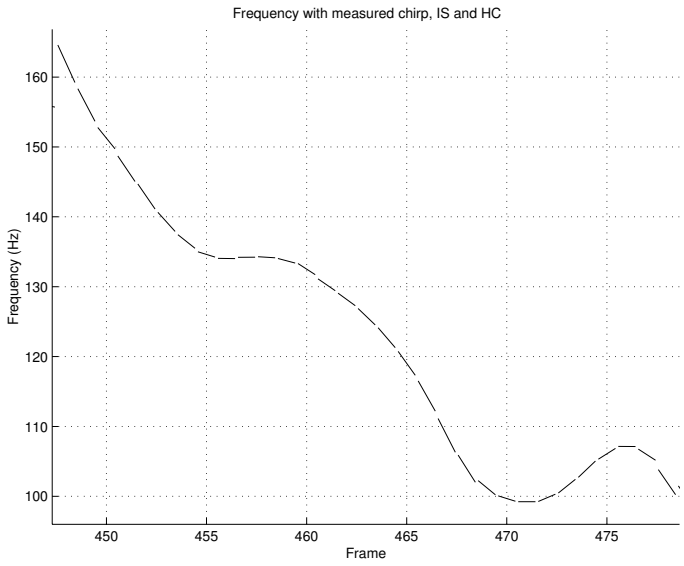


Figure 3.18: *Time-frequency plot of the fundamental frequency taken from the es02 (male speech) excerpt. The frequency chirp is illustrated as a slope.*

estimate of the fundamental frequency and obtain accurate estimates of the frequency chirp.

Informal listening experiments indicated that the decoded speech signal, obtained from the sinusoidal coder utilising the refined model parameters, sounded more natural than the decoded speech signal obtained by utilising the initial parameter estimates.

3.7.2 Individual sinusoids

The performance of the Individual Sinusoidal Analysis (ISA) module, see Figure 3.1 for a block diagram of this module, is illustrated by considering the time-frequency plot of a track over a number of frames from the sc03 (contemporary pop music) excerpt. Figure 3.19 illustrates the time-frequency plot of the initial estimates of θ_2 obtained from the IFE module. Observe that the transition in frequency from frame 170 to 171 is abrupt, as is the transition from frame 172 to 173. Furthermore, we notice that the initial estimates lie on the DFT frequency grid.

The refined frequency θ_2 and frequency chirp θ_3 estimates, obtained from the SPE module, are illustrated in Figure 3.20. Compared to the results illustrated in Figure 3.19, the transition in frequency over the frames is much smoother in this figure. In particular, the rapid rise in frequency from frame 162 to 171, and the

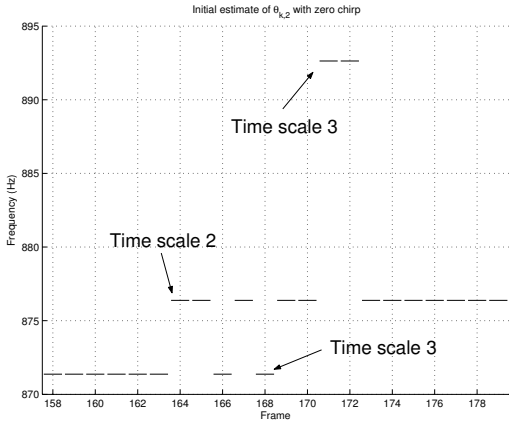


Figure 3.19: *Time-frequency plot of a track taken from the sc03 (contemporary pop music) excerpt. The frequency parameters are obtained from the IFE module. The time scales on which the spectral peaks are identified and from which the initial frequency parameters are obtained, are illustrated in this figure. Refer to Table 3.4 for more information regarding the time scales.*

following rapid decrease in frequency, are followed closely by the SPE module. We observe that the frequency behaviour of the partial in frames 171 and 172 appears to be quadratic, while our model only allows linear frequency variation in a frame.

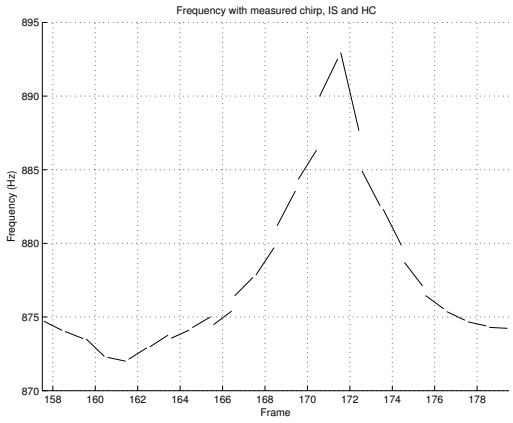


Figure 3.20: *Time-frequency plot of a track taken from the sc03 (contemporary pop music) excerpt. The frequency chirp is indicated as a slope. The frequency and frequency chirp parameters are obtained from the SPE module.*

Nevertheless, the SPE module is able to follow the quadratic frequency variation reasonably well with a piecewise linear approximation. Furthermore, it appears that the refined frequency estimates in frames 171 and 172 are too high when compared to neighbouring frames. The explanation for this is that the frequency estimates in these frames were obtained on a different time scale than the frequency estimates in neighbouring frames, see Figure 3.19 where the time scales are indicated. This apparent jump in frequency is avoided by forcing the SPE module to improve the initial estimates in frames 171 and 172 on the same time scale (namely time scale 2) as that utilised in frames 170 and 173. Similarly, the refined frequency estimate in frame 166 is brought closer to the estimates in neighbouring frames by forcing the SPE module to improve the initial estimate in this frame on time scale 2.

From Figures 3.19 and 3.20 we conclude that the SPE module is able to improve the initial frequency estimates provided by the IFE module, and to obtain accurate estimates of the frequency chirp.

Informal listening experiments indicated that the decoded signal, obtained from the sinusoidal decoder utilising the refined model parameters, sounded more natural than the decoded signal obtained by utilising the initial parameter estimates.

3.7.3 Entropy of sinusoidal parameters

The entropies of the sinusoidal parameters are given in Table 3.8. These entropies were determined from the complete sinusoidal component of all 12 excerpts comprising the test material. The test excerpts were given in Table 3.1. Observe that time-differential encoding results in a large coding gain, in particular for the fre-

Parameter	Levels ($\log_2$)	Entropy absolute	Entropy differential
$a_{k,1}$	6	5.35	3.48
$a_{k,2}$	4	2.68	
$\theta_{k,1}$	5	4.94	
N_{h}		4.47	2.60
θ_{f}	12	6.22	3.64
B	12	3.06	2.00
θ_{c}	5	1.88	
N_{c}		5.90	
$\theta_{k,2}$	9	8.55	1.46
$\theta_{k,3}$	5	1.48	

Table 3.8: *Entropy of sinusoidal parameters. The number of levels used to quantise the parameters is given in column two. The cost of absolute and time-differential encoding of quantisation indices are given in columns three and four, respectively. The amplitude sweep $a_{k,2}$, constant-phase $\theta_{k,1}$, frequency chirp θ_c and $\theta_{k,3}$, and number of individual sinusoids N_c are always absolutely encoded.*

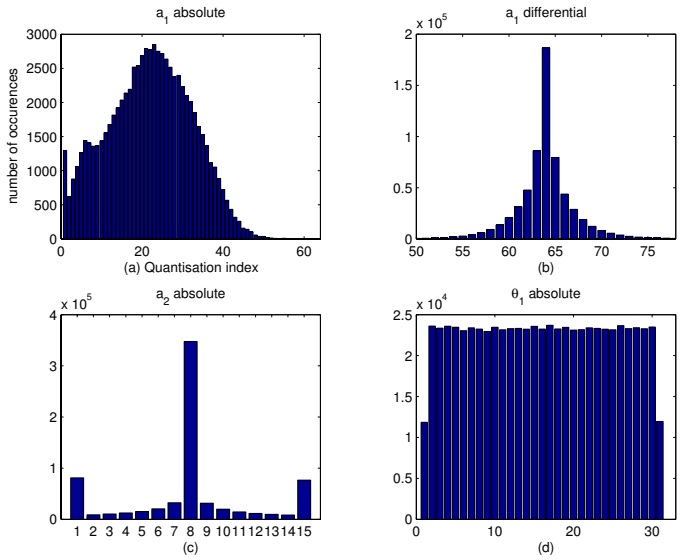


Figure 3.21: *Distribution of the amplitude parameters. (a) Absolute encoding of the constant amplitude $a_{k,1}$. (b) Differential encoding of the constant amplitude $a_{k,1}$. (c) Absolute encoding of the amplitude sweep $a_{k,2}$. (d) Absolute encoding of the constant phase $\theta_{k,1}$.*

quency parameters $\theta_{k,2}$. The distributions of the parameters in Table 3.8 are given in Figures 3.21 (amplitude parameters), 3.22 (individual-sinusoid parameters), and 3.23 (harmonic-complex parameters). A number of remarks concerning these distributions follow.

The distribution of the time-differentially encoded constant-amplitude parameters, illustrated in Figure 3.21 (b), is very peaked, resulting in a large coding gain. Several absolutely encoded amplitude-sweep parameters $a_{k,2}$ fall outside the allowable range specified by $a_{k,2,\texttt{max\%}} = 100\%$, see part (c). The phase parameters are uniformly distributed, see part (d). The distribution of the absolutely encoded frequency-chirp parameters, given in Figure 3.22 (b), is very peaked, substantiating our statement that the frequency chirps are usually small. Furthermore, the distribution of the time-differentially encoded frequency parameters, given in Figure 3.22 (d), is very peaked, implying that the frequency parameters of sinusoids on a track vary slowly from frame to frame. Time-differential coding of the number of harmonics results in a peaked distribution, as is evident from Figure 3.23 (b). The distribution of fundamental frequencies over the quantisation range, see part (c) of this figure, substantiates our claim that the range of fundamental frequencies [25, 1800] Hz is conservative. The time-differentially encoded fundamental frequencies, given in part (d), show a peaked distribution, similar to that of the time-differentially encoded frequency pa-

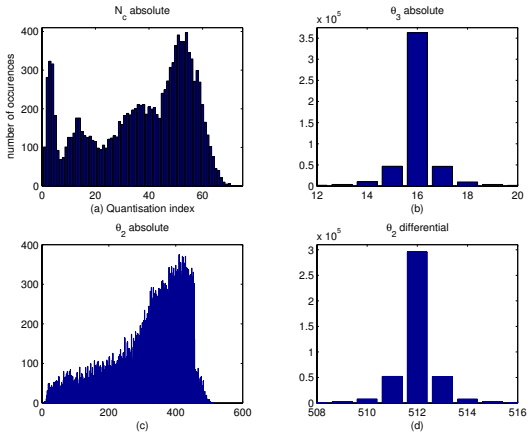


Figure 3.22: *Distribution of the individual-sinusoid parameters. (a) The number N_c of individual sinusoids per frame. (b) Absolute encoding of the frequency chirp $\theta_{k,3}$. (c) Absolute encoding of the frequency $\theta_{k,2}$. (d) Differential encoding of the frequency $\theta_{k,2}$.*

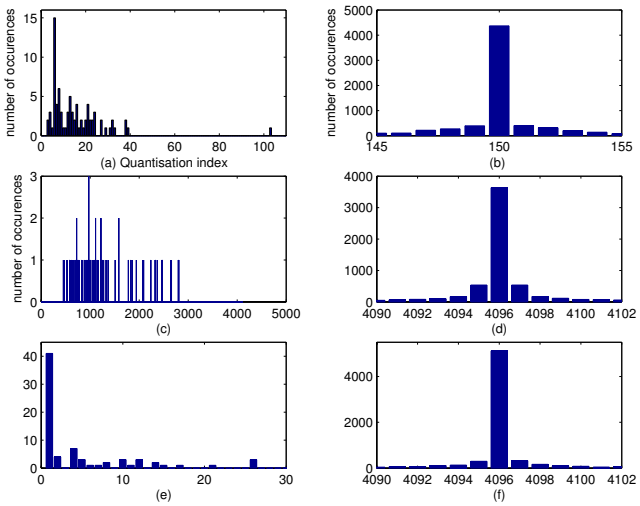


Figure 3.23: *Distribution of the harmonic-complex parameters. (a) Absolute encoding of number N_h of harmonics per frame. (b) Differential encoding of number N_h of harmonics per frame. (c) Absolute encoding of the fundamental frequency θ_f. (d) Differential encoding of the fundamental frequency θ_f. (e) Absolute encoding of the inharmonicity parameter B. (f) Differential encoding of the inharmonicity parameter B.*

rameters of individual sinusoids. As expected, the inharmonicity parameter assumes small values in most cases, see part (e), and time differential encoding of this parameter results in a very peaked distribution, meaning that the measured inharmonicity varies only slightly from frame to frame. The distribution of the absolutely encoded harmonic frequency-chirp parameters, not shown here, is very similar to that of the absolutely encoded frequency-chirp parameters of individual sinusoids.

3.8 Discussion

The parametric signal model utilised in the analysis does not include a transient signal component. During the development of the analysis process, transient coding by means of window switching was considered. This is described in Section 3.8.1. The limitations of the pitch detector are described in Section 3.8.2. The linking of individual sinusoids is considered in Section 3.8.3.

3.8.1 Transients

In the signal model, only the sinusoidal and noise components are considered, while no explicit transient component is utilised. Inspired by the approach taken in many transform coders, we considered window switching as an implicit way of modelling transients, as illustrated in Figure 3.24. When a transient is detected in a frame, window switching is applied. In the frame preceding the transient frame, a transition window is utilised. The transient frame is divided into a number of sub-frames, and a transition window is utilised in the following frame. Both the sinusoidal amplitude parameters and Laguerre prediction coefficients in the frame preceding the transient frame are re-calculated under the transition window, and the amplitude parameters and prediction coefficients in the transient frame are determined under each short window. The short windows utilised in the transient frame are identical to those utilised in measuring the temporal envelope.

In our approach, the estimation of non-constant polynomial-phase parameters of both individual sinusoids and harmonics remains unchanged in a transient frame. Keeping the estimation unchanged is a topic for further investigation, since the spectral image of partials in a transient frame is very degraded, hampering both the initial estimation and improvement of these parameters. Knowing the position of a transient onset will allow a suitable adaptation of analysis windows. Several techniques for estimating the exact position of the transient onset within the frame have been described in literature, see e.g. [76, 19, 119, 120]. Alternatively, explicit models of the transient component, see e.g. [12, 21, 19], can be combined with the sinusoidal and noise coders described in this chapter.

Informal listening experiments revealed that the utilisation of window switching

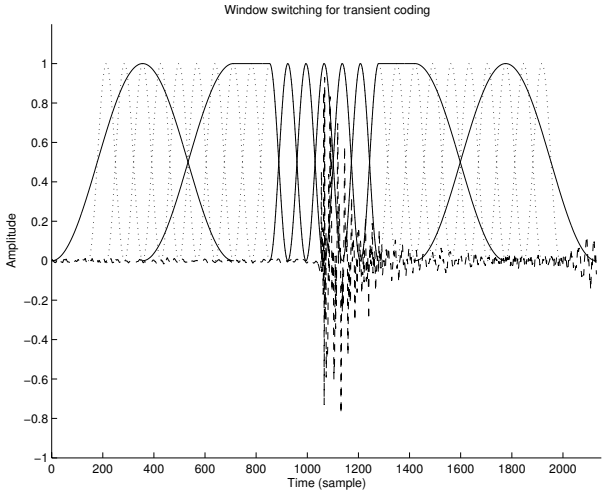


Figure 3.24: *Window switching for transient coding. Both the sinusoidal amplitude parameters and prediction coefficients are determined under each window denoted by a solid line. The windows used for the calculation of the short-time energy of the prediction residual remain unchanged, and are denoted by dotted lines. A real-world transient, taken from the castanet fragment (si02) and denoted by dashed lines, is included to complete the illustration.*

does not lead to a significant improvement in audio quality for critical excerpts like castanets (si02). For this reason, and in the interest of keeping the complexity of the encoder in check, no coding of transients is applied in the prototype coder.

3.8.2 Pitch detection

Although the pitch detector utilised in the sinusoidal coder proved to be reliable in practice and to be well-suited for our purposes, it does have a number of limitations. Firstly, multiple harmonic sounds in a frame are difficult to detect with a time-domain auto-correlation method. Secondly, the number of time scales (twelve) required to span the range of possible fundamental frequencies hampers the computational efficiency of the pitch detector. Lastly, the number of periods required (between eight and twelve) by the auto-correlation function diminishes our attempt to limit the degrading effect of temporal non-stationarity on the pitch-related peak in the auto-correlation function. Spectrally-based methods, like those utilising the Cepstrum [121], may allow more efficient detection of multiple harmonic complexes in a frame.

3.8.3 Parameter linking

The criterion for the linking of individual sinusoids should be extended to include amplitude and phase matching in addition to the frequency matching currently applied. This more thorough assessment of parameter matching should improve the linking of sinusoids in adjacent frames that are modelling the same partial.

3.9 Summary

The parametric signal model described in this chapter consists of a sinusoidal and a noise component. The sinusoidal component was divided into a harmonic-complex sub-component and an individual-sinusoids sub-component, and suitable models for both sub-components were defined. The model of individual sinusoids allows for temporal non-stationarity in both amplitude and frequency. In addition to temporal non-stationarity, the model of the harmonic complex includes inharmonicity encountered in stiff-stringed musical instruments. The presence of individual sinusoids was detected by identifying spectral peaks. For each sinusoid thus identified, the perceptual distortion was estimated, and the estimate was scaled to match the parameter update rate. Identifying the presence of a harmonic complex was based on detecting periodicity in the audio signal by means of the auto-correlation function. An optimisation procedure, based on Levenberg-Marquardt optimisation, was developed to obtain accurate estimates of the sinusoidal-model parameters. Analysis of individual sinusoids was carried out on multiple time scales, while the harmonic-complex parameters were estimated on a pitch-synchronous time scale. Individual sinusoids in adjacent frames were linked to obtain tracks. Linking was based on matching instantaneous frequencies at frame boundaries, where the frequency chirp parameters were utilised. Similarly, harmonic complexes in adjacent frames were linked to obtain harmonic-complex trajectories. The quantisation of sinusoidal parameters was carried out according to psycho-acoustical principles. Both absolute and time-differential encoding of sinusoidal parameters were applied. The performance of the analysis procedure was illustrated by considering both synthetic and real-world signals, and parameter statistics of the sinusoidal component were given.

The advantages of this sinusoidal coder over existing sinusoidal coders include the following. Flexible models of the harmonic complex and individual sinusoids allow the description of non-stationary signal behaviour, both in amplitude and in frequency. The analysis process provides accurate estimates of the model parameters, resulting in improved linking of sinusoids (or harmonics) over time. Furthermore, informal listening experiments indicated that decoded audio, obtained from the sinusoidal coder utilising the iterative optimisation of model parameters, sounded more natural than decoded audio obtained by utilising the initial parameter estimates. The utilisation of a harmonic complex allows a low-cost parameterisation of harmonic

sounds. This is advantageous when a high audio quality is required at low bit-rates.

Only the spectral and temporal envelopes of the residual are modelled by the noise coder, while the fine-structure of its waveform is not taken into account. The spectral envelope is modelled by Laguerre-based Pure Linear Prediction, where the Laguerre pole was chosen such that the spectral resolution thus obtained matched that of the human auditory system. The resolution with which the temporal envelope of the prediction residual is measured, is matched to the temporal resolution of the human auditory system. The quantisation of noise-model parameters is carried out according to psycho-acoustical principles.

3.10 Conclusion

This chapter describes the design of an audio coder and decoder, and provides the necessary design specifications to implement the codec. In the design of the analysis process, the need for a flexible model of the sinusoidal component, and an algorithm to obtain accurate estimates of the model parameters, as highlighted in Chapter 2, was addressed. In Chapter 4, the audio codec will be utilised to design a bit-rate scalable codec.

Chapter 4

Bit-Rate Scalability

4.1 Introduction

Bit-rate scalability is a desirable feature of an audio coder operating in a dynamic environment. A bit-rate scalable audio coder produces a bit stream containing a number of layers which can be removed to lower the bit rate. There are roughly two types of scalable bit streams. A *dependent-layer structure* contains one base layer and a number of enhancement or refinement layers, while the layers in an *independent-layer structure* are not mutually dependent.

In this chapter, we discuss the design of a bit-rate scalable parametric audio coder with a dependent-layer structure. In a parametric audio coder, a well-tuned trade-off between tones (or sinusoids) and noise in the decoded signal is required in order to obtain a high audio quality. In Section 2.3.3, we proposed the framework of a bit-rate scalable encoder and decoder which caters for this kind of well-tuned trade-off. In this chapter, concrete proposals for realising the bit-rate scalable encoder and decoder are proposed. The sinusoidal coder (SC), noise coder (NC), sinusoidal decoder (SD), and noise decoder (ND) modules, discussed in Chapter 3, are utilised to realise the bit-rate scalable coder. The target range of bit rates over which the scalable coder should operate is 10 – 40 kbits/s. When layers are removed from the bit stream to lower the bit rate, the resulting degradation in audio quality should be graceful. The results obtained in a listening test, conducted to validate the performance of the bit-rate scalable parametric audio coder, are given in Chapter 5.

In the design of the bit-rate scalable codec, we will pay attention to both the encoder and decoder in Sections 4.2 and 4.3, respectively. In Section 4.2, relevant existing approaches to bit-rate scalability are described. We will combine a number of these approaches with a rate-distortion optimisation mechanism to design a bit-rate scalable encoder. A suitable bit-stream syntax is proposed. The feature of the decoder which allows it to operate in bit-rate scalable mode, is matching the noise component to the sinusoidal component contained in the layers. This is the main topic of Section 4.3. Design specifications for both the encoder and decoder are given in Section 4.4, and relevant statistics describing the layers are given in Section 4.5.

The sinusoidal and noise parameters will be referred to frequently in this chapter, and are therefore summarised in Table 4.1.

Component	Parameter	Description
Noise	α	prediction coefficient
	$\alpha_{\mathtt{LAR}}$	prediction coefficient in LAR domain
	$O_{\mathtt{p}}$	prediction order
	E	temporal envelope of prediction residual
	$E_{\mathtt{dB}}$	temporal envelope on a dB scale
Harmonic complex	$\theta_{\mathtt{f}}$	fundamental frequency
	$\theta_{\mathtt{c}}$	frequency chirp
	B	inharmonicity
	$N_{\mathtt{h}}$	number of harmonics
Individual sinusoids	θ_2	frequency
	θ_3	frequency chirp
	D	perceptual distortion
	$N_{\mathtt{c}}$	number of sinusoids
Collective amplitudes	a_1	amplitude
	a_2	amplitude sweep
	θ_1	phase

Table 4.1: *Summary of the model parameters.*

4.2 Encoder

Our proposal for the overall structure of a bit-rate scalable encoder is illustrated in Figure 4.1. In the encoder, the SC is applied to the input audio signal $s[n]$ to obtain the sinusoidal-component parameters. Bit-rate scalability is achieved by the Bit-Rate Scalability (BRS) module, which distributes the sinusoidal parameters over the base and refinement layers in such a way, that the specified target bit rate of each layer is satisfied, and the highest possible audio quality for each layer is obtained. We denote the sinusoidal signal-component corresponding to the base layer by $\hat{s}_{\mathtt{sinusoid,bl}}[n]$. This signal is generated by the Base Layer Synthesis (BLS) module from the unquantised model parameters on a per-frame basis, and by applying overlap-add. The NC is then applied to the residual signal

$$s_{\mathtt{residual,bl}}[n] = s[n] - \hat{s}_{\mathtt{sinusoid,bl}}[n], \tag{4.1}$$

and the parameters generated by NC are placed in the base layer. Finally, the model parameters are multiplexed (MUX) in a scalable bit stream.

In contrast to the approaches taken in other scalable parametric audio coders, our proposal is based on making a dual description of part of the audio signal. When

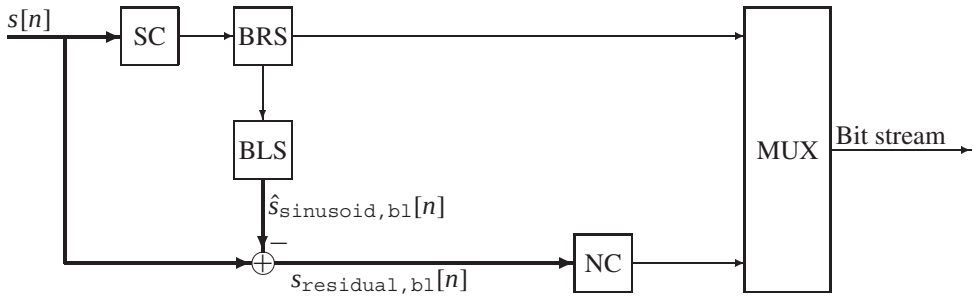


Figure 4.1: *The bit-rate scalable encoder consists of the Sinusoidal Coder (SC), Bit-Rate Scalability (BRS) module, Base Layer Synthesis (BLS) module, Noise Coder (NC), and bit-stream multiplexer (MUX).*

layers are removed from the scalable bit stream, the decoder will exploit this dual description to adapt the noise component in such a way that a well-tuned trade-off between sinusoids and noise is realised.

The primary purpose of this section is to describe the function of the BRS module. In Section 4.2.1, a number of existing approaches for lowering the bit rate, suitable for parametric audio coders, is discussed. These approaches are combined to scale the sinusoidal component in Section 4.2.2. In Section 4.2.3, we provide a substantiated proposal for creating a scalable bit stream.

4.2.1 Existing strategies for scaling the sinusoidal component

To obtain the highest possible quality with a sinusoidal coder operating at very low bit rates, as many sinusoids as possible should be coded in the bit stream. When too few sinusoids are present, the noise component will dominate and the decoded audio will not sound natural. Therefore, methods for lowering the encoding cost (in bits) of sinusoids are desirable. However, lowering the encoding costs by discarding information, leads to describing the parameters with reduced accuracy. To avoid audible artefacts, the sinusoidal parameters should be described with enough accuracy. As a result, a balance has to be found. In literature, several strategies are proposed to substantially lower the bit rate required by the sinusoidal component, while maintaining high audio quality; these are described in the following.

Phase continuation

Very little coding gain is achieved by applying time-differential coding of the phase parameters θ_1. From the entropy information given in Table 3.8 on page 98, we observe that the phase parameters are responsible for a large portion of the number of bits required to represent the sinusoidal parameters: 21% of the bits are needed for the start of a track, and 35% of the bits are needed for linking a track. These percentages do not include the side information, such as the number of individual sinusoids N_c, required to construct a meaningful bit stream. By transmitting phase parameters at the start of a track and for starting harmonics only, and by applying phase continuation in the decoder to estimate the lacking phase parameters of linked sinusoids, according to (3.73) on page 89, slow phase drifting is introduced. In this approach, the relatively low sensitivity of the human auditory system to slow phase drifting is exploited. This or a similar approach is taken in many parametric audio coders, see e.g. [12, 11, 19, 18]. While the penalty that is paid in terms of quality loss due to phase continuation in general audio is minor, speech quality degrades substantially. Phase continuation, in particular, often results in artificial, metallic-sounding speech.

Recently, a new technique for efficient phase transmission was proposed by den Brinker et al. [122]. This technique is based on coding the unwrapped phase of sinusoids on a track. The frequency parameters of linked sinusoids need not be transmitted in this approach, since they are derived in the decoder by differentiating the unwrapped phase. This technique is not considered further in our approach.

No transmission of amplitude sweep and frequency chirp parameters

When the phase parameters are transmitted only at the start of a track and for starting harmonics, the amplitude-sweep a_2 and frequency-chirp parameters θ_3 combined, consume almost 19% of the bits required to start up a track, while for linked sinusoids, these parameters consume 47% of the bit rate. These percentages do not include the required side information. In order to transmit as many sinusoids as possible at low bit rates, these parameters should only be transmitted at high bit rates. At low bit rates, the individual-sinusoid sub-component then reduces to the basic constant-amplitude and constant-frequency version utilised in many parametric audio coders [25, 54, 20, 75, 12]. Furthermore, the amplitude-sweep parameters of linked harmonics are not transmitted, and the harmonic-complex sub-component in the decoder thus becomes a constant-amplitude, linear-frequency model. (We recall that θ_3 plays an important role in the linking of individual sinusoids and that both a_2 and θ_3 aid in reducing the power of the residual.)

Low-pass filtering of amplitude and frequency parameters

To lower the cost of time-differential encoding of amplitude a_1 and frequency parameters θ_2 (and θ_f), these parameters are low-pass filtered before they are quantised and coded [18]. Since sinusoids on a track, or harmonics on a harmonic trajectory, model a stationary partial in an audio signal, low-pass filtering these parameters is not expected to result in a large degradation in audio quality. For non-stationary signals, like speech, the reduced temporal resolution, resulting from low-pass filtering of these parameters, will degrade the audio quality of the decoded signal.

Parameter quantisation

The minimum resolution with which sinusoidal parameters should be quantised is derived from data about just-noticeable differences, obtained in the field of psychoacoustics, as we noted in Section 3.5.1 on page 83. The quantisation resolution proposed in that section is near the lower end of the allowable range; lowering the quantisation resolution further will lead to annoying artefacts, and is therefore not an option. For long, steady tones, the current quantisation resolution should even be increased to maintain high audio quality [12].

Parameter interpolation

Frame removal and interpolation in the decoder, to lower the bit rate or to recover from frame erasures, has been applied in a number of (scalable) audio and speech coders. In scalable CELP coding of speech for example, prediction coefficients are not transmitted in every frame; instead, the missing prediction coefficients are estimated in the decoder by interpolating prediction coefficients from adjacent frames [123, 124]. In this way, the effective update rate of prediction coefficients is reduced. The effective update rate of sinusoidal parameters can be lowered in a similar way. Since the purpose of sinusoidal tracks is to model stationary partials contained in an audio signal, lowering the update rate of sinusoidal parameters on a track is an attractive possibility, and has been considered in parametric audio coding [18, 35].

In the approach of Levine, the amplitude a_1 and frequency parameters θ_2 of sinusoids on selected tracks were low-pass filtered, before down-sampling (by a factor of two) was applied to these parameters [18]. The lacking amplitude and frequency parameters were estimated in the decoder by up-sampling and low-pass filtering. In the approach of Myburg, the amplitude and frequency parameters of sinusoids on all tracks were transmitted at a variable rate which depended on the target bit rate [35]. The lacking parameters were estimated in the decoder by interpolation. We note that this strategy should only be applied when the update rate of model parameters is high enough to maintain a sufficiently high temporal resolution. Furthermore, it was

shown in [35] that this approach yields a graceful degradation in audio quality, and allows for a large reduction in bit rate.

Removing complete tracks

A common problem in parametric audio coding is the disproportionately high number of short tracks. In particular, histograms of track length exhibit an ostensibly exponential distribution [18, 35]. This observation stands in contrast to the purpose of tracks: to model stationary, tonal aspects of an audio signal. The mentioned problem stems from a core assumption made in parametric audio coding: an audio signal can be represented completely by tones and noise. Therefore, in practice, the sinusoidal coder often models signal components that fall in the tonal-noise transition region, as illustrated in Figure 2.5 on page 24.

An undesirable side-effect of many short tracks is an increase in bit rate, since starting up a track is much more expensive than continuing a track. It is therefore desirable to have a mechanism that selects only those tracks that do indeed model tonal aspects of an audio signal. Resulting selection criteria have most often been based on the relation between a sinusoid and the masked threshold, also called the signal-to-mask ratio (SMR in dB) [20, 21, 27, 25]. The higher the SMR, the more relevant a sinusoid is considered to be, and the higher the probability that it will not be discarded. In addition to the SMR, the duration of a track is an important consideration in determining whether a track is serving its intended purpose. A track-selection criterion, based on both the track duration and the time-averaged SMR of sinusoids on the track, was utilised by Levine [18]. In the bit-rate scalable coders described in [27, 25], only the SMR of sinusoids is used as track-selection criterion; the duration of sinusoidal tracks is not taken into account. In the next section, we propose a new, psycho-acoustically motivated measure for the perceptual relevance of a complete track.

4.2.2 Proposed strategy for scaling the sinusoidal component

Here, we describe how the sinusoidal component is partitioned into layers by the BRS module. We denote the total number of layers contained in the bit stream by $N_{\texttt{layers}}$, and the target bit rate for layer l by $R_{\texttt{target},l}$ kbits/s. Since the base layer contains both the noise component and part of the sinusoidal component, $R_{\texttt{target},1}$ can be written as $R_{\texttt{target},1} = R_{\texttt{noise}} + R_{\texttt{sin},1}$, where $R_{\texttt{noise}}$ is the bit rate of the noise component and $R_{\texttt{sin},1}$ the bit rate of the sinusoidal component. The refinement layers will contain only sinusoids, therefore, $R_{\texttt{target},l}$ can be written as $R_{\texttt{target},l} = R_{\texttt{sin},l} = R_{\texttt{hc},l} + R_{\texttt{is},l}$, where $R_{\texttt{hc},l}$ and $R_{\texttt{is},l}$ are the target bit rates of the harmonic-complex and individual-sinusoid sub-components in layer $l \geq 2$, respectively. We assume that $R_{\texttt{noise}}$ is constant, so that only the sinusoidal component needs to be

scaled.

In the following, we describe how the cost of the harmonic-complex and individual-sinusoid sub-components is lowered without removing tracks. If the cost of the sinusoidal component is still too high after these operations, complete tracks are selected from lower layers and moved to higher layers by a rate-distortion optimisation mechanism. The optimisation mechanism needs to know both the rate and (perceptual) distortion of a track. We propose a new method to estimate the perceptual distortion of a track.

Lowering the cost of the harmonic complex

A number of strategies from Section 4.2.1 are utilised to reduce the cost of the harmonic-complex sub-component, if present. The strategies utilised are:

1. The fundamental frequency $\theta_{\mathtt{f}}$, frequency chirp $\theta_{\mathtt{c}}$, inharmonicity parameter B, and the total number of harmonics N_{h}, are transmitted in the base layer. We do not utilise removal of $\theta_{\mathtt{f}}$, $\theta_{\mathtt{c}}$, and B in the encoder and estimation in the decoder by interpolation, for two reasons. The first reason is that encoding these parameters is relatively cheap. The second reason is that the temporal non-stationary character of voiced speech, in particular, necessitates a sufficiently high update rate of these parameters to maintain natural-sounding speech.

2. The un-quantised amplitude parameters a_1 of harmonics are low-pass filtered and quantised.

3. A distinction is made between starting harmonics and linked harmonics, analogous to the distinction made between starting and linked individual sinusoids on a track. The amplitude parameters a_1 of both starting and linked harmonics are transmitted, while the phase θ_1 parameters are only transmitted for starting harmonics. The decoder applies phase continuation, as described in Section 3.6 on page 89, to estimate the missing phase parameters.

4. No amplitude-sweep parameters a_2 are transmitted.

5. When a high number of harmonics are starting up in a frame, the bit rate required by the amplitude a_1 and phase θ_1 parameters may not fit into the budget of the base layer. Therefore, we choose to distribute the harmonic amplitude and phase parameters over the layers. In the interest of simplicity, we distribute the parameters in the following, pre-determined fashion. The number $N_{\mathrm{h},\mathtt{max},l}$ specifies the maximum (cumulative) number of amplitude and phase parameters in layer l. Then, layer $l = 1$ contains $a_{1,1}, \ldots, a_{N_{\mathrm{h},\mathtt{max},1},1}$ and $\theta_{1,1}, \ldots, \theta_{N_{\mathrm{h},\mathtt{max},1},1}$, layer $l = 2$ contains $a_{N_{\mathrm{h},\mathtt{max},1}+1,1}, \ldots, a_{N_{\mathrm{h},\mathtt{max},2},1}$ and $\theta_{N_{\mathrm{h},\mathtt{max},1}+1,1}, \ldots, \theta_{N_{\mathrm{h},\mathtt{max},2},1}$, etc. Only the first $N_{\mathrm{h},\mathtt{max},1}$ harmonics, belonging

to the base layer, are synthesised by the BLS module to generate the sinusoidal component, refer to Figure 4.1. The remaining harmonic partials are therefore contained in the residual and will be modelled by the noise coder. When not all layers are contained in the bit stream received by the decoder, the missing amplitudes of starting and linked harmonics are estimated from the spectral envelope of the noise component, see Section 4.3.1, and random phases are assigned to starting harmonics.

6. To reduce the cost of the harmonic complex further, the amplitude parameters a_1 of linked harmonics are not transmitted in every frame. This is signalled in the bit stream, and the decoder estimates the missing amplitude parameters by interpolation.

No further operations are applied to the harmonic complex in the base or refinement layers to lower the bit rate. The cost of the harmonic complex in layer l, after these operations have been applied, is denoted by $R_{\mathrm{hc},l}$ kbits/s.

Lowering the cost of individual sinusoids

A number of strategies from Section 4.2.1 are utilised to reduce the cost of tracks. The strategies utilised are:

1. The un-quantised amplitude parameters a_1 of sinusoids on a track are low-pass filtered and quantised. Low-pass filtering of the frequency parameters θ_2 on a track is not applied, since the cost of time-differential encoding is already low: 1.46 bits per parameter, see Table 3.8 on page 98.

2. Similar to the harmonic complex, only amplitude a_1, frequency θ_2 and phase parameters θ_1 for starting individual sinusoids are transmitted. Phase parameters of linked individual sinusoids are not transmitted, and the decoder applies phase continuation to estimate these parameters.

3. When the amplitude and frequency parameters of linked sinusoids are not transmitted in a frame, this is signalled in the bit stream, and the missing parameters are estimated by interpolation in the decoder.

We denote the cost of the individual-sinusoid sub-component, after these operations have been applied, by $R_{\mathtt{is}}$ kbits/s.

Perceptual distortion of a complete track

The rate-distortion optimisation mechanism needs to know the rate (in kbits/s) and (perceptual) distortion of each track. While obtaining the rate is a matter of counting

bits, estimating the perceptual distortion of a track requires closer inspection. To determine the perceptual distortion of a complete track, one has to take the ability of the human auditory system to integrate acoustical information over time into account. The temporal integration time is approximately 300 ms [69]. In the following, we make a distinction between tracks having a duration less than or equal to 300 ms and tracks having a duration in excess of 300 ms.

When a track has a briefer duration than 300 ms or 37 frames, given that $t_{\mathrm{UR}} \approx$ 8 ms, the distortion parameters D of individual sinusoids on the track are added to approximate the perceptual distortion that will be introduced when the track is not included in the decoded signal. We denote the integrated distortion associated with the complete track by $D_{\mathtt{integrated}}$, and we assume that $D_{\mathtt{integrated}}$ is a measure of the perceptual relevance of the track.

When a track has a duration in excess of 37 frames, a sliding integration window, spanning 37 frames, is applied to obtain the (time-dependent) perceptual relevance $D_{\mathtt{integrated}}$. Figure 4.2 provides an illustration of the progression of $D_{\mathtt{integrated}}$ by considering a track taken from the coded sm01 (bagpipes) excerpt. From this figure, we observe that $D_{\mathtt{integrated}}$ at the start of the track is substantially lower than near the end. In this case, the value of $D_{\mathtt{integrated}}$ at the start of the track would be an inaccurate parameterisation of the perceptual relevance of the complete track. Therefore, to reasonably parameterise the perceptual relevance of a complete track

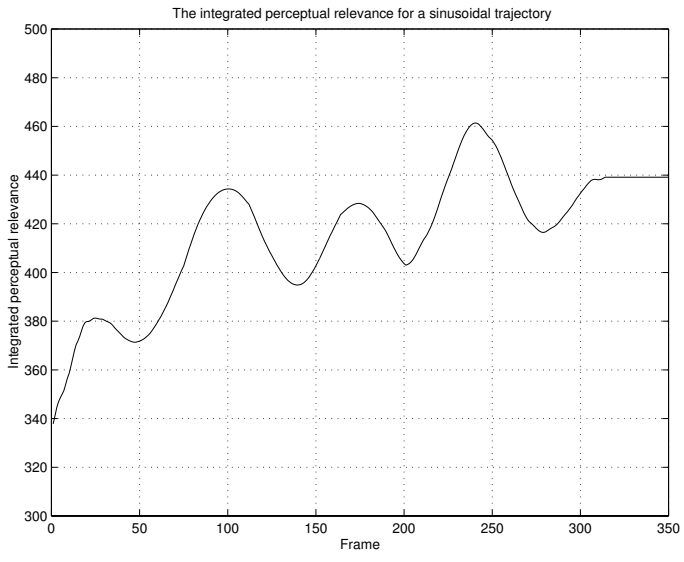


Figure 4.2: *The integrated distortion $D_{\mathtt{integrated}}$ of a long track in the coded sm01 (bagpipes) excerpt.*

by a single parameter, the possibility of a fluctuating $D_{\texttt{integrated}}$ has to be taken into account. A safe approach would be to assign the maximum value attained by $D_{\texttt{integrated}}$ to the complete track. Alternatively, the average value of $D_{\texttt{integrated}}$ can be assigned to the complete track. The main disadvantage of these approaches is that the resulting $D_{\texttt{integrated}}$ is not necessarily an accurate estimation of the time-localised perceptual relevance of the track. Furthermore, having to consider the complete track is a handicap when coding delay plays a role. To address the problem of (substantial) fluctuations in $D_{\texttt{integrated}}$ while providing a way of limiting the coding delay, we propose to re-start long tracks every 37 frames. The disadvantage is an increase in bit rate, since starting up a track is much more expensive than continuing a track, as we have observed. However, as we will show in Section 4.5, this increase in bit rate is minor since a relatively small number of tracks are long enough to be re-started. The advantage of this approach is that the value of $D_{\texttt{integrated}}$, obtained after re-starting long tracks, is uniquely determined in this approach. To avoid phase discontinuity in the decoder, the continued phase is transmitted instead of the original phase whenever a track is interrupted in this manner.

Rate-distortion optimisation

The rate-distortion optimisation mechanism selects those tracks that result in the highest possible audio quality while adhering to the constraint posed on the bit rate. The base layer is created first, after which the first refinement layer is created, etc. We denote the complete rate, including linking overhead, of a track k starting in frame i by $r_k^{(i)}$ kbits/s and its integrated distortion by $D_{k,\texttt{integrated}}^{(i)}$. Therefore, removing this track will result in a distortion $D_{k,\texttt{integrated}}^{(i)}$ and zero rate, while keeping the track will result in a rate $r_k^{(i)}$ and zero distortion. The desired selection of tracks to remove in order to satisfy the target bit rate is such that the smallest total distortion is introduced. The target bit rate for the individual-sinusoid sub-component in layer l is denoted by $R_{\texttt{is},l}$ kbits/s, where

$$R_{\texttt{is},l} = \begin{cases} R_{\texttt{target},l} - R_{\texttt{hc},l} & \text{if } l > 1 \\ R_{\texttt{target},l} - (R_{\texttt{hc},l} + R_{\texttt{noise}}) & \text{if } l = 1. \end{cases} \tag{4.2}$$

The resulting budget-constrained combinatorial optimisation problem is now formulated.

Problem 4.2.1 We have to find the selection of tracks over all frames for which

$$\sum_{i,k} b_k^{(i)} r_k^{(i)} \leq R_{\texttt{is},l}, \tag{4.3}$$

where the binary numbers $b_k^{(i)}$, indicating whether a track is kept or moved to a higher layer

$$b_k^{(i)} = \begin{cases} 1 & \text{when the track is kept} \\ 0 & \text{when the track is moved to a higher layer,} \end{cases} \tag{4.4}$$

are chosen in such a way that the total distortion

$$\sum_{i,k} (1 - b_k^{(i)}) D_{k,\texttt{integrated}}^{(i)} \tag{4.5}$$

is minimised.

The distortions are added in (4.5) as an approximation of the human ability to integrate distortions over a wide range of frequencies [67, 68]. Solving this optimisation problem by evaluating all possibilities results in an algorithm with non-polynomial complexity, and this approach is therefore not practically feasible.

The practical solution of this problem is based on a heuristic, a discrete version of Lagrangian optimisation [125]. Denote the Lagrange multiplier by $\lambda_l \geq 0$, and the Lagrangian cost of track k starting in frame i by

$$J_k^{(i)}(\lambda_l) = (1 - b_k^{(i)}) D_{k,\texttt{integrated}}^{(i)} + \lambda_l b_k^{(i)} r_k^{(i)}. \tag{4.6}$$

The set $\{\hat{b}_k^{(i)}\}$ which minimises $\sum_{i,k} J_k^{(i)}(\lambda_l)$ for a given λ_l, is the optimal solution to a budget-constrained problem with rate

$$R(\lambda_l) = \sum_{i,k} \hat{b}_k^{(i)} r_k^{(i)}. \tag{4.7}$$

The aim is to find that $\lambda_l = \lambda_{R_{\texttt{is},l}}$ for which $R(\lambda_{R_{\texttt{is},l}}) \approx R_{\texttt{is},l}$. The convexity of the rate-distortion curve simplifies the search for $\lambda_{R_{\texttt{is},l}}$. We explain our approach to finding $\lambda_{R_{\texttt{is},l}}$ in the design specifications in Section 4.4. Since the rate and perceptual distortion of each track were measured independently, the minimisation can be carried out independently:

$$\min_{\{b_k^{(i)}\}} \left(\sum_{i,k} J_k^{(i)}(\lambda_l) \right) = \sum_{i,k} \min_{b_k^{(i)}} \left((1 - b_k^{(i)}) D_{k,\texttt{integrated}}^{(i)} + \lambda_l b_k^{(i)} r_k^{(i)} \right), \tag{4.8}$$

resulting in an algorithm with polynomial complexity, which is practically feasible.

4.2.3 Proposed bit-stream syntax

After the contents of the layers are determined by the BRS module in the manner described above, the bit-stream multiplexer (the MUX module in Figure 4.1) creates

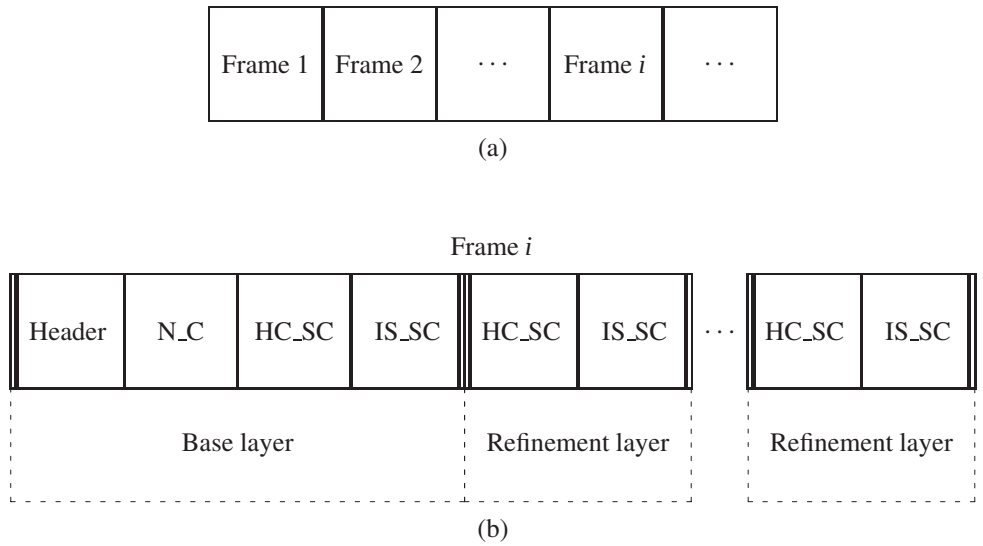


Figure 4.3: *(a) The bit-stream contains units filled with information from each frame. (b) The framework of the bit-stream syntax for a single frame. The frame contains information from the base layer and all refinement layers. See the text for more detail.*

the layered bit stream. In this section, we propose the framework of a suitable bit-stream syntax. Our developed prototype does not support the generation of an actual bit stream. We note, however, that this framework provides useful insight into the overhead required to construct a meaningful bit stream.

In the interest of simplicity, we state that the bit stream contains units filled with information from single frames, see Figure 4.3 (a). Alternatively, frames can be grouped into larger meta frames to form units of the bit stream. The utilisation of such meta frames leads to higher coding efficiency since less overhead is required [12]. The influence of meta frames on the bit rate is not investigated further.

We now consider the bit-stream syntax for a single frame. The information comprising a frame consists of the base layer and all refinement layers. Figure 4.3 (b) illustrates the build-up of a single frame, where the base layer contains the frame header, the complete noise component (N_C), a part of the harmonic-complex sub component (HC_SC), and a number of tracks from the individual-sinusoid sub component (IS_SC). The header contains the frame number, the number of refinement layers and pointers to each refinement layer. When layers are removed from the bit stream, the header should be adapted. The refinement layers contain additional parameters of the harmonic complex and tracks.

In the following, the syntax for the noise component (N_C), harmonic-complex sub component (HC_SC), and individual-sinusoid sub component (IS_SC) are given.

Noise component

The parameters of the noise component, contained in the base layer, are given in Figure 4.4. A refresh frame starts with a 0, followed by O_p absolutely-encoded LAR coefficients $\alpha_\mathtt{LAR}$ representing the prediction coefficients α. The first temporal-envelope measurement $E_\mathtt{dB}$, on a dB scale, is absolutely encoded, followed by differentially encoded $\delta E_\mathtt{dB}$s. When the noise-component parameters are differentially encoded with regard to the previous frame, the noise component starts with a 1. Both the LAR coefficients $\delta\alpha_\mathtt{LAR}$ and temporal-envelope measurements $\delta E_\mathtt{dB}$ are differentially encoded.

N_C (base layer)

0	$\alpha_\mathtt{LAR}$ ... $\alpha_\mathtt{LAR}$	$E_\mathtt{dB}$ $\delta E_\mathtt{dB}$ ... $\delta E_\mathtt{dB}$
1	$\delta\alpha_\mathtt{LAR}$ ... $\delta\alpha_\mathtt{LAR}$	$\delta E_\mathtt{dB}$ ... $\delta E_\mathtt{dB}$

Figure 4.4: *Syntax for the noise component. For convenience, the subscripts referring to the indices of parameters are omitted. See the text for more detail.*

Harmonic complex in the base layer

The harmonic complex follows the noise component in the bit stream, and is illustrated in Figure 4.5.

HC_SC (base layer)

<table>
<tr><td colspan="5">0 (no harmonic complex)</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>$\theta_\mathtt{f}$ $\theta_\mathtt{c}$ B N_h</td><td colspan="2">a_1 ... a_1
θ_1 ... θ_1</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">$\delta\theta_\mathtt{f}$ $\theta_\mathtt{c}$ δB δN_h</td><td>a_1 ... a_1</td><td>0 (interpolation)</td></tr>
<tr><td>θ_1 ... θ_1</td><td>1 δa_1 ... δa_1</td></tr>
</table>

Figure 4.5: *Syntax for the harmonic complex in the base layer. For convenience, the subscripts referring to the indices of parameters are omitted. See the text for more detail.*

When no harmonic complex is present in the current frame, this is indicated by a 0, while the presence of a harmonic complex is indicated by a 1.

If the harmonic complex in the current frame is starting up, this is indicated by a 0, and the fundamental frequency θ_{f}, frequency chirp θ_{c}, inharmonicity parameter B, and the number of harmonics N_{h} are absolutely encoded. The constant amplitude a_l and constant phase θ_l parameters of at most $N_{\mathrm{h,max},1}$ starting harmonics follow.

When the harmonic complex in the current frame is linked to the harmonic complex in the previous frame, this is indicated by a 1. In this case, the fundamental frequency $\delta\theta_{\mathrm{f}}$, inharmonicity parameter δB, and the number of harmonics δN_{h} are differentially encoded with regard to the previous frame. The frequency chirp θ_{c} is absolutely encoded. If $\delta N_{\mathrm{h}} > 0$, the constant-amplitude a_l and constant phase θ_l parameters of at most δN_{h} starting harmonics follow. The amplitude parameters of linked harmonics are not transmitted when the bit stream contains a 0, and the time-differential amplitudes δa_l in the next frame are differences with regard to the previous frame in this case. The decoder estimates the lacking amplitude parameters by interpolating amplitude parameters from adjacent frames. The amplitude parameters of linked harmonics are transmitted when the bit stream contains a 1. The time-differentially encoded amplitude parameters δa_l follow. The complete number of amplitude parameters in the base layer is upper-bounded by $N_{\mathrm{h,max},1}$.

Individual sinusoids

The syntax for the individual-sinusoid sub-component used in both the base and refinement layers is illustrated in Figure 4.6.

IS_SC (base and refinement layers)

<table>
<tr><td rowspan="3">N</td><td>a_1</td><td>0</td><td rowspan="3">$\cdots$</td><td>a_1</td><td>0</td><td colspan="6">0 (interpolation)</td></tr>
<tr><td>θ_1</td><td rowspan="2">1</td><td>θ_1</td><td rowspan="2">1</td><td rowspan="2">1</td><td>δa_1</td><td>0</td><td rowspan="2">$\cdots$</td><td>δa_1</td><td>0</td></tr>
<tr><td>θ_2</td><td>θ_2</td><td>$\delta\theta_2$</td><td>1</td><td>$\delta\theta_2$</td><td>1</td></tr>
</table>

Figure 4.6: *Syntax of the individual sub-component (IS_SC) used in both the base and refinement layers. For convenience, the subscripts referring to the indices of parameters are omitted. See the text for more detail.*

Firstly, the starting tracks are given. The number of starting tracks N is followed by the amplitude a_1, phase θ_1, and frequency θ_2 parameters of starting tracks. A linking bit indicates whether a starting track is continued to the next frame (1) or ends (0). The starting tracks are ordered according to frequency, in ascending order.

Secondly, the linked sinusoids are given. If the constant-amplitude and frequency parameters of linked sinusoids in the current frame are not transmitted, this is indicated by a 0. In this case, the track endings in the current frame are moved to the

next frame by the encoder to maintain the integrity of the bit stream, and the time-differential amplitudes δa_1 and frequencies $\delta\theta_2$, contained in the next frame, are differences with regard to the previous frame. The decoder estimates these parameters from adjacent frames by interpolation. If the constant-amplitude and frequency parameters of linked sinusoids are transmitted, this is indicated by a 1. In this case, the time-differentially encoded amplitude δa_1 and frequency parameters $\delta\theta_2$ of all linked sinusoids in the current layer follow, each with one linking bit. Once a track is started up in a particular layer, it remains in that layer for its complete duration.

Harmonic complex in a refinement layer

The syntax for the harmonic-complex sub-component used in refinement layers is illustrated in Figure 4.7.

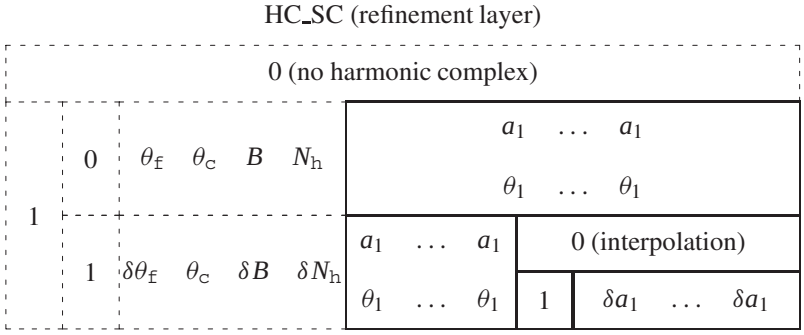


Figure 4.7: *Syntax for the harmonic complex in the base layer. For convenience, the subscripts referring to the indices of parameters are omitted. See the text for more detail.*

If the harmonic complex is starting up in the current frame, the amplitude a_1 and phase θ_1 parameters of a pre-determined number of harmonics are given.

If the harmonic complex is continued from the previous frame, and $\delta N_h > 0$, the amplitude and phase parameters of the starting harmonics are given first. If the constant-amplitude parameters of linked harmonics are not transmitted in this layer, this is indicated by a 0. The decoder estimates these amplitude parameters by interpolation from adjacent frames. If these parameters are transmitted, this is indicated by a 1, and the time-differential amplitudes δa_1 follow.

The complete number of amplitude parameters in layer l is upper-bounded by $N_{h,\max,l} - N_{h,\max,l-1}$.

4.3 Decoder

Our proposal for the overall structure of a bit-rate scalable decoder is illustrated in Figure 4.8. Layers may be removed from the bit stream received by the decoder. The bit-stream de-multiplexer (DEMUX) interprets that part of the bit stream received to obtain the parameters of the sinusoidal and noise components.

Recall that the parameters comprising the noise component are contained in the base layer, while the parameters comprising the sinusoidal component are distributed over the base and refinement layers. The noise component is obtained by applying the Noise Decoder (ND).

When not all layers are contained in the received bit stream, some harmonic-complex amplitude parameters may have to be estimated from the noise parameters. This is discussed in Section 4.3.1.

The sinusoidal signal contained in the bit stream, obtained by applying the Sinusoidal Decoder (SD), is denoted by $\hat{s}_{\texttt{sinusoid}}[n]$. Therefore, $\hat{s}_{\texttt{sinusoid}}[n]$ contains all harmonics and only those individual sinusoids contained in the layers received.

The dual description of the audio signal is exploited to obtain a well-tuned trade-off between sinusoids and noise. The feature of the decoder which adapts the noise signal $\hat{s}_{\texttt{noise,bl}}[n]$ to match $\hat{s}_{\texttt{sinusoid}}[n]$, and thus enables the decoder to operate in bit-rate scalable mode, is the Noise Adaptation (NA) module, described in Section 4.3.2. Recall from Section 3.6 that $\hat{s}_{\texttt{noise,bl}}[n]$ is a stochastic signal with a temporal and spectral envelope matched to that of $s_{\texttt{residual,bl}}[n]$. The adapted noise signal is denoted by $\hat{s}_{\texttt{noise,bl,adapted}}[n]$. The decoded signal $s_{\texttt{decoded}}[n]$ is the sum of $\hat{s}_{\texttt{sinusoid}}[n]$ and $\hat{s}_{\texttt{noise,bl,adapted}}[n]$.

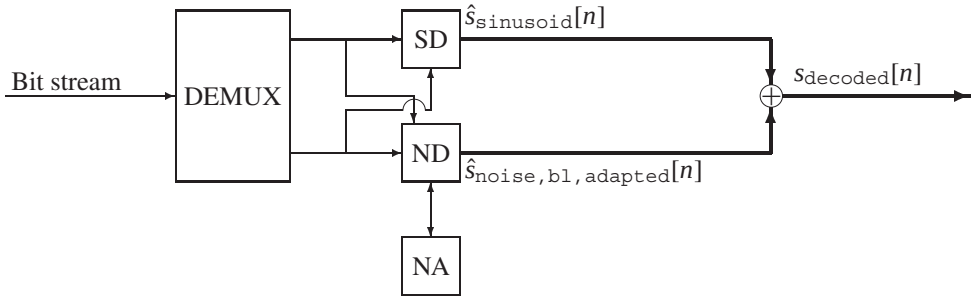


Figure 4.8: *The bit-rate scalable decoder consists of the bit-stream de-multiplexer (DEMUX), Sinusoidal Decoder (SD), Noise Decoder (ND), and Noise Adaptation (NA) modules.*

4.3.1 Strategy for estimating harmonic amplitude parameters

When not all refinement layers are contained in the bit stream received, some harmonic amplitude parameters will not be available if the number of harmonics N_{h} is too large. In this case, the required harmonic amplitude parameters in a frame are estimated from the amplitude-envelope of the noise component. The amplitude-envelope is given by

$$\sqrt{E_{\mathrm{c}}}|F_{\mathtt{synth}}(e^{j\theta})|,$$

where the energy E_{c} is derived from the temporal-envelope measurements E_i in sub-frame i. $F_{\mathtt{synth}}(e^{j\theta})$ is the transfer function of the prediction synthesis filter,

$$F_{\mathtt{synth}}(e^{j\theta}) = \frac{1}{1 - e^{-j\theta}\sum_{k=1}^{O_{\mathrm{p}}} \alpha_k H_k(\lambda; e^{j\theta})},$$

with prediction coefficients α_k, refer to (3.75) on page 90. Recall that the temporal-envelope measurements E_i and prediction coefficients α_k are contained in the base layer. A number of temporal-envelope measurements E_i are combined in the following way to obtain E_{c}:

$$E_{\mathrm{c}} = \left(\frac{\sum_{i=1}^{N_{\mathtt{short}}} \sigma_{r,i}^2}{N_{\mathtt{short}}}\right) \sum_n w_{E_{\mathrm{c}}}[n], \tag{4.9}$$

where $N_{\mathtt{short}}$ is the number of short windows ($w_{\mathtt{short}}[n]$) that fall under $w_{E_{\mathrm{c}}}[n]$ and the variance $\sigma_{r,i}^2$ is derived from E_i. Here, we have assumed that the random sources, with variance $\sigma_{r,i}^2$, are statistically independent. Recall that the temporal-envelope measurements were obtained under $w_{\mathtt{short}}[n]$. To choose the window $w_{E_{\mathrm{c}}}[n]$, we recall that the amplitude parameters were estimated in the encoder under the Hann window $w[n]$, as specified in (3.51) on page 70. For this reason, we choose $w_{E_{\mathrm{c}}}[n]$ to be the Hann window $w[n]$, which results in $N_{\mathtt{short}} = 9$, as illustrated in Figure 4.9. The only unknowns in (4.9) are the variances $\sigma_{r,i}^2$. These are obtained from the E_is by considering the prediction residual $r[n]$, obtained in the encoder, as a random source with variance $\sigma_{r,i}^2$ in sub-frame i. Thus, we can write E_i as

$$E_i = \sigma_{r,i}^2 \sum_n w_{\mathtt{short}}[n],$$

which yields $\sigma_{r,i}^2$.

The harmonic frequencies for which no amplitude parameters are contained in the bit stream are $\xi_k(\theta_{\mathtt{f}}, B)$, where $k = N_{\mathrm{h},\mathtt{max},l} + 1, \ldots, N_{\mathrm{h}}$ and where l is the highest layer contained in the bit stream. To determine the amplitude parameters corresponding to these frequencies, we observe that the spectrum of the cosine with amplitude

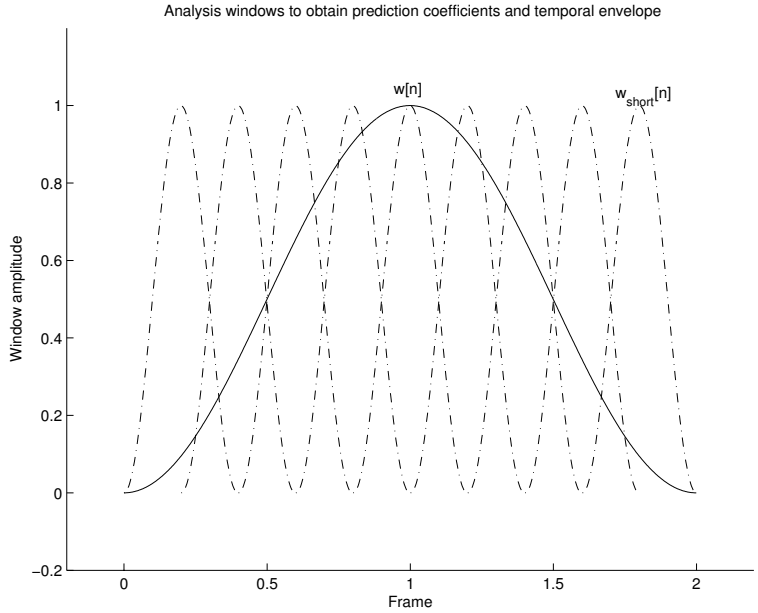


Figure 4.9: *The temporal envelope measurements are determined under the short Hann window $w_{\texttt{short}}[n]$ (dash-dotted line), while the prediction coefficients and sinusoidal amplitude parameters are obtained under the Hann window $w[n]$ (solid line).*

$a_{k,1}$ and frequency $\xi_k(\theta_{\texttt{f}}, B)$, $a_{k,1}\cos(\xi_k(\theta_{\texttt{f}}, B)n)$, under the window $w_{E_{\texttt{c}}}[n]$ is

$$\begin{aligned}\left(\mathrm{DFT}\, a_{k,1}\cos(\xi_k(\theta_{\texttt{f}}, B)n)w_{E_{\texttt{c}}}[n]\right)(e^{j\theta}) = \\ \frac{a_{k,1}}{2}\left[(\mathrm{DFT}\,\mathbf{w}_{E_{\texttt{c}}})(e^{j(\theta-\xi_k(\theta_{\texttt{f}},B))}) + (\mathrm{DFT}\,\mathbf{w}_{E_{\texttt{c}}})(e^{j(\theta+\xi_k(\theta_{\texttt{f}},B))})\right].\end{aligned} \quad (4.10)$$

Given this observation, the required amplitude parameters are then obtained by evaluating the amplitude envelope of the noise component in $\xi_k(\theta_{\texttt{f}}, B)$:

$$a_{k,1} = \frac{2\sqrt{E_{\texttt{c}}}|F_{\texttt{synth}}(e^{j\xi_k(\theta_{\texttt{f}},B)})|}{(\mathrm{DFT}\,\mathbf{w}_{E_{\texttt{c}}})(e^{j0}) + (\mathrm{DFT}\,\mathbf{w}_{E_{\texttt{c}}})(e^{j2\xi_k(\theta_{\texttt{f}},B)})}. \quad (4.11)$$

4.3.2 Noise adaptation

When more than the base layer is available to the decoder, the corresponding sinusoidal, $\hat{s}_{\texttt{sinusoid}}[n]$, and noise, $\hat{s}_{\texttt{noise,bl}}[n]$, signals do not match. Informal listening experiments confirmed that the sum $\hat{s}_{\texttt{sinusoid}}[n] + \hat{s}_{\texttt{noise,bl}}[n]$ sounds too noisy. Therefore, it is necessary to adapt $\hat{s}_{\texttt{noise,bl}}[n]$ appropriately. The purpose of the

noise adaptation module is to remove the parts of $\hat{s}_{\texttt{noise,bl}}[n]$ that correspond to the sinusoids in those refinement layers which are available to the decoder.

Adaptation is accomplished by applying a band-rejection filter to $\hat{s}_{\texttt{noise,bl}}[n]$ on a per-frame basis. The frequency response of the band-rejection filter is denoted by $F_{\texttt{br}}(\boldsymbol{\theta}_2; m)$, where the vector $\boldsymbol{\theta}_2$ contains the frequency parameters of sinusoids from the refinement layers and m is the index on the discrete frequency grid. The band-rejection operation is carried out in the frequency domain to obtain

$$\hat{s}_{\texttt{noise,bl,adapted,w}}[n] = \left(\mathrm{DFT}^{-1}(\mathrm{DFT}\,\hat{\mathbf{s}}_{\texttt{noise,bl,w}})[m] \cdot F_{\texttt{br}}(\boldsymbol{\theta}_2; m)\right)[n], \quad (4.12)$$

where $\hat{s}_{\texttt{noise,bl,w}}[n] = \hat{s}_{\texttt{noise,bl}}[n]w[n]$ and $w[n]$ is the Hann synthesis window. The complete $\hat{s}_{\texttt{noise,bl,adapted}}[n]$ is then obtained by applying overlap-add synthesis.

The rejection bandwidth of $F_{\texttt{br}}(\boldsymbol{\theta}_2; m)$ around a frequency $\theta_{k,2}$ is chosen to span one ERB. This is in line with the spectral resolution obtained through the choice of the Laguerre pole $\lambda = 0.7$, refer to Section 3.5.2 on page 85. In addition to the sinusoidal frequency parameters, the sinusoidal amplitude parameters $a_{k,1}$ are used to ensure that the energy removed from $\hat{s}_{\texttt{noise,bl}}[n]$ by the rejection filter is approximately equal to the energy contained in the corresponding windowed sinusoid. The band-rejection filter is then written as $F_{\texttt{br}}(\mathbf{A}; \boldsymbol{\theta}_2; m)$, where the elements of $\mathbf{A}$ have to be chosen.

In order to describe our approach to choosing the elements of $\mathbf{A}$, we first describe the filter $F_{\texttt{br}}(\mathbf{A}; \boldsymbol{\theta}_2; m)$ in more detail. The explicit form of the band-rejection filter is the product

$$F_{\texttt{br}}(\mathbf{A}; \boldsymbol{\theta}_2; m) = \prod_k F_{\texttt{br},k}(A_k; \theta_{k,2}; m) \cdot F_{\texttt{br},k}(A_k; -\theta_{k,2}; m), \quad (4.13)$$

where

$$F_{\texttt{br},k}(A_k; \theta_{k,2}; m) = \begin{cases} 1 - A_k F_{\texttt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}}) & \text{if } m \in M_{\theta_{k,2}} \\ 1 & \text{else.} \end{cases} \quad (4.14)$$

The rejection depth is limited by $A_k \in [0, 1]$ and $F_{\texttt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}}) \in [0, 1]$ is the generic bandpass-filter characteristic. The set $M_{\theta_{k,2}}$ contains the DFT bins belonging to the ERB band centred around $\theta_{k,2}$ and $m_{\theta_{k,2}} = \frac{\theta_{k,2}}{B_{\texttt{s}}}$ where $\theta_{k,2}$ is the frequency of the sinusoid and $B_{\texttt{s}}$ is the DFT bin-size under $w[n]$. The generic filter is considered in more detail in the design specifications in Section 4.4.2.

The energy contained in a windowed sinusoid with amplitude $a_{k,1}$ in the ERB band is

$$E_{\texttt{sin},k} = \frac{a_{k,1}^2}{4} \sum_{m \in M_{\theta_{k,2}}} |(\mathrm{DFT}\,\mathbf{w})(m - m_{\theta_{k,2}}) + (\mathrm{DFT}\,\mathbf{w})(m + m_{\theta_{k,2}})|^2.$$

Our aim is to choose A_k such that the energy in the ERB band is conserved:

$$\sum_{m \in M_{\theta_{k,2}}} |(\mathrm{DFT}\, \hat{\mathbf{s}}_{\mathtt{noise,bl,w}})[m]|^2 =$$
$$\sum_{m \in M_{\theta_{k,2}}} |(\mathrm{DFT}\, \hat{\mathbf{s}}_{\mathtt{noise,bl,w}})[m] \cdot (1 - A_k F_{\mathtt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}}))|^2 + E_{\mathtt{sin},k}. \quad (4.15)$$

We re-write this equation as

$$C_{k,3} A_k^2 - 2C_{k,2} A_k + C_{k,1} = 0 \quad (4.16)$$

where the coefficients

$$C_{k,1} = E_{\mathtt{sin},k},$$
$$C_{k,2} = \sum_{m \in M_{\theta_{k,2}}} |(\mathrm{DFT}\, \hat{\mathbf{s}}_{\mathtt{noise,bl,w}})[m]|^2 F_{\mathtt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}}), \text{ and}$$
$$C_{k,3} = \sum_{m \in M_{\theta_{k,2}}} |(\mathrm{DFT}\, \hat{\mathbf{s}}_{\mathtt{noise,bl,w}})[m] F_{\mathtt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}})|^2$$

are positive. The roots of (4.16) are real-valued when

$$C_{k,1} \leq \frac{C_{k,2}^2}{C_{k,3}}. \quad (4.17)$$

Therefore, A_k is chosen as

$$A_k = \begin{cases} \frac{C_{k,2} - \sqrt{C_{k,2}^2 - C_{k,3} C_{k,1}}}{C_{k,3}} & \text{if } C_{k,1} \leq \frac{C_{k,2}^2}{C_{k,3}} \text{ and } C_{k,3} \neq 0 \\ 1 & \text{else.} \end{cases} \quad (4.18)$$

After A_k is thus determined for each sinusoid, the complete transfer function of the band-rejection filter $F_{\mathrm{br}}(\mathbf{A}; \boldsymbol{\theta}_2; m)$ is constructed according to (4.13), and is applied to $\hat{s}_{\mathtt{noise,bl}}[n]$ according to (4.12), resulting in the adapted noise signal $\hat{s}_{\mathtt{noise,bl,adapted,w}}[n]$. Informal listening experiments revealed that this adapted noise signal is a much better match than $\hat{s}_{\mathtt{noise,bl}}[n]$ for the sinusoidal signal in the decoder.

4.4 Design specifications

The design specifications for the Bit-Rate Scalability module are provided in Section 4.4.1 and those for the Noise Adaptation module in Section 4.4.2.

4.4.1 Bit-Rate Scalability module

Low-pass filtering of amplitude parameters

For convenience, we denote the un-quantised amplitude parameters on a track (or a harmonic trajectory), by $a[l]$, where $l = 1, \ldots, L$ and L is the number of frames spanned by the track. The amplitude parameters of linked sinusoids and linked harmonics are low-pass filtered to reduce the cost of encoding these parameters. We denote the impulse response of the low-pass filter by $h[l]$, where

$$h[-1] = \frac{1}{4}, \quad h[0] = \frac{1}{2}, \quad h[1] = \frac{1}{4}, \quad h[2] = 0, \quad \ldots \tag{4.19}$$

The low-pass filtered amplitudes $a_{h^{(1)}}[l]$ are then obtained by applying

$$\begin{aligned} a_{h^{(1)}}[l] &= h[l] * a[l] \\ &= \frac{1}{4}a[l+1] + \frac{1}{2}a[l] + \frac{1}{4}a[l-1] \end{aligned}$$

where $l = 2, \ldots, L$, where $a[L+1]$ is defined as $a[L+1] = a[L]$, and where $*$ denotes convolution. The first amplitude is unchanged by specifying that $a_{h^{(1)}}[1] = a[1]$. It was observed after experimentation that applying the filter $h[l]$ five times in cascade resulted in very little degradation in audio quality. The low-pass filtered amplitude parameters obtained after this cascaded filtering are given by

$$a_{h^{(5)}}[l] = (\overbrace{h * h * \ldots * h}^{5 \text{ times}} * a)[l].$$

After low-pass filtering, the amplitudes $a_{h^{(5)}}[l]$ are quantised with the quantiser described in Section 3.5.1 on page 83. The resulting entropy of time-differential encoding is 1.92 bits, substantially lower than the original entropy of 3.48 bits (refer to Table 3.8 on page 98).

Transmission of amplitude and frequency parameters

Informal listening experiments revealed that the removal of amplitude and frequency parameters on a track results in a limited degradation in audio quality in the current system. To obtain a significant reduction in bit rate through this technique, the amplitude and frequency parameters of linked individual sinusoids are encoded every second frame. Similarly, the amplitude parameters of linked harmonics are encoded every second frame. Informal listening experiments showed that the resulting degradation in audio quality is very small. Since the time-differential parameters encoded in this approach are differences with regard to two frames back, the associated histograms will become less peaked. Therefore, the entropy information was

re-calculated, and is given in Table 4.2. From this table, we conclude that the average cost per frame of a time-differential amplitude transmitted every second frame is $2.68/2 = 1.34$ bits, a 30% reduction in encoding costs. A similar reduction is obtained for the frequency parameters.

Parameter	Original entropy	Entropy with interpolation
$a_{k,1}$	1.92	2.68
$\theta_{k,2}$	1.46	1.90

Table 4.2: *Entropy of sinusoidal amplitude and frequency parameters when they are transmitted every second frame.*

The layers

The lowest bit rate at which a reasonable trade-off between sinusoids and noise is achieved for all excerpts was determined after experimentation to be 16 kbits/s. Below this bit rate, important tracks that can not be represented by noise in a perceptually acceptable way, are removed. The result is that the dominant noise component results in an un-natural sounding decoded signal, and the audio quality degrades to such an extent below 16 kbits/s that one can not speak of a graceful degradation in audio quality. Therefore, the target bit-rate of 10 kbits/s can not be met. The highest bit rate is chosen as 40 kbits/s. The number of layers and the target bit rate for each layer are specified in Table 4.3.

Layer l	$R_{\mathtt{target},l}$ kbits/s	Cumulative target
1 (Base layer)	16	16
2 (Refinement layer 1)	4	20
3 (Refinement layer 2)	4	24
4 (Refinement layer 3)	8	32
5 (Refinement layer 4)	8	40

Table 4.3: *The bit stream contains five layers: one base layer and four refinement layers.*

The number of harmonic amplitude and phase parameters

The maximum number of harmonic amplitude and phase parameters $N_{\mathtt{h,max},l}$, for each layer l, is specified to restrict the size of the harmonic complex in each layer. This restriction is important when an audio signal is modelled by both (a high number of) harmonics and tracks. Examples of such audio signals include the si01 (harpsichord) and sm01 (bagpipes) excerpts. The values for $N_{\mathtt{h,max},l}$ were determined empirically, and are given in Table 4.4. The highest cost is incurred by the amplitude

and phase parameters of starting harmonics. Although the average number of starting harmonics per frame is low, the infrequent instances during which a high number of starting harmonics occur make this restriction necessary.

Layer l	$N_{\mathrm{h,max},l}$
1 (Base layer)	20
2 (Refinement layer 1)	30
3 (Refinement layer 2)	50
4 (Refinement layer 3)	80
5 (Refinement layer 4)	150

Table 4.4: *The maximum number of harmonic amplitude and phase parameters per layer. The numbers given in column two are cumulative.*

Coding delay and fluctuations in bit rate

In a practical application, it is not feasible to satisfy the target bit rate for a complete audio excerpt, since the resulting coding delay and possible fluctuations in momentary bit rate may be unacceptable. For example, in bidirectional communication applications, low coding delay is an important requirement. Furthermore, large fluctuations in the momentary bit rate are undesirable in most applications. To address these issues, the target bit rate is specified for brief periods, and thus becomes a short-time target bit rate. The short-time target bit rate of a particular layer is satisfied by applying the BRS module to a set of consecutive frames having a duration corresponding to the specified short time. The set of consecutive frames is referred to as a bit-rate block. We utilise non-overlapping bit-rate blocks, as illustrated in Figure 4.10. Deviations from the target bit rate in a bit-rate block are absorbed in a bit-reservoir for each layer.

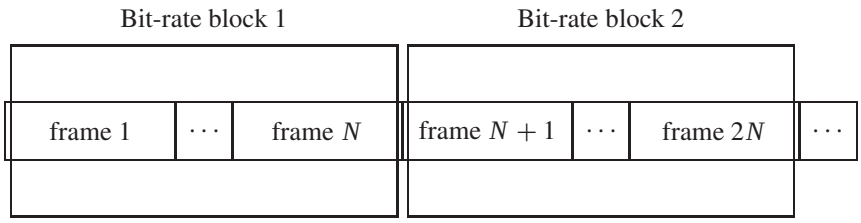


Figure 4.10: *Bit-rate blocks containing N frames each.*

In addition to bit-rate blocks, the integration of perceptual distortion parameters over 300 ms on a track introduces an additional coding delay. Reducing the integration period comes at the expense of less accurate estimates of the perceptual distortion

of complete tracks. The number of frames contained in a bit-rate block in the experiments was chosen as 150, which corresponds to ≈1.2 seconds. Together with the 300 ms integration time, the complete coding delay amounts to ≈1.5 seconds. For practical purposes, this may be too long. However, no extensive experimentation was carried out to determine the influence of lowering the duration of the bit-rate blocks and integration time, on the audio quality of each layer.

Estimating the slope in the Lagrangian optimisation

In this section, we describe how the optimal slope $\lambda_{R_{\mathrm{is},l}}$ is estimated in the Lagrangian optimisation process described in Section 4.2.2. Although several methods for estimating $\lambda_{R_{\mathrm{is},l}}$ have been described in literature, e.g., the bisection search [126, 127], the simple method described in the following proved to be sufficient for our purposes. Starting from an initial choice of $\lambda_l = \lambda_{\mathtt{initial}}$ in the first bit-rate block in layer l, λ_l is adapted as

$$\lambda_l := \begin{cases} \lambda_l + \delta_\lambda & \text{if } R(\lambda_l) > R_{\mathrm{is},l} \\ \max\{\lambda_l - \delta_\lambda, 0\} & \text{if } R(\lambda_l) < R_{\mathrm{is},l} \end{cases}$$

until $\lambda_l = 0$ or the target bit rate $R_{\mathrm{is},l}$ is satisfied. Here, we have exploited the fact that the rate-distortion curve is decreasing monotonically. Deviations from this target are reflected in the bit-reservoir, as described above. This search for $\lambda_{R_{\mathrm{is},l}}$ is carried out for each bit-rate block, where subsequent bit-rate blocks start from the $\lambda_{R_{\mathrm{is},l}}$ obtained in the previous bit-rate block. In practice, only small modifications of λ_l in subsequent bit-rate blocks are required in this approach. The initial choice of λ_l is

$$\lambda_{\mathtt{initial}} = \frac{D_{\mathtt{avg}}}{r_{\mathtt{avg}}},$$

where $r_{\mathtt{avg}}$ is the average rate of a track and $D_{\mathtt{avg}}$ the average distortion. Finally, δ_λ is chosen as $\delta_\lambda = 0.005$.

Obtaining an estimate of the bit rate of the noise component

In this section, we describe how the average bit rate of the noise component, $R_{\mathtt{noise}}$, is determined from all excerpts. Furthermore, a suitable approach to dealing with deviations from the average $R_{\mathtt{noise}}$ is proposed.

The bit rate of the noise component depends on the sinusoidal component in the base layer, while the sinusoidal component in the base layer is derived by assuming that the rate of the noise component is known. This is a chicken-and-egg problem. The rate $R_{\mathtt{noise}}$ is estimated in an iterative way. We denote the initial estimate by $R^{(0)}_{\mathtt{noise}}$. Given the estimate $R^{(i)}_{\mathtt{noise}}$ in iteration i, the sinusoids contained in the base

layer of all excerpts are determined by the Bit-Rate Scalability module. The Noise Coder is applied to all residuals, yielding the noise parameters for all excerpts. Next, the entropy information of the noise parameters is calculated, and the new estimate of the rate of the noise component, $R_{\mathtt{noise}}^{(i+1)}$, is determined from the entropy information. In practice, this iterative process converges after a few iterations. This process for estimating $R_{\mathtt{noise}}$ is carried out once.

In the developed prototype, the rate of the noise component is assumed constant. In practice, however, this rate will vary over time. To deal with deviations from $R_{\mathtt{noise}}$, we propose the following strategy. In the encoding process, the estimated $R_{\mathtt{noise}}$ is used in the first bit-rate block. The difference between $R_{\mathtt{noise}}$ and the realised rate in the first bit-rate block is absorbed by the bit-reservoir for the base layer and $R_{\mathtt{noise}}$ is updated. This approach is taken in each bit-rate block. The updated $R_{\mathtt{noise}}$ can be a weighted average of the realised rate over a number of preceding bit-rate blocks.

4.4.2 Noise Adaptation module

The generic band-pass filter characteristic $F_{\mathtt{g}}(|M_{\theta_{k,2}}|; m - m_{\theta_{k,2}})$, see Section 4.3.2, is considered in more detail in this section. For each frame, the DFT of the windowed signal $\hat{s}_{\mathtt{noise,bl,w}}[n]$ is computed to carry out the band-rejection operation. To allow the use of the FFT algorithm, the windowed signal is zero-padded up to a length of 1024 samples. (Recall from (3.50) and (3.51) on page 70 that the window $w[n]$ contains $2N_{\mathtt{s,UR}} + 1 = 711$ samples.) For simplicity, the index $m_{\theta_{k,2}}$ is rounded to the nearest integer,

$$m_{\theta_{k,2}} = \text{round}\left(\frac{\theta_{k,2}}{B_{\mathtt{s,zp}}}\right)$$

where $B_{\mathtt{s,zp}}$, being approximately 43 Hz, is the DFT bin-size corresponding to the zero-padded sequence. The width of the ERB band, denoted by $e_{\mathtt{BW}}(\theta_{k,2})$ (in radians), at a frequency $\theta_{k,2}$ is given by

$$e_{\mathtt{BW}}(\theta_{k,2}) = 24.7\left(4.37\frac{44.1}{2\pi}\theta_{k,2} + 1\right).$$

By specifying that $\theta_{k,2}$ lies at the centre of the critical band, the band spans the interval

$$\left[\theta_{k,2} - \frac{e_{\mathtt{BW}}(\theta_{k,2})}{2}, \theta_{k,2} + \frac{e_{\mathtt{BW}}(\theta_{k,2})}{2}\right].$$

The set $M_{\theta_{k,2}}$, containing DFT bins lying in this interval, is given by

$$M_{\theta_{k,2}} = \left\{\left\lfloor \frac{\theta_{k,2} - \frac{e_{\mathtt{BW}}(\theta_{k,2})}{2}}{B_{\mathtt{s,zp}}} \right\rfloor, \ldots, m_{\theta_{k,2}}, \ldots, \left\lceil \frac{\theta_{k,2} + \frac{e_{\mathtt{BW}}(\theta_{k,2})}{2}}{B_{\mathtt{s,zp}}} \right\rceil\right\}.$$

The filter with frequency response

$$F_{\mathrm{g}}(|M_{\theta_{k,2}}|, m) = \begin{cases} \cos(\pi \frac{m}{|M_{\theta_{k,2}}|-1}) & \text{if } -\frac{|M_{\theta_{k,2}}|-1}{2} \leq m \leq \frac{|M_{\theta_{k,2}}|-1}{2} \\ 0 & \text{else} \end{cases}$$

is chosen.

At low frequencies, the critical bandwidth $e_{\mathtt{BW}}$ is in the order of the bin-size $B_{\mathtt{s,zp}}$. In this case, the set $M_{\theta_{k,2}}$ will contain a small number of DFT bins, resulting in a very narrow band-rejection filter. This is undesirable, since the impulse response of the band-rejection filter will not fade out fast enough and the signal after band-rejection, $\hat{s}_{\mathtt{noise,bl,adapted},\mathbf{w}}[n]$, will not exhibit the same time-localised character as $\hat{s}_{\mathtt{noise,bl},\mathbf{w}}[n]$ before band-rejection. To avoid this problem, the minimum number of DFT bins is set to seven. Therefore,

$$M_{\theta_{k,2}} := M_{\theta_{k,2}} \bigcup \{m_{\theta_{k,2}} - 3, \ldots, m_{\theta_{k,2}} + 3\}.$$

After the band-rejection filter is applied to $(\mathrm{DFT}\,\hat{\mathbf{s}}_{\mathtt{noise,bl},\mathbf{w}})[m]$ according to Equation (4.12), the sequence $\hat{s}_{\mathtt{noise,bl,adapted},\mathbf{w}}[n]$ is truncated at those positions where zeros were added before overlap-add synthesis is applied.

4.5 Results

In this section, we provide a quantitative analysis of the contents of the layers. The characteristics of the individual sinusoids is considered in Section 4.5.1. The harmonic complex is described in Section 4.5.2, and the noise component in Section 4.5.3. In Section 4.5.4, the average bit-rate of each layer is given.

4.5.1 Individual sinusoids

An important characteristic of the individual-sinusoid sub-component is the average length of a track in each layer. As we have observed before, the high number of short tracks, common in parametric audio coding, is an undesirable phenomenon. By taking the length of a track into account when estimating its perceptual relevance, and by utilising a rate-distortion optimisation mechanism, we have strived to select the tracks that are indeed serving their intended purpose. The average and standard deviation of the track length per layer, obtained from all test material, are given in Table 4.5. The average track length decreases from 11 frames in the base layer (layer 1) to 4 frames in the highest refinement layer (layer 5). This is an indication of the ability of the BRS module to select the longer tracks first. Furthermore, we observe that the average track length decreases rapidly with each increasing layer.

Layer	Average track length	Standard deviation
1 (Base layer)	11.10	11.65
2 (Refinement layer 1)	8.04	8.70
3 (Refinement layer 2)	7.42	7.84
4 (Refinement layer 3)	5.58	5.85
5 (Refinement layer 4)	3.86	3.21

Table 4.5: *Average and standard deviation of the track length in each layer.*

Figure 4.11 illustrates the track-length distribution over the five layers, where the maximum track-length of 37 frames is visible from the first four sub-plots in this figure. We observe that the track lengths appear to be exponentially distributed in all layers. In particular, the track lengths in the higher layers are more un-evenly distributed than those in the lower layers.

Inspection of the characteristics of individual excerpts revealed that the exponential distribution is pronounced for all excerpts in the refinement layers. However, for the base layer, individual excerpts can be classified as belonging to one of two groups. The first group (consisting of the excerpts es01 [Vocal], es02 [German speech], es03 [English speech], si01 [Harpsichord], si02 [Castanets], si03 [Pitch pipe], and sm02 [Glockenspiel]), exhibits a pronounced exponential distribution of track length in the base layer, while the track lengths of the second group (sc01 [Trumpet solo and orchestra], sc02 [Orchestral piece], sc03 [Contemporary pop music], sm01 [Bagpipes],

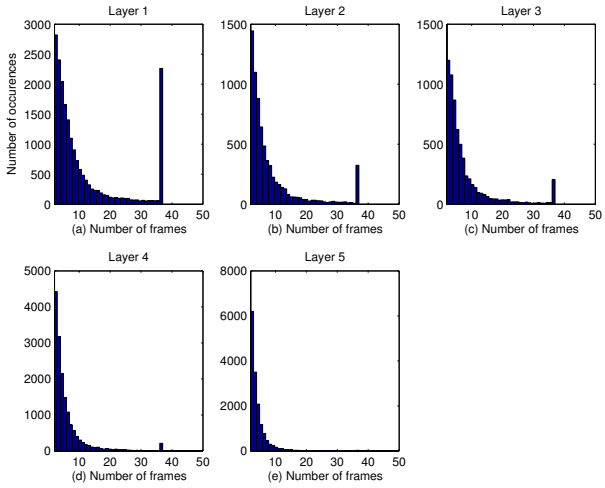


Figure 4.11: *Distributions of the track lengths over the layers for all excerpts. (a) Base layer. (b) Refinement layer 1. (c) Refinement layer 2. (d) Refinement layer 3. (e) Refinement layer 4.*

and sm03 [Plucked strings]) are more evenly distributed in the base layer. Figure 4.12 illustrates the track-length distributions of the first group, and Figure 4.13 those of the second group.

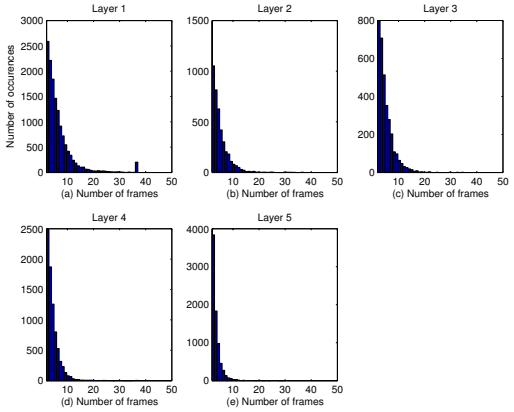


Figure 4.12: *Distributions of the track lengths over the layers for the excerpts es01 (Vocal), es02 (German speech), es03 (English speech), si01 (Harpsichord), si02 (Castanets), si03 (Pitch pipe), and sm02 (Glockenspiel). (a) Base layer. (b) Refinement layer 1. (c) Refinement layer 2. (d) Refinement layer 3. (e) Refinement layer 4.*

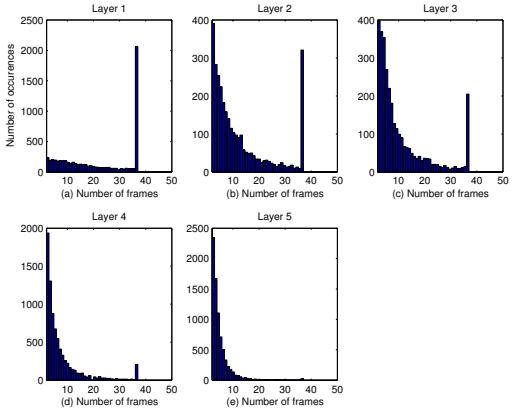


Figure 4.13: *Distributions of the track lengths over the layers for the excerpts sc01 (Trumpet solo and orchestra), sc02 (Orchestral piece), sc03 (Contemporary pop music), sm01 (Bagpipes), and sm03 (Plucked strings). (a) Base layer. (b) Refinement layer 1. (c) Refinement layer 2. (d) Refinement layer 3. (e) Refinement layer 4.*

In most of the excerpts belonging to the first group, the harmonic complex models the major part of the tonal component of the signal, while the harmonic complex plays a lesser role in most of the excerpts belonging to the second group. Therefore, it is not surprising that the individual-sinusoid sub-component in the base layer of the first group exhibits the same unfavourable characteristics as those exhibited by the higher layers. We conclude that the most important part of the individual-sinusoid sub-component is captured in the base layer.

The average number of individual sinusoids per frame, for each layer, is given in Table 4.6. A distinction is made between the average number of starting and linked individual sinusoids (columns three and four), and the average bit-rate of the individual-sinusoid sub-component in each layer is given in column five. From this table we observe that the largest number of individual sinusoids is contained in the base layer, and corresponds to a bit-rate of 7.6 kbits/s. An average number of 27 sinusoids is contained in all layers combined, corresponding to a bit-rate of 22 kbits/s. The cost per sinusoid, per frame, increases from 0.6 kbits/s in the base layer to 1.2 kbits/s in refinement layer 4, and a sinusoid costs 0.8 kbits/s on average over all layers.

Layer	Number	Starting	Linked	Bit rate (kbits/s)
1 (Base layer)	12.50	1.13	11.37	7.58
2 (Refinement layer 1)	3.27	0.41	2.86	3.09
3 (Refinement layer 2)	2.72	0.37	2.35	2.75
4 (Refinement layer 3)	5.07	0.91	4.16	4.93
5 (Refinement layer 4)	3.43	0.89	2.54	4.12
Total	26.99	3.71	23.28	22.47

Table 4.6: *The average number of individual sinusoids per frame, for each layer, is given in column two. The number of starting tracks is given in column three. Column four contains the number of linked tracks. The bit-rate given in column five includes the overhead.*

4.5.2 Harmonic complex

The average duration of the harmonic complex is 89 frames, substantially longer than the average track duration. The distribution of the harmonic complex duration, over all excerpts, is illustrated in Figure 4.14. Three harmonic complexes have a duration in excess of 900 frames. These are due to the pitch pipe (si03) excerpt. The overall trend indicates an ostensibly exponential distribution of harmonic-complex duration. Closer inspection revealed that the speech excerpts are responsible for the largest portion of harmonic complexes with relatively brief duration, while the duration of harmonic-complexes from the remaining excerpts are more evenly distributed.

In calculating the average bit-rate of the harmonic complex, we make a distinction

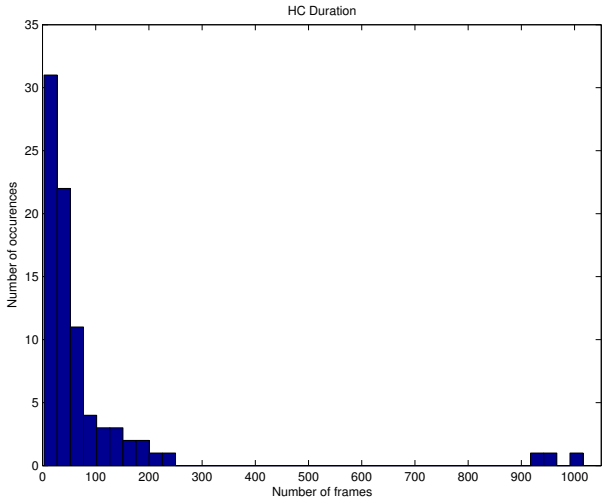


Figure 4.14: *Distribution of the harmonic-complex duration. Distribution for all harmonic complexes.*

between the cost of the non-constant polynomial-phase parameters (θ_{f}, θ_{c}, B, and N_{h}) and the amplitude and constant phase parameters (a_1 and θ_1). Over all excerpts, an average of 0.00473 harmonic complexes are starting up per frame, while an average of 0.408 harmonic complexes are linked per frame. This corresponds to a bit-rate of 0.82 kbits/s for the non-constant polynomial-phase parameters, where the overhead is included. The average number of harmonic amplitude and phase parameters per frame, for each layer, is given in Table 4.7. From this table, we observe that there is an average of 6.57 harmonic amplitude parameters per frame in the base layer, and

Layer	Number of Harmonics	Starting	Linked	Bit rate (kbits/s)
1 (Base layer)	6.57	0.17	6.40	2.08
2 (Refinement layer 1)	2.25	0.05	2.20	0.70
3 (Refinement layer 2)	1.99	0.09	1.90	0.67
4 (Refinement layer 3)	0.36	0.06	0.30	0.16
5 (Refinement layer 4)	0.26	0.01	0.25	0.09
Total	11.43	0.38	11.05	3.70

Table 4.7: *The average number of harmonic amplitude and phase parameters per frame, for each layer, is given in column two. The number of starting harmonics is given in column three. Column four contains the average number of linked harmonics. The corresponding bit rates, given in column 5, include the overhead associated with the amplitude and phase parameters.*

11.43 harmonic amplitude parameters per frame in all refinement layers combined. Furthermore, we observe that the number of starting harmonics make up only a small fraction of the total number of harmonics over all layers. The average cost of the amplitude and phase parameters per harmonic is approximately 0.33 kbits/s in all layers. When combined with the cost of the non-constant polynomial-phase parameters, the complete cost per harmonic is approximately 0.4 kbits/s on average, half the cost of an individual sinusoid.

4.5.3 Noise

The entropy information of the prediction coefficients and temporal envelope is given in Table 4.8. From this table, the cost of the noise component, when differential coding is applied and overhead is accounted for, is calculated as

$$R_{\texttt{noise}} = R_{\texttt{overhead}} + R_{\alpha} + R_{E} = 0.12 + 3.49 + 1.32 = 4.93 \text{ kbits/s}, \quad (4.20)$$

where $R_{\texttt{overhead}}$ is the bit rate required by the overhead (one bit per frame according to the bit-stream syntax), R_{α} the bit rate required by the prediction coefficients, and R_E the bit rate required by the temporal envelope. Therefore, the parameters of the noise component consume slightly more than 30% of the bit-rate of the base layer.

Parameter	Absolute	Differential
E	5.05	2.13
α_1	4.06	1.89
α_2	3.91	1.87
α_3	3.21	1.56
α_4	3.08	1.53
α_5	2.55	1.39
α_6	2.65	1.41
α_7	2.36	1.29
α_8	2.32	1.28
α_9	1.99	1.22
α_{10}	2.15	1.20
α_{11}	1.85	1.16
α_{12}	1.96	1.12

Parameter	Absolute	Differential
α_{13}	1.52	1.07
α_{14}	1.66	1.08
α_{15}	1.47	1.04
α_{16}	1.56	0.99
α_{17}	1.34	0.97
α_{18}	1.38	0.94
α_{19}	1.27	0.91
α_{20}	1.27	0.90
α_{21}	1.08	0.86
α_{22}	1.13	0.84
α_{23}	0.98	0.79
α_{24}	1.32	0.79

Table 4.8: *Entropy (in bits) of noise-component parameters. Entropies for both absolute and differential coding are given for the temporal leven E and prediction coefficients α.*

4.5.4 Bit rate

The average bit-rate of each layer is summarised in Table 4.9. From this table, we observe that the realised bit-rates of the base and first two refinement layers are rea-

sonably close to the target bit-rates. For refinement layers 3 and 4, however, the realised bit-rate lags behind the target, indicating that the size of the complete sinusoidal component is not large enough to fill these layers.

Layer	Realised bit rate (kbits/s)	Target bit rate (kbits/s)
1 (Base layer)	15.41	16
2 (Refinement layer 1)	3.79	4
3 (Refinement layer 2)	3.42	4
4 (Refinement layer 3)	5.09	8
5 (Refinement layer 4)	4.21	8

Table 4.9: *Average realised bit-rate of each layer.*

4.6 Discussion

The lowest bit-rate at which a reasonable audio quality is attained by the bit-rate scalable coder, namely 16 kbits/s, may still be too high for certain applications. To obtain a reasonable quality at even lower bit-rates, additional strategies should be applied. The problem is that the noise component becomes too dominant below 16 kbits/s, since it consumes more than 30% (4.93 kbits/s) of the bit-rate, which leaves an insufficient share of the room for the sinusoidal component. Furthermore, in contrast to the sinusoidal component, no operations were applied to the noise component to reduce its size; the noise component was placed as a whole in the base layer. Therefore, considering strategies for lowering the cost of the noise component is the obvious approach.

We now discuss three suitable strategies for obtaining a reasonable quality below 16 kbits/s. The *first* strategy is not to transmit the Laguerre prediction coefficients in every frame, similar to the approach taken to lower the cost of the sinusoidal component. This should be signalled in the bit-stream, and the decoder then estimates the missing prediction coefficients by interpolation in the LAR domain. The bit-stream syntax for the noise component should be adapted accordingly. From the calculation of entropy information, transmitting the prediction coefficients every second frame reduces the bit-rate of the noise component from 4.93 kbits/s to 3.85 kbits/s. Informal listening experiments suggested that the degradation in audio quality is minor. The decrease in bit-rate, however, is relatively small, and we conclude that this strategy alone is not suitable if a larger reduction in bit-rate is required. The *second* strategy is to lower the Laguerre prediction order O_{p}. Reducing the prediction order from $O_{\mathrm{p}} = 24$ to $O_{\mathrm{p}} = 12$ results in a bit-rate of 3.51 kbits/s for the noise component, see Table 4.8. Informal quality assessment suggested that, even though the difference between the decoded signals with $O_{\mathrm{p}} = 24$ and $O_{\mathrm{p}} = 12$ is audible, the degradation

in quality for speech is minor, while the degradation in quality for more complex sounds, like the sc02 (orchestra) and sc03 (contemporary pop music) excerpts, is more pronounced. The LAR representation of the prediction coefficients allows the simple scaling of the noise component in this manner. Like the first strategy, the second strategy alone is not suitable if a larger reduction in bit-rate is required. The *third* strategy is to combine the first two strategies. When only $O_{\mathrm{p}} = 12$ prediction coefficients are transmitted every second frame, the bit-rate of the noise component reduces to 2.94 kbits/s, almost half its original rate, while the resulting audio quality is still acceptable. When this strategy is applied, the bit-rate of the base layer can be lowered to approximately 14 kbits/s without having to reduce the number of bits available for the sinusoidal component. The noise parameters not transmitted in the base layer can be placed in the first refinement layer.

4.7 Summary

In this chapter, the design of a bit-rate scalable parametric audio coder is described. The layers comprising the bit stream are inter-dependent. The design of the encoder and decoder is based on the considerations discussed in Chapter 2, and we have strived to obtain a well-tuned balance between tones and noise in the decoded signal. A number of techniques are combined to reduce the cost of the sinusoidal component. In addition, each track is parameterised by a rate and a perceptual distortion. The sinusoidal component is then distributed over the base and refinement layers to satisfy the target bit rate of each layer, while the complete noise component is placed in the base layer. The decoder adapts the noise signal to the sinusoidal signal in order to obtain a well-tuned trade-off between sinusoids and noise.

The desired range of bit rates over which the scalable coder should operate is 10 – 40 kbits/s. However, informal listening experiments revealed that 16 kbits/s is the lowest bit rate at which a reasonable trade-off between sinusoids and noise can be achieved in the approach described in this chapter. Below 16 kbits/s, the noise component becomes dominant, resulting in an un-natural sounding decoded audio signal. Therefore, the range is re-defined as 16 – 40 kbits/s. Five layers were created to cover this range, and a framework of the scalable bit-stream syntax was provided. Coding delay and momentary fluctuations in bit rate were addressed by the definition of bit-rate blocks. The rate-distortion optimisation mechanism assigns longer tracks to the lower layers.

4.8 Conclusion

This chapter describes the design of a bit-rate scalable parametric audio coder. The scalable coder is based on the coder developed in Chapter 3. In the design of the

scalable coder, the need for a well-tuned trade-off between sinusoids and noise, as highlighted in Chapter 2, was addressed. In Chapter 5, the performance of our coder will be evaluated by means of listening tests.

Chapter 5

Listening test

5.1 Introduction

In this chapter, the audio quality delivered by the bit-rate scalable audio coder is assessed at each bit-rate. Subjective listening tests, carried out by a panel of experienced individuals, are still considered to be the most reliable method of audio quality assessment [38]. Objective quality-assessment methods, like PEAQ [45], are not considered to be mature and reliable enough to test coded material featuring low and intermediate quality [40]. Given these considerations, the audio quality of our bit-rate scalable parametric audio codec is assessed by conducting subjective listening tests. A suitable method for the assessment of medium to large impairments in audio quality is the MUSHRA (MUlti Stimulus test with Hidden Reference and Anchor) method [38]. Version RM3 of PPC (Reference Model 3 of PPC in MPEG-4 Extension 2) is included in the test.

In Section 5.2, an overview of the MUSHRA methodology is given. The setup of the test is described in Section 5.3. In Section 5.4, the results from the listening test are provided. These results are discussed in Section 5.5 and conclusions are drawn in Section 5.6. Section 5.7 provides recommendations concerning the approach that should be taken to improve the overall audio quality and to obtain bit-rates below 16 kbits/s.

5.2 Overview of the MUSHRA methodology

For an in-depth discussion of the MUSHRA methodology, we refer to [38, 40, 128]. In the following, a short description of the MUSHRA methodology is derived from these references. MUSHRA is a double-blind multi-stimulus test method with hidden reference and hidden anchor(s), especially designed for the assessment of medium to large impairments in audio quality. The listening test is conducted over one or more *sessions*. In each session, the audio material to be graded is presented during a number of *trials*. The *stimuli* presented during a trial comprise multiply processed versions of an audio excerpt. In a *blind* test, the only source of information on the

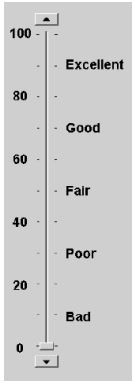


Figure 5.1: *The five-interval Continuous Quality Scale (CQS) used in MUSHRA listening tests.*

trials accessible to the individual are the stimuli, while a *double-blind* test is a blind test in which there is no possibility of uncontrolled interactions between the experimenter and the listening test [37]. In a *multi-stimulus* test, more than one stimulus is presented during a trial to the individual for quality assessment. The number of stimuli in a trial should not exceed 15. In a MUSHRA test, the uncoded original signal is provided as the reference. The material to be graded comprises a hidden reference, hidden anchor(s), and the coded material. The purpose of the hidden reference is to allow the experimenter to evaluate an individual's ability to successfully detect artefacts. In a test comprising material with medium to large impairments, the usefulness of the hidden reference is questionable. The purpose of the hidden anchor(s) is to provide an indication of how the coded material compares to well-known audio quality levels. At least one anchor, being a low-pass filtered version of the reference, should be used. The bandwidth of this anchor should be 3.5 kHz, which corresponds to control circuits used for supervision and coordination purposes in broadcasting. Additional anchors may also be used. The grading scale used in MUSHRA tests is the five-interval Continuous Quality Scale (CQS) illustrated in Figure 5.1. The quality ratings on this scale range from "bad" (corresponding to a score between 0 and 20) to "excellent" (corresponding to a score between 80 and 100). During a trial, the individual may select each of the stimuli in any order for playback. The individual is asked to assess the quality of all stimuli in a trial. In Figure 5.2, an illustration is provided of an interface through which the individual records his or her assessment of audio quality. The interface is that provided by the SEAQ tool, developed by the CRC (Communications Research Centre) in Ottawa, Canada. The reference is indicated as "REF" in this figure, while the stimuli (A through H in this illustration) contain the hidden reference, hidden anchor(s), and coded material. In each trial,

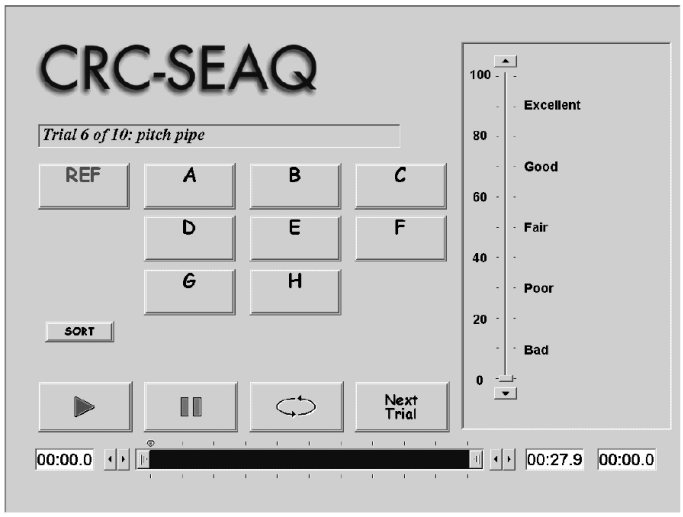


Figure 5.2: *The CRC-SEAQ user interface for the MUSHRA tests.*

the signals under test are randomly assigned to the stimuli. For each stimulus, the individual records a score by setting the slider to the appropriate value on the CQS. During playback, the individual can switch instantaneously between stimuli. The advantage of this approach is that the individual can compare all stimuli at will.

To obtain reliable results, individuals, experienced in the critical assessment of audio quality, are preferred. Furthermore, the actual test should be preceded by a training phase, where each individual becomes familiar with the full range and nature of possible impairments, as well as with all excerpts used in the test. The test may either be performed on headphones or loudspeakers, though the use of both during a session is not permitted. When presenting the results of a test, all mean scores should be accompanied by corresponding confidence intervals.

5.3 The test setup

The test material used in the listening test is given in Table 5.1, and is a selection of the material in Table 3.1 on page 68. One excerpt from each of the categories *speech* (es∗), *single instruments with only one note sounding at a time* (si∗), *simple sound mixtures* (sm∗) and two excerpts from the category *complex sound mixtures* (sc∗) are selected for the listening test. All excerpts are mono, where the left channel from the original stereo excerpt is copied to the right channel prior to encoding. The sampling frequency remains unchanged at 44.1 kHz.

The audio quality is measured for each bit rate at which the scalable parametric

Name	*Description*	*Duration (seconds)*
es02	German speech	8.6
si03	Pitch pipe	27.9
sm01	Bagpipes	11.1
sc02	Orchestral piece	12.7
sc03	Contemporary pop music	11.6

Table 5.1: *Test material used in the listening tests.*

audio coder, described in this thesis, operates. The base layer is referred to as "Codec 16 kb/s mono." The base and first refinement layer is referred to as "Codec 20 kb/s mono," etc. The quality impairments introduced at each bit-rate are easily detectable. Furthermore, it was ensured prior to the listening test that all individuals in the listening panel were familiar with the test material and that all individuals could detect the artefacts present in the coded material. For this reason, no hidden reference was included in the test. Version RM3 of PPC (Reference Model 3 of PPC in MPEG-4 Extension 2), operating at 24 kb/s mono, is included as an anchor in the test. This coder is referred to as "PPC 24 kb/s mono." Two additional anchors, being low-pass filtered versions of the reference, are also included. The first anchor, referred to as "Anchor 3.5 kHz mono," has a 3.5 kHz bandwidth. The second anchor, "Anchor 7 kHz mono," has a 7 kHz bandwidth. Therefore, each trial contains eight stimuli.

Per individual, only one session, containing 10 trials, was conducted, where each excerpt was presented twice. The CRC-SEAQ tool, illustrated in Figure 5.2, was used to control the session. In each trial, the reference and eight stimuli, described above, are presented. On average, an individual required 45 minutes to complete the session. Eleven experienced individuals, with ages ranging from 25 to 38 years, participated in the test.

The tests were conducted in a sound attenuated listening room with only one individual at a time, and with the use of *Stax Lambda Pro* headphones and a *Sound Enhancer IS 5022 Professional Mk ISP* D/A converter unit.

5.4 Results

A summary of the overall results is provided in Figure 5.3. The mean score and 95% confidence intervals at each bit-rate and for the anchors, given in this figure, are determined in the following manner. First, the mean score for each presentation is calculated according to

$$\bar{\mu}_{jk} = \frac{1}{N} \sum_{i=1}^{N} \mu_{ijk},$$

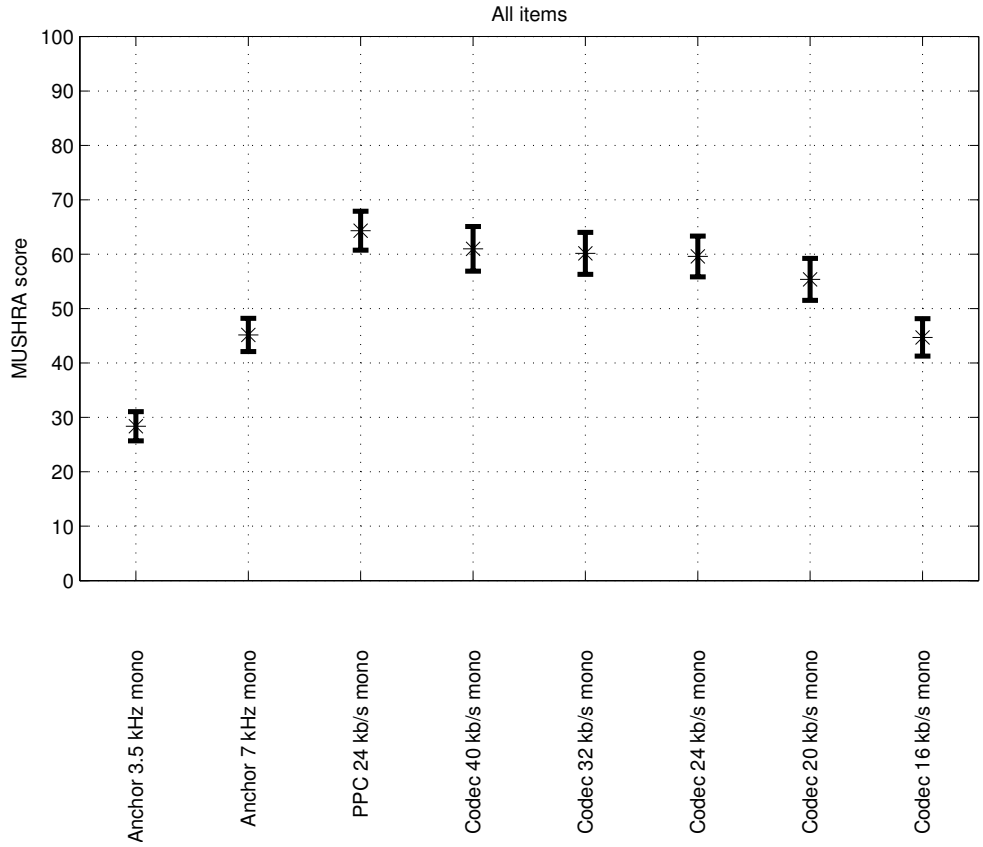


Figure 5.3: *Summary of all scores. The mean and 95% confidence intervals are given.*

where N is the number of individuals and the score given by individual i for codec (or anchor) j and excerpt k is denoted by μ_{ijk}. The overall mean scores $\bar{\mu}_j$ and $\bar{\mu}_k$ are calculated in a similar manner. Second, the standard deviation of each presentation is calculated according to

$$\sigma_{jk} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\bar{\mu}_{jk} - \mu_{ijk})^2}.$$

Finally, 95% confidence intervals are calculated as

$$\left[\bar{\mu}_{jk} - 1.96\frac{\sigma_{jk}}{\sqrt{N}}, \bar{\mu}_{jk} + 1.96\frac{\sigma_{jk}}{\sqrt{N}}\right].$$

From Figure 5.3, we observe that the 3.5 kHz anchor was rated as "poor," the 7 kHz anchor as "fair," and the PPC anchor as "good." The quality of Codec 40 kb/s

mono, Codec 32 kb/s mono, and Codec 24 kb/s mono is rated as "good" to "fair," while the quality of Codec 20 kb/s mono and Codec 16 kb/s mono is rated as "fair."

The bit-rate scalable codec is rated significantly higher than the 3.5 kHz anchor at all bit-rates. Furthermore, Codec 24 kb/s mono to Codec 40 kb/s mono are rated significantly higher than the 7 kHz anchor and are not significantly different from the PPC anchor. Codec 20 kb/s mono is rated significantly higher than the 7 kHz anchor and lower than the PPC anchor. Codec 16 kb/s mono is rated significantly lower than codec 20 kb/s mono and is rated as being similar to the 7 kHz anchor.

We observe that the largest improvement in audio quality is achieved by increasing the bit-rate from 16 to 20 kbits/s. Further increases in bit-rate do not lead to a significant increase in audio quality. At 24 kbits/s and higher, the bit-rate scalable codec is comparable in quality to PPC.

The Orchestral piece and Contemporary pop music are considered to be critical excerpts for parametric audio coders. In Sections 5.4.1 and 5.4.2, the results for these two excerpts are given and discussed. Appendix B contains the results for the remaining excerpts as well as the results for all excerpts at each bit-rate.

5.4.1 Orchestral piece

The results for the Orchestral piece excerpt are given in Figure 5.4. The audio quality attained by Codec 40 kb/s mono is rated as "good," significantly higher than the quality attained by the PPC anchor. Codec 32 kb/s mono is also rated as "good," and the difference in quality with regard to Codec 40 kb/s mono is not significant. Codec 24 kb/s mono is rated as "good" to "fair," in the same range as the PPC anchor. Codec 20 kb/s mono is rated as "fair," and is significantly lower in quality than Codec 32 kb/s mono. Codec 16 kb/s mono is rated as "fair" to "poor," in the same range as the 7 kHz anchor. The quality attained by Codec 16 kb/s mono is significantly higher than the quality of the 3.5 kHz anchor.

The improvement in quality attained by increasing the bit rate is a result of the rich tonal content, characterised by a large number of partials. Most of the partials in this excerpt are non-harmonically related, and individual sinusoids are therefore utilised to model them. Furthermore, the improvement in quality indicates that the Noise Adaptation module is able to adapt the noise component in such a way that a well-tuned trade-off between sinusoids and noise is achieved at all bit rates considered.

5.4.2 Contemporary pop music

The results for the Contemporary pop music excerpt are given in Figure 5.5. The audio quality attained by Codec 40 kb/s mono, Codec 32 kb/s mono, and Codec 24 kb/s mono is rated as "good." The quality obtained at these bit-rates is not significantly

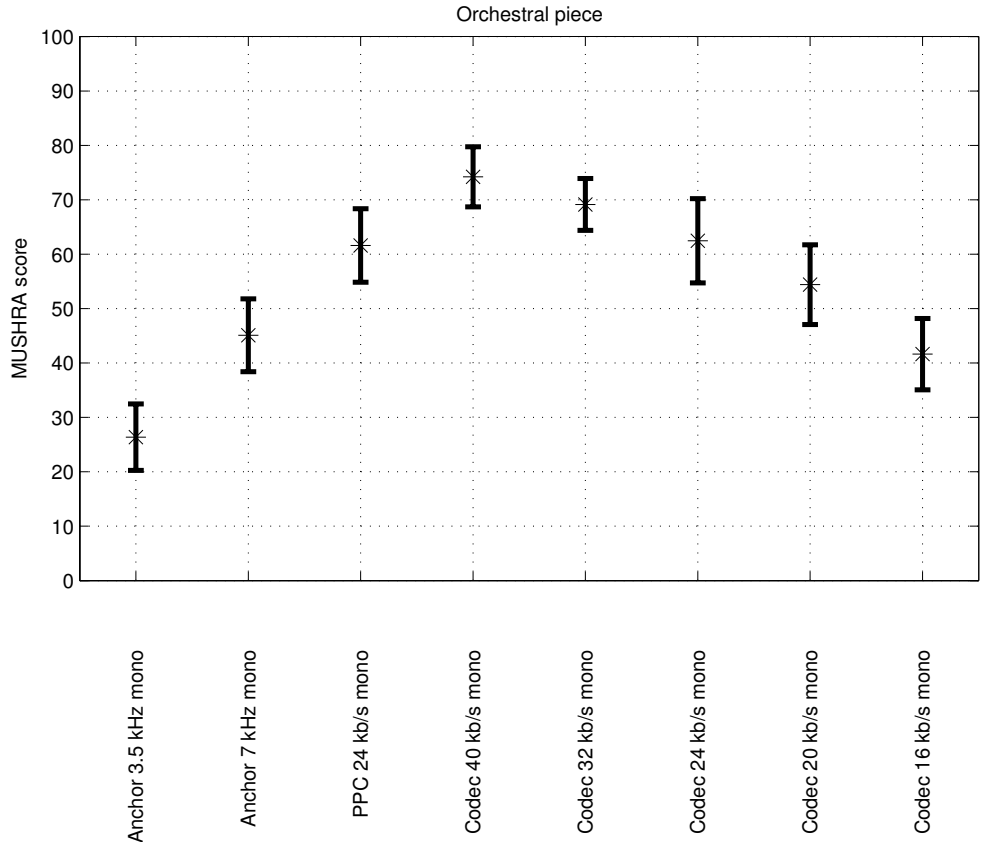


Figure 5.4: *Orchestral piece (sc02). All codecs and anchors.*

different from the quality attained by the PPC anchor. The audio quality of Codec 20 kb/s mono is rated as "good" to "fair," significantly higher than the 7 kHz anchor. Codec 16 kb/s mono is rated as "fair," significantly higher in quality than the 3.5 kHz anchor. Furthermore, even though Codec 16 kb/s mono is not rated significantly higher than the 7 kHz anchor, most individuals rated Codec 16 kb/s higher than the 7 kHz anchor.

From Figure 5.5, we observe that the audio quality increases from 16 to 24 kbits/s, after which the quality saturates. The improvement in quality attained from 16 to 24 kbits/s indicates that the tonal content in the excerpt is sufficiently modelled by sinusoids at 24 kbits/s. Adding more sinusoids does not lead to an improvement in quality. This excerpt contains a number of transients, and the absence of a transient component prohibits a further increase in quality. As in the case of the Orchestral piece (Figure 5.4), the increase in audio quality observed from 16 to 24 kbits/s indi-

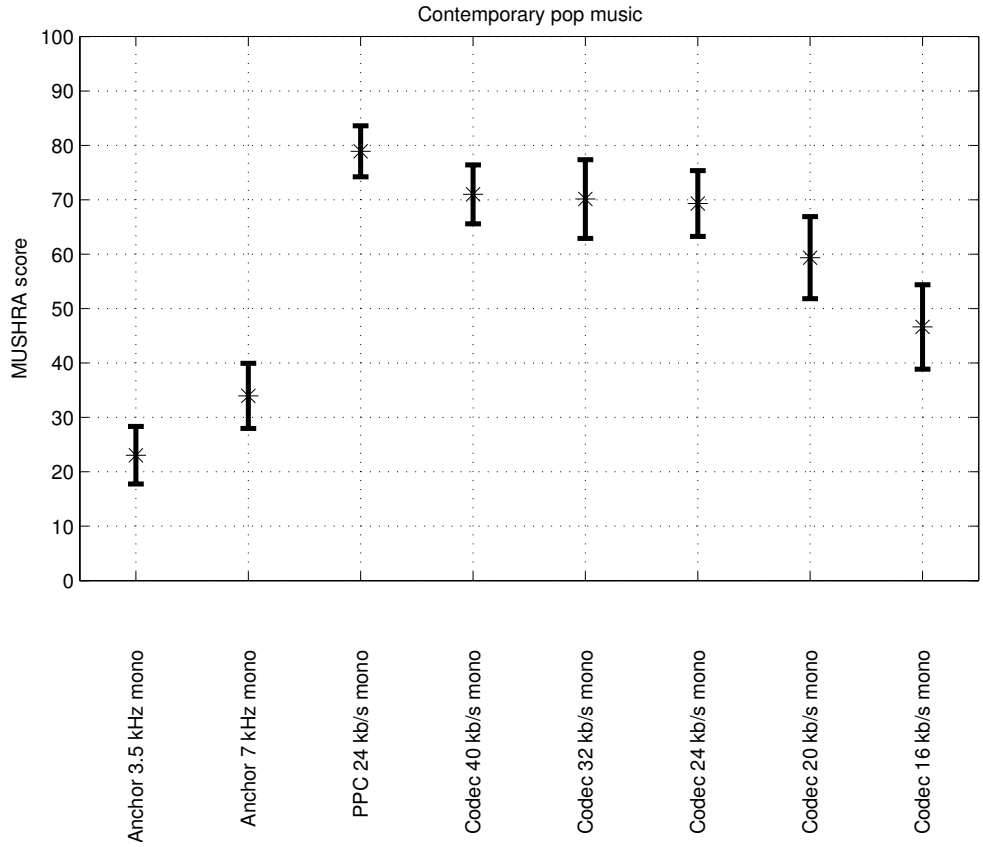


Figure 5.5: *Contemporary pop music (sc03). All codecs and anchors.*

cates that the Noise Adaptation module is capable of appropriately adapting the noise component such that a well-tuned trade-off between sinusoids and noise is obtained.

5.5 Discussion

The bit-rate scalable coder described in this thesis is able to maintain a "fair" to "good" audio quality in the range 16 to 40 kbits/s. The audio quality, averaged over all fragments, does not improve above 24 kbits/s, while only a modest degradation in quality takes place as the bit-rate is lowered to 16 kbits/s.

With exception of the German male excerpt, see Figure B.1 on page 161, the audio quality delivered by the bit-rate scalable coder is comparable or higher than the quality of the 7 kHz anchor at all bit-rates, and certainly higher than the quality

of the 3.5 kHz anchor. In a similar test conducted by the EBU on Internet audio codecs, all codecs scored significantly lower than the 3.5 kHz anchor at a bit-rate of 16 kbits/s mono [40]. Most of the codecs used in that test were waveform coders. We note that different test material was used and a different group of individuals participated in that test. However, regardless of these differences, this outcome at 16 kbits/s illustrates the potential of parametric audio coding at low bit rates.

To translate the overall results given in Figure 5.3 into a bit-rate versus audio-quality plot, an estimate of the bit-rate of the 3.5 kHz and 7 kHz anchors is made, where we assume that a waveform coder is applied to obtain transparent quality at these bandwidths. Johnston combined a spectral masking model with signal quantisation principles, resembling the approach taken in transform coders, to define perceptual entropy [129]. The perceptual entropy, expressed as a number of bits per sample, represents a lower bound on the bit rate required to obtain transparent audio quality. Johnston reported estimates of the perceptual entropy for narrow-band speech material, sampled at 8 kHz, up to wide-band audio, sampled at 32 kHz, of around 2.1 bits per sample [129, 66]. As far as real audio coders are concerned, we observe that at 96 kbits/s mono and a sampling frequency of 44.1 kHz, MP3 delivers near-transparent audio quality. This corresponds to $\approx$2.25 bits per sample, conform the estimate of perceptual entropy. With 2.25 bits per sample and a sampling frequency of 7 kHz, the bit-rate of the 3.5 kHz anchor is estimated as 15.8 kbits/s. Similarly, the bit-rate of the 7 kHz anchor is estimated as 31.5 kbits/s. We note that these are rough estimates only. Given these estimates, a bit-rate versus quality plot is given in Figure 5.6. From this figure, we observe that the bit-rate scalable codec clearly outperforms both the 3.5 and 7 kHz anchors at comparable bit-rates, which shows that bandwidth reduction is not the optimal approach to achieving a bit-rate reduction.

As we have observed before, the quality of the bit-rate scalable coder does not increase above 24 kbits/s. This indicates that an alternative approach should be applied in order to increase the audio quality above 24 kbits/s. In the bit-rate scalable parametric audio coder developed by Verma [19], phase parameters are encoded in the bit stream above 20 kbits/s. By adding tracks, which include absolutely encoded phase parameters for all sinusoids, to the bit stream, a maximum bit rate of 80 kbits/s is attained. The drawback of this approach is that differential encoding of phase parameters yields practically no coding gain, and the absolutely encoded phase parameters consume large portions of the bit rate as a result. The development of a new technique by den Brinker et al. [122] to code the unwrapped phase cost-efficiently, thereby requiring 2 bits per unwrapped phase parameter, is a promising alternative to overcome the high cost of encoding phase parameters. However, we do not believe that parametric audio coding will outperform waveform coding at high bit rates, even if the unwrapped phase of sinusoids is encoded.

We paid no attention to the coding of stereophonic audio in the design of the

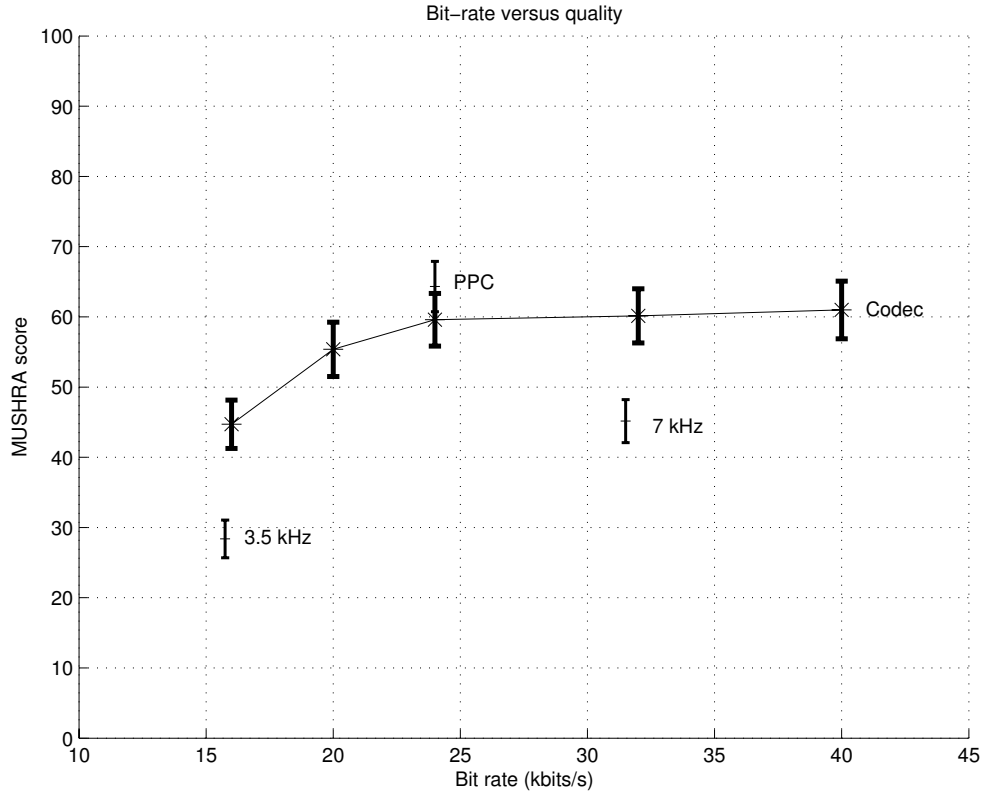


Figure 5.6: *Bit-rate versus quality for all anchors and the bit-rate scalable codec at the various bit-rates.*

scalable codec. Certainly, for audio codecs operating at bit rates of 32 kbits/s and above, stereo coding is an important functionality. While several techniques to code stereophonic audio are utilised in waveform coders, like Mid/Side [130] and Intensity stereo coding [131, 132], these techniques are not directly applicable to parametric audio coding[1]. More suitable parametric representations of stereo and multi-channel

[1] In *Mid/Side* stereo coding, the left (L) and right (R) channels are combined to form a mid channel ($M = \frac{L+R}{2}$) and a side channel ($S = \frac{L-R}{2}$), after which both the mid and side channels are coded by a waveform coder. In parametric audio coding, where a waveform match of the mid and side channels in the decoder can not be guaranteed due to phase continuation, Mid/Side stereo coding is likely to cause artefacts.

In waveform coding, *Intensity* stereo coding is applied to higher-frequency spectral bands; these spectral bands are obtained after the application of an analysis filterbank. In contrast to waveform coders, parametric audio coders do not employ such an analysis filterbank; instead, they model narrow-band signal components (sinusoids) across the whole spectrum, while spectral bands are at most utilised in some noise-coding strategies, see e.g. [18, 22].

audio signals are evolving. One such representation is Binaural Cue Coding (BCC), where a multi-channel signal is represented by a monaural audio signal and parameters comprising interaural level difference (ILD) and interaural time difference (ITD) measurements [133]. A more general parametric representation, additionally containing interaural cross-correlation (ICC) measurements, was utilised to enable PPC to efficiently code stereophonic audio at 32 kbits/s, where the stereo parameters required between 2.7 and 6.2 kbits/s [73]. The advantage of these parametric representations is that both a monaural signal and a set of stereo parameters are derived from the stereo (or multi-channel) signal. The monaural signal can then be coded by a parametric audio encoder, and the stereo parameters placed in the bit stream. After applying the parametric decoder, the stereo parameters are applied to the decoded (monaural) signal to obtain a stereo or multi-channel decoded signal. The parametric representation of stereo signals described in [73] can be applied to the scalable codec described in this thesis to enable stereo coding at an additional bit rate of between 2.7 and 6.2 kbits/s.

The quality achieved by PPC and the bit-rate scalable codec is comparable, even though the encoder structures are not similar. In the following, the main differences between these codecs are highlighted.

Transients The most prominent difference is that the bit-rate scalable codec does not utilise a transient signal component. As a result, excerpts in which prominent transients are present, like castanets (si02), are encoded with low quality by the bit-rate scalable codec. The contemporary pop music excerpt (sc03) contains a number of transients, and the feedback from individuals in the listening panel made clear that the lack of transients resulted in a decoded signal with less "punch."

Harmonic complex PPC does not utilise a harmonic complex. The harmonic complex is advantageous at low bit-rates, since it allows a low-cost parameterisation of harmonic sounds. The results for the speech excerpt (es02), given in Figure B.1 in the appendix on page 161, illustrates the advantage at low bit-rates.

Optimisation The Levenberg-Marquardt optimisation method utilised in the bit-rate scalable codec yields more accurate parameter estimates than those obtained by the parameter estimation utilised in PPC [134].

Tuning The analysis process in the bit-rate scalable codec as a whole, is not optimally tuned. In particular, we feel that the time scales chosen for the analysis of individual sinusoids require more attention. (The time scales are given in Table 3.4 on page 79.)

The German male speech excerpt (es02) received the lowest score of all excerpts, see Figure B.1 on page 161. The audio quality of the bit-rate scalable codec, at all bit-rates, received a similar rating as the 3.5 kHz and PPC anchors. This indicates that an alternative approach should be adopted to improve the quality of speech material.

In order to assess in how far the preference expressed for parametric audio coding over bandwidth reduction by the complete group is due to individuals being familiar with parametric audio coding, we classify two sub-groups of individuals. Of the eleven individuals, seven are familiar with parametric audio coding and the typical artefacts introduced by parametric audio coders. The scores given by these seven individuals are compared to the scores given by the remaining individuals in Figure 5.7. From this figure, we observe that the results for both sub-groups show the same trend.

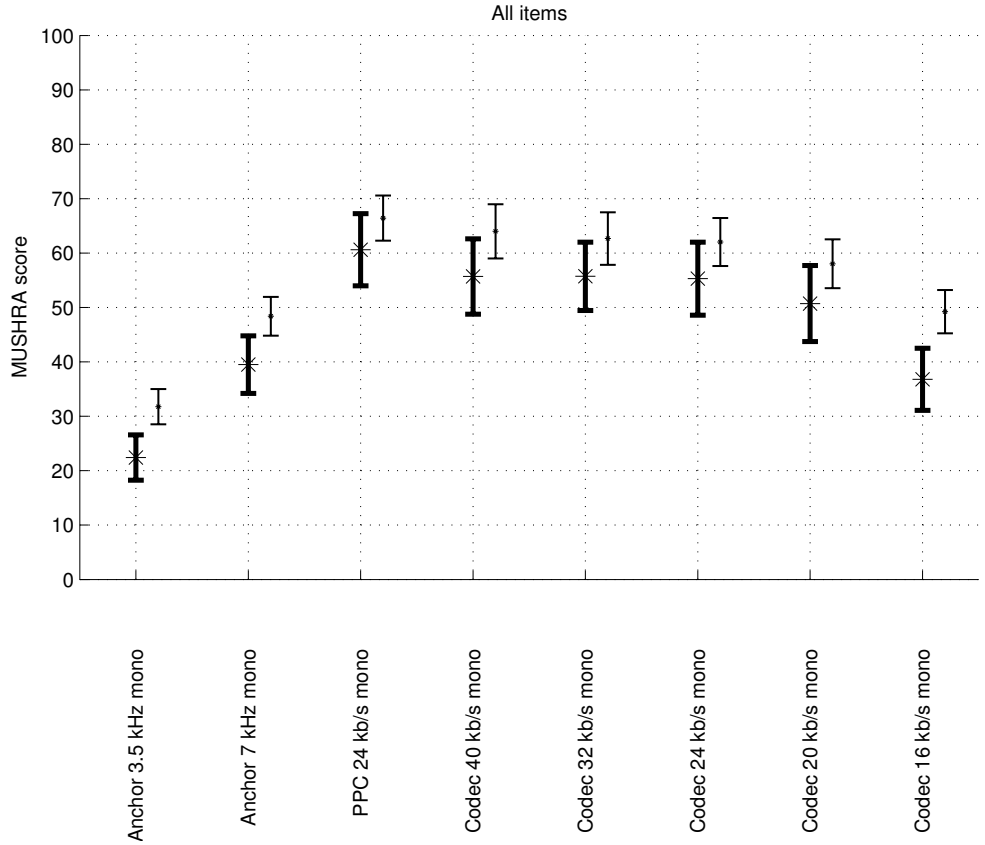


Figure 5.7: *The results for the sub-group of four individuals not familiar with parametric audio coding are given in thick lines. The results for the remaining seven individuals are given in thin lines.*

In particular, the audio quality of Codec 16 kb/s mono is rated significantly higher than the 3.5 kHz anchor and is rated as being similar to the 7 kHz anchor, for both sub-groups. Furthermore, the sub-group of individuals familiar with parametric audio coding gave less critical scores than those individuals not familiar with parametric audio coding.

5.6 Conclusions

The bit-rate scalable parametric audio codec delivers the same audio quality at a bit rate of 24 kbits/s as a state-of-the-art parametric coder dedicated to operate at the same bit rate, even though the scalable codec utilises no transient component in its current form. Therefore, a parametric audio coder can be made bit-rate scalable without sacrificing audio quality.

The range over which an increase in quality is obtained with the bit-rate scalable codec is 16 to 24 kbits/s. Above 24 kbits/s, an alternative approach should be applied to increase the audio quality. At 24 kbits/s and lower, a graceful degradation in audio quality is achieved, proving that the bit-rate scalable codec, in its current form, is able to maintain a high audio quality at lower bit-rates.

Full-band parametric audio coding at low bit-rates delivers a higher audio quality than bandwidth reduction applied to obtain a low bit-rate. Waveform coders usually apply bandwidth reduction to obtain low bit-rates. Therefore, we conclude that parametric audio coding outperforms waveform coding at low bit-rates.

5.7 Recommendations

Given the increase in quality observed above 16 kbits/s, one should optimise the bit-rate scalable codec at this bit-rate. In this way, the audio quality attained at higher bit-rates will be increased.

Waveform coding is a proven method to obtain high audio quality at high bit-rates. For this reason, the bit-rate scalable parametric audio coder should be combined with waveform coding of the residual above 24 kbits/s to improve the audio quality.

The audio quality attainable with the current mechanism at bit-rates lower than 16 kbits/s should be investigated further. Results from informal listening experiments indicate that the bit-rate of the noise component should be lowered to maintain a reasonable quality at lower bit-rates.

To improve the audio quality of speech material, the phase parameters should be encoded. The transmission of phase parameters in addition to frequency parameters is not a cost-efficient encoding method. Coding of the unwrapped phase is a more cost efficient technique [122].

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

The research leading to the design discussed in this thesis was assigned by and carried out at Philips Research in Eindhoven, the Netherlands, in the period June 2000 to October 2003 as a follow-up of Myburg's final project of the designers programme, Mathematics for Industry, at Technische Universiteit Eindhoven.

We described a bit-rate scalable parametric audio codec which is capable of encoding both audio and speech signals in the range 16 to 40 kbits/s. In the design of the coder, the following topics received attention.

The sinusoidal component was divided into a harmonic-complex and individual-sinusoids sub-components, and suitable models of both sub-components were defined. Iterative optimisation techniques were utilised to estimate the model parameters of the sinusoidal component. It was shown, by considering both synthetic and real-world signals, that the optimisation techniques are capable of improving initial estimates of the sinusoidal parameters. Furthermore, informal listening experiments revealed that the improved sinusoidal parameter estimates result in a higher audio quality than the quality obtained when the initial sinusoidal parameter estimates are utilised. We observed that the optimisation techniques suffer from high computational complexity. Suitable approaches for lowering the computational complexity were applied.

The spectral envelope of the residual component was modelled by Laguerre-based linear prediction, while the temporal envelope of the prediction residual was measured with a resolution matched to the temporal resolution of the human auditory system.

Quantisation strategies were developed and applied to the model parameters. The quantised model parameters were encoded either absolutely or time-differentially, and entropy coding was applied to exploit the redundancy.

To satisfy the bit-rate requirements of the layers comprising the scalable bit-stream, we chose to distribute the sinusoidal component over the base and refinement layers, and to place the noise component in the base layer. The choice of which si-

nusoidal parameters to transmit, and in which layer, was made such that the highest possible audio quality for each layer was obtained. We parameterised each complete track by a rate-distortion pair. Complete tracks were distributed over the layers in order to minimise the distortion at each rate. We have shown that the optimisation mechanism tended to select longer tracks at low bit-rates. Although the prototype does not support the creation of a bit-stream, we provided a suitable bit-stream syntax in this thesis. In the design of the decoder, we strived to maintain a well-tuned trade-off between sinusoids and noise by matching the noise component to the sinusoidal component through the means of a band-rejection filter.

The performance of the scalable codec was tested by conducting MUSHRA listening tests. The results from the listening test show that the scalable codec delivers the same audio quality at 24 kbits/s as a state-of-the-art parametric coder dedicated to operate at the same bit rate. The range over which an increase in quality is obtained with the bit-rate scalable codec is 16 to 24 kbits/s. In the range 24 to 40 kbits/s, no increase in quality was observed. The results from the listening test also show that parametric audio coding outperforms waveform coding at low bit-rates.

6.2 Recommendations

The main recommendations following from the research described in this thesis are:

- The prototype bit-rate scalable parametric audio coder should be fine-tuned and tested on more excerpts.

- The audio quality attainable at bit-rates below 16 kbits/s should be investigated further. Lowering the cost of the noise component below 16 kbits/s is the most promising strategy.

- Further attention should be paid to lowering the computational complexity of the optimisation methods used to refine the parameter estimates of the sinusoidal component.

- Coding of the unwrapped phase of sinusoids is recommended as a suitable strategy to improve the quality of speech material.

- The bit-rate scalable parametric audio codec has to be combined with waveform coding of the residual above 24 kbits/s.

- The sinusoidal and noise signal components utilised in the bit-rate scalable parametric audio codec should be supplemented with a transient component to improve the modelling of transients.

- The bit-stream syntax proposed in this thesis should be extended to cater for the transmission of the transient parameters, unwrapped phase, and waveform information.
- The programme code in MATLAB should be ported to a programming language more suitable for commercial applications, like the C programming language [135].

Appendix A

Optimisation tools

Appendix to Chapter 3

In this appendix, an overview of the Gauss-Newton and Levenberg-Marquardt optimisation methods is given. Since the Levenberg-Marquardt method is an extension of the classical Gauss-Newton method, we start by describing the Gauss-Newton method. Section A.1 considers the Gauss-Newton optimisation method while Section A.2 describes the Levenberg-Marquardt optimisation method.

A.1 Gauss-Newton optimisation

The Gauss-Newton method utilises the linearised version (or affine model) of the non-linear model around estimates of the model parameters. We denote the non-linear model by $\mathbf{s}_{\texttt{model}}(\mathbf{x}) \in \mathbb{C}^{N_\text{s} \times 1}$, where the column vector

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^\text{T} \tag{A.1}$$

contains the N model parameters. The estimates of the model parameters are contained in

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \dots & \hat{x}_N \end{bmatrix}^\text{T}, \tag{A.2}$$

and the estimation errors in

$$\boldsymbol{\Delta}_\mathbf{x} = \begin{bmatrix} x_1 - \hat{x}_1 & x_2 - \hat{x}_2 & \dots & x_N - \hat{x}_N \end{bmatrix}^\text{T} = \begin{bmatrix} \Delta_{x_1} & \Delta_{x_2} & \dots & \Delta_{x_N} \end{bmatrix}^\text{T}. \tag{A.3}$$

The linearised version of $\mathbf{s}_{\texttt{model}}$ around the model-parameter estimates $\hat{\mathbf{x}}$ is

$$\mathbf{s}_{\texttt{model,lin}}(\mathbf{x}) = \mathbf{s}_{\texttt{model}}(\hat{\mathbf{x}}) + J(\hat{\mathbf{x}}) \cdot \boldsymbol{\Delta}_\mathbf{x}, \tag{A.4}$$

where the estimation errors $\boldsymbol{\Delta}_\mathbf{x}$ appear linearly in this expression, and where the Jacobian $J(\hat{\mathbf{x}}) \in \mathbb{C}^{N_\text{s} \times N}$ is given by

$$J(\hat{\mathbf{x}}) = \begin{bmatrix} \left(\frac{\partial \mathbf{s}_{\texttt{model}}}{\partial x_1}\right)(\hat{\mathbf{x}}) & \left(\frac{\partial \mathbf{s}_{\texttt{model}}}{\partial x_2}\right)(\hat{\mathbf{x}}) & \dots & \left(\frac{\partial \mathbf{s}_{\texttt{model}}}{\partial x_N}\right)(\hat{\mathbf{x}}) \end{bmatrix}. \tag{A.5}$$

The linearised $\mathbf{s}_{\texttt{model,lin}}(\mathbf{x})$ is a reasonable approximation of the non-linear $\mathbf{s}_{\texttt{model}}(\mathbf{x})$ in the direct vicinity of the model parameter estimates $\hat{\mathbf{x}}$ only. The errors $\boldsymbol{\Delta}_{\mathbf{x}}$ are estimated by solving the linear least-squares problem

$$\begin{aligned} &\min_{\boldsymbol{\Delta}_{\mathbf{x}}} \|\mathbf{s} - \mathbf{s}_{\texttt{model,lin}}(\mathbf{x})\|_{\mathbf{w}}^2 \\ \Longleftrightarrow \quad & H\hat{\boldsymbol{\Delta}}_{\mathbf{x}} = \mathbf{P}, \end{aligned} \tag{A.6}$$

where the system matrix $H \in \mathbb{C}^{N\times N}$ is given by

$$H = J(\hat{\mathbf{x}})^{\mathrm{H}} \cdot \mathrm{diag}(\mathbf{w}) \cdot J(\hat{\mathbf{x}}), \tag{A.7}$$

and

$$\mathbf{P} = J(\hat{\mathbf{x}})^{\mathrm{H}} \cdot \mathrm{diag}(\mathbf{w}) \cdot (\mathbf{s} - \mathbf{s}_{\texttt{model}}(\hat{\mathbf{x}})), \tag{A.8}$$

according to the definition of the inner product given in Equation (3.5) on page 42. The Cholesky decomposition is a suitable method for solving the normal equations in (A.6). The solution yields $\hat{\boldsymbol{\Delta}}_{\mathbf{x}}$, and the estimates contained in $\hat{\mathbf{x}}$ are updated as $\hat{\mathbf{x}} := \hat{\mathbf{x}} + \hat{\boldsymbol{\Delta}}_{\mathbf{x}}$, after which the process is repeated. The main drawback of the Gauss-Newton method is its failure to converge in some cases.

A.2 Levenberg-Marquardt optimisation

The main difference between the Gauss-Newton and Levenberg-Marquardt methods is that regularisation of the system matrix H, by adding the term λI to H,

$$(H + \lambda I)\hat{\boldsymbol{\Delta}}_{\mathbf{x}} = \mathbf{P}, \tag{A.9}$$

is applied. Regularisation has two key advantages. *Firstly*, in the case that H is ill-conditioned, regularisation will lower its condition number. The eigenvalues of H provide a convenient way of measuring the conditioning of H since H is positive semidefinite. The condition number of H is then given by the ratio of its largest and smallest eigenvalues

$$\kappa_H = \frac{\mu_{\texttt{max}}}{\mu_{\texttt{min}}}.$$

The condition number of the regularised matrix $H + \lambda I$, with $\lambda > 0$, is then given by

$$\kappa_{H+\lambda I} = \frac{\mu_{\texttt{max}} + \lambda}{\mu_{\texttt{min}} + \lambda} < \kappa_H.$$

However, it is worthwhile to keep the condition number of H in mind, and to find ways of lowering it to avoid unnecessary regularisation. *Secondly*, for a properly chosen λ at each iteration step, the cost function will decrease at each iteration step.

Thus, the Gauss-Newton algorithm described in the previous section is extended as follows. Denote the model-parameter estimates from the previous iteration $i-1$ by $\hat{\mathbf{x}}^{(i-1)}$. The cost function at the end of iteration step $i-1$ is

$$c^{(i-1)} = \|\mathbf{s} - \mathbf{s}_{\mathtt{model}}(\hat{\mathbf{x}}^{(i-1)})\|^2_{\mathbf{w}}. \tag{A.10}$$

During iteration step i, the parameters are improved by solving the normal equations

$$(H^{(i)} + \lambda I)\hat{\mathbf{\Delta}}^{(i)}_{\mathbf{x}} = \mathbf{P}^{(i)}, \tag{A.11}$$

where a suitable $\lambda = \lambda^{(i)}$ has to be chosen. For each λ, the cost function associated with the updated parameter estimates is

$$c_i(\lambda) = \|\mathbf{s} - \mathbf{s}_{\mathtt{model}}(\mathbf{x}^{(i-1)} + \hat{\mathbf{\Delta}}^{(i)}_{\mathbf{x}}(\lambda))\|^2_{\mathbf{w}}. \tag{A.12}$$

The suitable $\lambda^{(i)}$ is the smallest λ for which $c_i(\lambda^{(i)}) \leq c^{(i-1)}$. To this end, the strategy proposed by Marquardt in [102] is employed. Compute $c_i(\lambda)$ for both $\lambda = \lambda^{(i-1)}/\nu$ and $\lambda = \lambda^{(i-1)}$, where $\nu > 1$ a constant. The choice of ν is arbitrary, a value of $\nu = 10$ has been found to be a good choice in practise. If

1. $c_i(\lambda^{(i-1)}/\nu) \leq c^{(i-1)}$, let $\lambda^{(i)} = \lambda^{(i-1)}/\nu$.
2. $c_i(\lambda^{(i-1)}/\nu) > c^{(i-1)}$ and $c_i(\lambda^{(i-1)}) \leq c^{(i-1)}$, let $\lambda^{(i)} = \lambda^{(i-1)}$.
3. $c_i(\lambda^{(i-1)}/\nu) > c^{(i-1)}$ and $c_i(\lambda^{(i-1)}) > c^{(i-1)}$, multiply $\lambda^{(i-1)}$ by powers of ν until, for some smallest m, $c_i(\lambda^{(i-1)}\nu^m) \leq c^{(i-1)}$. Let $\lambda^{(i)} = \lambda^{(i-1)}\nu^m$.

We note that this algorithm is not attractive if Step 3 has to be carried out many times with a large system matrix $H^{(i)}$, since solving the normal equations in (A.11), for each choice of λ, is computationally inefficient. More suitable algorithms for this scenario include the dogleg algorithm of Powell [136]. However, we will divide the complete non-linear problem into a number of smaller non-linear problems to reduce the computational burden. The system matrix, resulting from each sub-problem, can then be solved with little effort, making the algorithm proposed by Marquardt still attractive. Reducing the computational burden is described in Section 3.3.4 on page 62. To make this algorithm feasible for practical implementation, it is worthwhile to limit λ by $\lambda_{\mathtt{min}} \leq \lambda \leq \lambda_{\mathtt{max}}$. When the parameter estimates $\mathbf{x}^{(i-1)}$ are very close to a minimum of the cost function $c^{(i-1)}$, Step 3 of the algorithm will be repeated many times without any real practical benefit. Hence the upper bound $\lambda_{\mathtt{max}}$ on λ. The lower bound $\lambda_{\mathtt{min}}$ on λ should be chosen such that a small condition-number of the system matrix H is guaranteed. A value of $\nu = 10$ and an initial value of $\lambda^{(0)} = 10^{-2}$ were suggested by Marquardt [102]. The iterations have converged when

$$\max_k \frac{|\hat{\Delta}^{(i)}_{x_k}|}{|\hat{x}^{(i)}_k|} \leq \Delta_{\mathtt{min}}. \tag{A.13}$$

A suitable value for $\Delta_{\mathtt{min}}$ is 10^{-5}.

Appendix B

Listening test results

Appendix to Chapter 5

The remainder of the results from the listening test are given in this appendix. Figure B.1 presents the results for the German male speech excerpt.

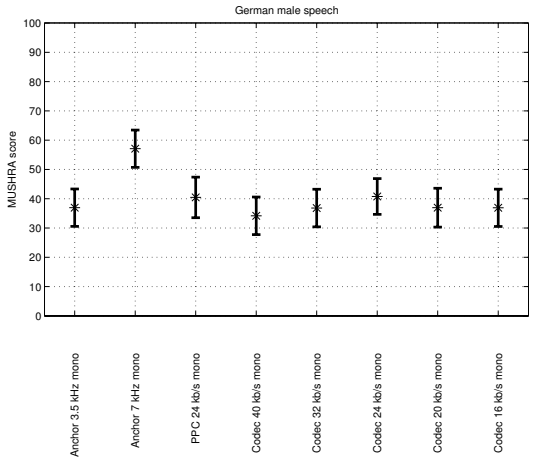


Figure B.1: *German male speech (es02). All codecs and anchors.*

The audio quality of the bit-rate scalable parametric audio codec did not improve with increasing bit-rate. The audio quality attained is comparable to that of the PPC and 3.5 kHz anchors. The 7 kHz anchor is rated significantly higher than the bit-rate scalable codec. For no other excerpt did the scalable codec attain a poorer audio quality over all bit rates. The inability of parametric audio coders to code speech material with a high quality is illustrated by the performance of both the scalable codec and the PPC anchor. The results for this excerpt illustrate the usefulness of the harmonic complex at low bit rates. By utilising a harmonic complex, the audio quality attained by Codec 16 kb/s mono is similar to that of the PPC anchor, operating at 24 kbits/s.

The harmonic complex captures the tonal content of the speech signal, while the addition of individual sinusoids at higher bit rates does not lead to an increase in audio quality.

In Figure B.2, the results for the pitch pipe excerpt are given.

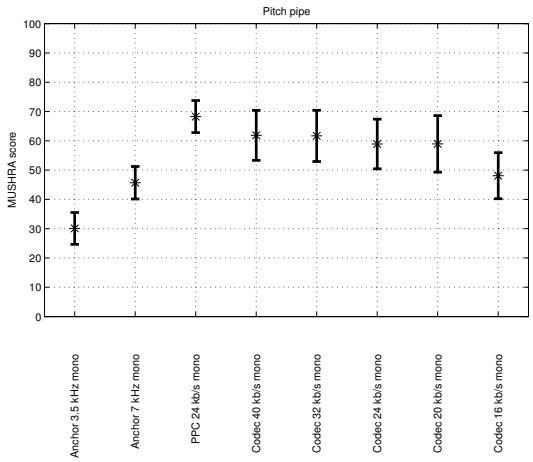


Figure B.2: *Pitch pipe (si03). All codecs and anchors.*

The audio quality attained by Codec 16 kb/s mono is significantly higher than the 3.5 kHz anchor, and comparable to the 7 kHz anchor. The audio quality attained by the scalable codec does not increase from 20 kbits/s and above, and is comparable to the PPC anchor. The audio quality attained by Codec 32 kb/s mono and Codec 40 kb/s mono is rated significantly higher than the 7 kHz anchor. The tonal aspects of this excerpt are captured by the harmonic complex. A large number of harmonics per frame are present in the intervals where a note is sounding. As a result, the restriction on the maximum number of harmonics per frame in a layer, as formulated in Section 4.2.2 on page 111, is applied to this excerpt. The penalty paid for having to estimate the lacking amplitude parameters in the decoder from the noise component, refer to Section 4.3.1 on page 121 for a description of this approach, is a slightly lower quality at 16 kbits/s than at 20 kbits/s.

In Figure B.3, the results for the bagpipes excerpt are given. The audio quality attained by Codec 16 kb/s mono is significantly higher than the 3.5 kHz anchor, and comparable to the 7 kHz anchor. The audio quality attained by the scalable codec does not increase from 20 kbits/s and above, is comparable to the PPC anchor, and is rated significantly higher than the 7 kHz anchor at these bit-rates. The restrictions on both the maximum number of harmonics per frame in the base layer and the number of tracks cause the audio quality at 16 kbits/s to be lower than the quality at 20 kbits/s.

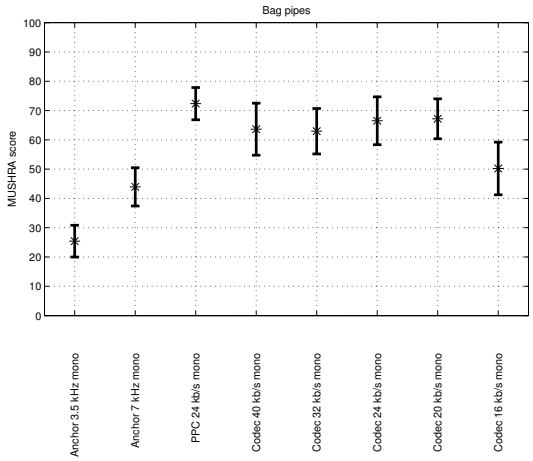


Figure B.3: *Bagpipes (sm01). All codecs and anchors.*

At 20 kbits/s, the tonal content of the signal, comprising both harmonically related and a few non-harmonically related partials, is captured by the harmonic complex and individual sinusoids. At higher bit rates, the addition of more individual sinusoids does not improve the audio quality.

In the following, the results for each bit rate are considered for all excerpts. The results for Codec 16 kb/s mono are given in Figure B.4. The German male speech

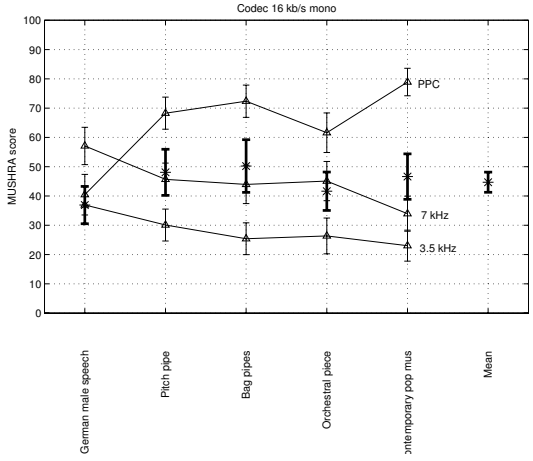


Figure B.4: *Codec 16 kb/s mono. All excerpts. The results for the anchors are given by thin lines.*

excerpt was given the lowest score, and is ranked as "poor," in the same range as the 3.5 kHz and PPC anchors. All the other excerpts were rated as "fair," in the same range as the 7 kHz anchor, and significantly lower than the PPC anchor.

The results for Codec 20 kb/s mono are given in Figure B.5.

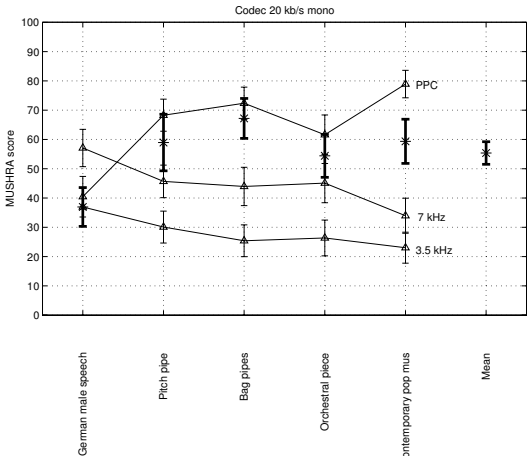


Figure B.5: *Codec 20 kb/s mono. All excerpts. The results for the anchors are given by thin lines.*

Again, the German male speech excerpt is ranked as "poor." The bag pipes excerpt is rated as "good," while the remaining excerpts are rated as "fair" to "good." The results for Codec 24 kb/s mono are given in Figure B.6. The German male speech excerpt received the lowest score, and is ranked as "poor." Only the pitch pipe is ranked as "fair" to "good," while the remainder of the excerpts are ranked as "good." The results for Codec 32 kb/s mono are given in Figure B.7. Compared to the results for Codec 24 kb/s mono, no clear improvement in quality is observed. The results for Codec 40 kb/s mono are given in Figure B.8. Compared to the results for Codec 32 kb/s mono, no clear improvement in quality is observed.

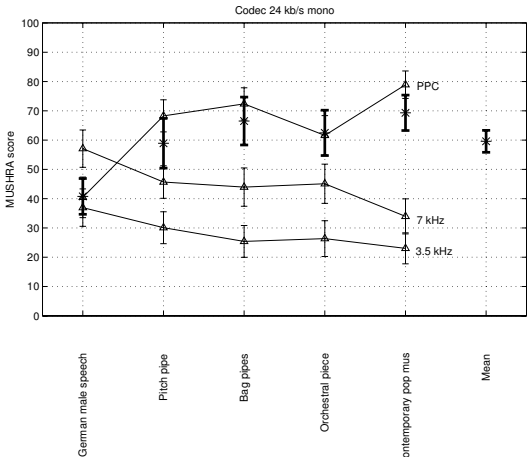


Figure B.6: *Codec 24 kb/s mono. All excerpts. The results for the anchors are given by thin lines.*

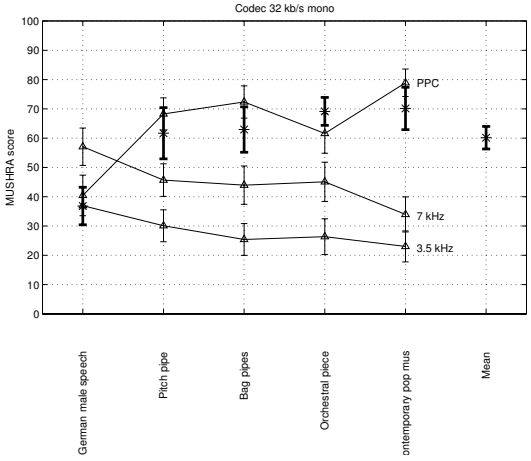


Figure B.7: *Codec 32 kb/s mono. All excerpts. The results for the anchors are given by thin lines.*

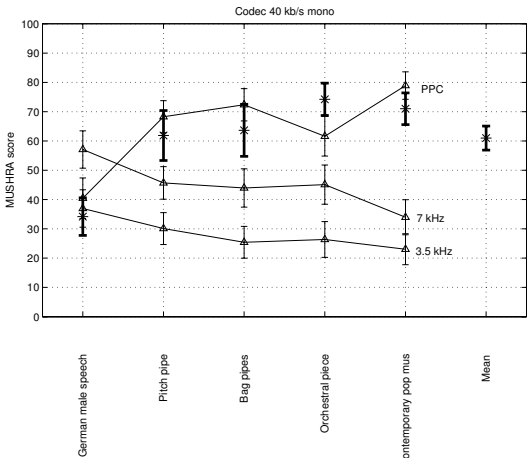


Figure B.8: *Codec 40 kb/s mono. All excerpts. The results for the anchors are given by thin lines.*

Bibliography

[1] E. Janssen and D. Reefman, "Super-Audio CD: an introduction," *IEEE Signal Processing Mag.*, vol. 20, no. 4, pp. 83–90, July 2003.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley-Interscience, 1991.

[3] H. Dudley, "Remaking speech," *J. Acoust. Soc. Am.*, vol. 11, p. 169, 1939.

[4] ——, "The vocoder," *Bell Labs. Rec.*, vol. 17, p. 122, 1939.

[5] CCITT Recommendation G.721, "32kb/s adaptive differential pulse code modulation (ADPCM)," in *Blue Book*. Comité Consultatif International Téléphonique et Télégraphique (CCITT), Oct. 1988, vol. III, Fascicle III.3.

[6] T. E. Tremain, "The government standard linear predictive coding algorithm," *Speech Technology*, pp. 40 – 49, Apr. 1982.

[7] R. Koenen (Editor), "Overview of the MPEG-4 standard," ISO/IEC, Tech. Rep. JTSC1/SC29/WG11 N4668, Mar. 2002.

[8] M. Nishiguchi, "MPEG-4 speech coding," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 139 – 146.

[9] K. Brandenburg, "MP3 and AAC explained," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 99 – 110.

[10] B. Grill, "The MPEG-4 general audio coder," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 147 – 156.

[11] H. Purnhagen and N. Meine, "HILN - the MPEG-4 parametric audio coding tools," in *ISCAS*, Geneva, May 2000, pp. III.201 – III.204.

[12] A. C. den Brinker, E. G. P. Schuijers, and A. W. J. Oomen, "Parametric coding for high-quality audio," in *AES* 112th *Convention*, Munich, Germany, May 10–13 2002, Convention Paper 5554.

[13] Audio Subgroup, "Call for proposals for new tools for audio coding," ISO/IEC, Tech. Rep. JTSC1/SC29/WG11 N3793, Jan. 2001.

[14] B. Edler and H. Purnhagen, "Concepts for hybrid audio coding schemes based on parametric techniques," in *AES* 105^{th} *Convention*, San Francisco, USA, Sept. 1998, preprint 4808.

[15] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397 – 3415, Dec. 1993.

[16] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modeling using psychoacoustic-adaptive matching pursuits," *IEEE Signal Processing Lett.*, vol. 9, no. 8, pp. 262–265, 2002.

[17] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[18] S. N. Levine, "Audio representation for data compression and compressed domain processing," Ph.D. dissertation, Department of Electrical Engineering of Stanford University, 1998.

[19] T. S. Verma, "A perceptually based audio signal model with application to scalable audio compression," Ph.D. dissertation, Department of Electrical Engineering of Stanford University, 1999.

[20] M. Ali, "Adaptive signal representation with application in audio coding," Ph.D. dissertation, University of Minnesota, 1995.

[21] B. Edler, H. Purnhagen, and C. Ferekidis, "ASAC - analysis/synthesis audio codec for very low bit rates," in *AES* 100^{th} *Convention*, Copenhagen, Denmark, May 1996, preprint 4179.

[22] M. M. Goodwin, "Adaptive signal models: Theory, algorithms, and audio applications," Ph.D. dissertation, Department of Electrical Engineering and Computer Science at the University of California, Berkeley, 1997.

[23] X. Serra, "Musical sound modelling with sinusoids plus noise," in *Musical Signal Processing*, C. Roads, S. Pope, A. Picialli, and G. D. Poli, Eds. Swets & Zeitlinger, 1997.

[24] A. C. den Brinker and A. W. J. Oomen, "Fast ARMA modelling of power spectral density functions," in *Tenth European Signal Processing Conference (EUSIPCO)*, Tampere, Finland, Sept. 4–8 2000, pp. 1229–1232.

[25] T. S. Verma and T. H. Y. Meng, "A 6kbps to 85kbps scalable audio coder," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Istanbul, Turkey, 2000, pp. 877–880.

[26] B. Feiten, R. Schwalbe, and F. Feige, "Dynamically scalable audio internet transmission," in *AES* 104^{th} *Convention*, Amsterdam, The Netherlands, May 16–19 1998, preprint 4686.

[27] H. Purnhagen, B. Edler, and N. Meine, "Error protection and concealment for HILN MPEG-4 parametric audio coding," in *AES* 110^{th} *Convention*, Amsterdam, The Netherlands, May 12–15 2001, preprint 5300.

[28] S. A. Ramprashad, "High quality embedded wideband speech coding using an inherently layered coding paradigm," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Istanbul, Turkey, 2000, pp. 1145–1148.

[29] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, no. 5, pp. 74–93, Sept. 2001.

[30] S. A. Ramprashad, "Understanding the quality losses of embedded speech and audio coders," in *IEEE Speech Coding Workshop*, Tsukuba City, Japan, Oct. 6–9 2002, pp. 11–13.

[31] L. Lastras and T. Berger, "All sources are nearly successively refinable," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 918–926, 2001.

[32] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 269–275, 1991.

[33] E. Tuncel and K. Rose, "Additive successive refinement," *IEEE Trans. Inform. Theory*, vol. 49, no. 8, pp. 1983–1991, 2003.

[34] H. Purnhagen, "An overview of MPEG-4 audio version 2," in *AES* 17^{th} *International Conference on High-Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 157–168.

[35] F. P. Myburg, "Introducing bit-rate control and scalability in a sinusoidal audio coder," Stan Ackermans Institute, Eindhoven University of Technology, Eindhoven, The Netherlands, Tech. Rep., June 2000, final report of the postgraduate programme Mathematics for Industry.

[36] ITU-R, "Methods for the subjective assessment of sound quality – general requirements," ITU-R, Tech. Rep. BS.1284, 1997, Recommendation.

[37] ——, "Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems," ITU-R, Tech. Rep. BS.1116-1, 1997, Recommendation.

[38] ——, "Method for the subjective assessment of intermediate quality level of coding systems," ITU-R, Tech. Rep. BS.1534-1, 2003, Recommendation.

[39] G. T. Waters (Editor), "Sound quality assessment material recordings for subjective tests. Users' handbook for the EBU - SQAM compact disc," Technical centre of the European Broadcasting Union, Tech. Rep. 3253-E, Apr. 1998.

[40] G. Stoll and F. Kozamernik, "EBU listening tests on internet audio codecs," *EBU Technical Review*, June 2000.

[41] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech codecs by exploiting masking properties of the human ear," *J. Acoust. Soc. Am.*, vol. 66, pp. 1647–1652, 1979.

[42] M. Karjalainen, "A new auditory model for the evaluation of sound quality of audio systems," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Tampa, USA, 1985, pp. 608–611.

[43] K. Brandenburg, "Evaluation of quality for audio encoding at low bit rates," *J. Audio Eng. Soc. (Abstracts)*, vol. 44, p. 653, May 1987, preprint 5300.

[44] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, C. Colomes, M. Keyhl, G. Stoll, K. Brandenburg, and B. Feiten, "PEAQ - the ITU-standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, vol. 48, pp. 3–29, Jan./Feb. 2000.

[45] ITU-R, "Method for objective measurements of perceived audio quality," ITU-R, Tech. Rep. BS.1387-1, 2001, Recommendation.

[46] W. C. Treurniet and G. A. Soulodre, "Evaluation of the ITU-R objective audio quality measurement method," *J. Audio Eng. Soc.*, vol. 48, no. 3, pp. 164–173, 2000.

[47] G. A. Soulodre, T. Grusec, M. Lavoie, and L. Thibault, "Subjective evaluation of state-of-the-art two-channel audio codecs," *J. Audio Eng. Soc.*, vol. 46, no. 3, pp. 164–177, 1998.

[48] H. Fletcher, "Normal vibration frequencies of a stiff piano string," *J. Acoust. Soc. Am.*, vol. 36, pp. 203–209, 1964.

[49] *MATLAB® The Language of Technical Computing*, Natic (MA), 2002.

[50] F. P. Myburg, "User's manual for the bit-rate scalable parametric audio coder," Philips Research Laboratories, Eindhoven, The Netherlands, Tech. Rep. Nat.Lab. Technical Note 2003/00592, 2003.

[51] E. B. George and M. J. T. Smith, "A new speech coding model based on a least-squares sinusoidal representation," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Dallas, USA, 1987, pp. 1641–1644.

[52] J. L. Flanagan and R. M. Golden, "Phase vocoder," Bell Laboratories, Tech. Rep. Bell Syst. Tech. J. 45, 1966.

[53] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 8, pp. 1223–1235, Aug. 1988.

[54] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, Department of Music of Stanford University, 1989.

[55] J. Jensen and R. Heusdens, "Optimal frequency-differential encoding of sinusoidal model parameters," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Orlando, USA, 2002, pp. III 2497–2500.

[56] E. Zwicker and H. Fastl, *Psychoacoustics Facts and Models*. Berlin: Springer Verlag, 1990.

[57] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[58] B. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Res.*, vol. 47, pp. 103–138, 1990.

[59] R. Heusdens and S. van de Par, "Rate-distortion optimal sinusoidal modelling of audio and speech using psychoacoustical matching pursuits," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Orlando, USA, 2002, pp. II 1809–1812.

[60] W. B. Kleijn and K. K. Paliwal, *Speech Coding and Synthesis*. Amsterdam: Elsevier, 1995.

[61] R. J. Sluijter and A. J. E. M. Janssen, "A time warper for speech signals," in *IEEE Workshop on Speech Coding Proceedings*, Porvoo, Finland, 1999, pp. 150–152.

[62] ISO/IEC, "Information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 3: Audio," ISO/IEC, Tech. Rep. 11172-3, 1993, JTSC1/SC29/WG11 MPEG.

[63] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–513, Apr. 2000.

[64] J. E. Hawkins and S. S. Stevens, "The masking of pure tones and of speech by white noise," *J. Acoust. Soc. Am.*, vol. 22, pp. 6–13, 1950.

[65] E. Zwicker and A. Jaroszewski, "Inverse frequency dependance of simultaneous tone-on-tone masking patterns at low levels," *J. Acoust. Soc. Am.*, vol. 71, pp. 1508–1512, 1982.

[66] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Select. Areas Commun.*, vol. 6, no. 2, pp. 314–323, Feb. 1988.

[67] A. Langhans and A. Kohlrausch, "Spectral integration of broadband signals in diotic and dichotic masking experiments," *J. Acoust. Soc. Am.*, vol. 91, pp. 317–326, 1992.

[68] S. Buus, E. Schorer, M. Florentine, and E. Zwicker, "Decision rules in detection of simple and complex tones," *J. Acoust. Soc. Am.*, vol. 80, pp. 1646–1657, 1986.

[69] G. van den Brink, "Detection of tone pulse of various durations in noise of various bandwidths," *J. Acoust. Soc. Am.*, vol. 36, pp. 1206–1211, 1964.

[70] S. van de Par, A. Kohlrausch, G. Charestan, and R. Heusdens, "A new psychoacoustical masking model for audio coding applications," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Orlando, USA, 2002, pp. II 1805–1808.

[71] H. W. Strube, "Linear prediction on a warped frequency scale," *J. Acoust. Soc. Am.*, vol. 68, pp. 1071–1076, 1980.

[72] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Linear prediction on a warped frequency scale," *J. Audio Eng. Soc.*, vol. 48, pp. 1011 – 1031, 2000.

[73] E. G. P. Schuijers, A. W. J. Oomen, A. C. den Brinker, and A. J. Gerrits, "Advances in parametric coding for high-quality audio," in *First IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA)*, Leuven, Belgium, Nov.15 2002, pp. 73 – 79.

[74] N. H. van Schijndel, T. Houtgast, and J. M. Festen, "Intensity discrimination of Gaussian-windowed tones: Indications for the shape of the auditory frequency-time window," *J. Acoust. Soc. Am.*, vol. 105, pp. 3425–3435, 1999.

[75] H. Purnhagen, B. Edler, and C. Ferekidis, "Object-based analysis/synthesis audio coder for very low bit rates," in *AES* 104th *Convention*, Amsterdam, The Netherlands, May 1998, preprint 4747.

[76] P. Masri, "Computer modelling of sound for transformation and synthesis of musical signals," Ph.D. dissertation, University of Bristol, 1996.

[77] J. F. Claerbout, *Fundamentals of Geophysical Data Processing*. New York: McGraw-Hill, 1976.

[78] W. Hess, *Pitch Determination of Speech Signals*. Berlin: Springer-Verlag, 1983.

[79] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[80] M. M. Sondhi, "New methods of pitch extraction," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 262–266, June 1968.

[81] J. Lattard, "Influence of inharmonicity on the tuning of a piano – measurements and mathematical simulation," *J. Acoust. Soc. Am.*, vol. 94, pp. 46–53, 1993.

[82] A. Galembo and A. Askenfelt, "Signal representation and estimation of spectral parameters by inharmonic comb filters with application to the piano," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 2, pp. 197–203, Mar. 1999.

[83] A. Askenfelt and A. Galembo, "Study of the spectral inharmonicity of musical sound by the algorithms of pitch extraction," *Acoustical Physics*, vol. 46, no. 2, pp. 121–132, 2000.

[84] S. W. Kim, R. Sperschneider, H. Purnhagen, Y. B. T. Kim, J. Herre, M. Dietz, S. Hotani, T. Moriya, M. Nishiguchi, T. Mlasko, and B. Grill, "ISO/IEC 14496-3 (MPEG-4 Audio) Amd. 1/FPDAM," ISO/IEC, Vancouver, Canada, Tech. Rep. JTSC1/SC29/WG11 N2803, July 1999.

[85] B. Friedlander and A. Zeira, "Oversampled Gabor representation for transient signals," *IEEE Trans. Signal Processing*, vol. 43, pp. 2088–2094, 1995.

[86] R. Heusdens and K. Vos, "Rate-distortion optimal exponential modeling of audio and speech signals," in *Proceedings of the 21st Symposium on Information Theory in the Benelux*, Wassenaar, The Netherlands, 2000, pp. 77–84.

[87] J. Nieuwenhuijse, R. Heusdens, and E. F. Deprettere, "Robust exponential modelling of audio signals," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Seattle, USA, 1998, pp. VI 3581–3584.

[88] L. Cohen, *Time-Frequency Signal Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[89] M. Desainte-Catherine and S. Marchand, "High-precision Fourier analysis of sounds using signal derivatives," *J. Audio Eng. Soc.*, vol. 48, no. 7/8, pp. 654–667, July/Aug. 2000.

[90] P. Masri and N. Canagarajah, "Extracting more detail from the spectrum with phase distortion analysis," in *Digital Audio Effects (DAFX) Workshop*, Barcelona, Spain, Nov. 1998, pp. 119–122.

[91] K. N. Hamdy and A. H. Tewfik, "Audio coding using steady state harmonics and residuals," in *International Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, Sept. 13–15 1999.

[92] Ph. Depalle and T. Hélie, "Extraction of spectral peak parameters using a short-time Fourier transform modeling and no sidelobe windows," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, USA, 1997, pp. 19–22.

[93] K. Vos, R. Vafin, R. Heusdens, and W. B. Kleijn, "High-quality consistent analysis-synthesis in sinusoidal coding," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 244–250.

[94] S. A. Tretter, "Estimating the frequency of a noisy sinusoid by linear regression," *IEEE Trans. Inform. Theory*, vol. 36, no. 6, pp. 832–835, 1985.

[95] S. Kay, "A fast and accurate single frequency estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 1987–1990, 1989.

[96] S. Peleg and B. Porat, "Estimation and classification of polynomial-phase signals," *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 422–430, 1991.

[97] S. Barbarossa, A. Scaglione, and G. B. Giannakis, "Product high-order ambiguity function for multicomponent polynomial-phase signal modeling," *IEEE Trans. Signal Processing*, vol. 46, no. 3, pp. 691–707, 1998.

[98] M. Z. Ikram and G. T. Zhou, "Estimation of multicomponent polynomial phase signals of mixed orders," *Signal Processing*, vol. 81, no. 11, pp. 2293–2308, 2001.

[99] R. A. Rasch and V. Heetvelt, "String inharmonicity and piano tuning," *Music Percept.*, vol. 3, pp. 171–190, 1985.

[100] H. A. Conklin Jr., "Generation of partials due to nonlinear mixing in a stringed instrument," *J. Acoust. Soc. Am.*, vol. 105, pp. 536–545, 1999.

[101] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[102] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society of Industrial Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[103] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia: SIAM, 1996.

[104] D. J. Thomson, "Spectrum estimation and harmonic analysis," *Proc. IEEE*, vol. 70, no. 9, pp. 1055–1096, 1982.

[105] F. P. Myburg, "Sinusoidal analysis of audio with polynomial amplitude and phase," Philips Research Laboratories, Tech. Rep. Nat.Lab. Technical Note 2001/309, 2001.

[106] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[107] R. Viswanathan and J. Makhoul, "Quantisation properties of transmission parameters in linear predictive systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, no. 3, pp. 309–321, June 1975.

[108] Z. Xiong, C. Herley, K. Ramchandran, and M. T. Orchard, "Flexible time segmentations for time-varying wavelet packets," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, Philadelphia, USA, Oct. 1994, pp. 9 – 12.

[109] Z. Xiong, K. Ramchandran, C. Herley, and M. T. Orchard, "Flexible tree-structured signal expansions using time-varying wavelet packets," in *IEEE Trans. Signal Processing*, vol. 45, no. 2, Feb. 1997, pp. 333 – 345.

[110] P. Prandoni, M. Goodwin, and M. Vetterli, "Optimal time segmentation for signal modeling and compression," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Munich, Germany, 1997, pp. 2029 – 2032.

[111] H. Hermansky, B. A. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Tampa, USA, 1985, pp. 509–512.

[112] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Am.*, vol. 87, pp. 1738–1752, 1990.

[113] A. C. den Brinker and F. Riera-Palou, "Pure linear prediction," in *AES* 115^{th} *Convention*, New York, USA, Oct. 10–13 2003, Convention Paper 5924.

[114] J. O. Smith III and J. S. Abel, "Bark and ERB bilinear transforms," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 6, pp. 697–708, 1999.

[115] A. C. den Brinker, "Stability of linear predictive structures using IIR filters," in *Proceedings of the 12th ProRISC Workshop*, Veldhoven, The Netherlands, Nov. 29–30 2001, pp. 317–320.

[116] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 26, no. 6, pp. 587–596, 1978.

[117] V. Voitishchuk, A. C. den Brinker, and S. L. J. van Eijndhoven, "Alternatives for warped linear predictors," in *Proceedings of the 12th ProRISC Workshop*, Veldhoven, The Netherlands, Nov. 29–30 2001, pp. 710–713.

[118] A. C. den Brinker and F. Riera-Palou, "Quantisation and interpolation of Laguerre prediction coefficients," in *Proceedings of the 13th ProRISC Workshop*, Veldhoven, The Netherlands, Nov. 29–30 2002, pp. 317–320.

[119] K. N. Hamdy, M. Ali, and A. H. Tewfik, "Low bit rate high quality audio coding with combined harmonic and wavelet representations," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Atlanta, USA, 1996, pp. 1045 – 1048.

[120] R. Vafin, R. Heusdens, S. van de Par, and W. B. Kleijn, "Improved modeling of audio signals by modifying transient locations," in *Proc. IEEE Workshop on Applications of Signal Proc. to Audio and Acoustics*, New York, USA, 2001, pp. 143–146.

[121] A. M. Noll, "Short-time spectrum and cepstrum techniques for vocal-pitch detection," *J. Audio Eng. Soc.*, vol. 36, no. 2, pp. 296–302, 1964.

[122] A. C. den Brinker, A. J. Gerrits, and R. J. Sluijter, "Phase transmission in a sinusoidal audio and speech coder," in *AES* 115^{th} *Convention*, New York, USA, Oct. 10–13 2003, Convention Paper 5983.

[123] R. Taori, R. J. Sluijter, and A. Gerrits, "On scalability in CELP coding systems," in *IEEE Speech Coding Workshop*, Pocono Manor, USA, Sept. 7–10 1997, pp. 67–68.

[124] R. M. Schwartz and S. E. Roucos, "A comparison of methods for 300–400 b/s vocoders," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Boston, USA, 1983, pp. 69–72.

[125] H. Everett, "Generalised Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399–417, 1963.

[126] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantisers," in *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, Sept. 1998, pp. 1445 – 1453.

[127] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," in *IEEE Trans. Image Processing*, vol. 2, Apr. 1993, pp. 160 – 175.

[128] G. A. Soulodre and M. C. Lavoie, "Subjective evaluation of large and small impairments in audio quality," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept. 1999, pp. 329–336.

[129] J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, New York, USA, 1988, pp. 2524–2527.

[130] J. D. Johnston and A. J. Ferreira, "Sum-difference stereo transform coding," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, San Francisco, USA, 1992, pp. II 569 – 572.

[131] J. D. Johnston and K. Brandenburg, "Wideband coding - perceptual considerations for speech and music," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi, Eds. New York: Marcel Dekker, 1992, ch. 4, pp. 109–140.

[132] J. Herre, K. Brandenburg, and D. Lederer, "Intensity stereo coding," in *AES 96th Convention*, Amsterdam, The Netherlands, Feb. 1994, preprint 3799.

[133] C. Faller and F. Baumgarte, "Binaural cue coding: A novel and efficient representation of spatial audio," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Orlando, USA, 2002, pp. II 1841–1844.

[134] M. G. Muzzi, "Improvement of the audio quality of a parametric audio coder," Master's thesis, IRCAM, Paris, 2003.

[135] B. W. Kernighan and D. M. Ritchie, *The C Programming Language, Second Edition*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[136] M. J. D. Powell, "A hybrid method for nonlinear equations," in *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, Ed. London, England: Gordon and Breach, 1970, pp. 87 – 114.

Samenvatting

Het comprimeren, of coderen, van digitaal geluid maakt transmissie en opslag van geluid via kanalen met beperkte bandbreedte en opslagmedia met beperkte opslagcapaciteit mogelijk. Conventionele technieken voor het coderen van geluid trachten de golfvorm van het audiosignaal te behouden. Deze technieken worden dan ook golfvorm audiocoders genoemd. Golfvorm audiocoders behalen een reductie in bitrate door gebruik te maken van modellen van het menselijk gehoor, zogenaamde perceptuele modellen. Parametrische audiocoders maken gebruik van zowel een parametrisch signaalmodel als een perceptueel model. Het parametrisch signaalmodel maakt het mogelijk om, met behoud van geluidskwaliteit, lagere bitrates te behalen dan met golfvorm audiocoders bereikbaar is. De parameters van het signaalmodel zijn gerelateerd aan het ontbinden van een signaal in componenten, zoals sinussen en ruis.

Een belangrijke vereiste van een audiocoder die in een dynamische omgeving wordt gebruikt, is een adequaat aanpassingsvermogen. In dit onderzoek worden het ontwerp, de implementatie en de testen van een modulaire, parametrische audiocoder beschreven, die in staat is zich aan te passen aan variërende omstandigheden door het ondersteunen van de productie van een schaalbare bitstroom. Met een schaalbare bitstroom is een audiocoder in staat om op meerdere bitrates te functioneren zonder dat het nodig is om het audiosignaal te hercoderen voor elk van deze bitrates. Een schaalbare bitstroom bestaat uit een aantal lagen, die ieder een bepaald beslag leggen op de totale bitrate. Uitgaand van de totale bitstroom kan de bitrate worden gereduceerd door lagen uit deze bitstroom te verwijderen. De decoder is in staat om de overblijvende lagen te decoderen. Een audiocoder die een schaalbare bitstroom produceert maakt het mogelijk om van muziek te genieten, ongeacht de toegangssnelheid tot het netwerk of de capaciteit van het opslagmedium.

De ontwikkelde audiocoder maakt gebruik van een flexibel, parametrisch signaalmodel dat het beschrijven van dynamische audiosignalen mogelijk maakt. Optimalisatie technieken zijn ontworpen die nauwkeurige schattingen van de modelparameters opleveren. Een hoge geluidskwaliteit wordt behaald door het verwezenlijken van een goede balans tussen sinussen en ruis voor iedere bitrate waarop de schaalbare audiocoder werkt. Resultaten verkregen uit luisterexperimenten wijzen erop, dat de schaalbare audiocoder in staat is een geluidskwaliteit te leveren die gelijkwaardig is aan de geluidskwaliteit van bestaande, niet-schaalbare audiocoders. Dit is opmerke-

lijk, omdat schaalbare audiocoders minder efficiënt zijn dan niet-schaalbare audiocoders. Het onderzoek en de ontwikkelde software zijn ook van belang voor bestaande parametrische audiocoders. Immers, de in dit project ontwikkelde modules kunnen mogelijk worden ingepast in andere parametrische audiocoders. Relevante modules in dit kader zijn de optimalisatie- en ruisaanpassings modules.

Acknowledgements

First of all, I want to thank Bert den Brinker and Stef van Eijndhoven for initiating this research project. Their effort, guidance, and kind support throughout the duration of the project are highly appreciated, as are the countless fruitful discussions and valuable suggestions. I am grateful to the Digital Signal Processing group at Philips Research in Eindhoven, headed by Carel-Jan van Driel, for placing their excellent facilities at my disposal and for creating such a special atmosphere to work in. I would also like to thank the members of the Audio and Speech Processing cluster, as well as Steven van de Par, for their interest and many worthwhile discussions from which I learned a lot. A warm word of thanks to Werner Oomen at Philips Digital Systems Labs in Eindhoven for his support in setting up and carrying out the listening tests.

I would like to express my gratitude to prof.dr.ir. Malo Hautus and prof.dr. Ton Kalker for their valuable suggestions that helped to improve this thesis considerably. I am also grateful to the remaining members of the review board for their constructive comments.

To my friends, I would like to express my appreciation for their interest and support. My family deserves a special word of thanks for their prayer and encouragement throughout. I am especially grateful to Merrel Reitsema senior for proof-reading this thesis. Last, but certainly not least, I would like to express my heartfelt gratitude to my wife for all that she is.

Francois Myburg
27 October 2003

Curriculum Vitae

Francois Philippus Myburg was born on August 8, 1974 in Wolmaransstad, South Africa. He completed his high school education in 1992 at the Potchefstroom Gimnasium Hoërskool, Potchefstroom, South Africa. In 1995, he completed the B.Sc. degree in Mathematics, Applied Mathematics, and Computer Science (cum laude), and in 1997 the B.Sc. degree in Electrical en Electronic Engineering, both at the Potchefstroom University for Christian Higher Education. He completed the M.Sc. degree in Applied Mathematics (cum laude) at the Potchefstroom University for Christian Higher Education in 2000. In the same year, he completed the post-graduate designers programme Mathematics for Industry at the Technische Universiteit Eindhoven. The final project of this programme, entitled "Introducing Bit-Rate Control and Scalability in a Sinusoidal Audio Coder," was carried out at Philips Research in Eindhoven. From June 2000 to October 2003, he was a Ph.D. student in the Systems and Control group in the department of Mathematics and Computer Science at the Technische Universiteit Eindhoven; his research activities in the field of Parametric Audio Coding were carried out at Philips Research in Eindhoven in the Digital Signal Processing group.