Very Low Bit Rate Parametric Audio Coding

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Dipl.-Ing. Heiko Purnhagen

geboren am 2. April 1969 in Bremen

2008

1. Referent:	Prof. Dr.-Ing. H. G. Musmann
2. Referent:	Prof. Dr.-Ing. U. Zölzer
Tag der Promotion:	28. November 2008

Acknowledgments

This thesis originates from the work I did as member of research staff at the Information Technology Laboratory of the University of Hannover.

First of all, I would like to thank my supervisor, Professor Musmann, for the opportunity to work in the inspiring environment of his institute and the Information Technology Laboratory, for enabling me to participate in the MPEG standardization activities, for the freedom he gave me to pursue my own ideas, and for everything I learned during these years. I would also like to thank Professor Zölzer and Professor Ostermann for being on my committee.

I'm very grateful for the fruitful interactions I had with my colleagues and students. In particular, I would like to thank Bernd Edler, Nikolaus Meine, Charalampos Ferekidis, Andree Buschmann, and Gabriel Gaus, who all contributed, in their own way, to the success of this work. I would also like to thank Frank Feige and Bernhard Feiten (Deutsche Telekom, Berlin) and Torsten Mlasko (Bosch, Hildesheim) for good cooperation and research funding. My work was closely related to the MPEG-4 standardization activities, and I would like to thank Schuyler Quackenbush, Masayuki Nishiguchi, and Jürgen Herre for their support in MPEG.

Many thanks go to Lars "Stockis" Liljeryd, Martin Dietz, Lars Gillner, and all my colleagues at Coding Technologies (now Dolby) for their patience, confidence, and support during the time I needed to finalize this thesis.

Furthermore, I would also like to thank all those people who continue to create and to craft sounds that help me believing that there are still audio signals around that are worthwhile to deal with. These are people like Joachim Deicke, Paul E. Pop, Bugge Wesseltoft, Sofia Jernberg, Fredrik Ljungkvist, Joel Grip, Paal Nilssen-Love, and many other musicians and artists.

Last but not least, I would like to thank my parents and friends for all their support.

Stockholm, December 2008

Det er én måde at forstå en anden kultur på. At leve den. At flytte ind i den, at bede om at blive tålt som gæst, at lære sig sproget. På et eller andet tidspunkt kommer så måske forståelsen. Den vil da altid være ordløs. Det øjeblik man begriber det fremmede, mister man trangen til at forklare det. At forklare et fænomen er at fjerne sig fra det.

Peter Høeg: Frøken Smillas fornemmelse for sne (1992)

Kurzfassung

In dieser Arbeit wird ein parametrisches Audiocodierungsverfahren für sehr niedrige Datenraten vorgestellt. Es basiert auf einem verallgemeinerten Ansatz, der verschiedene Quellenmodelle in einem hybriden Modell vereinigt und damit die flexible Verwendung einer breiten Palette von Quellen- und Wahrnehmungsmodellen ermöglicht. Das entwickelte parametrische Audiocodierungsverfahren erlaubt die effiziente Codierung von beliebigen Audiosignalen mit Datenraten im Bereich von etwa 6 bis 16 kbit/s.

Die Verwendung eines hybriden Quellenmodells setzt voraus, daß das Audiosignal in Komponenten zerlegt wird, die jeweils mit einem der verfügbaren Quellenmodelle angemessen nachgebildet werden können. Jede Komponente wird durch einen Satz von Modellparametern ihres Quellenmodells beschrieben. Die Parameter aller Komponenten werden quantisiert und codiert und dann als Bitstrom vom Encoder zum Decoder übermittelt. Im Decoder werden die Komponenten-Signale wieder gemäß der übertragenen Parameter synthetisiert und dann zusammengefügt, um das Ausgangssignal zu erhalten.

Das hier entwickelte hybride Quellenmodell kombiniert Sinustöne, harmonische Töne und Rauschkomponenten und verfügt über eine Erweiterung zur Beschreibung von schnellen Signal-Transienten. Der Encoder verwendet robuste Algorithmen zur automatischen Zerlegung des Eingangssignals in Komponenten und zur Schätzung der Modellparameter dieser Komponenten. Ein Wahrnehmungsmodell im Encoder steuert die Signalzerlegung und wählt die für die Wahrnehmung wichtigsten Komponenten für die Übertragung aus. Spezielle Codierungstechniken nutzen die statistischen Abhängigkeiten und Eigenschaften der quantisierten Parameter für eine effiziente Übertragung aus.

Der parametrische Ansatz ermöglicht die Erweiterung des Codierungsverfahrens um zusätzliche Funktionen. Die Signalsynthese im Decoder erlaubt es, Wiedergabegeschwindigkeit und Tonhöhe unabhängig voneinander zu verändern. Datenratenskalierbarkeit wird erzielt, indem die wichtigsten Komponenten in einem Basis-Bitstrom übertragen werden, weitere Komponenten dagegen in Ergänzungs-Bitströmen. Robustheit für fehlerbehaftete Übertragungskanäle wird durch ungleichförmigen Fehlerschutz und Techniken zur Minimierung der Fehlerfortpflanzung und zur Fehlerverdeckung erzielt.

Das resultierende Codierungsverfahren wurde als *Harmonic and Individual Lines plus Noise* (HILN) parametrischer Audiocoder im internationalen MPEG-4 Audio Standard standardisiert. Hörtests zeigen, daß HILN bei 6 und 16 kbit/s eine Audioqualität erzielt, die vergleichbar mit der von etablierten transformationsbasierten Audiocodern ist.

Schlagworte: Parametrische Audiocodierung, Signalzerlegung, Parameterschätzung, Quellenmodell, Wahrnehmungsmodell, MPEG-4 HILN

Abstract

In this thesis, a parametric audio coding system for very low bit rates is presented. It is based on a generalized framework that combines different source models into a hybrid model and thereby permits flexible utilization of a broad range of source and perceptual models. The developed parametric audio coding system allows efficient coding of arbitrary audio signals at bit rates in the range of approximately 6 to 16 kbit/s.

The use of a hybrid source model requires that the audio signal is being decomposed into a set of components, each of which can be adequately modeled by one of the available source models. Each component is described by a set of model parameters of its source model. The parameters of all components are quantized and coded and then conveyed as bit stream from the encoder to the decoder. In the decoder, the component signals are resynthesized according to the transmitted parameters. By combining these signals, the output signal of the parametric audio coding system is obtained.

The hybrid source model developed here combines sinusoidal trajectories, harmonic tones, and noise components and includes an extension to support fast signal transients. The encoder employs robust algorithms for the automatic decomposition of the input signal into components and for the estimation of the model parameters of these components. A perceptual model in the encoder guides signal decomposition and selects the perceptually most relevant components for transmission. Advanced coding schemes exploit the statistical dependencies and properties of the quantized parameters for efficient transmission.

The parametric approach facilitates extensions of the coding system that provide additional functionalities. Independent time-scaling and pitch-shifting is supported by the signal synthesis in the decoder. Bit rate scalability is achieved by transmitting the perceptually most important components in a base layer bit stream and further components in one or more enhancement layers. Error robustness for operation over error-prone transmission channels is achieved by unequal error protection and by techniques to minimize error propagation and to provide error concealment.

The resulting coding system was standardized as *Harmonic and Individual Lines plus Noise* (HILN) parametric audio coder in the international MPEG-4 Audio standard. Listening tests show that HILN achieves an audio quality comparable to that of established transform-based audio coders at 6 and 16 kbit/s.

Keywords: parametric audio coding, signal decomposition, parameter estimation, source model, perceptual model, MPEG-4 HILN

Contents

List of Figures

List of Tables

Symbols and Abbreviations

$\mathrm{E}\{x\}$	Expectation operator
Ω	Angular frequency (continuous time) [rad/s]
ω	Angular frequency (discrete time) [rad]
π	3.14159265358979323846264338327950 29...
φ	Phase [rad]
a	Amplitude
e	2.71828182845904523536028747135266 25...
f	Frequency [Hz]
f_s	Sampling rate [Hz]
$H(\Omega)$	Transfer function (continuous time)
$H(z)$	Transfer function (discrete time), $z = e^{j\omega}$
j	$\sqrt{-1}$
n	Discrete time index
t	Time [s]
T_f	Frame length (stride) [s]
T_s	Sampling interval [s]
$x(t)$	Time-continuous signal
$x[n]$	Time-discrete signal
AAC	Advanced Audio Coding
ACF	Autocorrelation Function
AD envelope	Attack/Decay temporal amplitude envelope
AOI	Audio on Internet
AR process	Autoregressive process
ASAC	Analysis/Synthesis Audio Codec
CE	Core Experiment
CELP	Code Excited Linear Prediction
cent	$1/100$th semitone interval (frequency ratio $1{:}2^{1/1200}$)

CI	Confidence Interval
CMOS	Comparison Mean Opinion Score
CR bound	Cramér-Rao bound
CWL	Codeword Length
ESW	Excitation Similarity Weighting
FFT	Fast Fourier Transform
HILN	Harmonic and Individual Lines plus Noise
HVXC	Harmonic Vector Excitation Coding
JND	Just-Noticeable Difference
LAR	Logarithmic Area Ratio
LPC	Linear Predictive Coding
MLE	Maximum Likelihood Estimator
MOPS	Million Operations Per Second
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group (ISO/IEC JTC 1/SC 29/WG 11)
MSE	Mean Square Error
PCM	Pulse Code Modulation
PCU	Processor Complexity Units
PDF	Probability Density Function
PE	Perceptual Entropy
PSD	Power Spectral Density
RMS	Root Mean Square
SDC	Subdivision Coding
SEM	Spectral Envelope Model
SFM	Spectral Flatness Measure
SMR	Signal-to-Mask Ratio
SNR	Signal-to-Noise Ratio
STFT	Short Time Fourier Transform
TwinVQ	Transform-Domain Weighted Interleave Vector Quantization
UEP	Unequal Error Protection
VQ	Vector Quantizer

1 Introduction

The digital representation of high-quality audio signals requires high bit rates if straightforward pulse code modulation (PCM) is employed. The compact disc (CD), for example, uses a bit rate of approximately 1.4 Mbit/s to store a stereo audio signal sampled at 44.1 kHz and quantized with 16 bit per sample. This bit rate would be too high for many other applications, such as mobile audio players with solid-state storage or systems that provide transmission of audio over dial-up telephone lines, over the Internet, or over radio channels. To enable such applications, more compact digital representations of audio signals are required, so that an audio signal can be conveyed at a significantly lower bit rate with no or tolerable impairment of the subjective (perceived) audio quality. This problem of efficient audio coding is addressed by two well-established paradigms.

State of the Art in Audio and Speech Coding

Traditionally, the problem of coding of arbitrary high-quality audio signals at low bit rates has been addressed by perceptual audio coding systems following the *transform coding* paradigm [10], [11], [38], [54], [92]. These systems are based on the coding of spectral components in combination with signal adaptive quantization in order to exploit statistical properties of the signal as well as properties of human perception. The spectral components are derived by a time-frequency decomposition implemented as transform or filter bank, and the quantization is controlled by a psychoacoustic model. The advanced audio coding standard (AAC) [9], [42], developed by ISO's moving picture experts group (MPEG), is a typical implementation of this paradigm. It achieves a quality perceptually equivalent to the CD at a bit rate of 128 kbit/s for a stereo audio signal [80], [118]. The transform coding paradigm can be successfully applied at bit rates down to approximately 16 kbit/s for a monaural signal. Even lower bit rates down to 6 kbit/s are possible with dedicated systems such as transform-domain weighted interleave vector quantization (TwinVQ) [52], which is a part of the MPEG-4 standard [44]. At such low bit rates, however, the audio bandwidth is substantially reduced, typically 5 kHz at 16 kbit/s or 3 kHz at 6 kbit/s, and further artifacts severely affect the subjective quality of audio signals like speech [81], [83].

The *speech coding* paradigm, on the other hand, originates from telephony applications [58], [119]. It exploits knowledge of the speech generation process in the human vocal tract, which can be modeled as a combination of voiced or unvoiced excitation followed by a resonator that shapes the signal spectrum. Properties of human perception, however, are only utilized to a relatively small degree. Typical implementations, like those

used for mobile telephones in the global system for mobile communication (GSM) [24] or defined in ITU-T standards like [51], or the MPEG-4 standard [21], [44], are mostly based on code excited linear prediction (CELP) coding. For narrow-band speech (3.5 kHz bandwidth), CELP coding can be used at bit rates down to approximately 6 kbit/s. For wide-band speech (7 kHz bandwidth), 16 kbit/s and more are common.

General Problem

For various applications, the transmission of audio signals at very low bit rates in the range of 6 to 16 kbit/s is of interest. This includes, for example, the carriage of audio signals over very narrow channels, like mobile radio links, or the streaming audiovisual content over slow Internet connections, where a large part of the available transmission bandwidth is needed for the video signal, leaving only a small part for the audio signal.

Comparison tests of coding systems operating at bit rates in the range of approximately 6 to 16 kbit/s have shown that, for speech signals, the speech coding paradigm generally outperforms the transform coding paradigm [14], [77], [81], [82]. For most other signals, however, transform coding outperforms speech coding. This observation reveals that none of the established coding paradigms is able to convey *arbitrary* signals satisfactorily at very low bit rates.

In order to understand this problem and derive possible approaches for a solution, it is necessary to examine the underlying model assumptions upon which these audio coding paradigms are based.

- Assumptions made about the source of the signal are referred to as *source model*. Such assumption can relate to physical properties of the actual source or to simplified mathematical models of it. In a coding system based on a given source model, the corresponding model assumptions are known *a priori* to both the encoder and decoder. This enables the encoder to remove *redundant* information contained in the audio signal which can be reconstructed in the decoder with help of the source model. Thus, less information has to be conveyed.
- Assumptions made about the perception of the signal by a human listener are referred to as *perceptual model*. Such assumptions imply that certain small deviations from the original audio signal are not noticed by the listener. This indicates that *irrelevant* information is contained in the original audio signal which does not need to be conveyed. Larger deviations that exceed the threshold of perceptibility are rated by a perceptual model in terms of a distortion measure.

It is important to note that both redundancy and irrelevancy reduction have to be considered jointly when assessing the overall coding efficiency of a system, that is, its bit rate vs. perceived distortion characteristics [6].

The transform coding paradigm utilizes the first and most prominent class of source models, which is based on linear correlation of subsequent samples, corresponding to the

spectral unflatness of the signal [53]. Assuming short-term stationarity, these characteristics can be exploited by linear prediction or by spectral decomposition accomplished by a filter bank or transform, which generates a time-frequency representation of the signal.

The speech coding paradigm utilizes the second major class of source models, which is rooted in the physics of the actual sound generation process at the signal's source. Typically, this process can be described as a random, periodic, or impulse-like excitation of a resonant system. For speech signals, such a source model characterizes the excitation from the vocal cords and the resonances in the vocal tract [58]. Similar *physical models* are also widely used for synthesis of musical instrument sounds [117].

The transform coding paradigm utilizes an advanced perceptual model to exploit the effect of auditory masking [130], that is, the increased threshold of audibility of a test sound (maskee) in presence of another, usually louder, sound (masker). Such a perceptual model, which calculates this masked threshold as a function of frequency for the current input signal, allows to assess the audibility of distortion added by a coding system. It is employed in transform coding to control the adaptive quantization and to ensure that the quantization distortion is below this threshold. The speech coding paradigm, on the other hand, utilizes only a fairly simple perceptual model.

This analysis of source and perception models explains the performance of the transform and speech coding paradigms for different types of signals observed above. While more general source models, like a time-frequency decomposition model, are applicable to arbitrary signals, specialized models, like a physical speech generation model, can achieve a more efficient and compact representation of a signal. This, however, is only true as long as the actual signal conforms sufficiently well to the assumptions of the specialized model, and explains why non-speech signals can lead to severe artifacts when processed by a speech coding system.

Motivation of the New Approach

The objective of this thesis is the efficient coding of *arbitrary* audio signals at very low bit rates in the range of approximately 6 to 16 kbit/s. An apparently simple approach to improve the coding of arbitrary signals would be to combine the both source models utilized in the transform and speech coding paradigms into a *hybrid model* in order to cover a larger class of signals efficiently. In such a system, the encoder would have to decompose the incoming signal into one or more speech components and residual non-speech components. However, this task is extremely difficult because the signal spaces covered by the two source models are by no means orthogonal so that a well-defined solution to the decomposition problem does not exist.

To address this decomposition problem, it is of interest to consider further alternative source models for signal components in addition to the two classes of source models outlined above. These additional source models should enable the design of a hybrid model in such a way that the signal decomposition into components, as required in the encoder, is feasible. This can be facilitated by using simple component source models with only

a low number of free parameters. If possible, the components into which the signal is decomposed should be orthogonal to permit independent estimation of the component parameters. Given these considerations, four additional classes of source models are of interest.

Sinusoidal models, which constitute an important class of source models, describe a signal as superposition of a set of sinusoids, each of which is characterized by parameters for amplitude and frequency that can be slowly time-varying [26], [36], [70], [111], [116]. Such models are popular because most real-world audio signals are dominated by tonal signal components. Different coding systems based on some form of sinusoidal model and primarily intended for speech signals have been proposed. This includes the *phase vocoder* by Flanagan in 1966 [26], a speech coder utilizing a generalized sinusoidal model by Hedelin in 1981 [36], and the sinusoidal coders described by McAulay and Quatieri [70] and by Serra and Smith [116] in 1986.

Sinusoidal models can be extended to form *harmonic models* [1], [88]. These address the whole set of partials of a harmonic tone, that is, a set of sinusoids whose frequencies are approximately integer multiples of a fundamental frequency. A speech coder utilizing *harmonic coding* for voiced speech was proposed by Almeida and Tribolet in 1982 [1].

Sinusoidal source models can be complemented by models for transient and noise-like signals. The class of *transient models* aims at appropriate representation of impulse-like signals [12], [87], [125]. An isolated impulse can be characterized by its temporal location and shape. An example for this is a damped sinusoid starting at a certain onset time.

The class of *noise models* addresses the representation of noise-like signals [31], [113]. These models characterize a noise-like signal by means of the parameters of an underlying stochastic process that could generate the observed audio signal. Typically, these parameters are related to the spectral and temporal envelope of the noise signal. As will be discussed below, noise models rely on the assumption of certain perceptual properties, namely that signals constituting different realizations of such a stochastic process *sound* the same.

These four additional classes of source models can all be regarded as *parametric models*. Even though a precise definition of the term parametric model is difficult (see, e.g., [33, p. 4] or [58, p. 26]), it is used here to refer to source models which characterize the signal by parameters that are more abstract than samples taken from a time-frequency representation of the signal. In this sense, the time-frequency decomposition model utilized by a transform coder is not a parametric source model.

Obviously, many of the source models mentioned here can only be applied to a limited class of signals. Hence, in order to allow the efficient representation of arbitrary audio signals, different source models with complementary characteristics have to be combined to form a hybrid model. Various hybrid models have been proposed in literature [35], [37], [61], [90], [112], [113], [126], combining two or more of the six different classes of source models outlined above.

In order to achieve efficient coding at very low bit rates, which is necessary to address the objective of this thesis, and in order to deal with the decomposition problem men-

tioned earlier, the combination of different *parametric* source models into a hybrid model appears to be the most promising approach. These considerations lead to the concept of *parametric audio coding*, where the input signal is decomposed into a set of simple components, like, for example, sinusoids and noise. Each of these components is described by an appropriate component source model and represented by model parameters. The model parameters can be quantized and coded and then conveyed to a decoder. There, the parameters of the components are decoded and then the component signals are resynthesized according to the transmitted parameters. By combining these signals, the output signal of such a coding scheme is obtained.

Very low bit rate operation requires that also perceptual models have to be taken into account extensively. Traditionally, most audio and speech coding systems strive to approximate the *waveform* of the original signal at their output such that a perceptual distortion measure for the residual error signal is minimized. For many applications, however, it is sufficient if the decoded signal *sounds* the same as the original signal. Considering a noise-like signal, it becomes obvious that waveform approximation is not necessarily required to achieve the same sound. Abandoning the requirement of waveform approximation allows to utilize a wider range of source models and enables a more efficient and compact parametric representation of a signal. Generalized perceptual models assessing the similarity of two sounds, however, are unfortunately very difficult and complex [5], [120]. Nevertheless, for several specific aspects of this problem, such as just-noticeable sound changes, simple models are known from literature [130].

In a parametric audio coding system operating at very low bit rates in the range of 6 to 16 kbit/s, where audible impairments might be unavoidable, it can be beneficial to utilize perceptual models in order to assist the decomposition into components and to control the quantization of the parameters. In particular in situations where the available bit rate does not permit to convey all components found by decomposition, a perceptual model can help to select the perceptually most important components for transmission.

The concept of a sinusoidal audio coder that makes extensive use of a psychoacoustic model for efficient coding at very low bit rates was first proposed by Edler, Purnhagen, and Ferekidis in 1996 [18], where it is referred to as *analysis/synthesis audio codec* (ASAC). In parallel with the work presented in this thesis, other parametric audio coding systems have been proposed. These systems are a sinusoids plus noise coder that switches to transform coding for transients proposed by Levine in 1998 [62], a scalable sinusoids plus transients plus noise coder proposed by Verma in 2000 [129], and a sinusoids plus transients plus noise parametric coder for high quality audio proposed by Brinker in 2002 [12]. However, all these three systems are aiming at bit rates that are significantly higher than those considered here.

Hence, in order to enable efficient parametric audio coding at very low bit rates, four major problems have to be solved. An optimized hybrid source model should be designed which allows a harmonic tone to coexist simultaneously with individual sinusoidal components, transients, and noise. To achieve robust signal decomposition and parameter estimation, all sinusoidal components, including the partials of a harmonic tone, should

be tracked over time to reliably build sinusoidal trajectories. Perceptual models should be utilized for both signal decomposition and component selection. Last but not least, model parameters should be coded efficiently, and in particular, joint coding of a set of parameters should be considered.

Outline of the New Approach

In order to develop a complete system for very low bit rate parametric audio coding, considering the four problems to be solved, two major tasks have to be accomplished.

The **first task** addresses the design of an appropriate hybrid source model and the development of algorithms for signal decomposition and parameter estimation. It can be divided into four steps.

In a **first step**, a hybrid parametric source model has to be designed that integrates different source models for signal components. Suitable source models known from literature as well as new source models aimed at very low bit rate audio coding should be considered. However, there is yet no theory that allows to calculate an optimum source model [86]. To assess the efficiency of a source model, it has to be tested in the context of a complete coding system. Hence, source model design actually should be treated in conjunction with the overall coding system design and not as an isolated task.

In a **second step**, techniques for the robust and yet accurate estimation of source model parameters of signal components in presence of interfering signals or noise have to be developed. In addition, mechanisms for adequate decomposition of the audio signal into signal components have to be devised. A simple approach to the decomposition problem is to divide the signal into a sequence of short segments and apply a greedy, iterative analysis-by-synthesis algorithm independently to each segment. In many cases, however, signal components can be non-orthogonal, which leads to mutual dependencies between the estimated model parameters and indicates that the problems of signal decomposition and parameter estimation are tightly coupled. Furthermore, in order to improve the robustness of the estimation of sinusoidal components, tracking of components across consecutive signal segments should be investigated.

In a **third step**, a perceptual model has to be designed that allows the selection of the perceptually most relevant signal components. This enables efficient operation of the coding system at very low bit rates, where the parameters of only a few signal components can be conveyed to the decoder. In case of component source models that are based on certain perceptual assumptions, like noise models, it is necessary to consider perceptual aspects already during the signal decomposition process. For example, a perceptual model could be used to discriminate between those parts of the input signal that are perceived as noise and those parts that are perceived as tonal.

In a **fourth step**, the problems of parameter estimation, signal decomposition, and component selection should be considered jointly in combination with constraints imposed by parameter quantization and coding (as addressed by the second task) in order to achieve overall optimization in a rate distortion sense. For example, the optimal signal

decomposition for a coding system operating at 6 kbit/s can be different from the optimal decomposition for the 16 kbit/s case. Furthermore, also constraints concerning the computational complexity of the encoder should be considered.

The **second task** addresses the development of a complete audio coding system based on the parametric signal representation derived in the first task. It can be divided into three steps.

In a **first step**, an appropriate quantization of model parameters has to be designed, taking into account a perceptual model. Furthermore, efficient schemes for entropy coding of the quantized parameters have to be devised. This can include predictive techniques to exploit statistical dependencies between parameters as well as variable length codes to exploit non-uniform probability distributions of parameters. In particular, joint coding of the set of model parameters of individual sinusoidal components should be investigated in order to improve coding efficiency. Finally, bit allocation strategies in the encoder have to be considered to optimize performance at very low bit rates.

In a **second step**, aspects of the decoding process have to be considered. While parameter decoding and dequantization usually is not difficult, the design of the signal synthesis algorithms can have a significant impact on the computational complexity of the decoder.

In a **third step**, possibilities to extend a parametric coding system by different additional functionalities should be investigated. These are, in particular, *time-scaling* and *pitch-shifting* in the decoder, *bit rate scalability* (that is, hierarchical embedded coding), and improved *error robustness* for operation over error-prone transmission channels.

Finally, the performance of the complete coding system, which will be referred to as *Harmonic and Individual Lines plus Noise* (HILN) coding, should be verified by means of subjective listening tests in order to allow comparison to other coding systems. Furthermore, the results should be contributed to the MPEG-4 standardization activities.

Organization of the Thesis

This thesis is organized as follows. Chapter 2 briefly reviews the fundamentals of parametric representations of audio signals and devises a generalized framework that describes how source and perceptual models can be utilized in a parametric audio coding system. In Chapter 3, a hybrid source model suitable for very low bit rate coding is designed and parameter estimation techniques for its component models are developed. This is followed by a study of techniques for signal decomposition and component selection in the encoder that take into account a perceptual model. Chapter 4 addresses the design of appropriate parameter quantization and entropy coding techniques and discusses bit allocation strategies in the encoder. Moreover, signal synthesis in the decoder is discussed and extensions of the coding system required in order to provide additional functionalities are presented. In Chapter 5, the development of HILN parametric audio coding in the context of the MPEG-4 standardization activities is briefly reviewed and the performance of this coder is verified by experimental results. Finally, Chapter 6 concludes the thesis with a summary and suggests directions for future research.

2 Fundamentals of Parametric Audio Coding

The design of an audio coding system is based on model assumptions for the signal source and signal perception. By utilizing such models, a coding system is able to exploit redundant and irrelevant information in the audio signal in order to obtain a compact coded representation of the signal. The term *parametric model* is essential here and, as stated earlier, refers to source models which characterize the signal by parameters that are more abstract than samples taken from a time-frequency representation of the signal.

This chapter briefly reviews the principles of parametric representations of audio signals and introduces a generalized framework for parametric audio coding that describes how source and perceptual models are utilized in order to build a complete coding system.

2.1 Parametric Representations of Audio Signals

The concept of describing an audio signal or its components by means of abstract parameters like pitch, duration, or loudness has long been used and is well established, for example in musical notation. The advent of powerful computing systems for digital signal processing, however, has enabled many new applications that utilize some form of parametric representations of audio signals. This section gives a brief overview of the principles and applications of parametric audio signal representations.

Even though parametric representations of audio signals can be employed in a wide range of applications, they share a common motivation and are based on the same principles. There are two major reasons why a parametric representation of an audio signal can be advantageous compared to a conventional sampled representation of the same signal.

Firstly, a parametric representation is potentially *more compact* than a sampled representation. That is, less parameters are required to describe the signal. This property is related to the underlying source model and indicates the potential to exploit redundancy contained in the audio signal.

Secondly, a parametric representation can describe the signal on a *higher semantic level* than a sampled representation. That is, the parameters can be closely related to physical or perceptual properties of the signal or its source. This property enables meaningful interpretation of the parametric representation by humans and can help to assess the perceptual relevance of a given parameter. At the same time, this property allows for flexible modification of the signal in its parametric representation.

In principle, a system employing a parametric representation of audio signals comprises three major components.

- **Analysis** To begin with, a parametric representation of the considered audio signal is generated. Typically, an automatic signal analysis is used to derive a parametric representation from the waveform of a given audio signal. This allows to handle arbitrary real-world signals. Alternatively, the parametric representation can be created manually or semi-automatically. This enables, for example, to create and compose arbitrary new sounds and is of interest for music or speech synthesis applications.
- **Processing** The parametric representation permits flexible processing of the audio signal that would not be possible with a conventional sampled representation of the signal. This property is in fact the primary motivation to employ parametric representations. Two different objectives for the processing are of interest. On the one hand, a parametric representation simplifies the modification of the audio signal in a perceptually meaningful way and enables flexible effects processing like, for example, time-scaling and pitch-shifting. On the other hand, the potentially compact nature of a parametric representation can be utilized for efficient transmission or storage of an audio signal at low bit rates. This can be achieved by quantizing the actual parameters, followed by entropy coding. For optimal performance, the quantization process should be controlled by a perceptual model, and parameters deemed irrelevant can be omitted completely.
- **Synthesis** In order to play back and listen to an audio signal given by means of a parametric representation, the waveform of the signal has to be reconstructed from the parameters. This is accomplished by automatic signal synthesis. The synthesis process is directly based on the source model employed by a given system.

Even though the musical notation of a song, as score for a human musician, or, for example, as piano roll for a mechanical player-piano, can be seen as a form of parametric representation of an audio signal, the following brief review of the history of parametric representations of audio signals will be limited to electronic systems employing such representations.

Early work on parametric representations for analysis and synthesis of sound focused on the human voice. The *voder* and the *vocoder* were systems for the synthesis and analysis/modification/synthesis, respectively, of human voice by Dudley in 1939 [16]. They employed unvoiced excitation and voiced excitation with variable pitch. The spectral shaping resulting from the resonances in the vocal tract was simulated by a filter bank with variable gains per filter bank channel (i.e., per frequency band). Hence, this system is also referred to as *channel vocoder*. The voiced/unvoiced mode, the pitch, and the gains for the filter bank channels are the time-variant parameters of this representation.

This approach evolved, and time-variant filters were introduced for spectral shaping, where the filter coefficients in an appropriate representation serve as parameters. Typically, all-pole filters were used, and this approach is also known as linear predictive coding (LPC) [65]. Also for the excitation, various alternative representations were introduced. This includes code excited linear prediction (CELP) [3] and regular pulse excitation (RPE) [60], which strive to describe the waveform of the excitation in a compact

manner and address both voiced and unvoiced speech. An alternative approach to represent voiced speech is the concept of *harmonic coding* proposed by Almeida and Tribolet [1], [2] in 1982, where, besides the pitch (i.e., the fundamental frequency of the harmonic tone) the magnitudes and phases of the partials serve as parameters.

Parametric representation of music and of the sound of musical instruments gained much interest when it became possible to convert such a representation into audible sounds by means of a programmable synthesizer [68] in the 1960s. Computer music languages like *MUSIC V* by Mathews [68] permit to define almost arbitrary synthesis algorithms, or virtual instruments. Such an instrument is described by combining simple digital signal processing (DSP) blocks like oscillators and envelope generators, an approach similar to analog modular Moog synthesizers [74]. Besides the instrument definitions, a program written in such a computer music language has a second part that contains the score to play and control the virtual instruments.

In order to re-create sounds of natural instruments by a program written in a computer music language, recordings of real instruments must be analyzed and modeled. In a prominent example by Risset in 1969 [111], sounds of wind and string instruments are modeled by additive synthesis of sinusoidal partials, with proper temporal amplitude envelopes for each partial and, if required, amplitude modulation, frequency modulation, or noise addition. The parameters of these models were found empirically by comparing synthesized and original sounds subjectively, using, for example, data from pitch-synchronous analysis of dedicated recordings as a basis.

The additive synthesis of partials, as mentioned above, is a special form of a *sinusoidal model*. Interestingly, the original (analog) channel vocoder was further developed into the (digital) *phase vocoder* by Flanagan in 1966 [15], [26], which is also a special form of sinusoidal modeling. The analysis in the phase vocoder can be considered as a complex modulated filter bank, typically with 30 channels (i.e., frequency bands) spaced at 100 Hz. Equivalently, the analysis can also be seen as a short time Fourier transform (STFT) with a sliding window. Synthesis is accomplished by a bank of 30 sine wave generators, one per channel, which are controlled by the magnitude and phase parameters of the 30 complex-valued channels from the filter bank. Maintaining sufficiently high bandwidth for the magnitude and phase parameters, the phase vocoder achieves almost perfect reconstruction of arbitrary signals.

A first generalized sinusoidal model was applied to the base band (100–800 Hz) of speech signals by Hedelin in 1981 [36]. This model describes the base band signal as a sum of sinusoids with the time-varying amplitude and the time-varying frequency/phase of each sinusoid (or tone) as model parameters. For this system, the simultaneous estimation of the parameters of all sinusoids using an recursive algorithm was studied.

Generalized analysis/synthesis based on a sinusoidal representation was further developed for speech and non-harmonic sounds by McAulay and Quatieri [70] and by Serra and Smith [116] in 1986. In these systems, noise-like signals are modeled by a large number of sinusoids with random parameters. Peak-picking in the short time magnitude spectrum obtained from a fast Fourier transform (FFT) is used to estimate the frequency and

amplitude parameters of the sinusoids during analysis. Sinusoidal trajectories are build by parameter matching between consecutive signals segments and are synthesized with continuous phase. To improve efficient parameter estimation for sinusoidal models, an analysis-by-synthesis/overlap-add (ABS/OLA) approach was proposed by George [29], [30] in 1992. The overlap-add paradigm is here also employed for synthesis, as opposed to the notion of time-varying parameters in the generalized sinusoidal representation.

In order to achieve better modeling of noise-like signal components, a sound analysis/synthesis system based on a *deterministic plus stochastic decomposition* was proposed by Serra in 1990 [114], [115]. This system is also referred to as spectral modeling synthesis (SMS). The deterministic component is modeled as a sum of quasi-sinusoidal components, i.e., sinusoids with piecewise linear amplitude and frequency variation [70], [116]. The stochastic component is modeled by its time-variant power spectral density (PSD) [31]. The proposed analysis assumes that the residual obtained by subtracting the deterministic component from the original signal can be considered as the stochastic component.

Furthermore, various concepts to improve the representation of transients signal components have been proposed since 1996. One approach is to improve the temporal resolution of sinusoidal models by using adaptive windows [34], [67], damped sinusoids [32], [87], or transient amplitude envelopes [12]. Other approaches use wavelets [35] or transform coding [61], [62], [63] to represent transients. A further approach is based on sinusoidal modeling in a real-valued frequency domain [125]. Finally, the concept of decomposition in sinusoids, transients, and noise [126], [128], [129] can also be seen as a representation based on frequency contours, time contours, and textures of the time-frequency plane [85].

2.2 Generalized Framework for Parametric Audio Coding

Figure 2.1 presents the generalized framework of an audio coding system in order to show how a source model and a perceptual model can be utilized in the encoder and decoder. This generalized framework devised here is a refinement of earlier proposals [20] [22] [104]. In the encoder, the original audio signal is first described in terms of parameters of a source model (or signal model) [33]. Then, these parameters are quantized according to distortion thresholds found by a perceptual analysis of the original signal according to a perceptual model. Finally, the quantized parameters are entropy coded to exploit their statistical properties. A perceptual model can also indicate that transmission of certain parameters is not necessary and that they can be substituted in the decoder by, for example, random values with a certain distribution. Typically, the quantizer configuration selected by the perceptual analysis in the encoder is transmitted as *side information* in the bit stream.

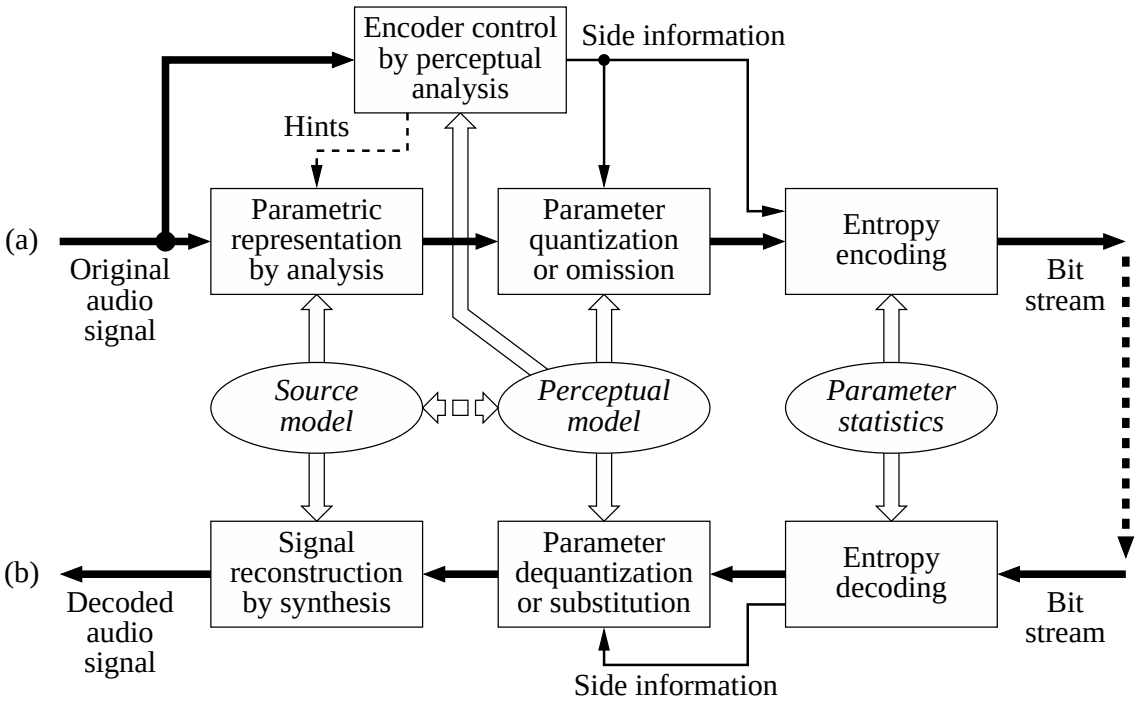

Figure 2.1: Generalized framework of an audio coding system showing the utilization of model assumptions for source and perception in the encoder (a) and decoder (b).

It is interesting to note that there can be mutual dependencies between the source model and the perceptual model. Furthermore, different aspects of a source model can be exploited by different parts of a coding system, and the same holds true for the perceptual model. Hence, it is in most cases not possible to clearly distinguish between those parts of a coding system doing redundancy reduction and those parts doing irrelevancy reduction [10]. A prominent example for this situation is a vector quantizer (VQ) [64], where signal statistics and a (perceptual) distortion measure can be exploited simultaneously.

In order to apply the general concept shown in Figure 2.1 in the design of a parametric audio coding system, the encoding process has to be studied in detail. The *encoder control by perceptual analysis* does not only control the parameter quantization but can also give *hints* to the *parametric representation by analysis* of the incoming original audio signal. In this way, the encoder control can supervise the complete encoding process. The hints to the parametric signal analysis can address two different issues. Firstly, the signal analysis can be configured to achieve best performance at the requested target bit rate. Secondly, for source models (like a sinusoidal or hybrid model) that require a decomposition of the input signal into components, the hints can provide guidance to achieve a perceptually optimized decomposition.

Furthermore, such hints can be used to implement an *analysis-by-synthesis* paradigm. This is a powerful though complex approach to jointly optimize (in a rate distortion

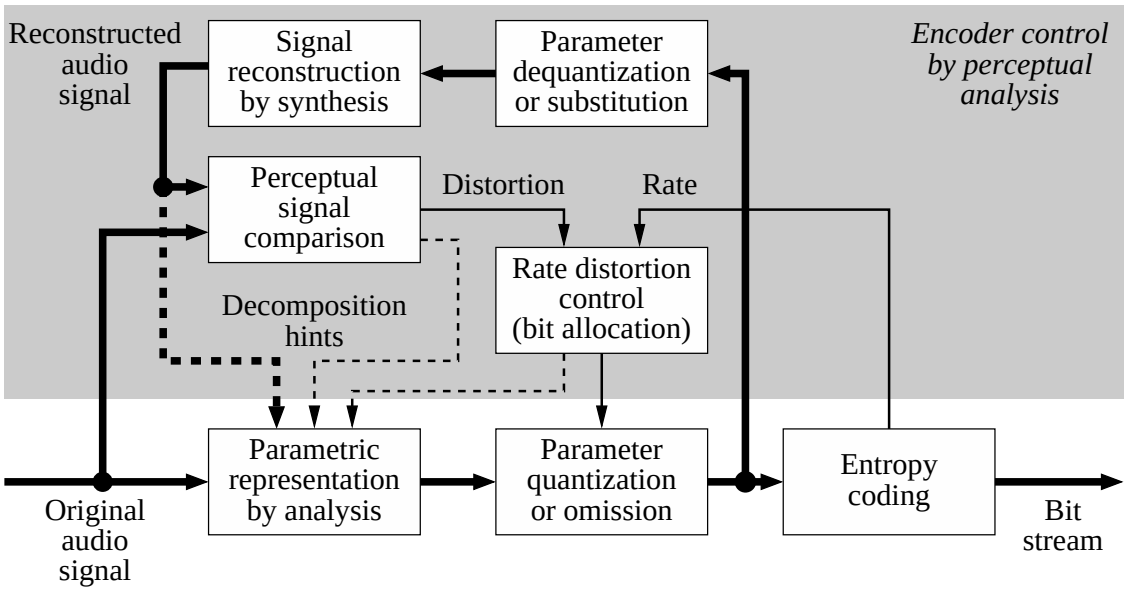

Figure 2.2: Parametric audio encoder implementing the analysis-by-synthesis paradigm.

sense) parameter estimation, signal decomposition, and component selection under the constraints imposed by parameter quantization and coding. A generalized diagram of an encoder implementing this paradigm is shown in Figure 2.2. The upper gray part corresponds to the *encoder control by perceptual analysis* in Figure 2.1 and the *side information* data flow is omitted for clarity. The basic idea of the analysis-by-synthesis paradigm is to find that bit stream with the desired rate which, when processed by the decoder, results in an audio signal that is perceptually as similar to the original signal as possible. To accomplish this, the quantized parameters that would be available in the decoder are dequantized again locally and signal synthesis is used to reconstruct the audio signal as it would be done by the decoder. The reconstructed signal is then compared to the original input signal, resulting in a perceptual distortion measure. This distortion measure, as well as the actual bit rate observed by the *entropy coding* is then used by the *rate distortion control* to optimize bit allocation by refining the configuration of the *parameter quantization or omission* and to provide hints to assist the signal decomposition in the parametric signal analysis. In this way, a full *analysis-by-synthesis loop* is formed. Also the reconstructed audio signal and data from the perceptual signal comparison can be provided to the parametric signal analysis to assist signal decomposition. These aspects will be discussed in detail in Chapter 3.

Compared to the encoding process, the decoding process is quite straight-forward, since parameter decoding and dequantization are determined by the corresponding steps in the encoder. The synthesis process that reconstructs the audio signal from its parametric representation is determined by the source model (or signal model) itself.

3 Signal Analysis by Decomposition and Parameter Estimation

In order to build a parametric audio coder that is capable of handling arbitrary real-world audio signals, it is necessary to design an appropriate hybrid source model and to develop algorithms for parameter estimation and signal decomposition. This chapter explains the design of the hybrid source model and studies algorithms for robust yet accurate estimation of the model parameters of signal components. These algorithms are then integrated in a framework that enables the decomposition of the original audio signal into components according to the hybrid source model. In order to optimize signal decomposition in the context of audio coding applications, it is necessary to integrate also perceptual models into the signal decomposition and component selection process. Finally, effects of rate distortion constraints are studied and two examples of complete parametric encoders are given.

3.1 Design of a Hybrid Source Model for Very Low Bit Rate Audio Coding

The hybrid source model developed during the course of the work presented here was designed with focus on application in very low bit rate audio coding, with target bit rates in the range of approximately 6 to 16 kbit/s. To assess the efficiency of such a source model, it is necessary to employ it in the context of a complete coding system. Hence, the design of the hybrid source model has been carried out in parallel with the development of the coding system itself.

This development lead to a hybrid parametric source model which will be described in detail in the remainder of this section. It includes elements from various models for sinusoidal, harmonic, transient, and noise components.

In order to enable encoding with a limited look-ahead (or latency) and to permit real-time encoding of audio signals, a frame-based approach was chosen for the hybrid source model. Hence, the audio signal is handled as a sequence of frames with a fixed frame length T_f that represent overlapping signal segments. For each frame q, the corresponding signal segment $x^{(q)}(t)$, which is centered around $t = qT_\mathrm{f}$, is described by a set of model parameters of the hybrid source model. Such a frame-based approach is common to practically all audio and speech coding systems.

The optimal frame length depends on many factors, including the target bit rate. In general, however, the frame length is chosen such that the signal can be assumed to be

stationary within a frame. The hybrid parametric model presented here is typically used with a frame length T_f of 32 ms since this was found to give the best overall quality for the given application. When tuning the system towards significantly lower or higher bit rates, it can be beneficial to increase or decrease the frame length, respectively, as will be discussed in Section 3.4.1.

3.1.1 Modeling of Sinusoidal Trajectories

The most prevalent signal components in the hybrid source model are sinusoidal trajectories. A relatively simple mathematical approach to model a tonal signal $x(t)$ is to treat it as the superposition of I individual sinusoidal components $x_i(t)$.

$$x(t) = \sum_{i=1}^{I} x_i(t) = \sum_{i=1}^{I} a_i(t) \cos\left(\varphi_{0,i} + 2\pi \int_0^t f_i(\tau)\, d\tau\right) \tag{3.1}$$

Each of the components is described by slowly varying parameters for its amplitude $a_i(t)$ and frequency $f_i(t)$ and a start phase $\varphi_{0,i}$, thus forming a sinusoidal trajectory. Note that the frequency parameter is regarded as the instantaneous frequency [7]

$$f(t) = \frac{1}{2\pi} \frac{d\varphi(t)}{dt} \tag{3.2}$$

of a signal $\cos(\varphi(t))$ with the instantaneous unwrapped phase $\varphi(t)$. While the signal $x(t)$ is treated in continuous time here, it can be easily converted to discrete time by sampling $x[n] = x(nT_\mathrm{s})$ at a rate of $f_\mathrm{s} = 1/T_\mathrm{s}$. This assumes that all sinusoidal components have frequencies below $f_\mathrm{s}/2$ to avoid aliasing problems.

For each frame q, the corresponding signal segment is described by the parameters for amplitude $a_i^{(q)} = a_i(qT_\mathrm{f})$, frequency $f_i^{(q)} = f_i(qT_\mathrm{f})$, and phase $\varphi_i^{(q)} = \varphi_i(qT_\mathrm{f})$ of the $I^{(q)}$ sinusoidal components that are present in this frame. The number $I^{(q)}$ of sinusoidal signal components can vary over time, i.e., from frame to frame. In order to form sinusoidal trajectories that span several frames, component i in the current frame q can be flagged as the continuation of component $k_i^{(q)}$ in the previous frame $q-1$, which thus becomes its predecessor. If a component in the current frame has no predecessor in the previous frame, this constitutes the start of a new trajectory in the current frame. Correspondingly, if a component in the previous frame is not continued by a successor in the current frame, this constitutes an end of a trajectory in the previous frame. Of course it is also possible that a sinusoidal component exists just in a single frame and neither has a predecessor nor a successor.

Given this parameterization of the sinusoidal trajectories, the signal can be resynthesized from this description. To ensure smooth synthesis, both the amplitude and frequency parameters $a(t)$ and $f(t)$ of a sinusoidal trajectory are interpolated in between frames [70],

[116]. Linear interpolation for the amplitude results in

$$a(t) = a(qT_\mathrm{f}) + \frac{a((q+1)T_\mathrm{f}) - a(qT_\mathrm{f})}{T_\mathrm{f}}(t - qT_\mathrm{f}), \quad qT_\mathrm{f} \le t \le (q+1)T_\mathrm{f}. \tag{3.3}$$

For simplicity of notation, without loss of generality, using $\Omega(t) = 2\pi f(t)$, assuming $q = 0$ and denoting $T = T_\mathrm{f}$, $\Omega_0 = \Omega(0)$, and $\Omega_1 = \Omega(T)$, the linear interpolation of the frequency can be written as

$$\Omega(t) = \Omega_0 + \frac{\Omega_1 - \Omega_0}{T}t, \quad 0 \le t \le T. \tag{3.4}$$

This results in a quadratic function for the unwrapped phase

$$\varphi(t) = \varphi_0 + \int_0^t \Omega(\tau)\,d\tau = \varphi_0 + \Omega_0 t + \frac{\Omega_1 - \Omega_0}{2T}t^2. \tag{3.5}$$

The terminal phase $\varphi(T)$ at the end of the interpolation interval is determined by the three parameters φ_0, Ω_0, and Ω_1. This terminal phase is used as start phase in the next frame to ensure phase-continuous synthesis of a sinusoidal trajectory spanning several frames. Hence only the *start phase* $\varphi_i^{(q)}$ for the first frame of a trajectory is utilized while the values of the phase parameter for the following frames are obsolete. If no phase parameters at all are conveyed from the encoder to the decoder, a fixed or random start phase is used instead. For starting or ending trajectories, the left or right half of a Hann window

$$w_\mathrm{h}(t) = \begin{cases} \cos^2(\frac{\pi}{2}t/T_\mathrm{f}), & |t| \le T_\mathrm{f} \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

is used for fade-in or fade-out, respectively, while the frequency is kept constant.

If signal decomposition and parameter estimation in the encoder is carried out by means of an analysis-by-synthesis approach, it can be advantageous to use different synthesis methods in the encoder and the decoder. Synthesis in the encoder should be as accurate as possible to minimize the analysis-by-synthesis residual, and it can even utilize additional model parameters that are not conveyed to the decoder.

3.1.2 Modeling of Harmonic Tones

A subset of the sinusoidal components that are present in a frame can form a harmonic tone if their frequencies are (approximately) integer multiples of the fundamental frequency f_h of the harmonic tone. In this case, the parameters of the components in this subset are are denoted $a_{i,\mathrm{h}}^{(q)}$, $f_{i,\mathrm{h}}^{(q)}$, and $\varphi_{i,\mathrm{h}}^{(q)}$, while all remaining sinusoidal components that do not belong to the harmonic tone are referred to as *individual sinusoids*. A harmonic tone comprises the $I_\mathrm{h}^{(q)}$ partials $i = 1,\ldots,I_\mathrm{h}^{(q)}$, and actual frequencies $f_{i,\mathrm{h}}^{(q)}$ of the partials

can be approximated sufficiently good by means of the fundamental frequency $f_\mathrm{h}^{(q)}$ and the first order stretching parameter $\kappa_\mathrm{h}^{(q)}$ according to

$$f_{i,\mathrm{h}} = i f_\mathrm{h}(1 + \kappa_\mathrm{h} i) \tag{3.7}$$

where the parameter κ characterizes the amount of stretching. For $\kappa = 0$, the plain harmonic relationship $f_{i,\mathrm{h}} = i f_\mathrm{h}$ is obtained. The stretching parameter allows to accommodate for the slight inharmonicities observed e.g. for free vibration of stiff or loaded strings [27, Section 2.18].

The parameters $I_\mathrm{h}^{(q)}$, $f_\mathrm{h}^{(q)}$, and $\kappa_\mathrm{h}^{(q)}$ can vary over time, and a harmonic tone in the current frame can be flagged to be the continuation of a harmonic tone in the previous frame, which means that the partials form corresponding trajectories. In principle, it would be possible to have two or more harmonic tones present simultaneously in a frame. However, it was found that reliable decomposition into a single harmonic tone plus remaining individual sinusoids is already difficult enough.

The primary advantage of a harmonic tone over representing the partials as individual sinusoids is that this allows for a more compact parameterization. To exploit these benefits, the partials' frequencies are conveyed by means of the fundamental frequency $f_\mathrm{h}^{(q)}$, the stretching parameter $\kappa_\mathrm{h}^{(q)}$, and a parameter indicating the total number $I_\mathrm{h}^{(q)}$ of partials. The partials' amplitudes are conveyed by means of an overall amplitude parameter $a_\mathrm{h}^{(q)}$ in combination with the parameters of spectral envelope model (SEM).

To characterize the overall amplitude of the harmonic tone independently from its spectral shape, the root of the summed squares of the partials' amplitudes is used as amplitude parameter

$$a_\mathrm{h} = \sqrt{\textstyle\sum_{i=1}^{I_\mathrm{h}} a_{i,\mathrm{h}}^2} \tag{3.8}$$

where I_h is the number of partials of the harmonic tone. Thus, a_h is directly related to the total power of the harmonic tone.

The spectral envelope model describes the spectral envelope $|H(\Omega)|$ of a signal. Hence, the partials' amplitudes are given by

$$a_{i,\mathrm{h}} = a_\mathrm{h} |H_\mathrm{s}(2\pi i f_\mathrm{h})| \tag{3.9}$$

where an appropriately scaled version $H_\mathrm{s}(\Omega)$ of the spectral envelope $H(\Omega)$ is employed to account for Equation (3.8).

Since the spectral envelopes considered here have a limited bandwidth, the time-discrete notation $|H(z)|$ with $z = e^{j\omega}$ and a sampling rate of $f_\mathrm{s,SEM}$ is used. This sampling rate can be that of the original signal, i.e., $f_\mathrm{s,SEM} = f_\mathrm{s}$. However, here it is chosen to be twice the frequency of the next partial tone directly above the highest partial tone present in the harmonic tone, i.e., $f_\mathrm{s,SEM}^{(q)} = 2(I_\mathrm{h}^{(q)} + 1) f_\mathrm{h}^{(q)}$, since this allows for a compact parameterization also for harmonic tones with a low bandwidth.

In order to model the spectral envelope $|H(z)|$, $H(z)$ is assumed to be the transfer function of an all-pole IIR filter

$$H(z) = \frac{1}{1 + \sum_{p=1}^{P} a_p z^{-p}} \tag{3.10}$$

of order P, where the filter coefficients a_p constitute the model's parameters. Similar all-pole filters are commonly used in speech coding systems, where this approach is known as linear predictive coding (LPC) [65].

Instead of the filter coefficients a_p, it is advantageous to use the logarithmic area ratio (LAR) [65] representation

$$g_p = \ln \frac{1 + k_p}{1 - k_p} = 2 \operatorname{arctanh}(k_p), \quad 1 \leq p \leq P \tag{3.11}$$

of the reflection coefficients $-1 < k_p < 1$ as parameters of the spectral envelope model. The reflection coefficients k_p can be derived by solving

$$a_p^{(p)} = k_p \tag{3.12}$$

$$a_i^{(p)} = a_i^{(p-1)} + k_p a_{p-i}^{(p-1)}, \quad 1 \leq i \leq p-1 \tag{3.13}$$

for $p = 1, 2, \ldots, P$ with the original filter coefficients being $a_p = a_p^{(P)}$. These equations are part of the Levinson-Durbin algorithm [65] that is commonly used to calculate the coefficients of an all-pole model and that will be discussed in Section 3.2.3.3. The reflection coefficients k_p have the interesting property that they do not depend on the predictor order P, which is not the case for the predictor coefficients $a_p^{(P)}$.

The all-pole model was found to be more efficient than an all-zero model spectral envelope (DCT approach) or a simple grouping of harmonic lines (i.e., a piece-wise constant line-segment approximation) [96]. Furthermore, it allows to easily adapt the filter order $P_{\mathrm{h}}^{(q)}$ (and thus the detailedness) of the spectral envelope modeling, even after LAR parameters have been calculated. The phase parameters $\varphi_{i,\mathrm{h}}^{(q)}$ are usually not conveyed.

By merging the data for the partials of the harmonic tone model with the data for the individual sinusoids, the complete information about all sinusoidal trajectories can be reconstructed. Synthesis of these trajectories can be carried out using the synthesis method discussed above. An interesting situation concerning predecessor information $k_i^{(q)}$ of such trajectories arises if the partials of a harmonic tone are described by the harmonic tone model in one frame, and as individual sinusoids in an adjacent frame. A solution to this problem will be discussed in Section 4.2.1.

3.1.3 Modeling of Transient Components

Due to the comparably long frame length T_{f} of typically 32 ms, additional means to model fast or transient parameter changes within a frame are required. For this purpose, a temporal amplitude envelope $a_{\mathrm{e}}(t)$ can be applied to selected individual sinusoids or a harmonic

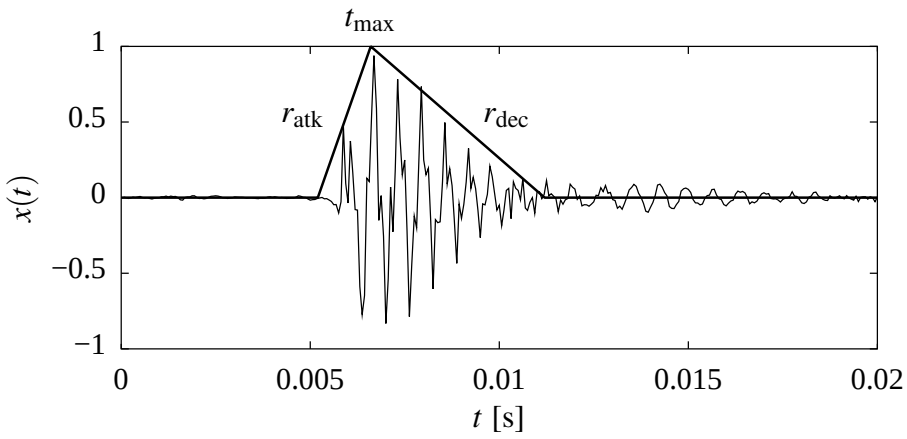

Figure 3.1: Example of waveform of a transient signal $x(t)$ (last click of *castanets*) and approximation of its temporal envelope by an AD envelope model $a_{\text{e}}(t)$ (thick line) described by the three parameters for position $t_{\max}$, attack rate r_{atk}, and decay rate r_{dec}.

tone. To characterize the typical shape of such envelopes in a compact manner, the attack/decay temporal amplitude envelope (AD envelope) model is introduced here

$$a_{\text{e}}(t) = \begin{cases} \max(0, 1 + r_{\text{atk}}(t - t_{\max})), & t \leq t_{\max} \\ \max(0, 1 - r_{\text{dec}}(t - t_{\max})), & \text{otherwise} \end{cases} . \tag{3.14}$$

It is described by three parameters: $t_{\max}$ indicates the temporal position of the maximum amplitude, the attack rate r_{atk} describes the slope of the attack phase with values ranging from 0 (flat) to ∞ (abrupt start, i.e., hard onset), and the decay rate r_{dec} describes the slope of the decay phase with values ranging from 0 (flat) to ∞ (abrupt end). The AD envelope allows to model short transient impulses as well as the abrupt start or end of a signal component. Figure 3.1 shows how such an AD envelope can be used to approximate the temporal shape of a short transient signal, the click of a castanet.

For each frame q, there is a set of envelope flags $e_i^{(q)}$ that for each individual sinusoid indicate whether the AD envelope is to be applied to that sinusoid or whether just the normal parameter interpolation between adjacent frames is needed. Also the harmonic tone has such an envelope flag, which, if set, indicates that the AD envelope is to be applied to all partials of the harmonic tone.

3.1.4 Modeling of Noise Components

In order to enable compact parameterization of the noise component, an all-pole spectral envelope model, Equation (3.10), is employed to described the spectral envelope of the noise component together with an amplitude parameter $a_{\text{n}}^{(q)}$ specifying the total power $\sigma_x^2 = a_{\text{n}}^2$. The resulting power spectral density (PSD) of the noise component is

$$S_{xx}(\omega) = \sigma_x^2 |H_{\text{s}}(e^{j\omega})|^2 \tag{3.15}$$

where an appropriately scaled version $H_{\mathrm{s}}(e^{j\omega}) = H(e^{j\omega})/g$ of the spectral envelope is employed which fulfills

$$1 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_{\mathrm{s}}(e^{j\omega})|^2 \, d\omega. \tag{3.16}$$

The parameters of the all-pole model are the reflection coefficients $k_{p,\mathrm{n}}^{(q)}$, represented as LARs $g_{p,\mathrm{n}}^{(q)}$, and the order $P_{\mathrm{n}}^{(q)}$ of the model can be adapted over time. This is the same parameterization as used to describe the spectral envelope of a harmonic tone. Since noise components can have the same bandwidth as the audio signal that is encoded, the sampling rate of the input signal is also used for the noise spectral envelope model, i.e., $f_{\mathrm{s,SEM}} = f_{\mathrm{s}}$.

For each frame q, there is one parameter set describing the noise component. These parameters characterize an autoregressive process (AR process) that is used to synthesize a noise signal $x^{(q)}(t)$ with the desired power spectrum (or PSD). An overlap-add synthesis

$$x(t) = \sum_{q} w_{\mathrm{l}}(t - qT_{\mathrm{f}})x^{(q)}(t) \tag{3.17}$$

with a low-overlap (or tapered) window

$$w_{\mathrm{l}}(t) = \begin{cases} 1, & \frac{|t|}{T_{\mathrm{f}}} \leq \frac{1-\lambda}{2} \\ \cos(\frac{\pi}{2}(\frac{1}{2} + \frac{1}{\lambda}(\frac{|t|}{T_{\mathrm{f}}} - \frac{1}{2}))), & \frac{1-\lambda}{2} < \frac{|t|}{T_{\mathrm{f}}} \leq \frac{1+\lambda}{2} \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

is used to handle parameter variation between frames. The value of $\lambda = 1/4$ for the shape of the low-overlap window was found to give the best temporal resolution without audible discontinuities at frame boundaries. To improve modeling, an optional temporal amplitude envelope $a_{\mathrm{e}}(t)$ can be applied to the noise signal prior to overlap-add synthesis. This is an additional AD envelope (independent from the one used for sinusoidal components) described by the three parameters $t_{\mathrm{max,n}}^{(q)}$, $r_{\mathrm{atk,n}}^{(q)}$, and $r_{\mathrm{dec,n}}^{(q)}$.

3.2 Parameter Estimation for Single Signal Components

This section studies in detail the estimation of the parameters of a single signal component. All the different components types supported by the chosen hybrid source model are addressed, i.e., sinusoidal trajectories, harmonic tones, transients, and noise components. The parameter estimation should be both accurate and robust. Good accuracy is important in order to allow perceptually equivalent resynthesis of a component and to facilitate analysis-by-synthesis based signal decomposition which could otherwise suffer from increased residuals due to inaccurate resynthesis. Good robustness allows to estimate the parameters of a signal component in spite of non-model signals (e.g. other signal

components) being present simultaneously, which is helpful for the segregation of signal components during signal decomposition. In accordance with the frame-based approach employed by the hybrid source model, parameter estimation is carried out on overlapping signal segments $x^{(q)}(t)$ centered around $t = qT_\mathrm{f}$, i.e., the middle of the frame q for which component parameters are being estimated.

3.2.1 Estimation of Sinusoidal Trajectory Parameters

The development of the algorithms for estimation of the parameters of a sinusoidal trajectory went through different stages. Firstly, the most basic case is treated here, the estimation of frequency, amplitude, and phase of a complex sinusoid in complex white Gaussian noise. It forms the basis of a robust algorithm for frequency estimation that is guided by an initial coarse estimate of the sinusoidal component's frequency. This algorithm is then adapted to allow also the estimation of the sweep rate of a chirp. An extension of this algorithm that allows to track a sinusoidal component over consecutive frames will be presented in Section 3.2.2.2.

3.2.1.1 Fundamentals of Frequency and Amplitude Estimation

Fundamental aspects of parameter estimation for a sinusoidal signal can be studied best for the basic case of a single complex sinusoid in complex white Gaussian noise. The estimation takes an observed signal segment $x[n]$ of length N as input

$$x[n] = a_0 e^{j(\varphi_0 + \omega_0 n)} + z[n], \quad n = 0, 1, \ldots, N-1, \tag{3.19}$$

where the unknown constants a_0, $\omega_0 = 2\pi f_0 / f_\mathrm{s}$, and φ_0 are the parameters of the sinusoid that are to be estimated, and where $z[n]$ represents a complex white Gaussian noise process with zero mean and variance σ_z^2.

3.2.1.1.1 Maximum Likelihood Estimator The optimal (and hence also unbiased) maximum likelihood estimator (MLE) for the frequency ω_0 is known to be given by the location of the peak of the periodogram [56, Section 13.3]

$$\hat{\omega}_0 = \operatorname*{argmax}_{\omega} P_{xx}(\omega). \tag{3.20}$$

An MLE, as discussed here, has no a-priori knowledge about the estimated parameter, i.e., the parameter's PDF is assumed to be flat $p(\omega_0) = \frac{1}{2\pi}, \quad -\pi < \omega_0 \leq \pi$. The periodogram $P_{xx}(\omega)$ is an estimate of the power spectrum (or PSD) $S_{xx}(\omega)$ of the signal segment $x[n]$ [95, Section 12.1.2] and is given by

$$P_{xx}(\omega) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\omega n} \right|^2 = \frac{1}{N} \left| \mathscr{F}\{x[n]\}(\omega) \right|^2 \tag{3.21}$$

as a scaled version of the squared magnitude of the Fourier transform $\mathscr{F}\{x[n]\}(\omega)$.

Once the frequency estimate $\hat{\omega}_0$ is found, amplitude a_0 and phase φ_0 can be easily estimated. To simplify notation, the complex amplitude $A_0 = a_0 e^{j\varphi_0}$ is introduced, which allows to give the MLE for amplitude $\hat{a}_0 = |\hat{A}_0|$ and phase $\hat{\varphi}_0 = \arg(\hat{A}_0)$ as

$$\hat{a}_0 e^{j\hat{\varphi}_0} = \hat{A}_0 = \frac{1}{N}\sum_{n=0}^{N-1} x[n] e^{-j\hat{\omega}_0 n}. \tag{3.22}$$

Thus, the estimated complex amplitude $\hat{A}_0$ is the cross-correlation between the observed signal segment and a complex sinusoid having the estimated frequency. This estimate $\hat{A}_0$ also has the property that it minimizes the energy $\sum_{n=0}^{N-1} |r[n]|^2$ of the modeling error $r[n] = x[n] - \hat{A}_0 e^{j\hat{\omega}_0 n}$.

The accuracy of the estimated parameters depends on the signal-to-noise ratio (SNR) a_0^2/σ_z^2 and number N of samples in the observed signal segment. The optimal MLEs, as given above, attain the Cramér-Rao bound (CR bound) for the variance of the estimation error. Specifically, the CR bounds are [56, Section 13.4.1]

$$\operatorname{var}(\hat{\omega}_0) \geq \frac{6\sigma_z^2}{a_0^2 N(N^2-1)}, \tag{3.23}$$

$$\operatorname{var}(\hat{a}_0) \geq \frac{\sigma_z^2}{2N}, \tag{3.24}$$

$$\operatorname{var}(\hat{\varphi}_0) \geq \frac{\sigma_z^2(2N-1)}{a_0^2 N(N+1)}. \tag{3.25}$$

For the case of a real-valued sinusoid in real-valued noise,

$$x[n] = a_\mathrm{r}\cos(\varphi_\mathrm{r} + \omega_\mathrm{r} n) + z[n] = a_\mathrm{r}\frac{1}{2}\left(e^{j(\varphi_\mathrm{r}+\omega_\mathrm{r} n)} + e^{-j(\varphi_\mathrm{r}+\omega_\mathrm{r} n)}\right) + z[n], \tag{3.26}$$

the same estimators as given in Equations (3.20) and (3.22) can be employed. Due to the symmetry of the spectrum of a real-valued signal, only the peak in the positive half of the periodogram has to be considered and the estimated amplitude has to be corrected according to Equation (3.26), i.e., $\hat{a}_\mathrm{r} = 2\hat{a}_0$. For $\hat{\varphi}_\mathrm{r} = \hat{\varphi}_0$ and $\hat{\omega}_\mathrm{r} = \hat{\omega}_0$ no corrections are needed. The CR bounds given in Equations (3.23), (3.24) (with $\operatorname{var}(\hat{a}_\mathrm{r}) = 4\operatorname{var}(\hat{a}_0)$), and (3.25) are attained if the SNR is sufficiently high.

3.2.1.1.2 Linear Regression of Phase Data If the observed signal segment has a sufficiently high SNR, the frequency estimator given in Equation (3.20) is equivalent to a linear regression of the phase data of the observed signal [57], [122]. This interpretation of the frequency estimator is of interest here since it can easily be extended to enable sweep rate estimation of chirps and tracking of sinusoids, which will be discussed in Sections 3.2.1.2 to 3.2.2. To utilize phase data, the signal model, Equation (3.19), is replaced

by an approximate model

$$x[n] = (1+v[n])a_0 e^{j(\varphi_0+\omega_0 n+u[n])} \approx a_0 e^{j(\varphi_0+\omega_0 n+u[n])}, \quad n = 0,1,\ldots,N-1 \tag{3.27}$$

where $u[n]$ is a real-valued zero mean white Gaussian noise with variance $\sigma_z^2/2a_0^2$. For this purpose, the complex noise $z[n]$ in the original model was decomposed into two real-valued orthogonal components, the phase noise $u[n]$ and the magnitude noise $v[n]$, which can be neglected if the SNR is high enough [122].

Denoting the phase of $x[n]$ by $\angle x[n]$, the phase data of the observed signal segment can be written as

$$\varphi[n] = \angle x[n] = \varphi_0 + \omega_0 n + u[n], \quad n = 0,1,\ldots,N-1. \tag{3.28}$$

Since the observed phase data $\arg(x[n])$ has a 2π ambiguity, proper phase unwrapping is necessary to obtain $\varphi[n]$. Assuming low frequencies $|\omega_0| \ll \pi$ and a sufficiently high SNR, this can be achieved by accumulating phase differences

$$\varphi[n] = \arg(x[0]) + \sum_{i=0}^{n-1} \arg(x^*[i]x[i+1]). \tag{3.29}$$

The observed phase data $\varphi[n]$ can now be approximated by the linearly increasing phase over time of a sinusoid with constant frequency

$$\hat{\varphi}[n] = \hat{\varphi}_0 + \hat{\omega}_0 n, \quad n = 0,1,\ldots,N-1. \tag{3.30}$$

The optimal match in a minimum mean square error (MSE) sense, i.e., minimizing $\sum_{n=0}^{N-1}(\varphi[n] - (\hat{\varphi}_0 + \hat{\omega}_0 n))^2$, is equivalent to a linear regression of the observed phase data. Thus, the optimal values of $\hat{\varphi}_0$ and $\hat{\omega}_0$ are given by the linear equation system

$$\begin{bmatrix} \sum_{n=0}^{N-1} n^2 & \sum_{n=0}^{N-1} n \\ \sum_{n=0}^{N-1} n & \sum_{n=0}^{N-1} 1 \end{bmatrix} \begin{bmatrix} \hat{\omega}_0 \\ \hat{\varphi}_0 \end{bmatrix} = \begin{bmatrix} \sum_{n=0}^{N-1} \varphi[n]n \\ \sum_{n=0}^{N-1} \varphi[n] \end{bmatrix} \tag{3.31}$$

with $\sum_{n=0}^{N-1} 1 = N$, $\sum_{n=0}^{N-1} n = (N-1)N/2$, and $\sum_{n=0}^{N-1} n^2 = (N-1)N(2N-1)/6$. It can be shown that this estimator for $\hat{\omega}_0$ is the same optimal MLE as given by Equation (3.20) if the SNR is sufficiently large [57], [122]. If the phase estimate $\hat{\varphi}_0$ from the linear regression, Equation (3.31), is of interest as well, the reference point of time should be in the middle of the analyzed data segment in order to make estimation errors for phase independent from frequency errors [122]. This means $n = -(N-1)/2,\ldots,(N-1)/2$ instead of $n = 0,\ldots,N-1$ in Equations (3.30) and (3.31). A major advantage of the estimator using linear regression of phase data, when compared to the estimator finding the location of the maximum in the periodogram, is that it allows for a computationally more efficient implementation. Furthermore, this estimator can be nicely extended towards parameter estimation for time-varying sinusoids as will be described in Section 3.2.1.3.

3.2.1.2 Guided Frequency Estimation

In order to apply the fundamental techniques of frequency estimation of sinusoids discussed in the previous section to real-world signals containing several sinusoidal and other signal components, the concept of guided frequency estimation is introduced here. It assumes that an initial coarse estimate of the frequency of a sinusoidal signal component is available, for example the location of a peak in the discrete Fourier transform of the current frame of the signal, which is typically provided by a signal decomposition framework, as will be discussed in Section 3.3.

Given such an initial coarse estimate $\hat{f}_0'$, it is possible to reduce interference from neighboring sinusoidal components and to meet the assumption of a single sinusoid in noise, as made in the previous section, by introducing a bandpass filter with a passband centered at the initial estimate of the sinusoid's frequency. In combination with the length (or duration) of the analyzed signal segment, estimation thus evaluates only a section of the time-frequency plane. The filter bandwidth has to be chosen carefully to achieve a good trade-off between maximum attenuation of neighboring components and minimum attenuation of the sinusoid to be estimated, which may be located off-center due to an error in the initial estimate. In the latter case, the bandpass filtering can introduce a bias of the frequency estimate towards the center of the passband, i.e., the initial frequency estimate. Effects of the trade-off between segment length and bandwidth will be discussed in Section 3.2.1.3 in connection with Figure 3.3.

To implement said bandpass filtering in a flexible and efficient manner, an approach known from heterodyne radio receivers is employed. A local oscillator signal

$$c(t) = e^{-j2\pi f_c t} \tag{3.32}$$

with the frequency of the initial estimate $f_c = \hat{f}_0'$ is used to heterodyne (i.e., down-mix) the original signal $x(t)$, which is then lowpass filtered to obtain a complex baseband signal

$$x_b(t) = (x(t)c(t)) * w(t). \tag{3.33}$$

Here, $w(t)$ is the impulse response of the lowpass filter and $*$ denotes convolution. In order to simplify further analysis, a delay-free lowpass filter with symmetric impulse response $w(t) = w(-t)$ is used. The down-mix process corresponds to a shift of the original signal $x(t)$ in the frequency domain by $-\hat{f}_0'$ such that the spectral peak representing the to-be-estimated sinusoid with frequency f_0 is now located in the baseband around frequency $f_{b,0} = f_0 - \hat{f}_0' \approx 0$. Since the original signal $x(t)$ is real-valued, there is another peak at $-f_0$, which does not carry additional information and which is shifted to $-f_0 - \hat{f}_0'$ and hence suppressed in the baseband signal $x_b(t)$ by the lowpass filter $w(t)$.

Due to the limited bandwidth of the baseband signal $x_b(t)$ after lowpass filtering, it can be represented at a much lower sampling rate f_d than the original signal, which is sampled at f_s. This is accomplished by down-sampling with a factor of $D = f_s/f_d$, i.e., with a sampling interval $T_d = DT_s = 1/f_d$. The downsampled baseband signal can be

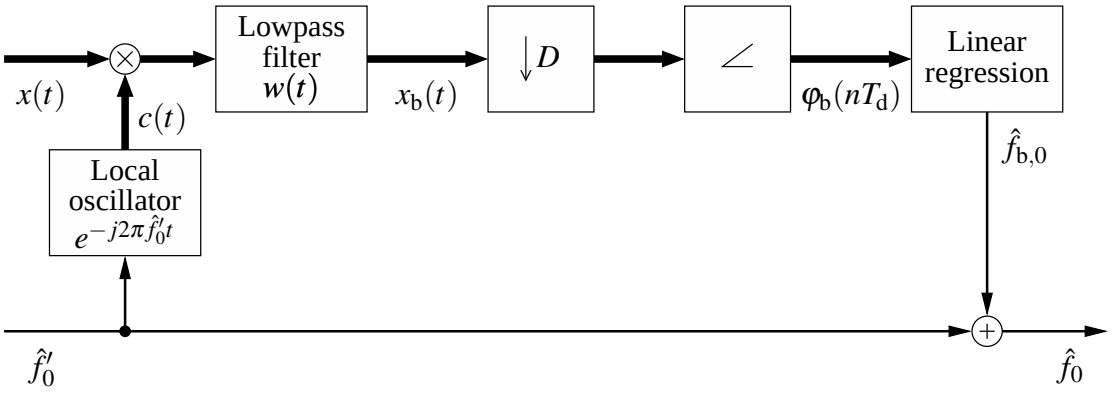

Figure 3.2: Heterodyne-based estimation of frequency $\hat{f}_0$ of sinusoidal component in signal $x(t)$ guided by initial coarse estimate $\hat{f}_0'$.

considered as a single complex sinusoid with a frequency $f_{\mathrm{b},0}$ close to zero plus a noise signal which accounts for in-band noise components and leakage from other components in the original signal that were not sufficiently suppressed by the lowpass filter. It thus conforms to the signal model given by Equation (3.19) as used in Section 3.2.1.1. The unwrapped phase $\varphi_\mathrm{b}(nT_\mathrm{d})$ of the baseband signal is now approximated by

$$\hat{\varphi}_\mathrm{b}(t) = \hat{\varphi}_{\mathrm{b},0} + \hat{\Omega}_{\mathrm{b},0} t \tag{3.34}$$

and the optimal parameters $\hat{\varphi}_{\mathrm{b},0}$ and $\hat{\Omega}_{\mathrm{b},0}$ are found by linear regression as outlined in Equation (3.31). The slope $\hat{\Omega}_{\mathrm{b},0}$ of this linear approximation of the phase over time is an accurate estimate of frequency $\hat{f}_{\mathrm{b},0} = \hat{\Omega}_{\mathrm{b},0}/2\pi$ of the complex sinusoid in the baseband signal. Finally, the initial estimate $\hat{f}_0'$ is added to obtain an accurate estimate of the frequency $\hat{f}_0 = \hat{f}_{\mathrm{b},0} + \hat{f}_0'$ of the sinusoidal component in the original signal. Figure 3.2 gives an overview of the complete guided frequency estimation technique presented here.

3.2.1.3 Parameter Estimation for Time-Varying Sinusoids

To accommodate for the time-varying frequency of a sinusoidal trajectory in case of vibrato or portamento, the heterodyne-based frequency estimator has been extended to permit also estimation of the sweep rate of linearly changing frequencies. For this purpose, the filter bandwidth had to be increased to cover the frequency range traversed during the duration of the signal segment.

Figure 3.3 visualizes the problem of trading off between the effective length of the analyzed signal segment and the bandwidth of the bandpass filter. It shows, on the time-frequency plane, the partials of a harmonic tone with a fundamental frequency that sweeps from 80 Hz to 120 Hz in the course of 100 ms, a moderate chirp rate that can be encountered e.g. in speech signals. For an effective segment length of $\Delta t = T_\mathrm{f} = 32$ ms typically used here, the minimum meaningful bandwidth according to the principle of time-frequency uncertainty $\Delta t \Delta f \geq 1$ is $\Delta f = 32$ Hz. This is shown as (a), and it can be seen

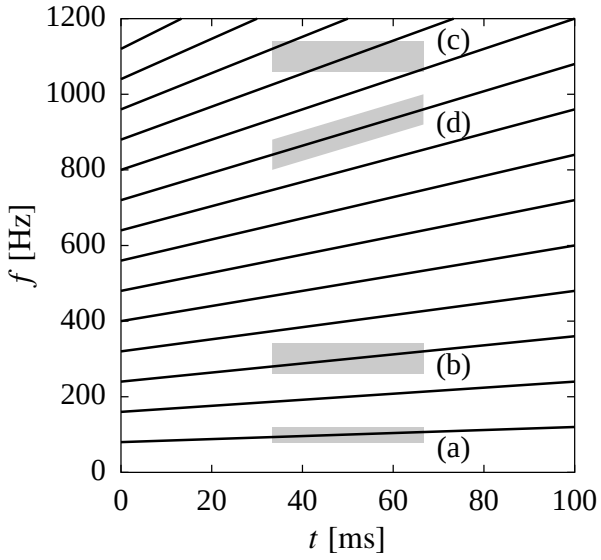

Figure 3.3: Rectangular (a), (b), (c), and slanted (d) sections of the time-frequency plane with length of 32 ms and width of 32 Hz (a), and 64 Hz (b), (c), (d), for analyzing the partials of a harmonic tone with fundamental frequency sweeping from 80 Hz to 120 Hz in 100 ms.

that this bandwidth is even sufficient to cover the sweeping first partial (i.e., fundamental frequency) component of this example. For higher partials an increased filter bandwidth is needed, and (b) shows the 3rd partial covered by a filter bandwidth of 64 Hz. For even higher partials with correspondingly higher absolute chirp rates, a bandwidth increase is insufficient as neighboring partials begin to appear in the passband due to the length of the analyzed signal segment, which is illustrated as (c) for the 11th partial. This problem can be addressed by analyzing a slanted section of the time-frequency plane, which is shown as (d) for the 9th partial and will be discussed in detail in Section 3.2.2.

3.2.1.3.1 Extensions for Chirp Rate Estimation It is possible to extend the heterodyne-based guided frequency estimation presented in Section 3.2.1.2 such that the chirp rate α_0 is estimated as a third parameter, in addition to frequency Ω_0 and phase φ_0. This assumes a sinusoid with sweeping frequency and a low to moderate chirp rate that can be covered by an increased bandwidth of the analyzed signal segment, as shown in Figure 3.3 (b). For this purpose, the original signal model of the linear regression based frequency estimation, Equation (3.34), is extended in a manner similar to Equation (3.5), which gives

$$\hat{\varphi}_{\mathrm{b}}(t) = \hat{\varphi}_{\mathrm{b},0} + \hat{\Omega}_{\mathrm{b},0} t + \hat{\alpha}_{\mathrm{b},0} t^2. \tag{3.35}$$

Using the unwrapped phase data $\varphi_{\mathrm{b}}(nT_{\mathrm{d}})$, the model parameters $\hat{\varphi}_{\mathrm{b},0}$, $\hat{\Omega}_{\mathrm{b},0}$, and $\hat{\alpha}_{\mathrm{b},0}$ achieving the best match in a minimum MSE sense correspond to a quadratic regression

of the data. Hence, the optimal parameters are given by the linear equation system

$$\begin{bmatrix} \sum_N n^4 & \sum_N n^3 & \sum_N n^2 \\ \sum_N n^3 & \sum_N n^2 & \sum_N n \\ \sum_N n^2 & \sum_N n & \sum_N 1 \end{bmatrix} \begin{bmatrix} \hat{\alpha}_{\mathrm{b},0} T_\mathrm{d}^2 \\ \hat{\Omega}_{\mathrm{b},0} T_\mathrm{d} \\ \hat{\varphi}_{\mathrm{b},0} \end{bmatrix} = \begin{bmatrix} \sum_N \varphi[n] n^2 \\ \sum_N \varphi[n] n \\ \sum_N \varphi[n] \end{bmatrix} \tag{3.36}$$

using N data points $\varphi[n] = \varphi_\mathrm{b}(nT_\mathrm{d})$ sampled at $t = nT_\mathrm{d}$, and hence $\hat{\omega}_{\mathrm{b},0} = \hat{\Omega}_{\mathrm{b},0} T_\mathrm{d}$.

When this quadratic regression is used in the heterodyne-based framework shown in Figure 3.2, the initial estimate $\hat{f}_0'$ must be added to obtain an accurate estimate of the frequency $\hat{f}_0 = \hat{f}_{\mathrm{b},0} + \hat{f}_0'$ of the sinusoidal component in the original signal. For the chirp rate $\hat{\alpha}_0$ no such correction is needed as long as the local oscillator signal $c(t)$ has a constant frequency.

3.2.1.3.2 Amplitude Estimation In general, not only the frequency but also the amplitude of a sinusoidal trajectory varies over time. Therefore, amplitude estimation should be confined to a resonably short segment of the original signal with help of a temporal window $w_\mathrm{a}(t)$. The evolution of amplitude over time can then be obtained by a sequence of amplitude estimates with shifted temporal windows.

Assuming that a sufficiently good estimate $\hat{\omega}_0$ of the constant frequency of a sinusoidal component in a real-valued signal $x(t)$ is available, the amplitude $\hat{a}_0$ and phase $\hat{\varphi}_0$ of this component can be estimated by minimizing the energy $\int r^2(t)\,dt$ of the windowed residual

$$r(t) = w_\mathrm{a}(t) \left(x(t) - a_\mathrm{e}(t) \left(\hat{c}_0 \cos(\hat{\Omega}_0 t) + \hat{s}_0 \sin(\hat{\Omega}_0 t) \right) \right) \tag{3.37}$$

where $w_\mathrm{a}(t)$ is the window determining the temporal location and extent of the examined signal segment. The gain factors $\hat{c}_0$ and $\hat{s}_0$ are the Cartesian representation of amplitude and phase as described by

$$\hat{a}_0 e^{j\hat{\varphi}_0} = \hat{c}_0 - j\hat{s}_0 \tag{3.38}$$

where the negative contribution of the imaginary component follows from Equation (3.26). The optional temporal amplitude envelope $a_\mathrm{e}(t)$ will be discussed later and defaults to $a_\mathrm{e}(t) = 1$ when not used.

The estimated gain factors $\hat{c}_0$ and $\hat{s}_0$ minimizing the residual are given by the linear equation system

$$\begin{bmatrix} \int w_\mathrm{a}^2(t) a_\mathrm{e}^2(t) \cos^2(\hat{\Omega}_0 t)\,dt & \int w_\mathrm{a}^2(t) a_\mathrm{e}^2(t) \cos(\hat{\Omega}_0 t) \sin(\hat{\Omega}_0 t)\,dt \\ \int w_\mathrm{a}^2(t) a_\mathrm{e}^2(t) \cos(\hat{\Omega}_0 t) \sin(\hat{\Omega}_0 t)\,dt & \int w_\mathrm{a}^2(t) a_\mathrm{e}^2(t) \sin^2(\hat{\Omega}_0 t)\,dt \end{bmatrix} \begin{bmatrix} \hat{c}_0 \\ \hat{s}_0 \end{bmatrix} = \begin{bmatrix} \int w_\mathrm{a}^2(t) a_\mathrm{e}(t) x(t) \cos(\hat{\Omega}_0 t)\,dt \\ \int w_\mathrm{a}^2(t) a_\mathrm{e}(t) x(t) \sin(\hat{\Omega}_0 t)\,dt \end{bmatrix}. \tag{3.39}$$

Given these estimates $\hat{c}_0$ and $\hat{s}_0$, the energy of the windowed residual $r(t)$ is

$$\begin{aligned}\int r^2(t)\,dt = \int w_\mathrm{a}^2(t)x^2(t)\,dt - \hat{c}_0 \int w_\mathrm{a}^2(t)a_\mathrm{e}(t)x(t)\cos(\hat{\Omega}_0 t)\,dt \\ -\hat{s}_0 \int w_\mathrm{a}^2(t)a_\mathrm{e}(t)x(t)\sin(\hat{\Omega}_0 t)\,dt\end{aligned} \tag{3.40}$$

where the first integral is the energy of the windowed signal $x(t)$ and the where the second and third integrals are the same as on the right side of Equation (3.39).

The estimation in Equation (3.39) differs slightly from the amplitude estimation of a complex sinusoid as given in Equation (3.22) because the windowed sin() and cos() components in Equation (3.37) are not necessarily orthogonal, which leads to non-zero values of the two non-diagonal elements of the matrix in Equation (3.39). This effect is pronounced at low frequencies, i.e., when the period $1/2\pi\hat{\Omega}_0$ of the signal gets into the same order of magnitude as the length of the effective window $w_\mathrm{a}(t)a_\mathrm{e}(t)$.

The amplitude estimation described here can be easily adapted to sinusoidal components with time-varying frequency. For this, a sufficiently good estimate of the frequency trajectory $\hat{\Omega}(t)$ of the component is required, as, for example, described by the estimated frequency and chirp rate parameters $\hat{\Omega}_0$ and $\hat{\alpha}_0$. The corresponding phase trajectory $\hat{\varphi}(t) = \int_0^t \hat{\Omega}(\tau)\,d\tau$ is then used instead of $\hat{\Omega}_0 t$ as the argument of the cos() and sin() functions in Equation (3.39).

If an estimate of the shape of the temporal amplitude envelope $a_\mathrm{e}(t)$ of the sinusoidal component is available, it can be taken into account for the amplitude estimation given by Equation (3.39). This, of course, assumes that the same amplitude envelope $a_\mathrm{e}(t)$ is also applied during signal synthesis. Such an estimated amplitude envelope can be used to improve modeling of transients, and techniques to estimate the parameters of an AD envelope model, Equation (3.14), will be discussed in Section 3.2.4.

The temporal amplitude envelope $a_\mathrm{e}(t)$ in Equation (3.39) can also be used to resemble the linear amplitude interpolation, Equation (3.3), typically used for signal synthesis. Assuming that coarse amplitude estimates for the previous and the current frame, $\hat{a}_0'^{(-1)}$ and $\hat{a}_0'^{(0)}$, respectively, are available, the effect of linear amplitude interpolation can be addressed by means of the constant-slope envelope

$$a_\mathrm{e}(t) = 1 + \frac{t}{T_\mathrm{f}} \frac{\hat{a}_0'^{(0)} - \hat{a}_0'^{(-1)}}{\hat{a}_0'^{(0)}} \tag{3.41}$$

with the current frame centered around $t = 0$ with $a_\mathrm{e}(0) = 1$.

3.2.1.3.3 Simulation Results In order to judge the accuracy of amplitude and phase estimation under conditions typical for the given application and to asses the effect of real-valued amplitude estimation and of the shape of the window $w_\mathrm{a}(t)$, various simulations were carried out for a synthetic signal comprising a single sinusoid with constant

frequency and amplitude in white noise. An SNR $a_\mathrm{r}^2/2\sigma_z^2$ of 0 dB was used in case of a real-valued signal, Equation (3.26), with $a_\mathrm{r} = 1$ and $\sigma_z^2 = 1/2$. The corresponding amplitude $a_0 = 1/2$ was used in case of a complex-valued signal, Equation (3.19), which translates to an SNR a_0^2/σ_z^2 of -3 dB. The typical segment length of $\Delta t = 32$ ms corresponds to a signal segment with $N = 512$ samples at $f_\mathrm{s} = 16$ kHz. For these simulations, the start phase $\varphi_\mathrm{r} = 1/2$ and the frequency $\Omega_\mathrm{r}/2\pi = f_\mathrm{r} = 5000$ Hz was used.

The results of these simulations together with the theoretical CR bounds according to Equations (3.23) to (3.25) are shown in Table 3.1. Mean and standard deviation of the estimated parameters were calculated based on a total of $K = 300$ simulations, each with independent noise. Because the true values of the estimated parameters are constant in this simulation, the standard deviation of the estimated parameter is identical to the standard deviation of the estimation error, i.e., $\sqrt{\mathrm{var}(\hat{f}_\mathrm{r})} = \sqrt{\mathrm{var}(\hat{f}_\mathrm{r} - f_\mathrm{r})}$, using the estimated frequency $\hat{f}_\mathrm{r}$ as an example. For all estimated parameters, the mean (not shown in Table 3.1) was always very close to the true value, i.e., $\overline{\hat{f}_\mathrm{r} - f_\mathrm{r}} \ll \sqrt{\mathrm{var}(\hat{f}_\mathrm{r} - f_\mathrm{r})}$, using again the frequency parameter as an example. This confirms that all estimators are unbiased. It should be noted that values measured for the standard deviation in these simulations themselves have an uncertainty of approximately $1/\sqrt{K}$ (i.e., $\approx 6\%$).

It can be seen that in case of complex amplitude estimation, Equation (3.22), the measured parameter estimation error variances are, as expected, very close to the values of the CR bounds for both a complex-valued signal in complex-valued noise and a real-valued signal in real-valued noise. If instead of the estimated frequency $\hat{f}_\mathrm{r}$ according to Equation (3.20) the true frequency f_r is used, there is no significant change of the amplitude error, while the variance of the phase error is reduced by a factor of approximately $1/4$. The latter effect is due to the fact that $n = 0$ is used as reference point for the phase φ_0 in Equation (3.19), which means that a frequency error results in a corresponding bias of the phase estimate, Equation (3.22). As explained earlier, this effect can be avoided by using the middle of the segment $(N-1)/2$ as reference point for the phase. The CR bound for this center-referenced phase $\hat{\varphi}_{0,\mathrm{c}} = \hat{\varphi}_0 + \hat{\omega}_0(N-1)/2$ can be derived from the CR bounds for $\hat{\varphi}_0$ and $\hat{\omega}_0$, Equations (3.25) and (3.23), as

$$\mathrm{var}(\hat{\varphi}_{0,\mathrm{c}}) = \mathrm{var}(\hat{\varphi}_0) - \mathrm{var}(\hat{\omega}_0)\left(\frac{N-1}{2}\right)^2 \geq \frac{\sigma_z^2}{a_0^2 2N}. \tag{3.42}$$

For large N, this is approximately $1/4$ of the CR bound of the variance for $\hat{\varphi}_0$. The last column in Table 3.1 shows that error of the center-referenced phase estimation is not significantly affected by using the true instead of the estimated frequency.

When real amplitude estimation, Equation (3.39), with a rectangular window $w_\mathrm{a}[n] = 1$ for $n = 0 \ldots N-1$ is used instead of complex amplitude estimation, the measured parameter variances are not significantly different from those for complex amplitude estimation. For the last pair of simulations shown in the table, a Hann window $w_\mathrm{a}[n] = \sin^2((n+1/2)\pi/N)$ for $n = 0 \ldots N-1$ of the same total length as the rectangular window

	$\sqrt{\mathrm{var}(\hat{f}_\mathrm{r} - f_\mathrm{r})}$ [Hz]	$\sqrt{\mathrm{var}(\hat{a}_\mathrm{r} - a_\mathrm{r})}$	$\sqrt{\mathrm{var}(\hat{\varphi}_\mathrm{r} - \varphi_\mathrm{r})}$ [rad]	$\sqrt{\mathrm{var}(\hat{\varphi}_\mathrm{r,c} - \varphi_\mathrm{r,c})}$ [rad]
CR bound	0.7614 (100%)	0.0442 (100%)	0.0883 (100%)	0.0442 (100%)
Complex sinusoid in complex noise, complex estimation				
estim. freq.	0.7619 (100%)	0.0472 (107%)	0.0878 (99%)	0.0456 (103%)
true freq.	0.0000 (0%)	0.0458 (104%)	0.0431 (49%)	0.0431 (97%)
Real sinusoid in real noise, complex estimation				
estim. freq.	0.7310 (96%)	0.0443 (100%)	0.0872 (99%)	0.0463 (105%)
true freq.	0.0000 (0%)	0.0459 (104%)	0.0427 (48%)	0.0427 (97%)
Real sinusoid in real noise, real estimation, rectangular window				
estim. freq.	0.7376 (97%)	0.0425 (96%)	0.0877 (99%)	0.0442 (100%)
true freq.	0.0000 (0%)	0.0437 (99%)	0.0420 (48%)	0.0420 (95%)
Real sinusoid in real noise, real estimation, Hann window				
estim. freq.	0.7043 (93%)	0.0594 (134%)	0.0899 (102%)	0.0607 (137%)
true freq.	0.0000 (0%)	0.0599 (135%)	0.0670 (76%)	0.0670 (152%)

Table 3.1: Estimation error, given as standard deviation $\sqrt{\mathrm{var}(\cdot)}$, for the parameters frequency $\hat{f}_\mathrm{r} = \hat{f}_0$, amplitude $\hat{a}_\mathrm{r} = 2\hat{a}_0$, phase $\hat{\varphi}_\mathrm{r} = \hat{\varphi}_0$, and center-referenced phase $\hat{\varphi}_\mathrm{r,c} = \hat{\varphi}_\mathrm{0,c}$ of a single sinusoid with constant frequency and amplitude in white noise for a signal segment with $N = 512$ samples at $f_\mathrm{s} = 16$ kHz. The signal is real- or complex-valued, with $a_\mathrm{r} = 2a_0 = 1$ and $\sigma_z^2 = 1/2$, corresponding to an SNR $a_\mathrm{r}^2/2\sigma_z^2$ of 0 dB in case of a real-valued signal, or an SNR a_0^2/σ_z^2 of -3 dB in case of a complex-valued signal. The CR bounds (100% reference) are given together with simulation results calculated from $K = 300$ independent measurements using complex or real amplitude estimation with rectangular or Hann window in combination with estimated or true frequency. Note that the errors (as observed in the simulation) themselves have an uncertainty of approximately $1/\sqrt{K}$ (i.e., $\approx 6\%$), explaining why they can be below the CR bound.

was used for the real amplitude estimation. In this case, the variance of the error of the estimated amplitude $\hat{a}_0$ and center-referenced phase $\hat{\varphi}_\mathrm{0,c}$ are almost doubled. This can be explained by the fact that the effective length of the Hann window is only approximately $N/2$. To compensate for this, longer Hann windows can be used, as will be exemplified in Section 3.2.2.

3.2.1.4 Computationally Efficient Frequency Estimation

The parameter estimation techniques presented in Sections 3.2.1.2 to 3.2.1.3 were designed primarily with focus on reliable and accurate estimation. Since parameter estimation is one of the modules that contributes most to the computational complexity of a complete encoder, it is also of interest to study alternative estimation techniques with focus on low computational complexity. These techniques, however, might require to

slightly compromise reliability and accuracy. A detailed description of a computationally efficient frequency estimation algorithm can be found in [105]. It employs a matching pursuit based atomic decomposition [66] using a dictionary comprising sinusoids, and is implemented in the frequency domain to achieve low computational complexity [127].

3.2.2 Building Sinusoidal Trajectories

The previous section treated the parameter estimation for a sinusoidal component within a short signal segment. Now the problem of building longer sinusoidal trajectories that span several segments is considered and two different approaches are presented.

3.2.2.1 Trajectory-Building by Parameter Matching

A simple trajectory-building approach is based on finding the best matches between frequency and amplitude parameters of the sinusoids estimated independently in consecutive segments. For this purpose, a quality measure $q_{k,i}$ is defined which assesses the similarity of frequency $f_k^{(-1)}$ and amplitude $a_k^{(-1)}$ of the kth sinusoid in the previous frame and frequency $f_i^{(0)}$ and amplitude $a_i^{(0)}$ of the ith sinusoid in the current frame:

$$q_f = 1 - \frac{\max(r_f, 1/r_f) - 1}{r_{f,\max} - 1} \quad \text{with} \quad r_f = \frac{f_i^{(0)}}{f_k^{(-1)}}, \tag{3.43}$$

$$q_a = 1 - \frac{\max(r_a, 1/r_a) - 1}{r_{a,\max} - 1} \quad \text{with} \quad r_a = \frac{a_i^{(0)}}{a_k^{(-1)}}, \tag{3.44}$$

$$q_{k,i} = \max(0, q_f) \cdot \max(0, q_a) \tag{3.45}$$

If frequency and amplitude do not change between the two frames then $q_{k,i} = 1$ is obtained. On the other hand, if the maximum permitted frequency ratio $r_{f,\max} = 1.05$ or amplitude ratio $r_{a,\max} = 4$ is exceeded then $q_{k,i} = 0$ is obtained. These maximum ratios were found appropriate for a typical frame length of $T_\mathrm{f} = 32$ ms.

Using this quality measure $q_{k,i}$, the best matching predecessor $k_{i,\mathrm{opt}}$ for the sinusoid i in the current frame can be found as the best match

$$k_{i,\mathrm{opt}} = \operatorname*{argmax}_k q_{k,i}. \tag{3.46}$$

Of course, only predecessors k can be considered here that were not already chosen as predecessors of other sinusoids in the current frame and thus already assigned to a different trajectory. If no predecessor can be found, i.e., $\max_k q_{k,i} = 0$, the sinusoid i starts a new trajectory in the current frame (birth). Correspondingly all sinusoids in the previous frame that are not selected as predecessor have reached the end of their trajectory (death).

The quality measure $q_{k,i}$ can be extended to also take into account sweeping components with chirp rate α and an optional temporal amplitude envelope $a_\mathrm{e}(t)$. Because

of the problems shown in Figure 3.3, the results of the parameter matching approach to trajectory building are not reliable in case of high sweep rates and especially for higher partials of a harmonic tone.

3.2.2.2 Phase-Locked Tracking of Sinusoidal Components

More reliable results can be obtained if the signal of a sinusoidal component is actually tracked between consecutive segments. For this purpose, the heterodyne-based frequency and chirp rate estimator developed in Section 3.2.1.3 is extended to take into account the frequency and phase parameters in the previous segment in order to provide phase-locked tracking of a supposed sinusoidal trajectory.

In the following, the kth sinusoid of the previous frame is used as a tentative predecessor to build a sinusoidal trajectory which, in the current frame, has approximately the initially estimated frequency $\hat{f}_0'$ that is given as guidance to the parameter estimation described here. The difference between predecessor's frequency $f_k^{(-1)}$ and the initial estimate $\hat{f}_0'$ can be used to derive an initial estimate of the chirp rate

$$\alpha_{\mathrm{c},0} = \pi \frac{\hat{f}_0' - f_k^{(-1)}}{T_\mathrm{f}}. \tag{3.47}$$

A correspondingly sweeping local oscillator signal $c(t)$ permits to extract a slanted section of the time-frequency plane for the accurate frequency and sweep rate estimation as shown as (d) in Figure 3.3. The local oscillator signal is given by

$$c(t) = e^{-j(\varphi_{\mathrm{c},0} + \Omega_{\mathrm{c},0}(t+T_\mathrm{f}) + \alpha_{\mathrm{c},0}(t+T_\mathrm{f})^2)} \tag{3.48}$$

where $t = 0$ refers to the center of the current frame and $t = -T_\mathrm{f}$ to the center for the previous frame. Start phase and start frequency of $c(t)$ at $t = -T_\mathrm{f}$ are set to the parameters of the predecessor, that is, $\varphi_{\mathrm{c},0} = \varphi_k^{(-1)}$ and $\Omega_{\mathrm{c},0} = 2\pi f_k^{(-1)}$. Using a cubic model for the phase trajectory

$$\hat{\varphi}_\mathrm{b}(t) = \varphi_{\mathrm{b},0}^{(-1)} + \Omega_{\mathrm{b},0}^{(-1)}(t+T_\mathrm{f}) + \hat{\alpha}_{\mathrm{b},0}(t+T_\mathrm{f})^2 + \hat{\beta}_{\mathrm{b},0}(t+T_\mathrm{f})^3 \tag{3.49}$$

of the baseband signal $x_\mathrm{b}(t)$, this means that both the start phase and start frequency are known to be zero, that is, $\varphi_{\mathrm{b},0}^{(-1)} = 0$ and $\Omega_{\mathrm{b},0}^{(-1)} = 0$, respectively.

The remaining two free parameters $\hat{\alpha}_{\mathrm{b},0}$ and $\hat{\beta}_{\mathrm{b},0}$ can now be estimated by regression such that the best match with unwrapped phase data $\varphi_\mathrm{b}(nT_\mathrm{d})$ is achieved in a minimum MSE sense. Similar to Equations (3.31) and (3.36), these optimal parameters are given by the linear equation system

$$\begin{bmatrix} \sum_N n^6 & \sum_N n^5 \\ \sum_N n^5 & \sum_N n^4 \end{bmatrix} \begin{bmatrix} \hat{\beta}_{\mathrm{b},0} T_\mathrm{d}^3 \\ \hat{\alpha}_{\mathrm{b},0} T_\mathrm{d}^2 \end{bmatrix} = \begin{bmatrix} \sum_N \varphi[n] n^3 \\ \sum_N \varphi[n] n^2 \end{bmatrix}. \tag{3.50}$$

In order to ensure reliable phase tracking from the predecessor, a set of N data points $\varphi[n] = \varphi_b(nT_d - T_f)$ sampled at $t = nT_d - T_f$ is used that spans one and a half frames. It starts with $n = 0$ at the center of the previous frame $t = -T_f$ and ends with $n = N-1$ at the end of the current frame $t = T_f/2$. With help of the estimated parameters $\hat{\alpha}_{b,0}$ and $\hat{\beta}_{b,0}$ the phase trajectory of the tracked sinusoidal component can be written as

$$\hat{\varphi}_0(t) = \varphi_{c,0} + \Omega_{c,0}(t+T_f) + (\alpha_{c,0} + \hat{\alpha}_{b,0})(t+T_f)^2 + \hat{\beta}_{b,0}(t+T_f)^3. \tag{3.51}$$

From this, finally the estimated phase and frequency parameters for the current frame can be calculated as

$$\hat{\varphi}_0^{(0)} = \hat{\varphi}(0) = \varphi_{c,0} + \Omega_{c,0}T_f + (\alpha_{c,0} + \hat{\alpha}_{b,0})T_f^2 + \hat{\beta}_{b,0}T_f^3, \tag{3.52}$$

$$\hat{f}_0^{(0)} = \frac{1}{2\pi}\left.\frac{d\hat{\varphi}(t)}{dt}\right|_{t=0} = \frac{1}{2\pi}\left(\Omega_{c,0} + 2(\alpha_{c,0} + \hat{\alpha}_{b,0})T_f + 3\hat{\beta}_{b,0}T_f^2\right). \tag{3.53}$$

In general, there is more than one candidate for the predecessor k available. In order to find the best predecessor k_{opt} for a given initial estimate $\hat{f}_0'$, all possible candidates are tried as tentative predecessor and for each of them, the tracking estimation described above is carried out. To assess the suitability of a tentative predecessors, the amplitude estimation, Equation (3.39), is performed using the phase trajectory $\hat{\varphi}(t)$, Equation (3.51). This allows to calculate the energy of the windowed residual $r(t)$, Equation (3.40), for all candidates. Now the candidate achieving minimum residual energy can be chosen as the best predecessor

$$k_{opt} = \operatorname*{argmin}_k \int r^2(t)\,dt \tag{3.54}$$

$$= \operatorname*{argmax}_k \left(\hat{c}_0 \int w_a^2(t)a_e(t)x(t)\cos(\hat{\varphi}(t))\,dt + \hat{s}_0 \int w_a^2(t)a_e(t)x(t)\sin(\hat{\varphi}(t))\,dt\right). \tag{3.55}$$

3.2.2.3 Guided Frequency Estimation Examples

Figures 3.4 and 3.5 illustrate the heterodyne-based guided frequency estimation and tracking for two different scenarios. In both cases, a synthetic real-valued test signal comprising a single sinusoidal trajectory with constant chirp rate and amplitude $a_r = 10^{-10/20}$ in white noise with $\sigma_z^2 = 1$ was used for the simulations. This corresponds to an SNR $a_r^2/2\sigma_z^2$ of -13 dB, i.e., a weak signal that is just about strong enough to allow for successful parameter estimation in these scenarios. Typical values were chosen for the sampling rate $f_s = 16$ kHz and the frame length $T_f = 32$ ms. The window functions for lowpass filtering and the sampling of the baseband signal given below are the same as used in the final encoder that will be described in Section 3.4.3.1.

The first scenario, shown in Figure 3.4, resembles case (a) in Figure 3.3 where the sinusoidal trajectory sweeps from 93 Hz to 107 Hz over the course of the current frame,

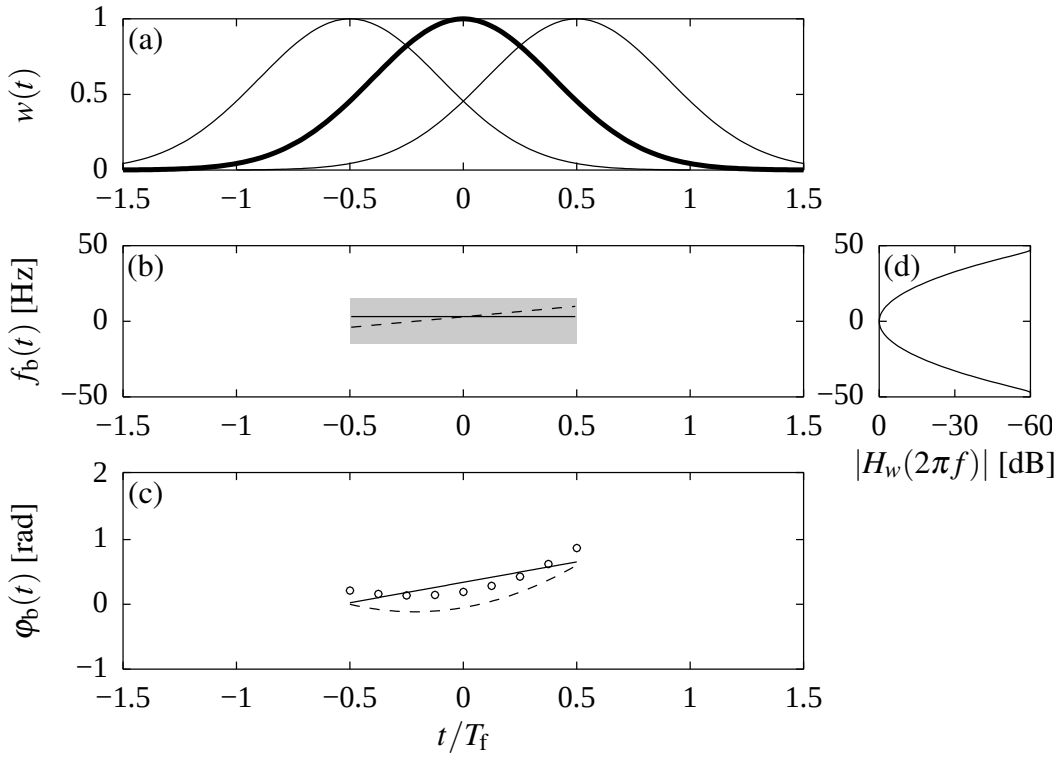

Figure 3.4: Trajectories of (b) frequency $f_\mathrm{b}(t) = f(t) - \hat{f}_0'$ and (c) phase $\varphi_\mathrm{b}(t)$ for baseband signal in heterodyne-based estimation of constant frequency (solid line) for simulation with synthetic signal (dashed line) at -13 dB SNR sweeping from 93 to 107 Hz in current frame corresponding to case (a) in Figure 3.3, assuming initial estimate $\hat{f}_0' = 97$ Hz and using $N = 9$ data points of unwrapped phase (circles) for linear regression of phase (solid line), using (a) Gaussian window $w(t)$ as (d) lowpass filter.

continuing with the same chirp rate outside of the frame boundaries. In this simulation, an initial estimate of $\hat{f}_0' = 97$ Hz is assumed, i.e., an estimate that is 3 Hz below the correct value for center of the frame $t = 0$. Linear regression of phase data, Equation (3.34), is applied to estimate a constant frequency, neglecting the sweeping character of the actual signal. In order to achieve high frequency selectivity, a narrow lowpass filter $w(t)$ with a -6 dB bandwidth of $\Delta f = 32$ Hz is used. Specifically, a Gaussian window $w(t) = e^{-\pi(t/T_\mathrm{f})^2}$ shown in panel (a) is employed here to avoid spectral sidelobes at the cost of a longer window length with t/T_f ranging from $-3/2$ to $3/2$. The main lobe of this Gaussian window closely resembles that of a Hann window $w_\mathrm{h}(t)$, Equation (3.6), and the transfer function $|H_w(\Omega)|$ is shown in panel (d). The baseband signal is sampled with rate $1/T_\mathrm{d} = 250$ Hz which gives a total of $N = 9$ data points for the regression covering the current frame. Panel (a) also indicates the shifted windows (thin lines) used for the first and last data points. Panel (b) shows the baseband frequency trajectory of the original signal (dashed line) and the estimated trajectory (solid line). The gray area approximately

indicates the section of the time-frequency plane with $\Delta t = 32$ ms and $\Delta f = 32$ Hz that is evaluated here. Panel (c) shows the phase trajectory of the original signal (dashed line) and the estimated trajectory (solid line). The circles indicate the $N = 9$ points of unwrapped phase data $\varphi_b(nT_d)$ used by the linear regression. The frequency estimated in this simulation is $\hat{f}_0 = 97$ Hz $+ 3.139$ Hz $= 100.139$ Hz, which is very close to the correct value in the center of the frame of $f_0 = 100$ Hz. It should be noted that this estimated frequency would be different in subsequent simulations due to the random nature of the white noise signal added to the sinusoidal trajectory. In case of a high SNR of 57 dB the estimated frequency would be $\hat{f}_0 = 97$ Hz $+ 2.499$ Hz $= 99.499$ Hz, which indicates that there is a slight bias (in this case approximately 0.5 Hz) towards the center frequency of the analyzed section, caused by the error in the initial estimate in combination with the narrow bandwidth of the window $w(t)$ and the sweeping nature of the original signal.

The second scenario, shown in Figure 3.5, resembles case (d) in Figure 3.3 where the sinusoidal trajectory sweeps from 840 Hz to 960 Hz over the course of the current frame, continuing with the same chirp rate outside of the frame boundaries. In this simulation, an initial estimate of $\hat{f}_0' = 880$ Hz is assumed, i.e., an estimate that is 20 Hz below the correct value for center of the frame $t = 0$. Tracking regression of phase data, Equation (3.49), is applied, assuming error-free data for the start phase (0 rad) and start frequency (780 Hz) from the tentative predecessor for the center of the previous frame $t = -T_f$. In order to achieve appropriate frequency selectivity, a lowpass filter $w(t)$ with a -6 dB bandwidth of $\Delta f = 64$ Hz is used. Specifically, a Hann window $w(t) = \cos^2(\frac{\pi}{4} t/T_f)$ is employed here with a window length of T_f, i.e., with t/T_f ranging from $-1/2$ to $1/2$. The baseband signal is sampled with rate $1/T_d = 500$ Hz which gives a total of $N = 25$ data points ranging from the center of the previous frame to the end of the current frame. In addition to the window function $w(t)$, panel (a) indicates the shifted windows (thin lines) used for the 1st, 9th, and 25th data point. Panel (b) shows, relative to the initial estimate $\hat{f}_0'$, the frequency trajectory of the original signal (dashed line) and the estimated trajectory (solid line). The gray area now indicates a slanted section with $\Delta t = 48$ ms and $\Delta f = 64$ Hz. As in Figure 3.4, panel (c) shows the unwrapped phase trajectories. The frequency estimated in this simulation is $\hat{f}_0 = 880$ Hz $+ 23.651$ Hz $= 903.651$ Hz, which is quite close to the correct value in the center of the frame of $f_0 = 900$ Hz.

3.2.2.4 Trajectory-Building Examples

To compare the performance of the two trajectory-building techniques discussed in Sections 3.2.2.1 and 3.2.2.2, a synthetic test signal with $I = 5$ sinusoids having a fixed amplitude $a_i(t) = 1$ and constant or varying frequencies $f_i(t)$ was used. The signal has a duration of 1 s and is repeated once, now with additional white noise with $\sigma_z^2 = 1/2$, corresponding to a SNR of 0 dB when an individual trajectory is considered. The signal was sampled at $f_s = 16$ kHz and a frame length of $T_f = 32$ ms was used. Figure 3.6 shows the sinusoidal trajectories estimated by the two different techniques. For panel (a), guided frequency and chirp rate estimation using the quadratic phase model, Equa-

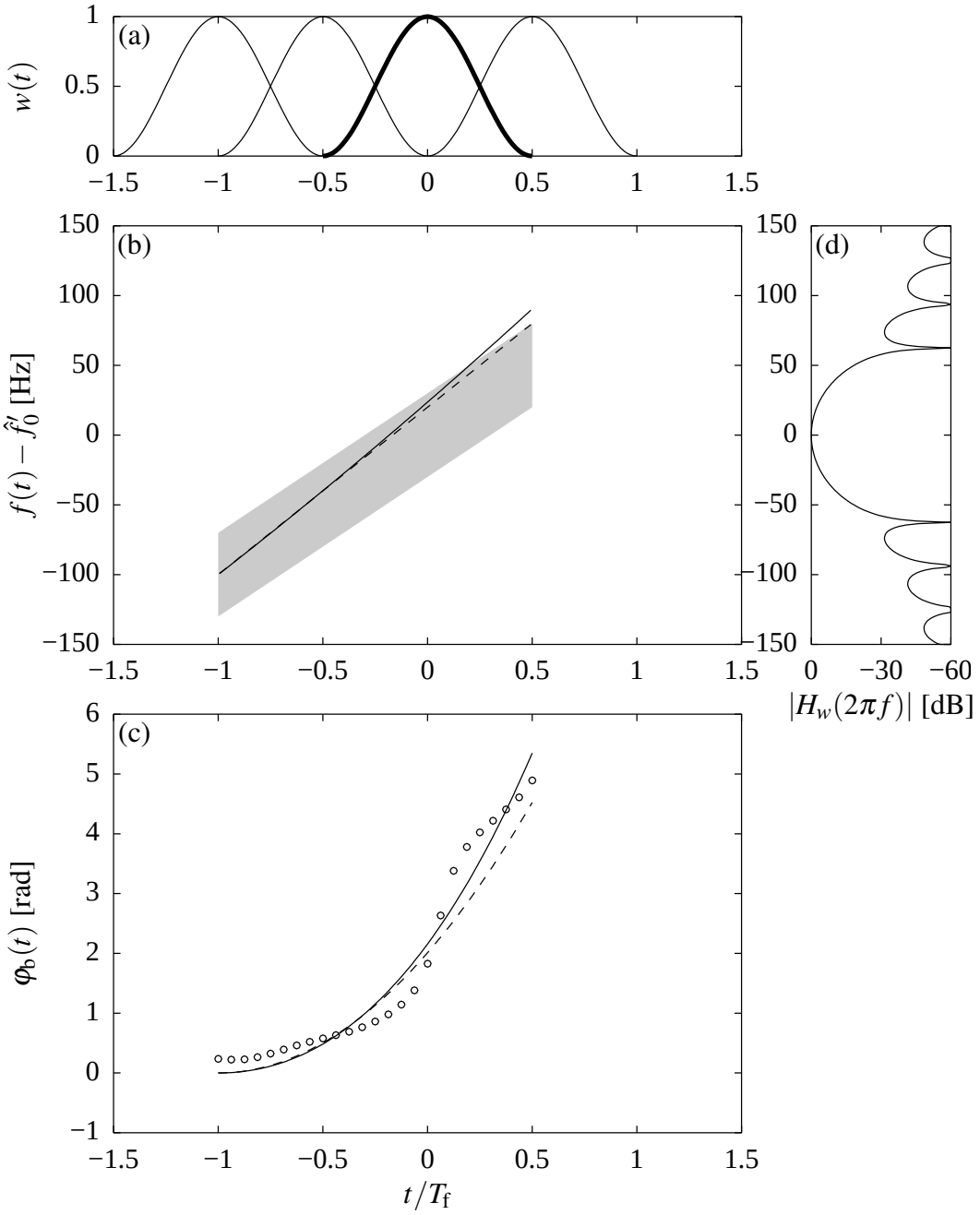

Figure 3.5: Trajectories of (b) frequency $f(t) - \hat{f}_0'$ and (c) phase $\varphi_b(t)$ for baseband signal in heterodyne-based phase tracking estimation (solid line) for simulation with synthetic signal (dashed line) at -13 dB SNR sweeping from 840 to 960 Hz in current frame corresponding to case (d) in Figure 3.3, assuming initial estimate $\hat{f}_0' = 880$ Hz and using $N = 25$ data points of unwrapped phase (circles) for linear regression of phase (solid line), using (a) Hann window $w(t)$ as (d) lowpass filter.

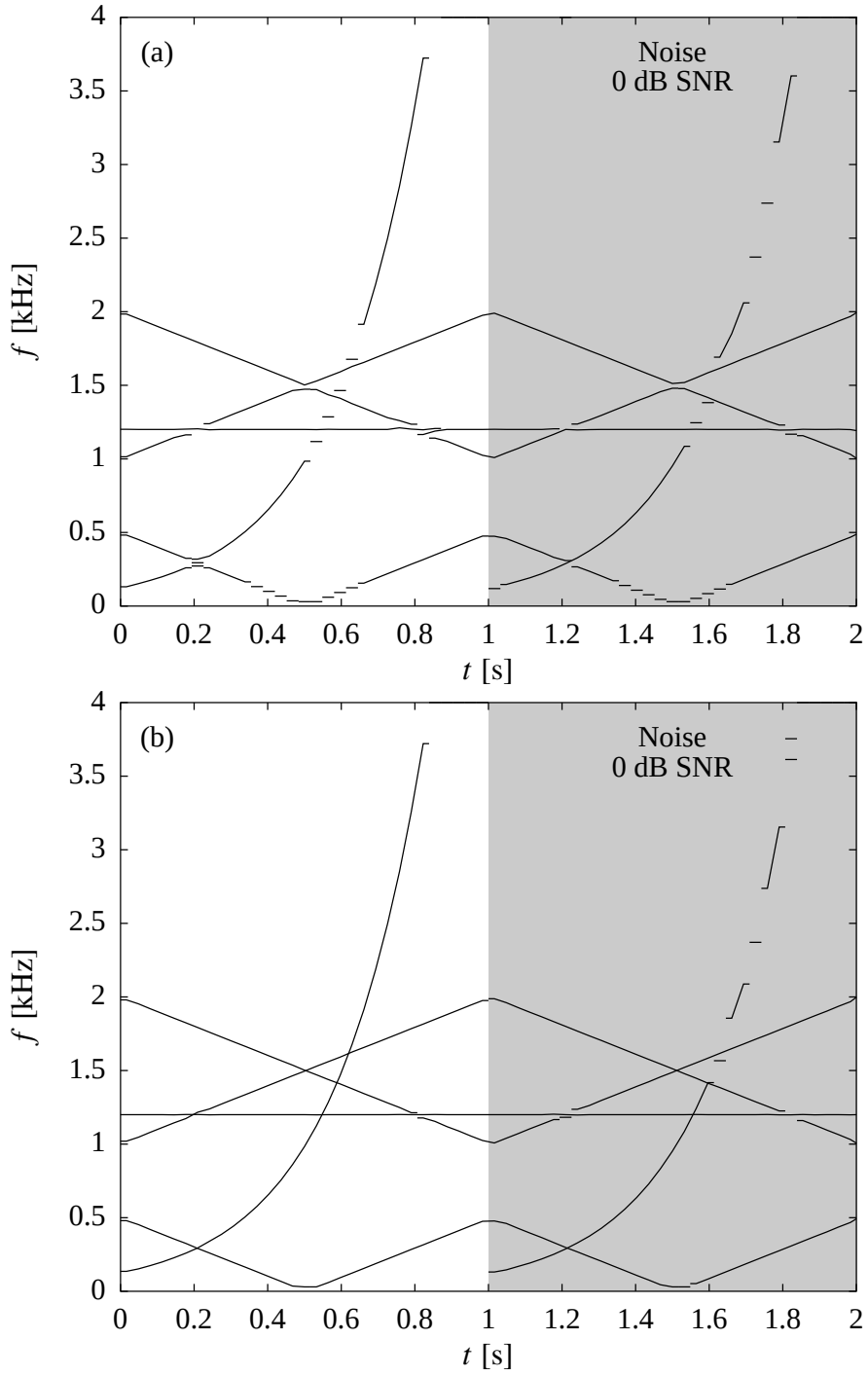

Figure 3.6: Sinusoidal trajectories found by parameter matching (a) and phase-locked tracking (b) for a synthetic signal comprising $I = 5$ sinusoids with $a_i(t) = 1$, using $f_s = 16$ kHz and $T_f = 32$ ms. The signal is repeated once after $t = 1$ s, now with additional white noise with $\sigma_z^2 = 1/2$ (i.e., 0 dB SNR) as indicated by the gray shading (after [110]).

tion (3.35), was employed in combination with trajectory-building by parameter matching, Equation (3.46), extended to consider also the chirp rate. For panel (b), phase-locked tracking of sinusoidal components, Equations (3.49) and (3.54), was employed.

The phase-locked tracking technique exhibits only one tracking error in the left half of Figure 3.6, at 0.8 s, 1.2 kHz. In contrast, the parameter matching technique fails for all crossing trajectories and has problems for low frequencies due to the maximum permitted frequency ratio $r_{f,\max}$, as can be seen between 0.2 s and 0.4 s around 100 Hz. In case of additional noise, shown in the right half of Figure 3.6, the performance of phase-locked tracking is reduced and some tracking errors can be seen, especially for high chirp rates. Nevertheless it still performs significantly better than parameter matching in the presence of noise. In this simulation, both techniques were operated in a complete signal decomposition framework, which is necessary to supervise the parameter estimation for signals where several sinusoids are present simultaneously. The decomposition framework will be described in detail in Section 3.3. In this simulation, a simple decomposition framework without any psychoacoustic model was employed (i.e., the component selection strategy SNR was used, which will be described in detail in Section 3.3.2.1).

3.2.2.5 Parameter Estimation Accuracy Experiments

In order to assess the parameter estimation error of the phase-locked tracking of sinusoids described above in combination with the windowed amplitude estimation, Equation (3.39), and to compare it with the CR bounds, a synthetic test signal was used. It comprises a single sinusoidal component with constant frequency f_r and amplitude $a_\mathrm{r} = 1$ and added white noise at different levels σ_z^2 corresponding to an SNR of -10 dB, 0 dB, 10 dB, and 20 dB. The typical frame length $T_\mathrm{f} = 32$ ms at $f_\mathrm{s} = 16$ kHz was used and a Hann window, Equation (3.6), with 50% overlap was employed in amplitude estimation, i.e., $w(t) = w_\mathrm{h}(t)$.

Table 3.2 shows the standard deviations measured in this experiment, calculated from four sets of $K = 300$ estimated parameters from consecutive frames (corresponding to 9.6 s) for four different frequencies $f_\mathrm{r} = 62.5$ Hz, 562.5 Hz, 575 Hz, and 5000 Hz. It should be noted that values measured for the standard deviation in these experiments themselves have an uncertainty of approximately $1/\sqrt{4K}$ (i.e., $\approx 3\%$). In addition, the CR bounds for frequency, amplitude, and center-referenced phase are given. The CR bound for frequency was calculated for a segment length of 1.5 frames ($N = 768$) as used by the phase-locked tracking estimation. For amplitude and phase, the effective length of the Hann window of one frame ($N = 512$) was used to calculate the CR bounds. It can be seen that measured standard deviation of the estimated parameters corresponds well to CR bounds, confirming the efficiency of the phase-locked tracking estimation developed here. The observation that the measured frequency error is slightly below the CR bound can be explained by the fact that the examined signal segment is actually somewhat longer than the 1.5 frames assumed here due to the size of the Hann window $w(t)$ used to calculate the phase data samples (see panel (a) of Figure 3.5). The slightly higher measured phase

SNR	**−10 dB**	**0 dB**	**10 dB**	**20 dB**
σ_z^2	5	0.5	0.05	0.005
Frequency error $\sqrt{\mathrm{var}(\hat{f}_\mathrm{r} - f_\mathrm{r})}$ [Hz]				
CR: $N = 768$	1.3105 (100%)	0.4145 (100%)	0.1311 (100%)	0.0414 (100%)
Measured	1.2308 (94%)	0.3581 (86%)	0.1093 (83%)	0.0334 (81%)
Amplitude error $\sqrt{\mathrm{var}(\hat{a}_\mathrm{r} - a_\mathrm{r})}$				
CR: $N = 512$	0.1397 (100%)	0.0442 (100%)	0.0140 (100%)	0.0044 (100%)
Measured	0.1369 (98%)	0.0434 (98%)	0.0137 (98%)	0.0044 (101%)
Phase error $\sqrt{\mathrm{var}(\hat{\varphi}_\mathrm{r,c} - \varphi_\mathrm{r,c})}$ [rad]				
CR: $N = 512$	0.1397 (100%)	0.0442 (100%)	0.0140 (100%)	0.0044 (100%)
Measured	0.1757 (126%)	0.0552 (125%)	0.0187 (134%)	0.0068 (153%)

Table 3.2: CR bounds (100% reference) for segment length N and measured parameter estimation errors, given as standard deviation $\sqrt{\mathrm{var}(\cdot)}$, for phase-locked tracking of sinusoids with constant frequency f_r and amplitude $a_\mathrm{r} = 1$ and added white noise at different levels σ_z^2 calculated from four sets of $K = 300$ measurements with $T_\mathrm{f} = 32$ ms at $f_\mathrm{s} = 16$ kHz (i.e., 9.6 s). Note that the measured parameter estimation errors themselves have an uncertainty of approximately $1/\sqrt{4K}$ (i.e., $\approx 3\%$), and see the text for a discussion of their relation to the CR bounds.

error, on the other hand, can be explained by the constraints in Equation (3.49) that ensure a phase trajectory smoothly continuing from the estimated predecessor parameters.

3.2.3 Estimation of Harmonic Tone Parameters

The most important parameter characterizing a harmonic tone component is its fundamental frequency. Reliable estimation of this parameter is essential because detecting an erroneous fundamental frequency and forcing sinusoidal signal components onto the corresponding incorrect harmonic grid can result in very annoying artifacts. There are two major approaches to detect the presence of a harmonic tone and to estimate its fundamental frequency. The first approach takes the original signal $x(t)$ as input and looks for periodicities in the magnitude spectrum of this signal. The second approach takes the parameters of all individual sinusoidal trajectories as input and looks for a pattern in this data that corresponds to the frequencies of the partials of a harmonic tone. In addition to the fundamental frequency also the stretching parameter has to be estimated. In order to enable analysis-by-synthesis based decomposition, the exact frequency, amplitude, and phase parameters of all partials are required as well. Finally, if a harmonic tone was found, the parameters of the spectral envelope model have to be calculated from the amplitudes of the partials.

3.2.3.1 Signal-Based Fundamental Frequency Estimation

The general idea of this approach to estimate the fundamental frequency f_h is to look for a periodic pattern in the magnitude spectrum of the signal $x[n]$, assuming that it is caused by a series of spectral peaks representing the partial tones at $f_{i,\mathrm{h}} = if_\mathrm{h}$. The period of this pattern, i.e., the distance between to partial tones, is then an estimate for the fundamental frequency. To detect such a periodic pattern and estimate its period length, usually a Fourier transform is applied to a representation of the magnitude spectrum.

A common approach is to utilize the power spectrum (i.e., squared magnitude) $S_{xx}(\omega)$, since its inverse Fourier transform is the autocorrelation function $r_{xx}[k]$ of the signal. The location k_h of the first peak in $r_{xx}[k]$ next to the maximum at $k = 0$ indicates the periodicity of a harmonic spectrum and is referred to as pitch lag, i.e., the duration $k_\mathrm{h} = f_\mathrm{s}/f_\mathrm{h}$ of one period of the harmonic signal $x[n]$ sampled at f_s.

An alternative approach makes use of the log spectrum, which results in a cepstrum analysis of the signal [8], [91]. The real cepstrum can be written as $\mathscr{F}^{-1}\{\ln|\mathscr{F}\{x[n]\}(\omega)|\}[k]$, where k is referred to as quefrency.

For the initial experiments on parameter estimation for harmonic tones in the course of this work, a cepstrum-based algorithm for fundamental frequency estimation was developed. It takes the current signal segment as input, typically using Hann windows with 50% overlap and a 32 ms stride. Zero-padding is applied to double the length of the log magnitude spectrum prior to an inverse DFT, and thus yields a smoother cepstrum that allows for easier and more precise detection of peaks. A search range for the pitch lag corresponding to a fundamental frequency range of 30 to 1000 Hz was found to be appropriate for this application. A typical problem encountered with this approach using the location of the largest peak in the cepstrum within the search range are octave errors, i.e., that the estimated pitch lag is e.g. a factor of 2 too high. This issue was addressed by correcting and refining the pitch lag estimate based on further peaks identified in the cepstrum. Using the resulting fundamental frequency estimate, the exact parameters of all partials can be estimated individually by means of the guided frequency estimation presented in Section 3.2.1.3 using multiples of the fundamental frequency as initial estimate. A fairly reliable initial estimate for the chirp rate of the fundamental and the higher partials is derived by comparing the fundamental frequency estimated in adjacent frames. Based on the partials' estimated frequencies, the final estimates for fundamental frequency and stretching according to Equation (3.7) can be found by means of regression minimizing the mean squared frequency error.

The cepstrum based fundamental frequency estimation developed here has show to work fairly reliably for a large class of signals. However, complex signals with a comparably dense tonal spectrum, e.g. the sound of a full orchestra, pose problems to this approach. In such situations, it can happen that a fairly low fundamental frequency is detected, typically in the range of 30 to 100 Hz, as there is a high likelihood that a dense tonal spectrum has numerous peaks that can be taken as partials for the erroneously detected fundamental frequency. Similar problems can occur if multiple harmonic tones are

present simultaneously, especially if their fundamental frequencies are related by simple ratios, as is the case when a chord is played.

3.2.3.2 Building a Harmonic Tone from Sinusoidal Trajectories

Instead of the signal $x[n]$ itself, the approach presented now takes the parameters of the complete set of individual sinusoidal trajectories as input, which in turn were estimated by the algorithms described in Section 3.2.1. The general idea is to look for the pattern of the frequencies of the partials of a harmonic tone in this data, similar to, e.g., the approaches proposed in [13] or [59]. As such, this approach is focused on an efficient representation of the available sinusoidal description of the signal rather than on an analysis of the signal itself. This has the advantage that it is much easier to achieve robust detection and estimation of harmonic tones, i.e., minimize the risk of audible artifacts due to erroneous harmonic tone parameters in case of difficult signals. On the other hand, it lacks the capability of estimating the parameters of difficult-to-detect partials that were missed in the initial sinusoidal modeling.

The problem at hand is to find a subset of the set of estimated sinusoidal trajectories that can be considered as the series of partials of a harmonic tone with a corresponding fundamental frequency f_h. For this purpose, a quality measure $q_i(\Delta f_i)$ is introduced to assess the difference Δf_i between the expected frequency $f_{i,\mathrm{h}}$ of the ith partial according to Equation (3.7) and the frequency parameter f_k of the nearest sinusoidal trajectory in the current frame. This quality measure has a simple triangular shape $q_i(\Delta f_i) = \max(0, 1 - |\Delta f_i / \Delta f_\mathrm{max}|)$ with a width of $\Delta f_\mathrm{max} = \min(0.1 f_\mathrm{h}, 10\ \mathrm{Hz})$, where the 10 Hz limit is reflecting the largest expected uncertainty of the sinusoidal frequency estimation. It is combined with an additional weight factor $w_i = 0.9^{k-1}$ that is related to the perceptual relevance of the kth sinusoidal trajectory considered as the ith partial here. This assumes that the trajectories are sorted in order of decreasing perceptual relevance, with the first trajectory $k = 1$ being the most important one, as will be discussed in Section 3.3.2. In case a trajectory is continued from the previous frame, the weight w_i is multiplied by 1.5 to put emphasis on sustained trajectories.

Based on the quality measure and the additional weight factor, the weighted total power q_h of the considered harmonic tone is calculated by accumulation over all partials

$$q_\mathrm{h}(f_\mathrm{h}, \kappa_\mathrm{h}) = \frac{1}{2} \sum_i a_k^2 w_i q_i(\Delta f_i) \tag{3.56}$$

Given the set of sinusoidal trajectories in the current frame, q_h is a function of the considered fundamental frequency f_h and stretching parameter κ_h. The dependency on the stretching parameter (see Equation (3.7)) becomes important for harmonic tones with a high number of partials. To determine the parameter pair $f_\mathrm{h}, \kappa_\mathrm{h}$ describing the dominating harmonic tone in the complete set of sinusoidal trajectories, the weighted total power measure q_h has to be maximized. To avoid the high computational complexity of a full search, a stepwise refined search approach is employed.

In the first step, a coarse search is performed over the full range of permissible fundamental frequencies, typically from 30 Hz to 1 kHz. Approximately 350 values for f_h are probed, spaced logarithmically with an increment ratio of 1.01. This corresponds to an increment of approximately 17 cent, where 100 cent denote the semitone interval $1{:}\sqrt[12]{2}$. No stretching is assumed, at most the first 25 partials i are considered, and the width Δf_max is increased by a factor of 5 to compensate for the sparse set of f_h probed in this coarse search. In the second step, the search is refined in the neighborhood of the maximum found in the first step. Approximately 200 values for f_h in the range from 0.9 to 1.1 relative to the maximum found in the previous step are probed with an increment ratio of 1.001 in combination with 11 values for κ_h covering the range ± 0.0002. This corresponds to a range of 3.5 semitones and an increment of 1.7 cent. The third and last step refines the search further and probes 100 values for f_h in the range from 0.99 to 1.01 relative to the maximum found in the previous step with an increment ratio of 1.0002 (0.35 cent increment) in combination with 25 values for κ_h in the ± 0.0002 range.

Once the best parameter pair f_h, κ_h is determined, the subset of sinusoidal trajectories that constitute the corresponding partial tones is collected. For each sinusoidal trajectory k, the index i of the partial tone closest to the frequency f_k is calculated according to Equation (3.7)

$$i = \mathrm{round}\left(\left(\sqrt{\kappa_\mathrm{h}\frac{f_k}{f_\mathrm{h}} + \frac{1}{4}} - \frac{1}{2}\right) / \kappa_\mathrm{h}\right), \tag{3.57}$$

where round() returns the nearest integer. For very small values of κ_h, this equation becomes ill-conditioned and the solution $i = \mathrm{round}(f_k/f_\mathrm{h})$ for $\kappa_\mathrm{h} = 0$ is used instead. If the frequency difference is within the window given above, i.e., $|f_k - f_{i,\mathrm{h}}| \leq \Delta f_\mathrm{max}$, the trajectory k is considered as the ith partial and flagged accordingly. Note that the window width Δf_max is doubled if the trajectory k is continued from the previous frame in order to put emphasis on sustained trajectories. After all K trajectories have been processed in this way, for each partial i either the corresponding trajectory k is known or, if no matching trajectory was found, it is marked as void. The index i of the highest non-void partial determines the total number of partials I_h of the harmonic tone. All trajectories not flagged as being a partial now form the remaining subset of individual sinusoidal trajectories.

Finally, the following additional rules are applied to ensure that a harmonic tone fulfills certain minimum requirements in order to improve reliability of the harmonic tone detection and estimation. The total power of the partials of the harmonic tone compared to the total power of all sinusoidal trajectories must be above a given threshold, typically in the range of 0.7 to 0.8, indicating that the harmonic tone is an important signal component. The power calculation is modified to include the weight factor $w_i = 0.9^{k-1}$ explained above in order to put emphasis on the perceptually more relevant sinusoidal trajectories. Furthermore, at least 6 partial tones must have been found in the set of sinusoidal trajectories. Lastly, no more than 2 adjacent partials are allowed to be void. In

order to simplify the calculation of the parameters of the spectral envelope model (see Section 3.2.3.3), non-zero "dummy" amplitudes are calculated for void partials that are not associated with a sinusoidal trajectory. The "dummy" amplitude of partial i is set to a value 20 dB below the average amplitude of the two lower and two higher partials, i.e., the four partials $i-2$, $i-1$, $i+1$, and $i+2$.

The various heuristic rules and thresholds included in the harmonic tone detection and estimation algorithm describe here were derived by extensive experimentation in the context of the complete coding system operated at typical target bit rates in the range of 6 to 16 kbit/s. It is possible that a sinusoidal trajectory is considered as a partial of a harmonic tone in one frame and as an individual sinusoid in an adjacent frame. These situations are taken care of prior to signal synthesis, as will described in detail in Section 4.2.1.

3.2.3.3 Spectral Envelope Model Parameter Calculation

While the frequencies $f_{i,\mathrm{h}}$ of the partials of a harmonic tone are sufficiently well described by fundamental frequency f_{h} and the stretching parameter κ_{h}, it is now of interest to represent the amplitudes $a_{i,\mathrm{h}}$ of the partials by means of a spectral envelope model. Given the amplitudes $a_{i,\mathrm{h}}$ of the partials $i = 1,\ldots,I_{\mathrm{h}}$, the problem at hand is now to find the parameters of the all-pole spectral envelope model as defined by Equations (3.9) and (3.10). The amplitude parameter a_{h} of the harmonic tone can be calculated easily as specified by Equation (3.8). The choice of an appropriate filter order P_{h} and the calculation of the filter coefficients a_p is however more difficult. For filter order $P_{\mathrm{h}} = I_{\mathrm{h}} - 1$ an exact representation of the amplitudes $a_{i,\mathrm{h}}$ can be possible, but the equation system is usually ill-conditioned, as is also the case for higher filter orders. For lower filter orders $P_{\mathrm{h}} < I_{\mathrm{h}} - 1$ usually only an approximation of the exact amplitudes $a_{i,\mathrm{h}}$ is possible, and an error measure is required to find the filter coefficients that results in the best approximation for a given filter order. A detailed discussion of this problem, known as discrete all-pole modeling (DAP), can be found in the literature [23].

Here, a simpler approach was chosen that strives for a smooth approximation of the spectral envelope of the harmonic tone and thereby also allows for easy dynamic adaption of the filter order using an LAR representation of the filter coefficients, as outlined in Section 3.1.2. It makes use of the autocorrelation function (ACF) $r_{xx}[k]$ of the harmonic tone. The autocorrelation function of a time-discrete stationary signal $x[n]$ is defined as

$$r_{xx}[k] = \mathrm{E}\{x^*[n]x[n+k]\}, \tag{3.58}$$

where $\mathrm{E}\{x\}$ is the expectation operator. It is directly related to the power spectrum (or PSD) $S_{xx}(\omega)$ of this signal by means of the Fourier transform

$$S_{xx}(\omega) = \mathscr{F}\{r_{xx}[n]\}(\omega) = \sum_{n=-\infty}^{\infty} r_{xx}[n]e^{-j\omega n}. \tag{3.59}$$

Using the spectral envelope model sampling rate $f_{\mathrm{s,SEM}} = 2(I_\mathrm{h}+1)f_\mathrm{h}$, the time-discrete signal of the harmonic tone can be written as

$$x[n] = \sum_{i=1}^{I_\mathrm{h}} a_{i,\mathrm{h}} \cos\left(\pi \frac{in}{I_\mathrm{h}+1}\right), \tag{3.60}$$

where the phase parameters $\varphi_{i,\mathrm{h}}$ are ignored as they are irrelevant for the spectral envelope, and where the stretching κ_h is neglected as well. This signal has a period of $2(I_\mathrm{h}+1)$ and its autocorrelation, Equation (3.58), can be written as

$$r_{xx}[k] = \frac{1}{2(I_\mathrm{h}+1)} \sum_{n=1}^{2(I_\mathrm{h}+1)} \left(\sum_{i=1}^{I_\mathrm{h}} a_{i,\mathrm{h}} \cos\left(\pi \frac{in}{I_\mathrm{h}+1}\right) \sum_{l=1}^{I_\mathrm{h}} a_{l,\mathrm{h}} \cos\left(\pi \frac{l(n+k)}{I_\mathrm{h}+1}\right) \right) \tag{3.61}$$

$$= \frac{1}{2} \sum_{i=1}^{I_\mathrm{h}} a_{i,\mathrm{h}}^2 \cos\left(\pi \frac{ik}{I_\mathrm{h}+1}\right). \tag{3.62}$$

Using this autocorrelation function $r_{xx}[k]$ as input, the Levinson-Durbin algorithm [65] can be used to calculate the coefficients of an all-pole spectral envelope model that approximates the shape of the power spectrum $S_{xx}(\omega)$ as good as possible with the desired filter order P_h, i.e., with P_h coefficients. The algorithm can be described as follows

$$E^{(0)} = r_{xx}[0] \tag{3.63}$$

$$k_p = -\left(r_{xx}[p] + \sum_{i=1}^{p-1} a_i^{(p-1)} r_{xx}[p-i] \right) / E^{(p-1)} \tag{3.64}$$

$$a_p^{(p)} = k_p \tag{3.65}$$

$$a_i^{(p)} = a_i^{(p-1)} + k_p a_{p-i}^{(p-1)}, \quad 1 \le i \le p-1 \tag{3.66}$$

$$E^{(p)} = (1-k_p^2) E^{(p-1)} \tag{3.67}$$

where Equations (3.64) to (3.67) are solved recursively for $p = 1, 2, \ldots, P_\mathrm{h}$. The final result is available both as filter coefficients $a_p^{(P_\mathrm{h})}$ and as reflection coefficients k_p.

Using the example of a harmonic tone with $I_\mathrm{h} = 24$ partials estimated at $t = 2.35$ s in signal *Suzanne Vega* (see also Figure 4.18), Figure 3.7 shows the spectral envelope as defined by the reflection coefficients derived by the Levinson-Durbin from $r_{xx}[k]$ according to Equation (3.62) for all-pole filters of order of $P_\mathrm{h} = 9$ and $P_\mathrm{h} = 23$ as dotted and dashed lines, respectively. While the modeling for $P_\mathrm{h} = 9$ is reasonable, though somewhat inaccurate due to the low filter order, it can be seen clearly that modeling becomes problematic when the filter order P_h approaches the number of partials $I_\mathrm{h} = 24$. This can be explained by the fact that $r_{xx}[k]$ according to Equation (3.62) represents the line spectrum of a periodic signal with a period $2(I_\mathrm{h}+1)$ samples, and the all-pole model parameters found by the Levinson-Durbin algorithm try to approximate this line spectrum nature by placing

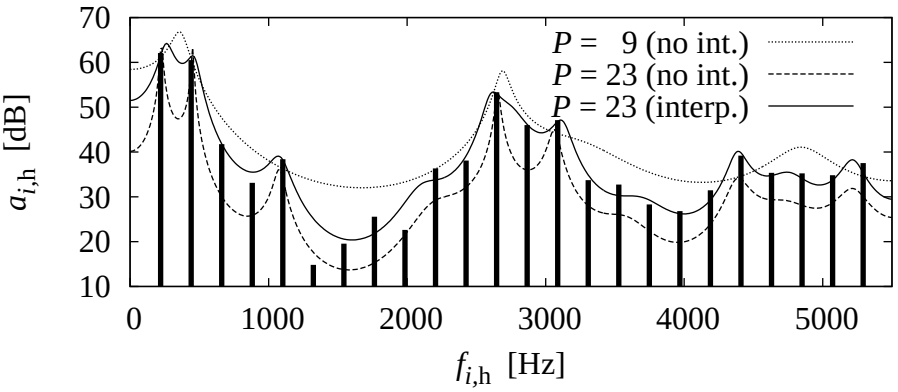

Figure 3.7: Amplitudes of the $I_\mathrm{h} = 24$ partials of a harmonic tone with $f_\mathrm{h} = 220.5$ Hz and $a_\mathrm{h} = 1766$ (64.9 dB) estimated at $t = 2.35$ s in signal *Suzanne Vega* and spectral envelope approximations with all-pole model of order $P_\mathrm{h} = 9$ and $P_\mathrm{h} = 23$ (without interpolation) and $P_\mathrm{h} = 23$ (with interpolation). Amplitudes are given relative to a reference level (0 dB) of $a = 1$ for signals represented as 16 bit PCM, i.e., with a full scale amplitude of $a = 32767$.

resonance peaks at the (dominant) partial tones. If the filter order P_h is low compared to the number of partials I_h, this undesired effect is usually absent.

Hence, to avoid this problem and ensure a smooth modeling of the spectral envelope of a harmonic tone, the number of partials used to calculate $r_{xx}[k]$ according to Equation (3.62) is doubled by inserting additional ("virtual") partial tones with interpolated amplitude values in between the real partial tones [71]. This new series of partial amplitudes $a'_{i',\mathrm{interp}}$ with $1 < i' < I' = 2I_\mathrm{h} + 1$ is defined by

$$a'_{i'} = \begin{cases} a_{(i'/2),\mathrm{h}} & i' = 2,4,\ldots,I'-1 \\ 0 & i' = 1,3,\ldots,I' \end{cases} \tag{3.68}$$

$$a'_{i',\mathrm{interp}} = a'_{i'} * h_{i',\mathrm{interp}}, \tag{3.69}$$

where $h_{i',\mathrm{interp}}$ is the half-band lowpass filter used for interpolation. For the experiments reported here, a 27 tap filter was used, but even the simplest approach, a 3 tap filter implementing linear interpolation, i.e., $h_{-1,\mathrm{interp}} = 0.5$, $h_{0,\mathrm{interp}} = 1$, $h_{1,\mathrm{interp}} = 0.5$, achieves already almost the same result. This interpolation approach strives to make the autocorrelation $r_{xx}[n]$ correspond to a smoothed spectral envelope of the harmonic tone and to conceal the line spectrum nature of the partials itself. The solid line in Figure 3.7 shows the spectral envelope for a filter order of $P_\mathrm{h} = 23$ derived using the interpolation approach described above. It can be seen that modeling accuracy is greatly improved compared to the dashed line derived without interpolation.

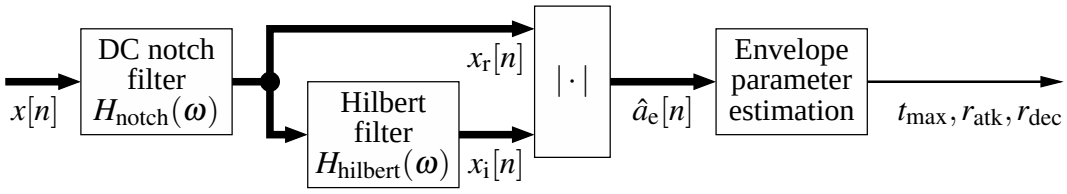

Figure 3.8: Parameter estimation for AD envelope based on analytic signal approximation from Hilbert filter.

3.2.4 Estimation of Transient Component Parameters

The temporal amplitude envelope of a transient signal component can be modeled by the attack/decay envelope as defined in Section 3.1.3, Equation (3.14), and shown in Figure 3.1. The shape of this AD envelope $a_{\text{e}}(t)$ is described by the three parameters t_{max}, r_{atk}, and r_{dec} and allows to model the envelope of short transient impulses as well as the abrupt start or end of a signal component. Estimation of the parameters of the AD envelope is based on a signal waveform $x(t)$ that is assumed to be dominated by the transient component of interest. This can either be the original input signal or an intermediate signal derived during signal decomposition, as will be discussed in Section 3.3.1. The parameter estimation comprises two steps, which are shown in Figure 3.8 and described in more detail below. Firstly, the envelope $\hat{a}_{\text{e}}(t)$ of the signal is calculated and then the three envelope parameters are found by fitting the AD model envelope to $\hat{a}_{\text{e}}(t)$.

This simplest approach to calculate the temporal envelope $\hat{a}_{\text{e}}(t)$ of a signal $x(t)$ would be to take the absolute value $|x(t)|$ and apply smoothing by means of a lowpass filter or peak detector with decay. However, contradicting requirements for the smoothing time constant make this approach difficult, because a fast response time has to be traded against performance for signals with low frequencies. Therefore, a different approach is employed that is based on the magnitude of an analytic signal $x_{\text{a}}(t)$ derived from the real-valued input waveform $x(t)$. Input is a segment of the time-discrete input signal $x[n]$ that covers not just the current frame of length T_{f} but also half of the previous and half of the next frame in order to avoid problems at frame boundaries. The signal segment is first processed by a DC notch (or bandpass) filter $H_{\text{notch}}(\omega)$ to remove signal components with frequencies close to 0 and $f_{\text{s}}/2$. Then, this real-valued signal $x_{\text{r}}[n]$ is processed by a Hilbert filter $H_{\text{hilbert}}(\omega)$ which applies a $\pi/2$ phase shift to derive the corresponding imaginary part $x_{\text{i}}[n]$ of the desired analytic signal $x_{\text{a}}[n] = x_{\text{r}}[n] + jx_{\text{i}}[n]$. Since a finite-length Hilbert filter inherently has a bandpass characteristic, the initial DC notch filter ensures that both the real and the imaginary part have similar spectra. Both the DC notch filter and the Hilbert filter are implemented as symmetric FIR filters, and for $f_{\text{s}} = 16$ kHz a filter length of 65 taps was chosen for both filters as a trade off between the performance for frequencies close to 0 and the required look-ahead (or delay). The magnitude responses of both filters are shown in Figure 3.9.

The normalized magnitude of the analytic signal $\hat{a}_{\text{e}}[n] = |x_{\text{a}}[n]| / \max_n |x_{\text{a}}[n]|$ over time

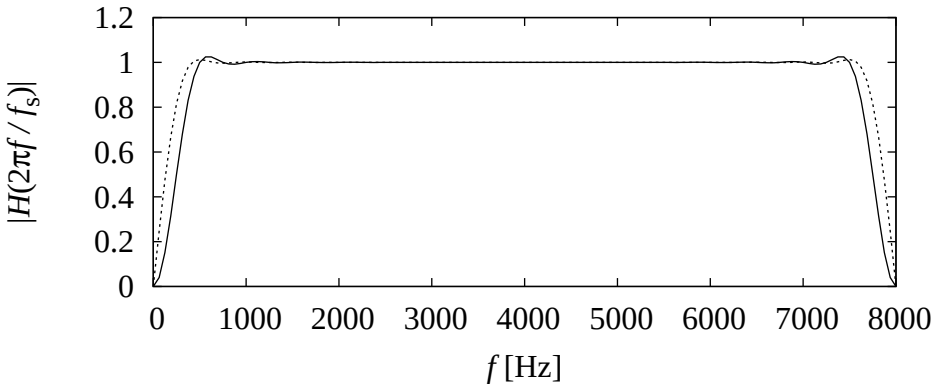

Figure 3.9: Magnitude response of Hilbert filter $H_{\text{hilbert}}(\omega)$ (dotted line) and DC notch filter $H_{\text{notch}}(\omega)$ (solid line) for $f_s = 16$ kHz.

constitutes the input to the second step, where the parameters $t_{\max}$, r_{atk}, and r_{dec} of the triangular AD envelope model are estimated. An example of the estimated envelope $\hat{a}_e(t)$ and the approximated envelope $a_e(t)$ as defined by the estimated envelope parameters is shown in Figure 3.10 for one click in signal *castanets*.

The three envelope parameters are estimated from $\hat{a}_e[n]$ using regression techniques with appropriate weighting depending on amplitude and distance from the maximum. As first parameter, $t_{\max}$ is obtained. It indicates the point in time at which the envelope $\hat{a}_e(t)$ reaches its maximum for the first time within in the current frame. Next the root mean square (RMS) amplitude of $\hat{a}_e(t)$ is calculated for a signal segment that slightly extends beyond the borders of the current frame. Specifically, this segment starts $T_f/16$ prior to the current frame and ends $T_f/16$ after the current frame, and is also used as input to the regression-based estimation of attack and decay rates. The attack rate r_{atk} is determined by linear regression as the slope of that line going through the maximum at $t_{\max}$ that fits best the original envelope prior to the maximum. To improve behavior of this regression, a weighting function

$$w(a,t) = \begin{cases} \left(1 + \frac{4|t-t_{\max}|}{T_F}\right)\left(\frac{a-a_{\text{RMS}}}{1-a_{\text{RMS}}}\right)^{1+\frac{4|t-t_{\max}|}{T_F}}, & a > a_{\text{RMS}} \\ 0, & a \leq a_{\text{RMS}} \end{cases} \tag{3.70}$$

is used, where $a = \hat{a}_e/\hat{a}_e(t_{\max})$ is a normalized amplitude in the range of 0 to 1. It gives no weight to those parts of the envelope that are below the RMS amplitude while the weight continuously increases with increasing amplitude. For large amplitudes, the weight also increases with increasing distance to $t_{\max}$ on the time axis. The decay rate r_{dec} is determined by linear regression as the slope of that line that fits best the original envelope after the maximum. The same weighting function is used as for the attack, but the line is not forced to go trough the maximum at $t_{\max}$. Only the slope of the line is used as parameter r_{dec} and the vertical position of the line is ignored.

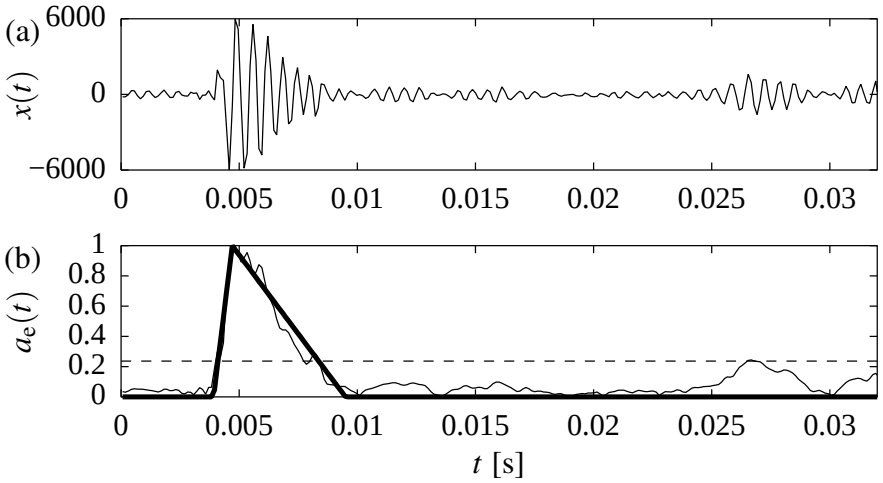

Figure 3.10: Panel (a) shows one frame of signal $x(t)$ containing a transient (one click in signal *castanets*). Panel (b) shows estimated envelope $\hat{a}_e(t)$ (thin line), RMS of $\hat{a}_e(t)$ (dashed line), and approximated AD envelope $a_e(t)$ (bold line) as described by the estimated parameters t_{max}, r_{atk}, and r_{dec} (after [101]).

If the attack as well as the decay rate are below a threshold of $1/T_f$ (which means that there are no fast changes of the signal amplitude within the current frame), the slope of a line that best fits the $\hat{a}_e(t)$ envelope for the whole frame is calculated by weighted linear regression. The weighting function

$$w(a) = \begin{cases} \frac{a-a_{RMS}}{1-a_{RMS}}, & a > a_{RMS} \\ 0, & a \leq a_{RMS} \end{cases} \tag{3.71}$$

now only depends on the amplitude, not on the time difference relative to t_{max}. If the slope is positive, it is used as r_{atk} so that together with $t_{max} = T_f$ and $r_{dec} = 0$ a continuously increasing amplitude is modeled. Correspondingly, if the slope is negative, its absolute value is used as r_{dec} together with $t_{max} = 0$ and $r_{atk} = 0$. A more detailed discussion of the envelope parameter estimation can be found in [18].

Once the envelope parameters t_{max}, r_{atk}, and r_{dec} are estimated, the corresponding temporal amplitude envelope $a_e(t)$ is calculated according to the AD envelope model, Equation (3.14). This amplitude envelope can then be utilized during amplitude estimation for transient signal components, as described by Equation (3.39) in Section 3.2.1.3.2.

The decision whether or not to apply the modeled amplitude envelope $a_e(t)$ to a signal component is usually taken based on the energy of the windowed residual $r(t)$ as defined in Equation (3.40). In the case that using the modeled envelope instead of the default $a_e(t) = 1$ leads to a smaller residual for the sinusoidal component i in frame q, the corresponding envelope flag $e_i^{(q)}$ is set. The envelope flag for a harmonic tone is set if partial tones that have the envelope flag set contribute with more than 70% to the total energy of the harmonic tone in the current frame.

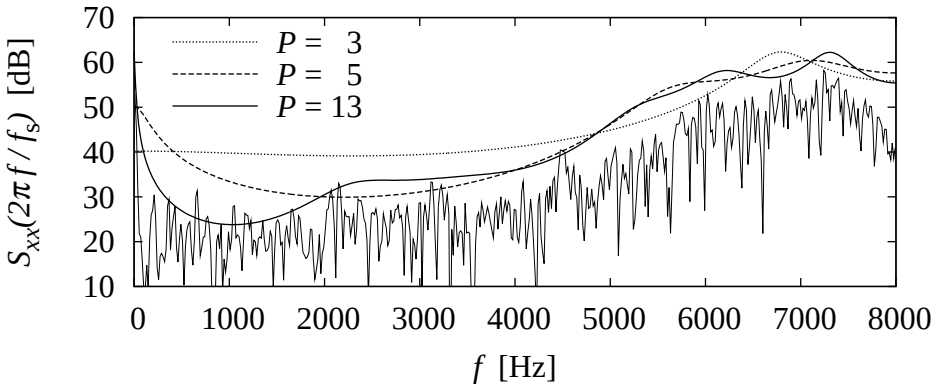

Figure 3.11: Spectrum $S_{xx}(\omega)$ of noise-like segment at $t = 2.15$ s in signal *Suzanne Vega* and spectral envelope approximations with all-pole model of order $P_\mathrm{n} = 3$, $P_\mathrm{n} = 5$, and $P_\mathrm{n} = 13$ with amplitude $a_\mathrm{n} = 459$ (53.2 dB) for $f_\mathrm{s} = 16$ kHz. Amplitudes are given relative to a reference level (0 dB) of $a = 1$ for signals represented as 16 bit PCM, i.e., with a full scale amplitude of $a = 32767$. Note that the signal spectrum is shown attenuated here for better visualization, and that this signal segment contains a significant amount of energy at very low frequencies.

3.2.5 Estimation of Noise Component Parameters

The noise component in a frame is characterized by an amplitude parameter a_n describing the total power $\sigma_x^2 = a_\mathrm{n}^2$ in combination with an all-pole spectral model described by a set of P_n reflection coefficients $k_{p,\mathrm{n}}$. These parameters are estimated based on the power spectrum $S_{rr}(\omega)$ of the residual as available after the analysis-by-synthesis based estimation of the sinusoidal components described in Sections 3.2.1 and 3.3.1.1. The inverse Fourier transform of this spectrum directly gives the autocorrelation $r_{rr}[n] = \mathscr{F}^{-1}\{S_{rr}(\omega)\}[n]$ needed to calculate the parameters of the all-pole model. Assuming that the power spectrum is scaled properly to compensate for the temporal windows applied when calculating $S_{rr}(\omega)$, the amplitude parameter can be found as $a_\mathrm{n} = \sqrt{r_{rr}[0]}$. The Levinson-Durbin algorithm, Equations (3.63) to (3.67), is used to find the reflection coefficients $k_{p,\mathrm{n}}$ up to the desired filter order P_n.

Figure 3.11 shows the power spectrum $S_{xx}(\omega)$ of a noise-like segment of signal *Suzanne Vega* (see Figure 4.18 at $t = 2.15$ s) together with the spectral envelope of the noise component as described by the noise parameters estimated for this frame. In addition to the modeling with filter order $P_\mathrm{n} = 13$, as chosen here, also the spectral envelope approximations with a reduced filter order of $P_\mathrm{n} = 3$ and $P_\mathrm{n} = 5$ are shown.

It is possible to apply a temporal amplitude envelope $a_\mathrm{e}(t)$ to the noise component, and the necessary parameters $t_{\mathrm{max,n}}^{(q)}$, $r_{\mathrm{atk,n}}^{(q)}$, and $r_{\mathrm{dec,n}}^{(q)}$ can be estimated from the residual signal $r(t)$ as described in Section 3.2.4. However, this combination was typically not found to be beneficial and therefore usually no temporal envelope is applied to the noise

component. It should also be noted that the noise parameter estimation described here assumes that the residual power spectrum $S_{rr}(\omega)$ used as input contains only noise-like components. If there are for example spectral peaks corresponding to remaining tonal signal components left over after the analysis-by-synthesis based estimation of the sinusoidal components, the estimated noise parameters may become inaccurate. This effect has to be considered when choosing the number of sinusoidal components estimated in the analysis-by-synthesis loop, as will be discussed in Section 3.3.1.1.

3.3 Signal Decomposition and Component Selection

In order to obtain a parametric description of an audio signal based on a hybrid source model, the audio signal has to be decomposed into its different components such that the model parameters of each component can be estimated. Here, an analysis-by-synthesis approach to signal decomposition is presented and specific aspects of the discrimination of noise and sinusoidal components are discussed. Furthermore, in view of the application to very low bit rate audio coding, it becomes important to determine the perceptual relevance of the different signal components. This information then allows to ensure that the perceptually most relevant components are selected for transmission in a bit stream. In the following, different approaches for component selection with help of perceptual models are presented and their performance is compared.

3.3.1 Signal Decomposition for Hybrid Models

A signal decomposition algorithm has to determine the different components that constitute the input signal and initiate parameter estimation for these components. For this purpose, deterministic components, i.e., components that are modeled by a deterministic signal model (like sinusoids), and stochastic components, i.e., components that are modeled by a stochastic signal model (like noise), have to be distinguished. Deterministic components permit subtractive signal decomposition, and thus allow for an iterative analysis-by-synthesis approach where in each step of the iteration a dominant deterministic component in the current residual is extracted. In contrast, stochastic components do not allow for subtractive decomposition. In principle, if all deterministic components have been extracted properly, only stochastic components are left over in the residual. In practice, however, additional techniques can improve the discrimination of noise and sinusoidal components.

3.3.1.1 Analysis-by-Synthesis Loop

This section describes the analysis-by-synthesis loop, an iterative approach used to extract deterministic signal components from the incoming audio signal. Given the hybrid source model utilized here, all deterministic components are described as sinusoidal trajectories

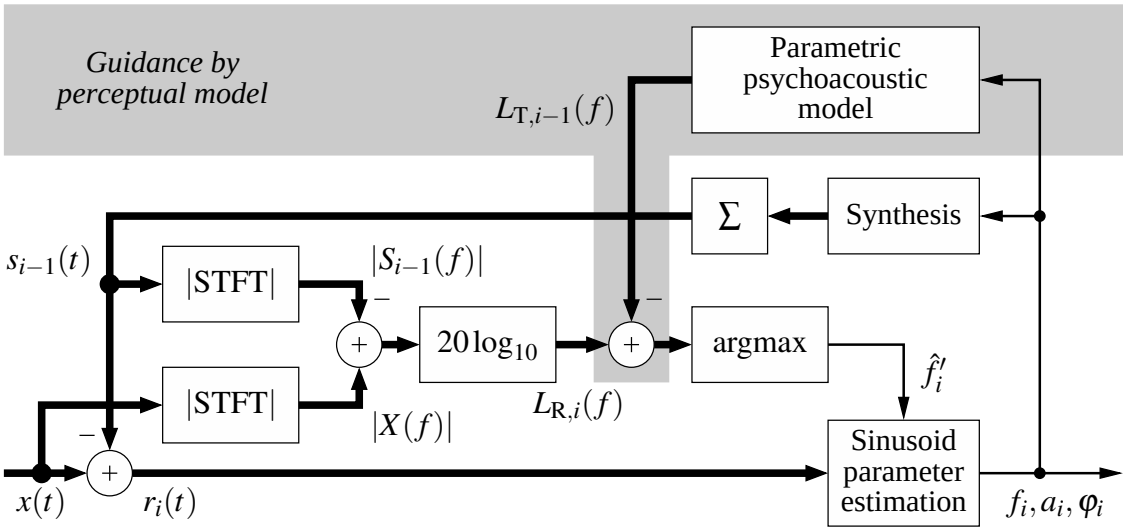

Figure 3.12: Analysis-by-synthesis loop for extraction of sinusoidal components with optional guidance by a psychoacoustic model.

with an optional temporal amplitude, either individual sinusoids or partials of a harmonic tone. The analysis-by-synthesis loop is shown in Figure 3.12. It takes a segment of the original audio signal $x(t)$ as input and generates as output a list of the I extracted sinusoidal components, specified by their estimated parameters f_i, a_i, and φ_i. The gray shaded section of the figure adds a mechanism for perceptual guidance that will be discussed later in Section 3.3.2.2.

Prior to the start of the iterative component extraction procedure, the magnitude spectrum $|X(f)|$ of the input signal $x(t)$ is calculated using a windowed STFT. Furthermore, the accumulator collecting the resynthesized versions of the extracted sinusoids is reset, that is, $s_0(t) = 0$. For each iteration cycle i of the loop, the magnitude spectrum $|S_{i-1}(f)|$ of the synthesized signal $s_{i-1}(t)$ containing all $i-1$ previously extracted sinusoids is calculated. This spectrum is subtracted from the original spectrum $|X(f)|$, and the resulting difference is limited to positive values and converted to a level on the logarithmic dB scale

$$L_{R,i}(f) = 20\log_{10}\max\left(|X(f)| - |S_{i-1}(f)|, 0\right). \tag{3.72}$$

This residual spectrum $L_{R,i}(f)$ indicates how much the original spectrum $|X(f)|$ exceeds the already synthesized spectrum $|S_{i-1}(f)|$. Then, the maximum difference of $L_{R,i}(f)$ compared to a given reference level $L_{T,i-1}(f)$ is identified. For the moment, a constant reference level of $L_{T,i-1}(f) = 0$ dB is assumed, which simply means that the maximum of $L_{R,i}(f)$ itself is identified. The reference level $L_{T,i-1}(f)$ is introduced here to allow for guidance by a perceptual model, as will be discussed below in Section 3.3.2.2. The location $\hat{f}_i'$ of the identified maximum on the frequency axis is used as a coarse frequency estimate for the ith sinusoid.

The accuracy of $\hat{f}_i'$ is determined by the STFT frequency resolution, which is in the

order of 15 Hz for a typical window length of 64 ms for $T_\mathrm{f} = 32$ ms and 50% window overlap. This initial estimate $\hat{f}_i'$ is then used to guide the accurate estimation of the sinusoid's frequency as described in Section 3.2.1.2. This guided frequency estimation is performed on the residual signal $r_i(t) = x(t) - s_{i-1}(t)$ in order to avoid undesired interference from other signal components that were already extracted. The guided frequency estimation employs phase-locked tracking of sinusoidal components, as described in Section 3.2.2.2, in order to identify and build sinusoidal trajectories that are continued from previous frames. It returns accurate estimates of the parameters f_i, a_i, and φ_i, and, in case of a continued trajectory, information about the predecessor. These parameters can then be used to resynthesize the extracted sinusoidal component. However, in order to make the resynthesis as accurate as possible, the estimated phase trajectory $\hat{\varphi}_0(t)$, Equation (3.51), is used for resynthesis instead, together with an interpolated time-varying amplitude. To prepare for the next cycle of the analysis/synthesis loop, the resynthesized signal is added to the previously extracted components, yielding the accumulated signal $s_i(t)$.

Support for transient components modeled as sinusoids with temporal amplitude envelope $a_\mathrm{e}(t)$ is included in this analysis-by-synthesis loop. The decision whether or not to apply an AD envelope to the sinusoidal component currently being extracted is taken such that the residual energy is minimized, as outlined at the end of Section 3.2.4. The AD envelope parameters t_max, r_atk, and r_dec are estimated in each cycle of the loop, based on the current residual $r_i(t)$, until the AD envelope is used for the first time, i.e., the corresponding envelope flag is set. For all following cycles of the analysis-by-synthesis loop for the current frame, the AD envelope parameters remain unchanged.

It is of interest to note that the analysis-by-synthesis loop described here actually makes simultaneously use of two different means to calculate a residual. The time-domain residual $r_i(t)$ is calculated in the usual way by subtraction of resynthesized components. The frequency-domain residual $L_{\mathrm{R},i}(f)$, however, is calculated as the difference of two magnitude spectra. Thus it is more robust against slight modeling errors, like frequency or phase errors, that affect the conventional time-domain residual. This frequency-domain residual enables improved decomposition, because it is used to identify the approximate frequency of the next component to be extracted. Furthermore, the final frequency-domain residual $L_{\mathrm{R},I+1}(f)$ found after the last iteration step I provides the residual spectrum that is considered as a stochastic component, i.e., as noise component.

Various criteria, or a combination thereof, can be used to terminate the iterative component extraction by the analysis-by-synthesis loop. A simple rule is to extract a fixed number of sinusoids in each frame. Alternatively, the number of extracted sinusoids can be determined such that a given budget of bits available for the current frame is not exceeded when the sinusoids' parameters are transmitted. In addition, the loop should also be terminated when the residual signal becomes almost zero, i.e., if the energy of the residual falls below an appropriately chosen and possibly signal dependent threshold. If the analysis-by-synthesis loop is controlled by a psychoacoustic model, as will be discussed in Section 3.3.2.2, an interesting criterion is to terminate the loop when all audible or "perceptually relevant" sinusoidal components have been extracted.

The choice of a termination criterion becomes more difficult when the analysis-by-synthesis loop is also utilized for decomposition into deterministic and stochastic components, that is, if *all* deterministic components are to be extracted such that the final residual contains only stochastic components. In this case, the loop needs to be terminated if it is not possible to extract any additional deterministic components that can be considered as being "sufficiently good." For this purpose, subsequent to the guided parameter estimation, a quality measure q_{sine} can be calculated that describes how well the estimated sinusoidal component represents the original signal in a narrow frequency band centered at the estimated frequency. Details of this quality measure will be explained in Section 3.3.1.2. Only if the quality measure is above a threshold, the estimated sinusoidal component is considered as being "sufficiently good" and is extracted, i.e., subtracted from the residual. Otherwise, the estimated component is discarded as "bad." In both cases, the frequency of the initial estimate is marked in order to prevent that it appears again in a later iteration cycle of the analysis-by-synthesis loop. While it is common that components are sporadically discarded in the analysis-by-synthesis loop, the loop can be terminated if too many "bad" components appear in a row.

Support of a harmonic tone component can be easily added to this analysis-by-synthesis loop. In case the fundamental frequency is estimated from the time-domain audio signal, as described in Section 3.2.3.1, first all partials related to the harmonic tone are extracted in the analysis-by-synthesis loop before extraction continues for the individual sinusoidal components. On the other hand, if the harmonic tone is built from the extracted sinusoidal components, as described in Section 3.2.3.2, no modifications of the analysis-by-synthesis loop are required. Only the termination criterion should be chosen such that all significant partials of a potential harmonic tone are being extracted.

3.3.1.2 Discrimination of Noise and Sinusoidal Components

The decomposition algorithm has to ensure that determination of the type of a component is reliable and robust, in particular with respect to deterministic versus stochastic components. This means that both possible types of errors, the modeling of noise-like signal components as sinusoids as well as the modeling of sinusoidal signal components as noise, should be avoided. Such a discrimination is closely related to perceptual models, i.e., the problem of "what is perceived as noise?" Furthermore, the decision what to model as sinusoids and what to model as noise also can depend on other constraints, like the target bit rate of a coding system. In the following, different extension to and refinements of the analysis-by-synthesis loop are presented that improve discrimination between sinusoidal and noise-like component during signal decomposition.

To avoid the first type of error, i.e., modeling of noise as sinusoids, the quality measure

q_{sine} is introduced. It can be written as

$$q_{\text{sine}} = \frac{a_i^2 \frac{1}{T_\text{f}} \int_0^{T_\text{f}} a_\text{e}^2(t)\,dt}{\int_{f_i-\Delta f/2}^{f_i+\Delta f/2} |X(f)|^2\,df} \tag{3.73}$$

where f_i and a_i are the estimated frequency and amplitude of the sinusoidal component in question, and where $X(f)$ denotes the STFT spectrum of the original signal $x(t)$, scaled appropriately to take into account effects of a temporal analysis window and the transform length. The width of the considered narrow frequency band is denoted Δf and chosen such that it spans the width of the main lobe of the frequency response of the analysis window, i.e., approximately 40 Hz for the typical frame length of $T_\text{f} = 32$ ms and 50% window overlap considered here. If the estimated component has a time-varying frequency, the width Δf of the frequency band is increased by $|\alpha_i T_\text{f}/\pi|$ according to the frequency range traversed within a frame for the estimated chirp rate α_i. The optional temporal amplitude envelope defaults to $a_\text{e}(t) = 1$ when not utilized by the considered sinusoid. A quality measure of $q_{\text{sine}} = 1$ implies that the complete energy of the original signal in this narrow frequency band is properly represented by the estimated sinusoidal component. As long as this quality measure is above an appropriate threshold, e.g., $q_{\text{sine}} > 0.5$, the estimated sinusoidal component can be considered as being "sufficiently good," while it is considered as "bad" and hence discarded in case of lower values of q_{sine}.

Furthermore, the first type of error, i.e., modeling of noise as sinusoids, is also addressed by an additional module that implements a perceptual model which, based on the spectral flatness measure (SFM) [53, p. 57], indicates whether a given narrow spectral band is perceived as a tonal or noise-like signal. Based on the STFT spectrum $X(f)$ of the original signal, the spectral flatness measure γ_x^2 can be calculated for a narrow frequency band with a width of typically $\Delta f = 250$ Hz centered at f_c as

$$\gamma^2(f_\text{c}) = \frac{e^{\left(\frac{1}{\Delta f}\int_{f\text{c}-\Delta f/2}^{f\text{c}+\Delta f/2} \ln|X(f)|^2\,df\right)}}{\frac{1}{\Delta f}\int_{f_\text{c}-\Delta f/2}^{f_\text{c}+\Delta f/2} |X(f)|^2\,df}. \tag{3.74}$$

The value γ_x^2 of the SFM ranges from 1 for a completely flat spectrum (i.e., white noise) down to 0 for a line spectrum (i.e., a tonal signal). If this flatness measure has a value close to 1, indicating the flat spectrum of a noise signal, signal components in the neighborhood of f_c can be considered as noise-like. The reliability of this measure for tonal vs. noise discrimination can be improved by smoothing the power spectrum $|X(f)|^2$, and a moving average operation with a window width of approximately 50 Hz has been found suitable for this purpose.

Excluding noise-like frequency regions completely from sinusoidal component extraction would be prone to artifacts. Hence, an improved approach was conceived that avoids such hard decisions. In this approach, the calculation of the residual spectrum that is considered as stochastic component after termination of the analysis-by-synthesis loop

is adapted to also take into account the narrow-band SFM $\gamma^2(f)$ defined above. Instead of $L_{\mathrm{R},I+1}(f)$ according to Equation (3.72), now the following residual spectrum

$$L_{\mathrm{noise}}(f) = 20\log_{10}\max\left(|X(f)| - w_{\mathrm{SFM}}\left(\gamma^2(f)\right)|S_I(f)|, 0\right) \tag{3.75}$$

is used, where $w_{\mathrm{SFM}}(x) = 1 - x^{10}$ is an empirically derived weighting function. For noise-like frequency bands with an SFM above approximately 0.7, the spectrum $|S_I(f)|$ of the extracted sinusoids is attenuated before subtraction, and for completely noise-like frequency bands with an SFM of 1, the original spectrum $|X(f)|$ is used directly a stochastic component. For all other frequency bands that do not have a distinct noise-like characteristic, the calculation of the frequency domain residual remains unchanged.

To avoid the second type of error, i.e., modeling sinusoids as noise, it is important to ensure that all significant sinusoidal components are extracted from the input signal even if not all of them will be transmitted to the decoder. To achieve this, an appropriate combination of the different termination criteria for the analysis-by-synthesis loop described in Section 3.3.1.1 is necessary. The optimal choice of criteria can actually depend on the target bit rate, which will be discussed in detail in Section 3.4.1.

All these extension and refinements for improved discrimination of noise and sinusoidal components presented here are used in the quality-optimized encoder that will be described in Section 3.4.3.1

3.3.2 Perception-Based Decomposition and Component Selection

In view of the application to very low bit rate audio coding, component selection becomes important in order to ensure that the perceptually most relevant components are conveyed in the bit stream. It is possible to determine the relevance of the different components by means of a perceptual model once the decomposition is completed. Alternatively, such a perceptual model can be integrated directly into the decomposition process. Both approaches are presented in the two following sections, respectively, and focus is put on the selection of sinusoidal components. Perception-related aspects that are of interest during signal decomposition in the context of discrimination of noise and sinusoidal components were already discussed in Section 3.3.1.2. And a more general discussion about how the optimal trade-off between individual sinusoids, harmonic tones, and noise components can be affected by the target bit rate will be given in Section 3.4.1

3.3.2.1 Component Selection by Perceptual Models

Various strategies can be used to select the I perceptually most relevant sinusoidal components in a given frame. These strategies constitute different approaches towards an appropriate perceptual model to assess the relevancy of signal components. In the context of the complete coding system, it is usually necessary to be able to adapt the number I of selected components dynamically to comply with bit rate constraints. Since the exact

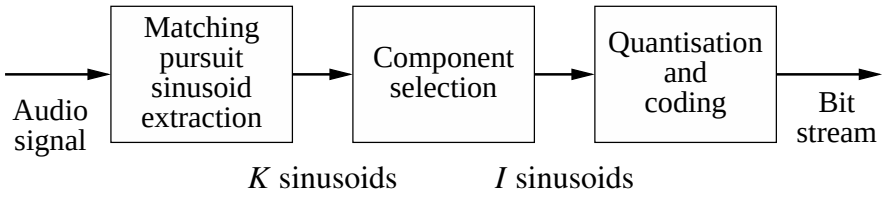

Figure 3.13: Experimental framework for comparison of different strategies for component selection using perceptual models.

value of I often is not yet known during signal decomposition, a common approach is to sort the extracted sinusoidal components according to their perceptual relevance and thus generate an ordered list of K sinusoids with $K \geq I$. When the actual bit stream is generated, the first I components on this list are selected for transmission.

3.3.2.1.1 Component Selection Strategies In order to compare different sinusoidal component selection strategies, a unified experimental framework is required. For this purpose, the sinusoidal matching pursuit decomposition presented in Section 3.2.1.4 is used as a basis here. It is operated in a simplified configuration where the noise and harmonic tone components as well as the optional temporal envelopes for transients are disabled. In this way, a list of K sinusoidal components is generated for each frame of the input audio signal. The task at hand is now to select I out of these K extracted sinusoids for transmission and resynthesis at the decoder, as illustrated in Figure 3.13. This framework allows to consider the resynthesis of all K sinusoids as the ground truth. Now, the optimal component selection is the one having the smallest impact on the perceived sound when the resynthesis of the selected I components is compared to the resynthesis of all K components.

Various strategies for the I out of K component selection have been investigated. They utilize different perceptual models based e.g. on the masked threshold or the auditory excitation level. Most of these strategies generate a list of components ordered according to their relevance, that is, these strategies just reorder the original list of K extracted components. In the following, the seven strategies considered here are described in detail.

- **Strategy SNR** The matching pursuit sinusoid extraction [66] used in the first block of Figure 3.13 is a greedy algorithm that iteratively extracts sinusoids from the current frame of the input signal in order to minimize the energy of the residual, i.e., maximize the signal-to-noise ratio (SNR) of the approximation. Hence, sinusoids are extracted in the order of decreasing amplitude, i.e., the sinusoid with the highest amplitude is extracted first. Since the K extracted sinusoids are already ordered according to the SNR strategy, the selection of I sinusoids is simply done by taking the first I entries of the ordered list of all K extracted sinusoids.
- **Strategy SMR** For this strategy [127], first the masked threshold $L_{\mathrm{T},K}(f)$ describing the simultaneous masking caused by all K extracted sinusoids is calculated using

a parametric psychoacoustic model as will be described in detail in Section 3.3.2.3. The sinusoids are then reordered according to their signal-to-mask ratio (SMR) so that the sinusoid k with maximum $L_k - L_{\mathrm{T},K}(f_k)$ is selected first, where $L_k = 20\log_{10} a_k$ denotes the level of sinusoid k in dB.

- **Strategy HILN** During the development of the parametric coder presented in this work, a novel selection strategy based on simultaneous masking was conceived. It is denoted as HILN here and will be described in detail in Section 3.3.2.2. This strategy employs an iterative algorithm where in the ith step the sinusoid with maximum $L_k - L_{\mathrm{T},i-1}(f_k)$ is selected, i.e., the one which is highest above the masked threshold $L_{\mathrm{T},i-1}(f)$ caused by the $i-1$ sinusoids that were already selected in the previous steps. The iteration is started with the threshold in quiet $L_{\mathrm{T},0}(f)$. As final result, this algorithm generates a reordered list of the extracted sinusoids.

- **Strategy ESW** This strategy was introduced in [93] and is named Excitation Similarity Weighting (ESW). It is based on the auditory excitation pattern [130, Section 6.3] and tries to maximize the matching between the auditory excitation pattern associated with the original signal and the auditory excitation pattern associated with the selected sinusoids. For the experiments reported here, the set of all K extracted sinusoids was regarded as the original signal. To measure the similarity of the excitation patterns, the difference between the excitation levels in dB of the original and the selected sinusoids is accumulated along the basilar membrane. In each step of this iterative procedure, the sinusoid is selected which results in the best improvement in similarity. Since the excitation level of the original is the same for all iteration steps, this procedure is equivalent to an iterative maximization of the overall excitation level Q_{ESW} in dB

$$Q_{\mathrm{ESW}} = \int_0^{24\ \mathrm{Bark}} L_{\mathrm{E}}(z)\,dz \tag{3.76}$$

  where $L_{\mathrm{E}}(z)$ is the excitation level in dB at critical-band rate z.

- **Strategy LOUD** Inspired by the ESW strategy, a new selection strategy LOUD was conceived, which tries to improve perceptual similarity even more. It uses the specific loudness $N'(z)$ in sone/Bark instead of excitation level $L_{\mathrm{E}}(z)$ in dB. Both $N'(z)$ and $L_{\mathrm{E}}(z)$ are non-linear functions of the excitation $E(z)$. Strategy LOUD results in a selection procedure that iteratively maximizes the loudness N in sone [130, Section 8.7] that is associated with the i selected sinusoids. Like ESW, this strategy generates a reordered list of the extracted sinusoids.

- **Strategy LREV** All selection strategies discussed until now make use of greedy reordering algorithms that start with the selection of the most relevant component. However, since the masking- and excitation-based selection strategies utilize non-linear and non-orthogonal quality measures, the greedy approach can lead to sub-optimal results. This can be illustrated using the example shown in Figure 3.14, where $K = 3$ sinusoidal components at 1000 Hz, 1250 Hz, and 1500 Hz with levels of 60 dB,

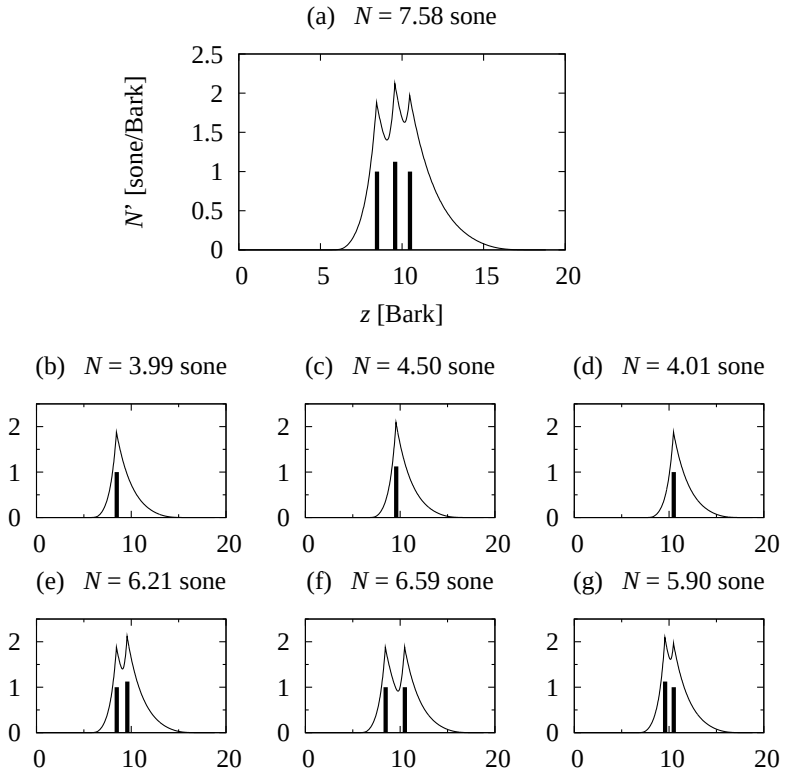

Figure 3.14: Specific loudness $N'(z)$ and loudness N for combinations out of 3 sinusoids ($f_1 = 1000$ Hz, $L_1 = 60$ dB; $f_2 = 1250$ Hz, $L_2 = 62$ dB; and $f_3 = 1500$ Hz, $L_3 = 60$ dB). In (a) all 3 sinusoids are present, as indicated by the vertical bars, in (b), (c), and (d) only a single sinusoid is present, and in (e), (f), and (g) a pair of two sinusoids is present.

62 dB, and 60 dB, respectively, are considered in panel (a). Both the calculated loudness N and subjective assessment indicate that choosing the sinusoids at 1000 Hz and 1500 Hz, as shown in panel (f), is the optimum selection for $I = 2$. All greedy reordering algorithms presented here, however, would select the sinusoid at 1250 Hz as first component, leaving only the sub-optimal alternatives (e) and (g) for $I = 2$. One approach to address this problem is to reverse the direction of the reordering procedures by starting from the full set of K sinusoids and iteratively de-selecting the components considered of lowest relevance. Strategy LREV uses this reversed selection procedure applied to the loudness measure N.

- **Strategy LOPT** It is obvious that also LREV is a greedy algorithm. To assess the sub-optimality of both strategies LOUD and LREV, a full search to find the best subset of I sinusoids that gives the highest loudness N was implemented as reference and is referred to as strategy LOPT here. However, the computational complexity

Strategy S	**Avg. loudness** $\overline{N}_S$	**Avg. diff.** $\overline{N}_S - \overline{N}_{\mathrm{LOPT}}$	**Max diff.** $N_S - N_{\mathrm{LOPT}}$
SNR	10.833 sone	−0.628 sone	−5.979 sone
SMR	11.269 sone	−0.192 sone	−4.593 sone
HILN	11.267 sone	−0.194 sone	−4.006 sone
ESW	11.415 sone	−0.046 sone	−0.925 sone
LOUD	11.459 sone	−0.003 sone	−0.395 sone
LREV	11.460 sone	−0.001 sone	−0.237 sone
LOPT	11.461 sone	0.000 sone	0.000 sone
all 16	12.303 sone	0.842 sone	5.570 sone

Table 3.3: Average loudness $\overline{N}_S$, average loudness difference $\overline{N}_S - \overline{N}_{\mathrm{LOPT}}$, and maximum loudness difference $N_S - N_{\mathrm{LOPT}}$ achieved by different strategies S for selection of $I = 8$ out of $K = 16$ sinusoids for 12 speech and music items with a total duration of 141 s.

of this search is of order $O(2^K)$, which becomes prohibitive for values of K above approximately 20. In addition, LOPT does not lead to a simple reordering of the list of sinusoids, as indicated in the example in Figure 3.14. Hence it cannot be easily combined with the bit allocation strategies typically employed in the quantization and coding block in Figure 3.13, where I is usually determined iteratively such that a given bit budget per frame is not exceeded.

3.3.2.1.2 Experimental Results To compare the performance of the seven selection strategies described above, they were applied to a set of 12 speech and music items with a total duration of 141 s, as listed in Table A.1 in Appendix A, which was used throughout the MPEG-4 Audio core experiment procedure (see Section 5.2.1). For these experiments, the typical configuration for a sampling rate $f_{\mathrm{s}} = 16$ kHz with frame length $T_{\mathrm{f}} = 32$ ms and 50% window overlap was used.

Due to the lack of a simple, well-established perceptual similarity measure, an objective comparison of the performance of the different selection strategies is not easy. However, since most systems for the objective measurement of perceived audio quality, like PEAQ [50], [121], internally utilize modeling of excitation patterns, the loudness measure N seems to be suitable for this purpose. Hence, the full search strategy LOPT is considered as reference here. To make the comparison computationally feasible, the selection of $I = 1,\ldots,16$ sinusoids out of $K = 16$ sinusoids extracted by the matching pursuit was assessed. Figure 3.15 shows the average loudness $\overline{N}_S$ and loudness difference $\overline{N}_S - \overline{N}_{\mathrm{LOPT}}$ achieved by different strategies S. Table 3.3 gives the numerical values for $I = 8$, including the maximum value of the loudness difference $N_S - N_{\mathrm{LOPT}}$ for all frames of the items.

It can be seen from Figure 3.15 and Table 3.3 that LOUD and LREV perform almost equal to LOPT, with a slight advantage for LREV. This means that extremely complex full

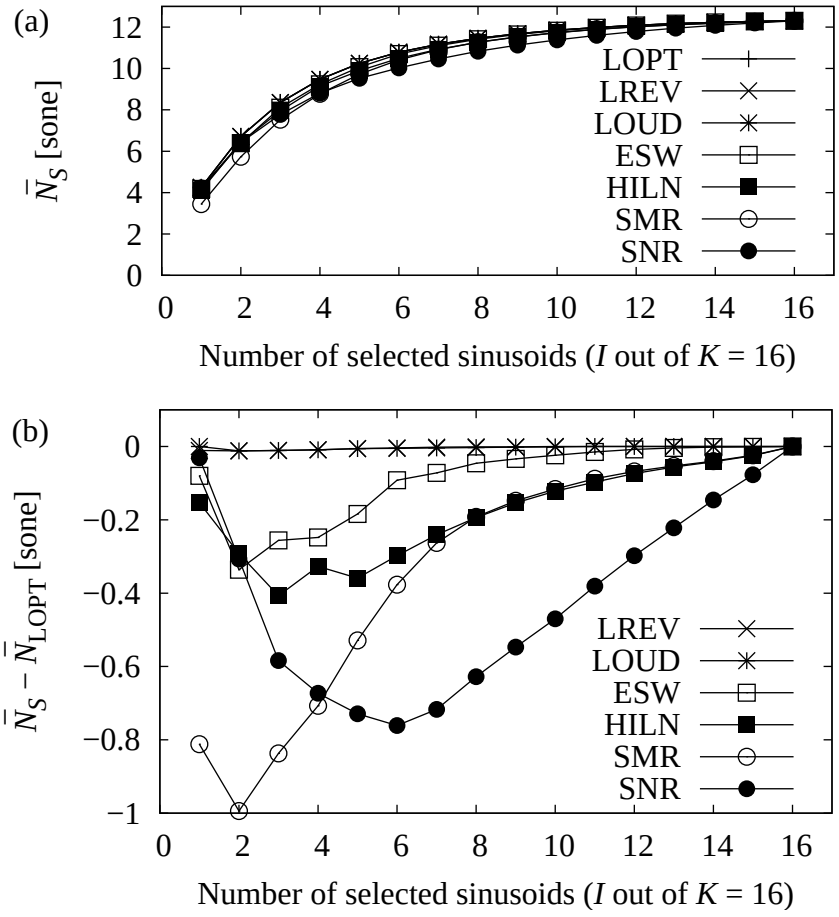

Figure 3.15: Average loudness $\overline{N}_S$ (a) and average loudness difference $\overline{N}_S - \overline{N}_{\mathrm{LOPT}}$ (b) achieved by different strategies S for selection of I out of $K = 16$ sinusoids for 12 speech and music items with a total duration of 141 s (after [109]).

search of LOPT gives only very little additional benefit. As expected, ESW behaves quite similar to LOUD, and only for small values of I, differences in the achieved loudness N are observed. The masking-based strategies SMR and HILN perform almost identical for $I = 7,\ldots,16$, but not as good as the excitation-based strategies LOUD and ESW. It is interesting to observe that SMR shows the worst performance of all strategies for $I = 1,\ldots,4$. Strategy SNR shows the worst performance of all strategies for $I = 5,\ldots,16$. Please note that the vanishing differences in performance when I reaches $K = 16$ are inherently caused by the experimental setup, i.e., the I out of K selection.

To illustrate the differences between the selection strategies discussed here, Figure 3.16 shows the selected $I = 10$ sinusoids out of $K = 40$ extracted sinusoids for one frame of a pop music item with vocals (*Tracy Chapman*). The reordered ranking is indicated by the labels 1 to 10 and the original spectrum as well as the masked threshold

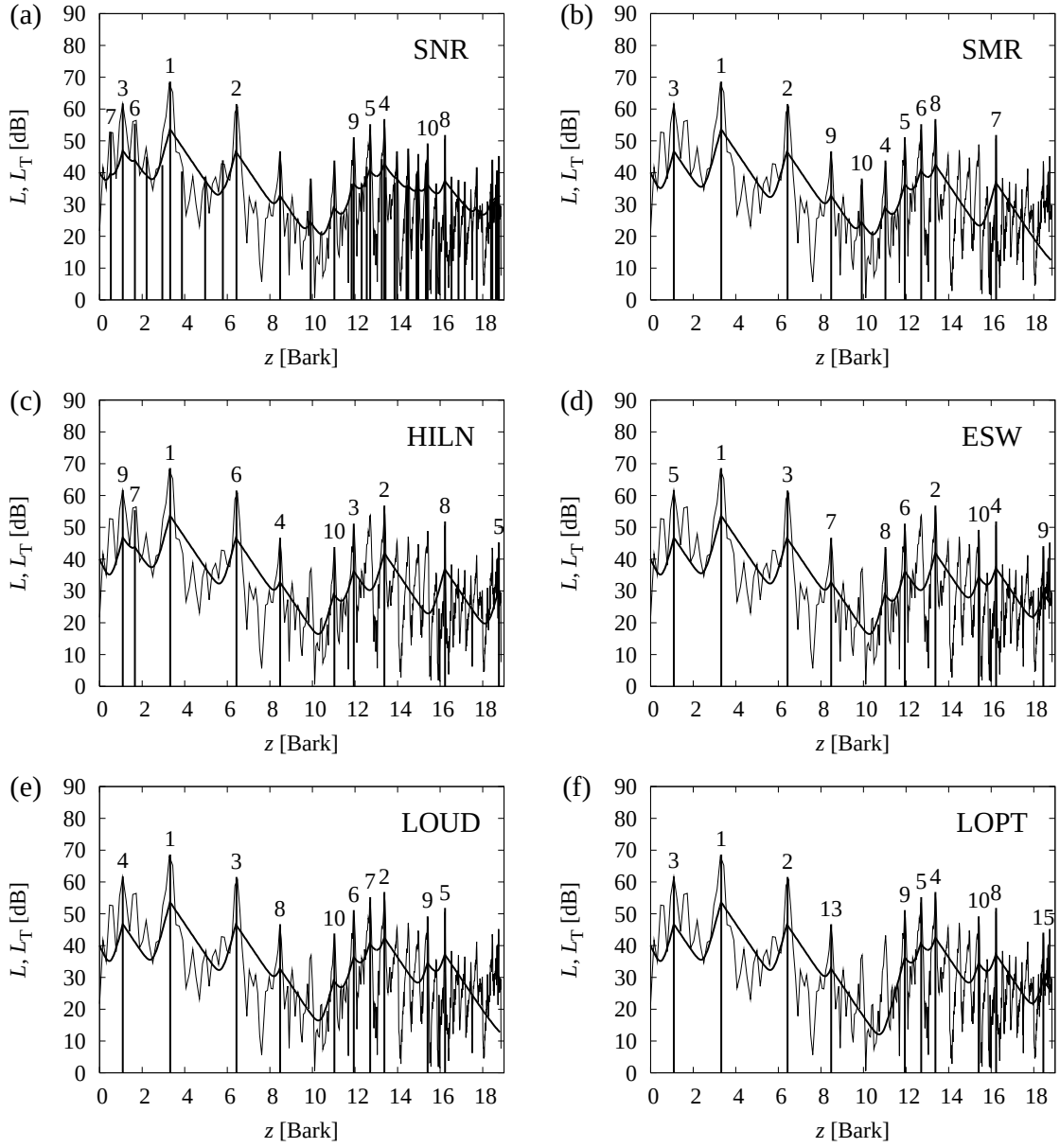

Figure 3.16: Signal spectrum for one frame of signal *Tracy Chapman* and masked threshold evoked by (a) 40 sinusoids selected by strategy SNR (i.e., original ordering from matching pursuit, first 10 sinusoids labeled); (b) 10 out of 40 sinusoids selected by SMR; (c) 10 out of 40 sinusoids selected by HILN; (d) 10 out of 40 sinusoids selected by ESW; (e) 10 out of 40 sinusoids selected by LOUD and LREV; (f) 10 out of 20 sinusoids selected by LOPT (labels show rank in original ordering from matching pursuit) (after [109]).

caused by the selected sinusoids are included in the graphs.

To allow subjective assessment of the different selection strategies, they were implemented in the HILN coder framework (see Section 5.1). Various test items were encoded at a bit rate of 6 kbit/s, i.e., using about 10 to 20 sinusoids per frame, and coding of the noise component was enabled. An informal listening test indicates that the strategy SNR results in a lower quality than all other strategies. However, the differences in subjective quality between the other strategies are fairly subtle and are not expected to result in statistically significant grading differences in a formal listening test.

In summary, strategies based on the masked threshold as well as strategies that seek to approximate the auditory excitation level were found to be suitable for this task. Differences in performance can be observed mainly for the case of a very small number I of selected components, where excitation-based strategies perform advantageous. The original matching pursuit (strategy SNR), which does not incorporate any perceptual model, results in the lowest performance according to both the loudness difference measure and subjective assessment.

3.3.2.2 Analysis-by-Synthesis Guided by Perceptual Models in the Loop

As outlined in Section 3.3.1.1 and shown in Figure 3.12, it is possible to include a psychoacoustic model in the analysis-by-synthesis loop. In this way, the extraction of sinusoidal components can be guided in such a way that the perceptually most relevant components are extracted first. The general idea of this approach was described as selection strategy HILN in Section 3.3.2.1.

The integration of a psychoacoustic model in the analysis-by-synthesis loop enables two potential advantages. On the one hand, the computational complexity can be reduced since only perceptual relevant components are extracted and the extraction of components that would be discarded after perceptual reordering is avoided. On the other hand, the components are extracted already in the order of their perceptual relevance, such that a subsequent reordering, as described in Section 3.3.2.1 is not necessary.

Compared to the analysis-by-synthesis loop discussed in Section 3.3.1.1, where a constant reference level of $L_{\mathrm{T},i-1}(f) = 0$ dB was assumed, the reference level $L_{\mathrm{T},i-1}(f)$ in the ith iteration step now represents the masked threshold for the simultaneous masking caused by all sinusoidal components that were extracted in the previous $i-1$ iteration steps. The iteration is started with the threshold in quiet $L_{\mathrm{T},0}(f)$. The location

$$\hat{f}_i' = \underset{f}{\operatorname{argmax}} \left(L_{\mathrm{R},i}(f) - L_{\mathrm{T},i-1}(f) \right) \tag{3.77}$$

where the residual spectrum is highest above the masked threshold is now used as a coarse frequency estimate for the ith sinusoid.

Given the typical analysis window used here to calculate the residual spectrum $L_{\mathrm{R},i}(f)$, a sinusoid will be represented by a peak at f_i in this spectrum with a main lobe

approximately 30 Hz wide. The masked threshold $L_{\mathrm{T},i-1}(f)$ caused by a previously extracted sinusoid with frequency f_{i-1} exhibits a steep slope for frequencies below f_{i-1}. Assuming f_i is located in the frequency region of this steep slope, the location of the maximum determined by Equation (3.77) is now biased towards lower frequencies, i.e., $\hat{f}_i' < f_i$, and the amount of this error is determined by the slope of $L_{\mathrm{T},i-1}(f)$ and the exact shape of the peak in $L_{\mathrm{R},i}(f)$. To avoid this bias, a post-processing of the coarse estimate $\hat{f}_i'$ according to Equation (3.77) is introduced. It identifies the nearest peak (i.e., local maximum) in residual spectrum $L_{\mathrm{R},i}(f)$ itself within a search range related to the width of the main lobe. Now, the location of this peak is used as coarse frequency estimate provided as input to the guided frequency estimation. An alternative way to address this bias problem, which is also present in the perceptually weighted matching pursuit [127], was later described in [39].

3.3.2.3 Parametric Psychoacoustic Model

A parametric psychoacoustic model derives the masked threshold $L_{\mathrm{T}}(f)$ from a parametric description of an audio signal. Input to a parametric psychoacoustic model is a list of components and their parameters representing the current signal. For each component, the simultaneous masking curve evoked by the component is calculated, and finally the accumulated masking effect is calculated, where the nonlinear "addition" of masking can be considered. For sinusoidal components and the partials of a harmonic tone, frequency and amplitude parameters are needed, and the simultaneous masking is primarily determined by the spreading function $\Delta L(\Delta z)$, i.e., the excitation level versus critical-band rate patterns discussed in [130, Section 6.3]. If a sinusoidal component is combined with a temporal amplitude envelope, this results in a correspondingly reduced level of the masked threshold (due to the energy reduction caused by the envelope) and increased spectral width (due to the bandwidth increase corresponding to the reciprocal of the transient duration). For noise-like components, parameters describing the amplitude and spectral envelope are required, and the masked threshold is calculated by convolution of the noise spectrum (appropriately mapped onto the critical-band rate scale z) with a spreading function.

For the experiments reported in Section 3.3.2.1, a simple parametric psychoacoustic model was used, implementing the basic models of simultaneous masking. For the approach discussed in Section 3.3.2.2, where the psychoacoustic model is included in the analysis-by-synthesis loop, an advanced parametric psychoacoustic model developed by Ferekidis [25] was used. It implements the level dependencies of the slopes of the spreading function $\Delta L(\Delta z)$ and uses a more detailed model for the nonlinear addition of masking, as described in [4].

3.4 Constrained Signal Decomposition and Parameter Estimation

In a coding system based on the parametric signal representation described here, the signal decomposition and parameter estimation can be affected by various external constraints. The most important constraint is the target bit rate at which the coding system is operated. In order to optimize the performance in a rate distortion sense, that is, to achieve the best perceptual audio quality at a given bit rate, various encoder settings need to be "tuned," like frame length, audio bandwidth, and details of the signal decomposition and component selection algorithms. In addition, constraints imposed on computational complexity and delay are also relevant in practical applications. Finally, two specific solutions for complete real-world encoder implementations are presented.

3.4.1 Rate Distortion Optimization

In order to optimize the coding system in a rate distortion sense, constraints imposed by the available bit rate have to be considered during signal decomposition and component selection. Unfortunately, the overall perceptual distortion of interest here is hard to quantify by objective means. Hence most of the optimization and "tuning" described in this section was derived empirically and is based on informal listening or simplified perceptual models.

The quantization and coding of the parameters conveyed in the bit stream will be discussed in Section 4.1, and the bit allocation and bit rate control in the encoder will be described in Section 4.1.4. For the following discussion, it is sufficient to anticipate that the bit rate required by the noise component and the harmonic tone component can be dynamically adapted by reducing the order of the all-pole model describing the spectral envelope. The remaining bit rate is used to transmit as many sinusoidal components as possible from a list where they are ordered according to their perceptual relevance, as described in Section 3.3.2.1.

Considering specifically the target bit rates of 6 and 16 kbit/s, the following aspects of signal decomposition, parameter estimation, and component selection in the encoder have been "tuned" to optimize the resulting perceptual audio quality.

- **Frame Length** As noted previously, typically a frame length of $T_\mathrm{f} = 32$ ms is employed. Significantly shorter frames lead to a reduced perceptual quality because less components can be conveyed if the parameters have to be sent more frequently, in particular at 6 kbit/s. For longer frames, on the other hand, the reduced temporal resolution starts to have a detrimental effect on perceptual quality.
- **Audio Bandwidth** Typically an audio bandwidth of 8 kHz is provided, corresponding to an input sampling rate of $f_\mathrm{s} = 16$ kHz. It was found that bandwidth limitation to 4 kHz did not reduce the coding artifacts that are audible at the target bit rates,

while the bandwidth limitation itself is perceived as an additional artifact. An audio bandwidth larger than 8kHz is possible but does in general not lead to an improved perceptual quality. In case of full 20 kHz audio bandwidth, the employed noise model does not seem to perform optimal and the increased spectral sparseness of sinusoidal signal representation becomes perceptually problematic.

- **Number of Extracted Sinusoids** The number of sinusoids extracted in the analysis-by-synthesis loop is determined by the termination criterion, as explained in Section 3.3.1.1. It is obvious that this number should always exceed the number of individual sinusoidal components that can be transmitted in the bit stream, of course only as long as the extracted components are "sufficiently good" as defined in Section 3.3.1.2. Furthermore, to allow a harmonic tone to be built from the extracted sinusoidal components as described in Section 3.2.3.2, the extracted sinusoids should include most or all partials of a potential harmonic tone. The maximum number of "sufficiently good" sinusoids extracted per frame is typically limited to about 90 to 120, depending upon target bit rate. In combination with the other termination criteria, an average of about 50 sinusoids is extracted per frame for typical audio material.
- **Noise Component** The resulting noise component is primarily determined by the termination criterion of the analysis-by-synthesis loop and the refinements described in Section 3.3.1.2. As stated above, the termination criterion depends upon target bit rate, leading to a slightly stronger noise component at lower bit rates. Nonetheless, in particular at lower bit rates, significantly more sinusoids are extracted than what can be conveyed in the bit stream in order to avoid that left-over sinusoidal components in the residual are modeled as noise.
- **Harmonic Tone** The harmonic tone component tends to be more beneficial at lower than at higher bit rates, since it allows to increase the total number of sinusoids that can be conveyed at lower bit rates, while this effect is less pronounced at higher bit rates. In order to reduce the risk of artifacts caused by miss-detection of a harmonic tone, the detection thresholds are tightened at higher bit rates.
- **Spectral Envelope Model** For the noise component, the order of the all-pole model describing the spectral envelope is typically limited to about 14 or 20 coefficients, depending upon target bit rate. For the harmonic tone, the order of the all-pole model describing the spectral envelope is limited by the number of partials of the harmonic tone. However, the final level of detail of the spectral envelope models for both the harmonic tone and noise component (i.e., the number transmitted LAR coefficients) is determined during bit allocation, as will be described in Section 4.1.4.

3.4.2 Encoder Implementation Constraints

In a real-world implementation of an audio encoder based on the parametric signal representation described here, aspects like computational complexity and delay become rel-

evant. The computational complexity of such an encoder is clearly dominated by the algorithms for signal decomposition and parameter estimation. However, the gain in coding efficiency, that is, the improvement in perceived audio quality at a given bit rate, which can be achieved by using more advanced and thus more complex algorithms is in most cases fairly moderate. Hence, careful optimization can reduce the computational complexity significantly with no or very little loss in coding efficiency. These observations will be exemplified in Section 3.4.3, where two specific encoder implementations are discussed that differ several orders of magnitude in their computational complexity.

The other major constraint imposed on an encoder is the permissible algorithmic delay. For live transmission or two-way communication, usually only short delay between the audio signal at the input of the encoder and the output of the decoder is acceptable. For file-based encoding, on the other hand, no such delay constraints exist and an encoder could take the complete audio signal into account during signal decomposition. An intermediate approach would be an encoder that uses a look-ahead of, for example, some seconds. However, the two specific encoder implementations presented in Section 3.4.3 only introduce a minimal algorithmic delay, which is due to the overlapping windows applied in signal analysis.

3.4.3 Complexity of HILN Encoder Implementations

In the following, two specific encoder implementations for the HILN parametric audio coding system as standardized in MPEG-4 are presented. They employ different techniques for signal decomposition and parameter estimation, whereas parameter quantization and coding (see Section 4.1) is determined by the bit stream format and hence basically the same for both encoders. The first implementation is optimized for best quality, i.e., best rate distortion performance, while the second implementation is optimized for minimum computational complexity, i.e., high execution speed of the encoder.

3.4.3.1 Quality-Optimized Encoder

Figure 3.17 shows the block diagram of the quality-optimized reference encoder that was used during the development of the HILN parametric audio coding system and its standardization in MPEG-4 (see Section 5.1). It was also used to prepare the test items for the subjective verification test reported in Section 5.2. The predominant module is the analysis-by-synthesis loop for the extraction of sinusoidal trajectories as described in Section 3.3.1.1. The parameter estimation is based on phase-locked tracking of sinusoidal components (see Section 3.2.2.2). To enable support of transient components, the temporal amplitude envelope estimation (see Section 3.2.4) is included in the loop. A parametric psychoacoustic model (see Section 3.3.2.3) is incorporated in the loop so that the perceptually most relevant sinusoids are extracted first. A harmonic tone component is supported by grouping of sinusoids, as explained in Section 3.2.3.2. Finally, the noise

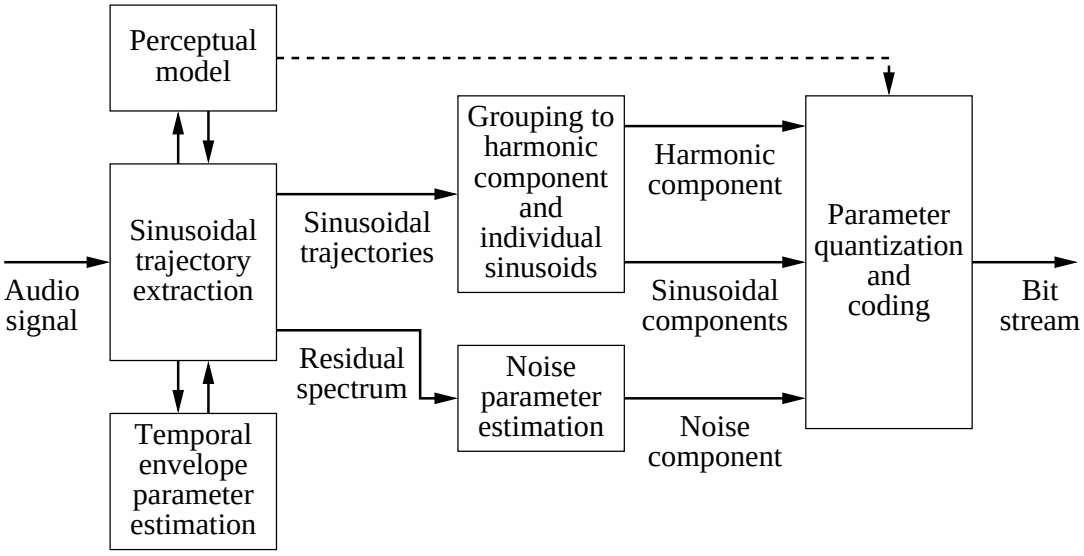

Figure 3.17: General block diagram of the quality-optimized HILN encoder.

component is derived from the residual spectrum according to Equation (3.75) in Section 3.3.1.2.

3.4.3.2 Speed-Optimized Encoder

Figure 3.18 shows the block diagram of an encoder optimized for high execution speed which was developed to demonstrate real-time transmission with the HILN parametric audio coding system [105]. It is based on the frequency-domain matching pursuit described in Section 3.2.1.4. The sinusoidal components extracted in this way are then reordered according to their perceptual relevance by means of strategy HILN, as explained in Section 3.3.2.1. Optionally, support for transient components can be enabled by providing the matching pursuit with an estimated temporal amplitude envelope. Parameters for the noise component are finally estimated from the spectrum of the residual signal. Support for a harmonic tone component was not included in this encoder to avoid the computational load of the modules that would be required. Since the partials of a harmonic tone can also be conveyed as individual sinusoids, usually only little loss in coding efficiency is caused by the lack of a harmonic tone component.

3.4.3.3 Encoder Complexity

To assess the computational complexity in a manner relevant to real-world applications, the average CPU load required for real-time HILN encoding of an audio signal sampled at 16 kHz with a bit rate of 6 or 16 kbit/s was measured on different workstations. These measurements were carried out for three different encoders: the quality-optimized

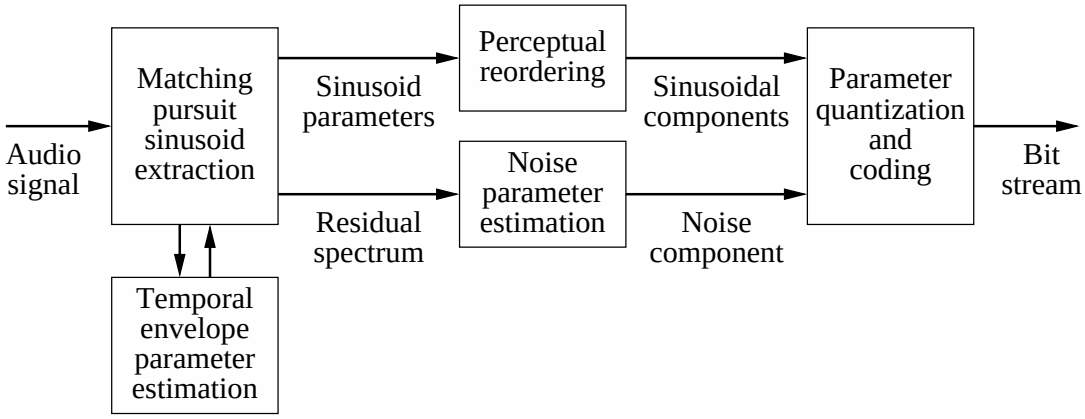

Figure 3.18: General block diagram of the speed-optimized HILN encoder.

Encoder	**Bit rate [kbit/s]**	**CPU load [MHz]**	**Encoding speed-up**
Quality-optimized encoder	6	26000	1
Speed-optimized encoder (no transient support)	6	24	1080
Speed-optimized encoder with transient support	6	51	510
Quality-optimized encoder	16	31000	1
Speed-optimized encoder (no transient support)	16	46	680
Speed-optimized encoder with transient support	16	100	310

Table 3.4: CPU load and encoding speed-up (when compared to the quality-optimized encoder) of three different encoder implementations for audio signals sampled at $f_s = 16$ kHz and operating at a frame length of $T_f = 32$ ms, measured on an Intel Pentium CPU based workstation (PIII 500 MHz) (after [105]).

encoder described in Section 3.4.3.1, the speed-optimized encoder described in Section 3.4.3.2 (without support for transients), and finally the speed-optimized encoder with support for transients enabled. All three encoders are implemented completely in ANSI C. Table 3.4 reports the CPU load measured on a workstation using an Intel Pentium CPU (PIII 500 MHz), and very similar results for the relative execution speed were obtained for workstations using other x86 CPUs or alternative CPU architectures like SUN UltraSPARC and Alpha 21264 [105]. This data indicates that real-time HILN encoding is easily possible with a CPU load in the range of 25 to 100 MHz on a normal PC or workstation using the speed-optimized encoder described here. The computational complexity of the speed-optimized encoder generating a 16 kbit/s bit stream is about twice as high as for 6 kbit/s due to the higher number of sinusoidal components being extracted. Enabling support for transients in this encoder roughly doubles the computational complexity. A more detailed analysis of the speed-optimized encoder can be found in [105].

4 Parameter Coding and Signal Synthesis

In order to build a complete audio coding system based on the parametric signal representation derived in the preceding chapter, suitable techniques for parameter quantization, coding, and transmission are required, together with techniques for efficient signal synthesis in the decoder. This chapter describes the design of a complete quantization, coding, and bit allocation scheme optimized for parametric audio coding at very low target bit rates. This is followed by a discussion of different techniques for efficient signal synthesis in the decoder. Finally, extensions of the parametric coding system are described that enable additional functionalities, addressing signal modification in the decoder (like time-scaling and pitch-shifting), bit rate scalable transmission (i.e., hierarchical embedded coding), and means to improve robustness against transmission errors.

4.1 Parameter Encoding and Bit Allocation

This section discusses the details of parameter quantization, entropy coding, and bit allocation for a parametric audio coding system based on the hybrid source model derived in Section 3.1 are discussed. Specific solutions are derived for a system targeted at very low bit rate operation.

4.1.1 Quantization of Model Parameters

This section elaborates on the parameter quantization for all parameters of the hybrid source model defined in Section 3.1. Most parameters are quantized by a non-uniform scalar quantizer, i.e., independent from other parameters. However, for the LAR parameters describing the spectral envelope of a harmonic tone or noise component, a predictive coding scheme is used to exploit dependencies between subsequent frames. This scheme combines prediction and quantization and is therefore also described in this section.

4.1.1.1 Quantization of Non-Predicted Parameters

The specific details of the scalar quantizers used for the parameters of the hybrid source model in the HILN coding system are discussed now. The quantization step size for all of these quantizers was verified by experimentation in the context of the complete coding system. In general, the step size was chosen such that a significantly coarser quantization would lead to clearly audible artifacts in the decoded audio signal that are directly related to the corresponding quantization error.

4.1.1.1.1 Amplitude Parameters The just-noticeable differences (JND) ΔL for the amplitude of tones or noise at medium levels is approximately 0.7 to 1 dB relative to the corresponding representative level L [130, Section 7.1]. Therefore amplitude parameters are quantized with a non-uniform quantizer that has a constant quantization step size on a logarithmic scale. For this purpose, the amplitude parameters are first converted into a logarithmic representation on the dB scale, where $a = 1$ is used as 0 dB reference level for signals represented as 16 bit PCM, i.e., with a full scale amplitude of $a = 32767$. Then, the desired quantizer can be implemented as uniform quantization on the dB scale.

- For the amplitude parameter a_i of a *sinusoidal trajectory*, quantization with a step size of 1.5 dB was found to be appropriate. Given the dynamic range of approximately 96 dB of a typical audio signal represented as 16 bit PCM, this corresponds to a total of 64 quantization steps. In case of an onset, i.e., for the first frame of a newly born sinusoidal trajectory, even coarser quantization with a step size of 3 dB is acceptable, and thus used in the default operation mode of the amplitude quantization scheme.
- For the amplitude parameter a_h of a *harmonic tone*, quantization with a step size of 1.5 dB was found to be appropriate.
- Similarly, also for the amplitude parameter a_n of a *noise component*, quantization with a step size of 1.5 dB was found to be appropriate.

4.1.1.1.2 Frequency Parameters The JND for the frequency of a tone can best be indicated on the perceptual frequency scale, the critical-band rate z, and is approximately $1/28$ Bark [130, Section 7.2]. For frequencies below 500 Hz, this corresponds to 3.5 Hz, and for frequencies above 500 Hz, this corresponds to a frequency change by a ratio of 1:1.007 (12 cent).

- For the frequency parameter f_i of a *sinusoidal trajectory*, quantization with a step size Δz of $1/32$ Bark on the critical-band rate scale was found to be appropriate. For audio signals sampled at 16 kHz, the maximum frequency of 8 kHz thus results in a quantizer with a total of 603 quantization steps.
- For the fundamental frequency parameter f_h of a *harmonic tone*, quantization with a step size of 1:1.0026 (4.5 cent) on a logarithmic scale was found to be appropriate. This quantizer can represent fundamental frequencies in the range from 20 to 4000 Hz and has 2048 quantization steps. Quantization on a logarithmic scale (instead of the critical-band rate scale) is advantageous because almost all harmonic tones have partials above 500 Hz. The relatively fine quantization reduces the risk of audible artifacts and comes at a very low cost, since there is at most one such parameter per frame.
- For the stretching parameter κ_h of a *harmonic tone*, uniform quantization with a step size of $1/16000$ was found to be appropriate. This quantizer has a total of 35 steps and covers the symmetric range from $-17/16000$ to $17/16000$.

4.1.1.1.3 Phase Parameters In normal operation, the phase parameters for *sinusoidal trajectories* and for the partials of a *harmonic tone* are not conveyed at all, and instead a random start phase in combination with smooth phase continuation is employed for signal synthesis in the decoder. The fact that the phase parameters can be considered perceptually irrelevant for this application was verified by experimentation in the context of the complete HILN coding system. However, if deterministic decoder behavior is required, start phases for *sinusoidal trajectories* and for the partials of a *harmonic tone* are quantized uniformly with $\pi/16$ step size, i.e., with a total of 32 quantization steps.

4.1.1.1.4 Temporal Envelope Parameters For *transient components*, the three parameters $t_{\max}$, r_{atk}, and r_{dec} describing the temporal amplitude envelope $a_{\text{e}}(t)$ are quantized as follows. For the temporal position $t_{\max}$ of the maximum, a uniform quantization with 16 steps covering the duration of the frame from 0 to T_{f} is used. For the quantization of the attack and decay rates r_{atk} and r_{dec}, an angular representation is utilized where an angle of 0 corresponds to a rate r of 0 (flat), an angle of $\pi/4$ corresponds to a rate r of $0.2/T_{\text{f}}$ (a ramp-up or ramp-down within $1/5$ of a frame), and an angle of $\pi/2$ corresponds to a rate r of ∞ (abrupt start or end). These angles in the range of 0 to $\pi/2$ are then quantized with 16 uniform steps.

4.1.1.2 Prediction and Quantization of Spectral Envelope Parameters

The quantization of the parameters describing the spectral envelope of a *harmonic tone* or *noise component* is carried out within a prediction-based coding scheme. In the hybrid source model defined in Section 3.1, the spectral envelope of a *harmonic tone* or *noise component* is described by means of set of P LAR coefficients g_p, where the model order $P^{(q)}$ can change from one frame q to the next frame $q+1$. In many cases, the spectrum of a component varies only slowly over time. Hence, strong dependencies can be observed between the LARs in two subsequent frames. To study these dependencies, the mean $\overline{g_p} = \mathrm{E}\{g_p\}$, the variance $\mathrm{var}(g_p) = \mathrm{E}\{(g_p - \overline{g_p})^2\}$, and the correlation coefficient $\rho_p = \mathrm{E}\{(g_p^{(q)} - \overline{g_p})(g_p^{(q+1)} - \overline{g_p})\}/\mathrm{var}(g_p)$ for the pth LAR in two subsequent frames were measured for a large training set of audio items using a typical encoder configuration with $T_{\text{f}} = 32$ ms and $f_s = 16$ kHz. The observed values for mean, variance, and correlation are shown as a function of the LAR index $p = 1,\ldots,8$ for both harmonic tone components and noise components in Tables 4.1 and 4.2, respectively.

To exploit these dependencies for an efficient coding of the LARs, a simple first order linear predictor with a single predictor coefficient a is used, and the non-zero mean of the LARs is compensated by subtracting a corresponding offset g_{mean} prior to prediction. The resulting predictor structure is shown in Figure 4.1. Note that the calculation of the predicted value $\hat{g}'$ is based on the quantized prediction error $\Delta g'$ instead of the unquantized prediction error Δg. In this way the predictor behaves identical in both encoder and decoder. The optimal predictor coefficient is $a = -\rho$ and the achieved prediction gain

LAR index p	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Mean $\overline{g_p}$	−4.017	1.699	−0.463	0.260	−0.065	0.135	−0.018	0.135
Variance var(g_p)	0.401	0.480	0.271	0.180	0.124	0.098	0.085	0.078
Corr. coeff. ρ_p	0.852	0.842	0.765	0.618	0.597	0.516	0.527	0.482
Pred. gain [bit]	0.934	0.891	0.636	0.347	0.318	0.223	0.235	0.191
Offset $g_{\text{mean},p}$	−5.0	1.5	0.0	0.0	0.0	0.0	0.0	0.0
Pred. coeff. a_p	−0.75	−0.75	−0.5	−0.5	−0.5	−0.5	−0.5	−0.5
Quant. step size	0.1	0.1	0.15	0.15	0.15	0.15	0.15	0.15
Number of steps	128	128	64	64	64	64	64	32

Table 4.1: Mean, variance, correlation, and prediction gain for LARs describing *harmonic tone* spectral envelope (after [71]). Predictor offset, predictor coefficient, quantizer step size, and number of quantizer steps used for coding (for LARs $p = 9,\ldots,25$ the same quantization and coding as for $p = 8$ is used).

LAR index p	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Mean $\overline{g_p}$	−2.536	0.816	−0.431	0.170	−0.160	0.100	−0.093	0.087
Variance var(g_p)	1.609	0.889	0.267	0.148	0.121	0.087	0.081	0.067
Corr. coeff. ρ_p	0.803	0.833	0.761	0.690	0.694	0.635	0.642	0.599
Pred. gain [bit]	0.746	0.855	0.626	0.467	0.474	0.373	0.383	0.321
Offset $g_{\text{mean},p}$	−2.0	0.75	0.0	0.0	0.0	0.0	0.0	0.0
Pred. coeff. a_p	−0.75	−0.75	−0.75	−0.75	−0.75	−0.75	−0.75	−0.75
Quant. step size	0.3	0.3	0.4	0.4	0.4	0.4	0.4	0.4
Number of steps	32	32	33	33	33	33	33	33

Table 4.2: Mean, variance, correlation, and prediction gain for LARs describing *noise component* spectral envelope (after [71]). Predictor offset, predictor coefficient, quantizer step size, and number of quantizer steps used for coding (for LARs $p = 9,\ldots,25$ the same quantization and coding as for $p = 8$ is used).

is $G = 1/(1-\rho^2)$. The gain can be expressed in bits as $\frac{1}{2}\log_2(G)$ and is also given in Tables 4.1 and 4.2.

Instead of the exact values $\overline{g_p}$ and ρ_p measured for the training set of audio items, simpler approximated values for the predictor offset $g_{\text{mean},p}$ and the predictor coefficient a_p are used. These values are given in the lower parts of Tables 4.1 and 4.2. The effect of this parameter prediction scheme on the probability distribution of the coded spectral envelope parameters can be seen in Figure 4.2. It shows the probability density function (PDF) $p(g_i)$ of the original LARs g_i and the PDF $p(\Delta g_i)$ of the resulting prediction errors Δg_i for the first 4 LARs $g_1,\ldots,g_4$ of a noise component. Note that i is used here as a subscript instead of p in order to avoid confusion with the PDF $p(\cdot)$. The non-zero mean of the first coefficients as well as the reduced variance of the prediction errors $\Delta g_1,\ldots,\Delta g_4$ is clearly visible from these figures.

One of the advantages of the LAR representation g_p is that it is a suitable domain to apply uniform quantization, and this holds true for the LAR prediction error Δg_p as well.

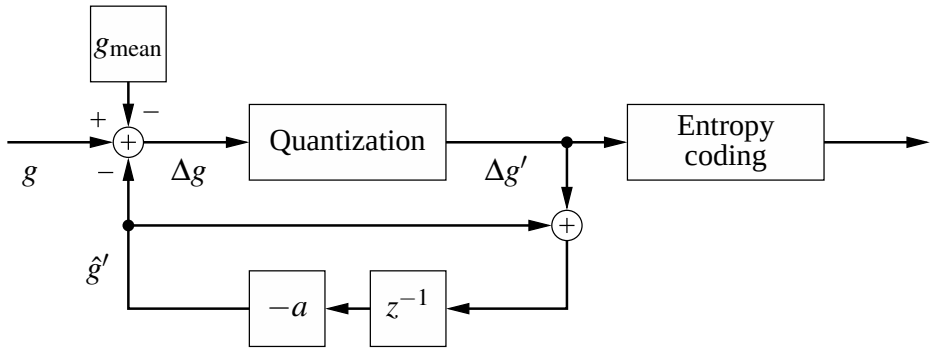

Figure 4.1: Prediction and coding of LARs g_p describing spectral envelopes using offset $g_{\text{mean},p}$ and predictor coefficient a_p (subscript p omitted for legibility).

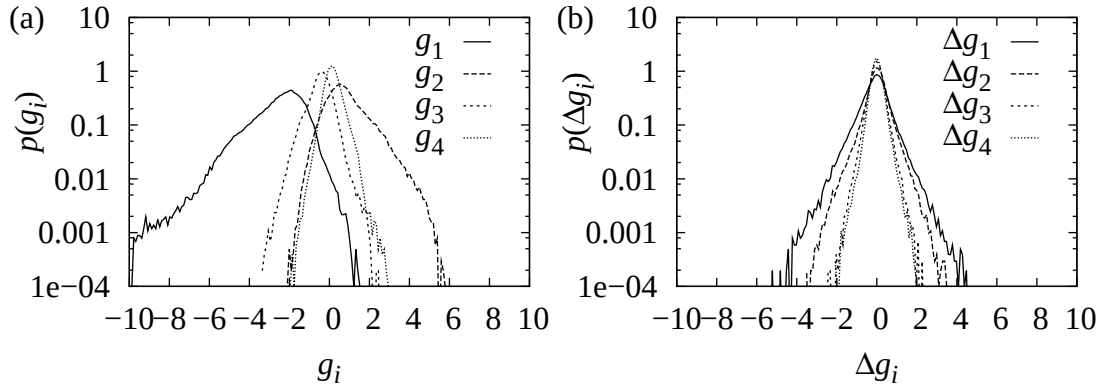

Figure 4.2: PDF $p(g_i)$ of distribution of LARs $g_1, \ldots, g_4$ (a) and PDF $p(\Delta g_i)$ of distribution of prediction error $\Delta g_1, \ldots, \Delta g_4$ (b) for predictive coding of LARs describing noise component spectral envelope (after [71]).

Appropriate quantization step sizes were found empirically by experimentation in the context of the complete parametric coding system. The step sizes chosen in this process, together with the total number of quantizer steps, are given in the lower part of Tables 4.1 and 4.2. It can be seen that the first two LARs g_1, g_2 use finer quantization than the remaining higher order LARs. Furthermore, it can be seen that much coarser quantization is permissible for the LARs describing the spectral envelope of a noise component than for the LARs that describe the spectral envelope of a harmonic tone, and hence in fact the amplitudes of the partials of this tone.

All the quantizers used for the LAR prediction errors are symmetric. And most of them are midrise quantizers, that is, 0 is a decision level and the closest quantized values are $+\frac{1}{2}\Delta$ and $-\frac{1}{2}\Delta$, where Δ denotes the quantization step size. The only exception is the quantizer applied to the LAR prediction errors $\Delta g_3, \ldots, \Delta g_{25}$ of a noise component. Here, a midtread quantizer is used, i.e., 0 is a possible quantized value and the closest

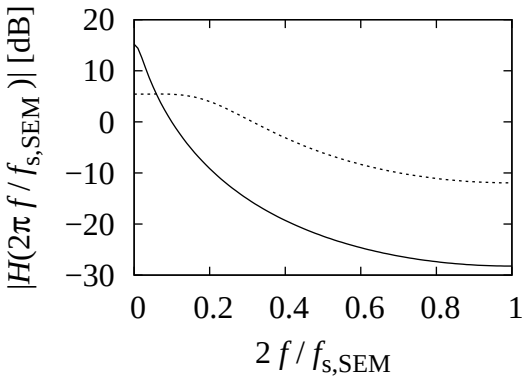

Figure 4.3: Normalized mean all-pole model spectra for harmonic tone (solid line) and noise component (dotted line).

decision levels are $+\frac{1}{2}\Delta$ and $-\frac{1}{2}\Delta$. And, different from all other quantizers used here, the quantized values are not located in the middle between the neighboring decision levels but moved $\frac{1}{8}\Delta$ closer towards 0 in order to minimize the variance of the quantization error, thus reflecting the step slope of the distribution shown in panel (b) of Figure 4.2. This approach is related to the quantizer design proposed by Max [69].

It should be noted that the predictive quantization and coding of LARs as shown in Figure 4.1 still allows to adapt the filter order P of the all-pole spectral envelope model from frame to frame. If the filter order is increased from one frame to the next, the state variables of the predictive coding (shown as delay z^{-1} in Figure 4.1) are simply initialized with 0 for the additional higher LARs that were not conveyed for the previous frame. In case of a reduced filter order, no special precautions are necessary.

Since the predictor offsets $g_{\mathrm{mean},p}$ were derived from the mean LARs observed for a large training set of audio items, they represent the average all-pole model spectral envelopes for harmonic tones and noise components. These lowpass-like spectra are shown in normalized form in Figure 4.3.

4.1.2 Entropy Coding of Model Parameters

After quantization, all the parameters of the hybrid source model are available as a set of integer indices, referring to entries in the corresponding dequantization tables. Now, entropy coding techniques are applied to convey this set of integer parameters in a compact manner from the encoder to the decoder, i.e., using a minimal number of bits per frame. To achieve this, first dependencies between parameters are exploited by means of differential coding, and then variable length codes are employed to minimize the average codeword length.

For the set of frequency and amplitude parameters of sinusoidal trajectories starting in the current frame, a sophisticated technique referred to as *subdivision coding* is applied

that integrates both differential coding and variable length codes. It will be described in Section 4.1.3.

4.1.2.1 Differential Coding of Model Parameters

Differential coding of quantized parameters is a simple yet efficient approach to exploit dependencies between the parameters within a frame and between the parameters in subsequent frames. Differential coding can be seen as the simplest form of predictive coding, where a first order predictor with a predictor coefficient of $-a = 1$ is employed.

- To exploit the *intra-frame* dependencies of the amplitude parameters of all components, the concept of a global amplitude parameter is applied. This new parameter is set to the maximum amplitude of all components in the current frame, and then the amplitudes of all components are described as difference with respect to this maximum. Only the amplitudes of new components (but not of components continued from the previous frame) are coded in this way.
- To exploit the *inter-frame* dependencies of the parameters of components that are continued from the previous frame, time-differential coding is applied. This means that for all amplitude and frequency parameters, only the difference with respect to the value in the previous frame is conveyed. The inter-frame dependencies for the LARs describing spectral envelopes are already taken into account by the predictive coding approach described in Section 4.1.1.2.

4.1.2.2 Variable Length Codes for Quantized Model Parameters

In case a quantized parameter (or its differentially coded representation) exhibits a non-uniform probability distribution over its range of possible values, a code with variable codeword length can be used to reduce the average codeword length (CWL) compared to a straight-forward binary representation of the integer-valued index. For a given distribution, an optimal code can be designed according to Huffman [41]. Such a code is then described by means of a codeword table (or a decision tree) and needs to be available to both the encoder and decoder.

However, if the distribution of the quantized parameters (or their differences) can be approximated by a model distribution, it is often possible to design a more regular code that allows for encoding and decoding using a simple algorithm and hence does not require a large codeword table. Such a code reduces storage and computation requirements compared to a Huffman code and is here referred to as an *algorithmic code*.

The variable length code used to represent the quantized stretching parameter κ_h of a harmonic tone (HFS code) is shown in Table 4.3. It is a good example to illustrate the concept of an algorithmic code. In this case, the codeword structure can be described as [Z S L 4] where Z denotes the zero bit, S denotes the sign bit, L denotes a bit signaling the codeword length, and 4 denotes a 4-bit integer. This code has 35 codewords for the

Codeword	Index	Codeword	Index
1 1 1 1111	−17	1 0 0	1
1 1 1 1110	−16	1 0 1 0000	2
1 1 1 1101	−15	1 0 1 0001	3
1 1 1 *xxxx*	$-y$	1 0 1 *xxxx*	y
1 1 1 0001	−3	1 0 1 1101	15
1 1 1 0000	−2	1 0 1 1110	16
1 1 0	−1	1 0 1 1111	17
0	0		

Table 4.3: Codeword table of algorithmic code (HFS code) used for quantized stretching parameter κ_h of a harmonic tone. Bits within a codeword are grouped to visualize the structure [Z S L 4] of the algorithmic code.

indices $i = -17, \ldots, 17$ with a CWL in the range $1 \leq l_i \leq 7$. This code would be optimal (i.e., the average CWL is equal to the entropy) if the probability distribution of the index i representing the quantized parameter would be the same as the underlying symmetric and staircase-like model distribution with the probabilities $P(i) = (1/2)^{l_i}$.

The algorithmic codes DIA, DHF, and DIF used for the time-differentially coded amplitude and frequency parameters of components continued from the previous frame are listed in Table 4.4. The DIA code is used for the amplitude index differences of all component types (harmonic tones, individual sinusoids, noise) but was optimized for the probability distribution observed for individual sinusoids, since they constitute more than 90% of parameters where the DIA code is used. For the frequency index differences of harmonic tones and individual sinusoids, two independent codes, DHF and DIF, are used. Table 4.4 also gives the entropy and average CWL for these codes, measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items with a total of 20866 frames (667.7 s). This set of 39 audio items is listed in Table A.2 and is used throughout this chapter for this type of measurements. It can be seen that the average CWL is only slightly larger than the entropy, indicating the good performance of these algorithmic codes. The only exception is the DIA code in case of harmonic tones, where the average CWL is more than 1 bit larger than the entropy. However, this is only a very small disadvantage, since it only affects less than 2% of the parameters where the DIA code is applied.

The codeword lengths of the DIA code together with the measured probability distribution of the corresponding parameter index differences are shown in Figure 4.4. The measured probabilities are given as $-\log_2(P)$ so that they can be compared directly with the codeword lengths. It can be seen that the codeword lengths closely match the measured probabilities, especially for the codes with highest probability (i.e., the short codewords). Also the staircase-like model distribution reflected in the design of the algorithmic codes is clearly visible. Also for the DHF and DIF codes, the codeword lengths closely match the measured probabilities.

To represent the predicted and quantized LAR spectral envelope parameters (see Sec-

Parameter	Code	Range	Entropy [bit]	Avg. CWL [bit]
ampl. index diff. (harm+indi+noise)	DIA	$-25,\ldots,25$	3.580	3.811
ampl. index diff. (harm, 1.8%)	DIA	$-25,\ldots,25$	1.970	3.129
ampl. index diff. (indi, 90.3%)	DIA	$-25,\ldots,25$	3.590	3.820
ampl. index diff. (noise, 7.9%)	DIA	$-25,\ldots,25$	3.651	3.867
freq. index diff. (harm)	DHF	$-69,\ldots,69$	2.793	3.259
freq. index diff. (indi)	DIF	$-42,\ldots,42$	2.582	2.928

Table 4.4: Algorithmic codes (DIA, DHF, DIF) used for time-differentially coded parameters of components continued from the previous frame. Entropy and average codeword length measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

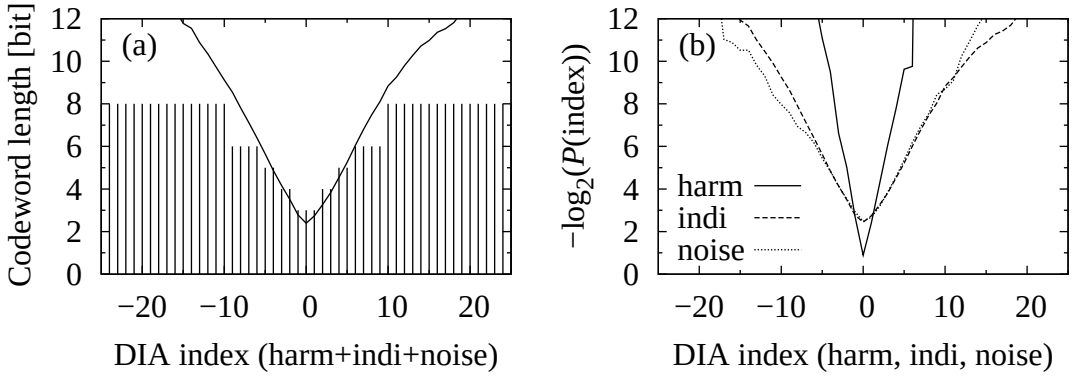

Figure 4.4: Codeword length of algorithmic DIA code (panel (a), vertical bars) and probability of amplitude index differences for all components shown as $-\log_2(P)$ (panel (a), solid line). Probability of amplitude index differences for the three different component types (panel (b)). Probabilities measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

tion 4.1.1.2), a set of five different algorithmic codes is used. These codes LARH1, LARH2, LARH3, LARN1, and LARN2 are listed in Table 4.5. Since the probability distribution depends on the order p of the LAR coefficient, a set of different codes is used for different ranges of p, as described in this table.

The inter-frame prediction error Δg of the LARs, as shown in panel (b) of Figure 4.2, has a PDF $p_x(x)$ that can be approximated closely by the Laplace distribution

$$p_x(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\frac{|x-m_x|}{\sigma_x/\sqrt{2}}}. \tag{4.1}$$

Since uniform quantization is used for these parameters, the codeword length of an optimal variable length code should be approximately proportional to the magnitude of the index representing the quantized parameter. For a probability distribution where the prob-

Parameter	Code	Range	Code structure	Entropy [bit]	Avg. CWL [bit]
harm LAR $p = 1,2$	LARH1	$-64,\ldots,63$	S R 2	5.326	5.899
harm LAR $p = 3,\ldots,7$	LARH2	$-32,\ldots,31$	S R 1	4.015	4.056
harm LAR $p = 8,\ldots,25$	LARH3	$-16,\ldots,15$	S R	3.261	3.349
noise LAR $p = 1,2$	LARN1	$-16,\ldots,15$	S R	3.317	3.375
noise LAR $p = 3,\ldots,25$	LARN2	$-16,\ldots,16$	Z S R	1.693	1.943

Table 4.5: Algorithmic codes (LARH1, LARH2, LARH3, LARN1, LARN2) used for predicted LAR parameters describing spectral envelope of harmonic tone and noise component. Entropy and average codeword length measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

ability decreases by a factor of $1/2$ when the absolute value of the index is increased by 1, an optimal code with the structure [S R] can be conceived. Here, S is the sign bit and R is a run of bits with value 0 terminated by a bit with value 1. The number of bits in this run then gives the magnitude of the index. In this case, a midrise quantizers is assumed, i.e., there is no codeword for the parameter value zero. The codes LARH3 and LARN1 have this structure. Note that, in order to avoid the gap at index 0, positive indices are denoted 0 to 15 here (instead of 1 to 16) while negative indices are denoted -1 to -16.

If the probability distribution is wider, e.g., if the index has to be increased by 4 in order for the probability decrease by a factor of $1/2$, a 2-bit integer can be appended after the run in the codeword, resulting in a algorithmic code with structure [S R 2]. This approach is applied for code LARH1, while for code LARH2, a 1-bit integer is used instead. Finally, code LARN2 handles parameters originating from a midtread quantizer. Therefore, the algorithmic code has the structure [Z S R], i.e., it starts with a bit that indicated whether the value of the quantized index is zero or not.

Table 4.5 also gives the entropy and average CWL for the codes LARH1, LARH2, LARH3, LARN1, and LARN2, measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items. It can be seen that the average CWL is only slightly larger than the entropy, indicating the good performance of these algorithmic codes. The codeword lengths of these five codes closely match with the measured probability distribution of the corresponding parameter indices. Only for LARH1, shown in Figure 4.5, the observed distribution is less symmetric than the Laplace model, which is related to the fact that these parameters represent the global nature (like highpass/lowpass/bandpass) of the spectral envelope of a harmonic tone. Also the linear relationship between the codeword length and the magnitude of the index can be clearly seen.

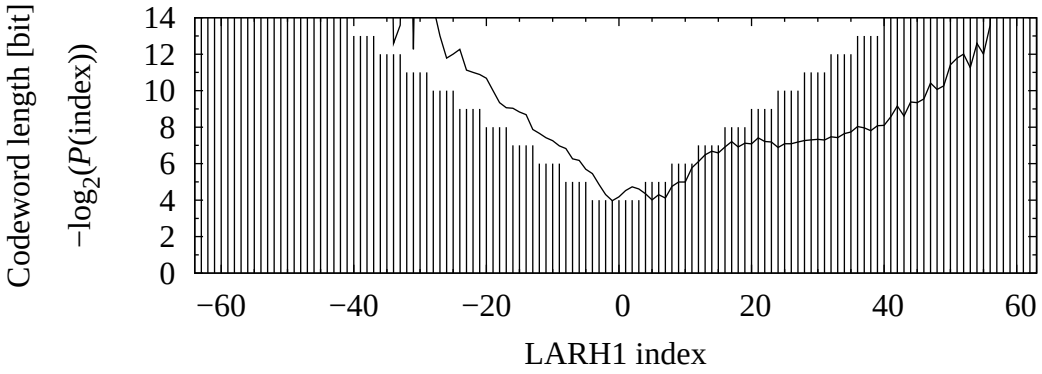

Figure 4.5: Codeword length of algorithmic LARH1 code (vertical bars) and distribution of measured probabilities shown as $-\log_2(P)$ (solid line) for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

4.1.3 Joint Coding of a Set of Model Parameters by Subdivision Coding

To encode the set of quantized frequency and amplitude parameters of sinusoidal trajectories starting in the current frame, a sophisticated technique has been devised that is referred to as subdivision coding. It is motivated by the fact that it is not necessary to convey the frequency/amplitude parameter pairs of new sinusoidal trajectories in any specific order. Assuming N new sinusoidal trajectories, there is a total of $N!$ possible permutations of this set of N parameter pairs. By choosing one specific permutation, the corresponding redundancy of $\log_2(N!)$ bit can be exploited when conveying a set of N parameter pairs. Figure 4.6 shows the possible bit rate reduction (given as number of bits saved per parameter) as a function of the total number N of parameters in the set. Subdivision coding utilizes this effect by arranging the parameter pairs in an order of increasing frequency, which constitutes a well-defined permutation of the set of parameter pairs. This is followed by intra-frame differential coding in combination with adaptive variable length codes which exploits the statistical dependencies of the frequency parameters that were introduced when sorting them. Furthermore, subdivision coding takes also advantage of the non-uniform probability distribution of the frequency and amplitude parameters.

To motivate the concept of subdivision coding, a set of N statistically independent integer parameters f_n (e.g., the set of indices representing quantized frequencies) with uniform distribution in the range $0 \leq f < F$ is considered. In a first step, the elements in the set are ordered according to their value, yielding $0 \leq f_{N-1} \leq \ldots \leq f_1 \leq f_0 < F$. The parameters are now coded in this order, starting with the smallest parameter f_{N-1}. Note that with this ordering, the index n indicates the number of parameters that remain to be coded. For each parameter f_n, $n = 0,\ldots,N-1$, the range of possible values is limited by the previously coded parameter f_{n+1} (or $f_N = 0$ for the first parameter $n = N-1$) and the

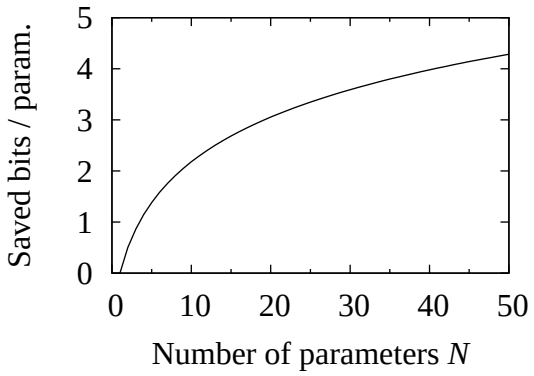

Figure 4.6: Theoretical bit rate reduction $\log_2(N!)/N$ in bits per parameter for subdivision coding of a set of N parameters.

highest possible value $F-1$. In order to devise a variable length code for the differential coding of this parameter, the distribution of the probabilities $P(f_n - f_{n+1})$ must be known. It can be approximated as

$$P_n(f_n - f_{n+1}) \approx \frac{1}{F - f_{n+1}} p_n\left(\frac{f_n - f_{n+1}}{F - f_{n+1}}\right) \tag{4.2}$$

by sampling a continuous PDF $p_n(x)$ describing the shape of the probability distribution normalized for the range $0 \le x < 1$. This shape can be determined with help of the cumulative distribution function

$$D_M(x_{\min}) = P(x \le x_{\min}) = 1 - (1 - x_{\min})^M \tag{4.3}$$

for the smallest value $x_{\min}$ of a set of M independent continuous random variables x_m, $m = 1,\ldots,M$ with uniform distribution in the interval $0 \le x_m < 1$. This cumulative distribution function $D_M(x_{\min})$ follows from a simple geometric construction in the M-dimensional space $[x_m]$, where the volume of the cube $(1 - x_{\min})^M$ represents the probability $P(x > x_{\min})$ that all random variables x_m are larger than $x_{\min}$. Figure 4.7 illustrates this cube as a shaded square for $M = 2$. Derivation of $D_M(x)$ for $M = n+1$ gives the wanted PDF

$$p_n(x) = \frac{dD_{n+1}(x)}{dx} = (n+1)(1-x)^n. \tag{4.4}$$

Figure 4.8 shows an example of the distribution of these probabilities $P_n(f_n - f'_{n+1})$ for the case of $N = 3$ parameters with the actual values $f'_2 = 0.15F$, $f'_1 = 0.49F$, and $f'_0 = 0.55F$.

To construct a variable length code for parameter f_n that is adapted to the assumed distribution of the probabilities $P_n(f_n - f_{n+1})$, a recursive subdivision approach is pursued. In a first step, the range of possible values for integer parameter f_n (which starts at f_{n+1} and ends at $F-1$) is divided into two parts of equal probability $1/2$. The first bit of the

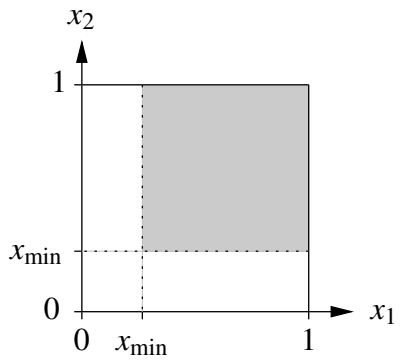

Figure 4.7: Geometric construction for cumulative distribution function $D_M(x_{\min})$ for $M = 2$. The shaded area corresponds to the probability $P(x > x_{\min}) = 1 - D_M(x_{\min}) = (1 - x_{\min})^2$ that both random variables x_1 and x_2 are larger than $x_{\min}$.

codeword now indicates in which of these two parts the actual parameter value is located. This part now determines the range of possible values for f_n in the second step. The new range is again divided into two parts of equal probability, and the second bit of the codeword is determined. This subdivision step is repeated recursively until the resulting range contains only a single possible value. The subdivision algorithm requires that boundaries of the considered ranges have integer values, which means that the probabilities of both parts in a subdivision step are not always exactly equal. In Figure 4.8, the boundaries for the first and second subdivision step are shown for all three parameters considered in this example. The codeword for the first parameter $f_2' = 0.15F$ begins with 01 since the actual parameter value is in the lower half of the first subdivision step (first bit in codeword is 0) but in the upper half of the second subdivision step (second bit in codeword is 1). The codewords for the second parameter $f_1' = 0.49F$ and the third parameter $f_0' = 0.55F$ in this example begin with 10 and 00, respectively.

A corresponding subdivision algorithm is operated in the decoder using the transmitted bits to select in each step the same part as in the encoder, thereby retrieving the actual parameter value when the recursion terminates because the resulting range contains only a single value. The exact positions of the boundaries between parts are stored in SDC boundary tables for the first 5 subdivision steps. Hence, these tables have $2^5 - 1 = 31$ entries. For the further subdivision steps, uniform distribution within the remaining range is assumed. The SDC boundary tables are available to both encoder and decoder and thus ensure identical behavior of the subdivision process on both sides. Since the shape of the PDF $p_n(x)$ depends on n, i.e., the number of the parameters in the set that remain to be coded, an appropriate SDC boundary table must be chosen for a given n. For the coding of frequency parameters, a total of 8 boundary tables were defined, one for each of the value $n = 0, \ldots, 6$ and an 8th table that is used for $n \geq 7$, i.e., if 7 or more frequency parameters remain to be coded.

Since the real PDFs differ slightly from the assumption of uniform and independent

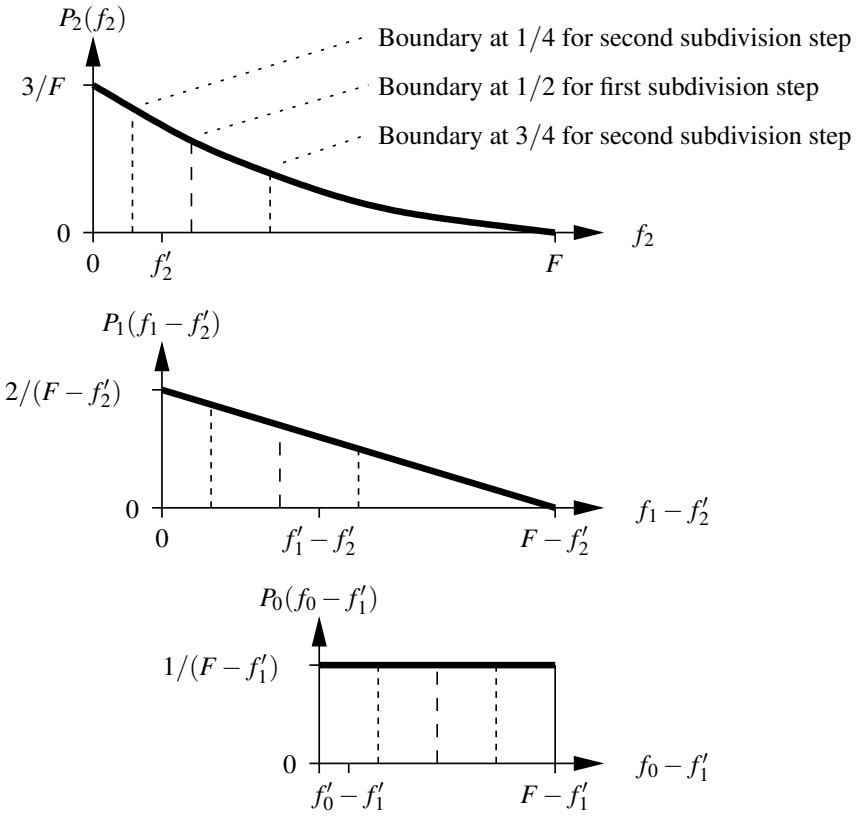

Figure 4.8: Example of subdivision coding of $N = 3$ parameters with the actual values $f_2' = 0.15F$, $f_1' = 0.49F$, and $f_0' = 0.55F$, assuming uniform and independent distribution of the parameters $0 \leq f_n < F$ prior to sorting. The dashed vertical lines indicate the boundaries for the first and second subdivision step used to generate the first and second bit of the codewords. The codewords for the parameters f_2', f_1', and f_0' begin with 01, 10, and 00, respectively. The bold lines indicate the shape of the distribution of the probabilities $P_n(f_n - f_{n+1}')$.

distribution that lead to Equation (4.4), empirically collected data was used in the design of the SDC boundary tables employed in the actual implementation. Panel (a) of Figure 4.9 shows the PDFs $p_n(x)$ of the probability distribution models described by the 8 SDC boundary tables for $n = 0,\ldots,7$. Panel (b) shows the corresponding PDFs $p_n(x)$ measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items. The complete decoding algorithm for subdivision coding [72] including the necessary SDC boundary tables is given in Appendix B.

For reasons of simplicity, subdivision coding is also used for the quantized amplitude parameters of the sinusoidal trajectories starting in the current frame. Like all other amplitude parameters, they are given as difference relative to the global amplitude parameter (i.e., the maximum amplitude of all components in a frame, see Section 4.1.2.1) and typically coarse quantization is applied to these difference as described in Section 4.1.1.1.1.

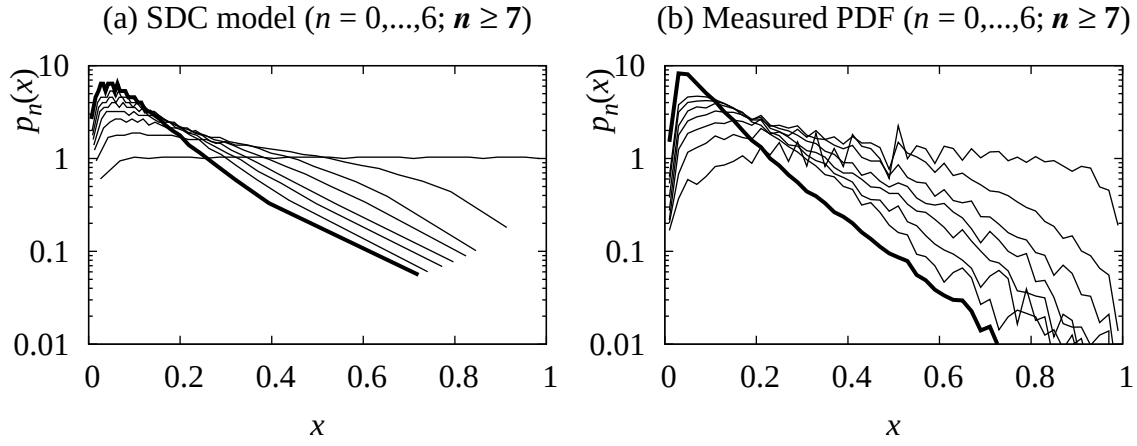

Figure 4.9: PDFs $p_n(x)$ of probability distribution models (a) as described by SDC tables for frequency indices for $n = 0$ (almost horizontal line) to $n \geq 7$ (bold line) and measured PDFs (b) $p_n(x)$ for $n = 0$ (almost horizontal line) to $n \geq 7$ (bold line) for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

The resulting amplitude parameter is limited to the range of 0 to 24, where 0 denotes the maximum amplitude (as given by the global amplitude parameter). Based on the non-uniform distribution that can be observed for this parameter, a special SDC boundary table was designed. Figure 4.10 depicts the probability distribution model described by this boundary table (shown as $-\log_2(p(x))$) together with the actual lengths of the 25 codewords generated by subdivision coding. Furthermore, the probability distribution measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items is included in this graph.

Table 4.6 gives the entropy and the average CWL measured for subdivision coding of quantized amplitude parameters based on HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items. It can be seen that the average CWL is only slightly larger than the entropy, indicating good performance of this approach.

Table 4.6 also gives the average CWL measured for the frequency parameters. For the bit streams considered here, the quantized frequency parameter can have $F = 603$ different values (corresponding to the range from 0 Hz to $f_s/2 = 8$ kHz). The average number N of new sinusoidal components in a frame is 5.9 or 21.4 for the bit streams encoded at 6 or 16 kbit/s, respectively (see Table 4.9). Assuming independent and uniform parameter distribution, this would correspond to a theoretical bit rate reduction $\log_2(N!)/N$ of approximately 1.6 or 3.1 bit per parameter, respectively. For reference, the entropy measured for direct non-differential coding of the absolute frequency indices and for intra-frame differential coding of the ordered frequency parameters as frequency index differences are given as well. The corresponding probability distributions for absolute frequency indices and the index differences in case of intra-frame differential coding are

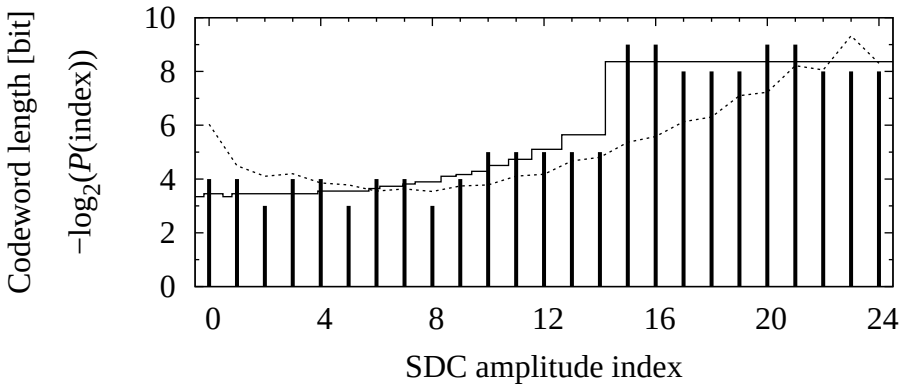

Figure 4.10: Codeword length for SDC of amplitude indices (vertical bars), probability distribution model described by SDC boundary table (staircase-like solid line), and distribution of measured probabilities shown as $-\log_2(P)$ (dotted line) for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

Parameter	**Range**	**Entropy [bit]**	**Avg. CWL [bit]**
SDC of amplitude indices	$0,\ldots,24$	4.193	4.491
SDC of set of frequency indices	$0,\ldots,602$		5.945
Absolute frequency indices	$0,\ldots,602$	8.826	
Frequency index differences	$0,\ldots,602$	6.534	

Table 4.6: Entropy and average CWL for subdivision coding of amplitude indices. Average CWL for subdivision coding of the set of frequency indices of new individual sinusoids and entropy for two simpler coding schemes (absolute and intra-frame differential coding) for frequency indices. Entropy and average codeword length measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

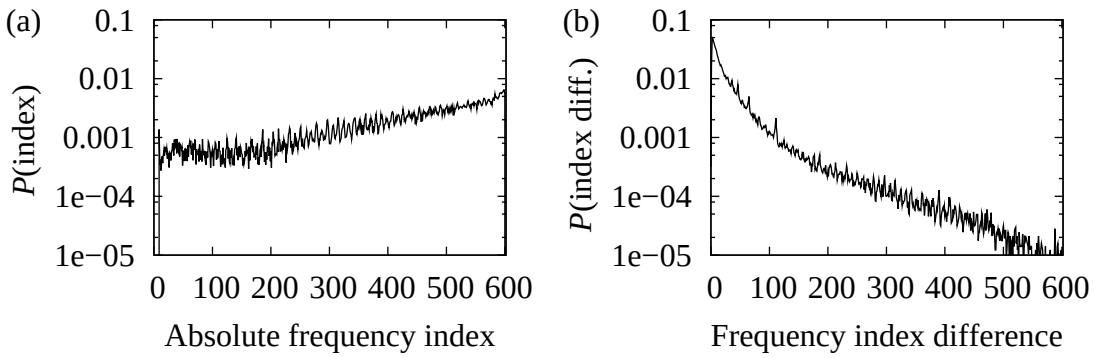

Figure 4.11: Distribution of probabilities (a) of frequency indices of new individual sinusoids (i.e., absolute coding) and (b) of frequency index differences (i.e., intra-frame differential coding of sorted list of frequencies) measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

shown in Figure 4.11. It is interesting to note that the average CWL of subdivision coding is even lower than the entropy measured for intra-frame differential coding. This can be explained by the fact that the differential coding did not take into account that the probability distribution is depending on the number n of parameters that remain to be coded.

As result, the subdivision coding approach presented here allows to reduce the number of bits needed to represent the sinusoidal trajectories beginning in a frame by more than 30% compared to direct non-differential coding [99]. This constitutes a significant gain in coding efficiency and is important since the parameters of new sinusoidal trajectories account for approximately 35% to 45% of the total bit rate (see Table 4.7).

4.1.4 Bit Allocation in HILN Encoder

In order to convey the binary codewords representing the quantized and coded parameters from the encoder to the decoder, they must be assembled to form a bit stream that can be properly parsed again in the decoder. The general structure of the HILN bit stream format is shown in Figure 4.12. It comprises a configuration header and a sequence of encoded frames. The configuration header carries all data that does not change from frame to frame, e.g., the frame length and the sampling rate. The encoded frame comprises the codewords representing all the parameters of the signal components that are conveyed in this frame. In order to be able to decode this data correctly, additional side information is conveyed in a frame header at the beginning of each frame. This side information signals which components are present in the current frame, the total number of individual sinusoids, which of these sinusoids are continued from the previous frame, and whether a temporal amplitude envelope for transient components is conveyed. For the noise component and the harmonic tone, the number LAR parameters used for the spectral envelope

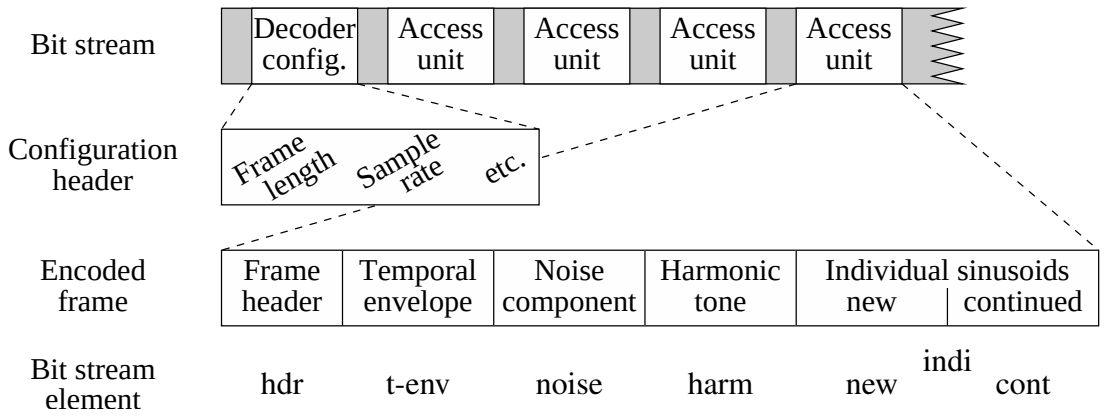

Figure 4.12: General structure of HILN bit stream comprising configuration header and a sequence of encoded frames. The bit stream elements of an encoded frame convey the parameters of the signal components present in that frame.

model is conveyed, and for a harmonic tone, also the total number of partials is conveyed.

The encoder typically has only a limited budget of bits available to convey the coded parameters for each frame, and the size of this budget depends primarily on the desired target bit rate and the frame length. Therefore, mechanisms for optimized bit allocation are needed in the encoder to achieve a high coding efficiency.

The objective of the bit allocation algorithm in a parametric audio encoder is to determine, for each frame, how many and which signal components are to be conveyed in the bit stream. Furthermore, for the all-pole spectral model for harmonic tone and noise components, the number of LAR parameters that are to be conveyed in the bit stream needs to be determined.

The number of bits required per frame in order to achieve an approximately constant perceived audio quality can vary substantially over time for a given audio signal. This effect is related to the notion of the time-variant perceptual entropy (PE) of an audio signal [55], which is a conceptual model for the minimum amount of information necessary to convey a signal without impairment of the perceived quality. In order to accommodate such a time variant bit rate requirement while at the same time achieve efficient transmission with no or only a few unused bits over a constant bit rate channel, a common approach is to introduce a bit reservoir (i.e., a bit buffer). This bit reservoir allows to smooth the peaky bit rate demand related to the signal's time-variant PE at the cost of an increased latency (i.e., end-to-end delay) of the complete coding and transmission system. In order to control utilization of the bits in the reservoir, a bit rate control mechanism is needed for the bit allocation process. Also in situations where a time-varying bit rate is allowed, such a bit rate control mechanism is required.

The mechanisms for bit allocation and bit rate control are closely related to the rate distortion aspects of signal decomposition discussed in Section 3.4.1. Given the signal components and their parameters for the current frame as derived by the signal decom-

position and parameter estimation techniques described in Chapter 3, there are basically three degrees of freedom during bit allocation that affect the number of bits required to encode the current frame. These are the two orders of the all-pole models describing the spectral envelope of harmonic tone and noise components, i.e., the number of transmitted LAR coefficients, and, as third degree, the number of individual sinusoidal components that are transmitted in the current frame.

In the following, the empirically designed algorithms for bit allocation and bit rate control employed by the quality-optimized HILN encoder outlined in Section 3.4.3.1 will be presented. Different from the classic SMR-based definition of PE [55] applicable in case of transform coding, the notion of PE is more complex in the context of parametric coding systems. In order to assess the contribution of the harmonic tone and noise component to the PE of the current frame, helper parameters are calculated that describe the relative contribution of the respective component to the total energy in the current frame. If such a helper parameter has a low value, then the maximum number of bits that can be allocated to the corresponding component is significantly reduced. This is achieved by reducing the number of transmitted LAR coefficients for the spectral envelope model of the respective component accordingly. For the noise component approximately 20% to 70% of the bits available for coded parameters in the current frame can be used, depending on the value of said helper parameter. For the harmonic tone component, approximately 50% to 80% of the bits available for coded parameters can be used, again depending on the value of the corresponding helper parameter. In case a harmonic tone or noise component is not present in the current frame, obviously no bits are spent for that component.

After bit allocation for the harmonic tone and noise component, in principle all remaining bits available in the current frame can be used for the parameters of individual sinusoidal components. However, care must be taken in order to utilize the bit reservoir in a meaningful manner, i.e., to save bits for future use in a “difficult-to-code” frame in case the current frame has a PE that is below average. Here, a quite simple approach is pursued that postulates that a frame has an above-average PE if a large portion of the signal components in the current frame are new components, and a below-average PE if most components are continued from the previous frame. To implement this approach, the percentage of new components amongst the 10 perceptually most relevant individual sinusoidal components in the current frame (i.e., the first 10 entries in the ordered list of components generated during signal decomposition) is determined. Assuming that the current frame would use the number of bits per frame available at the desired target bit rate, the number of bits remaining for individual sinusoidal components is calculated, given the actual number of bits used for the harmonic tone and noise components in the current frame. Now this number of remaining bits is multiplied with a factor in the range of 0.8 (only continued components) to 1.2 (only new components) to determine the actual number of bits available for individual sinusoids. Furthermore, it is always ensured that at least 30% of the bits available per frame at the target bit rate can be allocated to individual sinusoids. The actual number of transmitted individual sinusoidal components is determined in a bit allocation loop, where this number is increased until a further iteration

step would exceed the available bit budget. In this loop, the sinusoidal components are added in the order of perceptual relevance, as determined during signal decomposition.

Finally, the bit rate control algorithm ensures that the limits of the bit reservoir are never exceeded. For frames with almost no components, it thus can be necessary to insert padding bits in order to achieve the desired target bit rate if the bit reservoir is already full and no additional bits can be saved for future use. The size of the bit reservoir is, like the target bit rate, a configuration parameter of the encoder that can be set according to the requirements of the intended application.

Figure 4.20 shows in panel (a) an example of the actual bit allocation for the different elements in the bit stream for a short segment of an audio signal encoded with HILN at a target bit rate of 6 kbit/s, which corresponds to a average of 192 bit/frame for the typical frame length of $T_\mathrm{f} = 32$ ms used here.

Table 4.7 provides detailed bit allocation statistics for two typical target bit rates, 6 and 16 kbit/s, measured for 39 audio items. The noise component, which is present in almost all frames, requires an average bit rate in the range of 1.0 to 1.5 kbit/s, depending upon the target bit rate. The harmonic tone component requires an average bit rate of less than 1 kbit/s because it is only present in a minority of the frames. However, if present, it utilizes typically 2 to 4 kbit/s and can use more than 60% of the available bits in a frame at the lower target bit rate of 6 kbit/s. In average 50% to 70% of the available bit rate is used for the parameters of individual sinusoidal trajectories. Approximately $1/3$ of these bits are used for continued trajectories and $2/3$ are used for new ("born") trajectories. Header data and optional temporal amplitude envelope parameter finally account for approximately 17% of the available bit rate. Table 4.8 provides information about the average and maximum all-pole model order (i.e., number of transmitted LAR coefficients) for the harmonic tone and noise components. Table 4.9 provides information about the average and maximum number of sinusoidal components per frame, considering the partials of a harmonic tone as well as new and continued individual sinusoidal trajectories.

The measured distribution of the bit allocation for the different bit stream elements, of the all-pole model order, and of the number of sinusoidal components is shown in Figure 4.13 and Figure 4.14 for the target bit rates of 6 and 16 kbit/s, respectively. Panel (d) of Figure 4.13 shows that at a target bit rate of 6 kbit/s typically 10 to 20 simultaneous individual sinusoidal trajectories can be present. The peaky distribution in panels (b) and (f) of Figures 4.13 and 4.14 for the number of LAR parameters and the number of partials in a harmonic tone is caused by the mechanism used to code this side information. Specifically, the number of LAR parameters is conveyed as an index into a table with 16 different entries ranging from 1 to 25 (for noise components) or from 2 to 25 (for harmonic tones), and the number of partial tones is conveyed as an index into a table with 32 different entries ranging from 3 to 250. These entries are unequally spaced, with increasing steps towards higher numbers, similar to a companding quantizer.

Information about the utilization of the bit reservoir cannot be derived from the distribution of the total number of bits per frame shown in panel (a) of Figures 4.13 and 4.14. Therefore, the distribution of the number of bits available in bit reservoir for HILN bit

Bit stream element	6 kbit/s (192 bit/frame)			16 kbit/s (512 bit/frame)		
	Min	Max	Mean	Min	Max	Mean
hdr	14	37	24.2 (12.9%)	15	88	53.7 (11.2%)
t-env	0	34	9.2 (4.9%)	0	85	24.1 (5.0%)
harm	0	119	23.2 (12.4%)	0	154	21.6 (4.5%)
noise	0	79	34.0 (18.2%)	0	102	44.5 (9.3%)
indi (new+cont)	0	208	96.4 (51.5%)	0	643	334.5 (69.9%)
new	*0*	*208*	*67.7 (36.2%)*	*0*	*643*	*217.3 (45.4%)*
cont	*0*	*119*	*28.7 (15.3%)*	*0*	*336*	*117.2 (24.5%)*
total	14	290	187.0 (100.0%)	15	784	478.4 (100.0%)

Table 4.7: Bit allocation statistics in bit/frame for HILN bit streams encoded at 6 and 16 kbit/s with a frame length of $T_\mathrm{f} = 32$ ms and a bit reservoir size of 384 bit and 1024 bit, respectively, measured for 39 audio items.

Bit stream element	6 kbit/s			16 kbit/s		
	Max	Mean all frames	Mean if present (% of frames)	Max	Mean all frames	Mean if present (% of frames)
harm	25	3.7	11.9 (31.4%)	25	3.7	13.3 (27.7%)
noise	13	11.3	11.9 (95.0%)	19	16.6	17.5 (95.0%)

Table 4.8: Statistics for noise and harmonic tone all-pole model order for HILN bit streams encoded at 6 and 16 kbit/s with a frame length of $T_\mathrm{f} = 32$ ms, measured for 39 audio items.

Bit stream element	6 kbit/s		16 kbit/s	
	Max	Mean	Max	Mean
harm	43	4.3	43	3.9
indi (new+cont)	23	10.2	73	38.8
new	*21*	*5.9*	*73*	*21.4*
cont	*21*	*4.3*	*64*	*17.3*
total	50	14.5	87	42.6

Table 4.9: Statistics for number of sinusoids per frame (harmonic tone partials as well as new and continued individual sinusoids) for HILN bit streams encoded at 6 and 16 kbit/s with a frame length of $T_\mathrm{f} = 32$ ms, measured for 39 audio items.

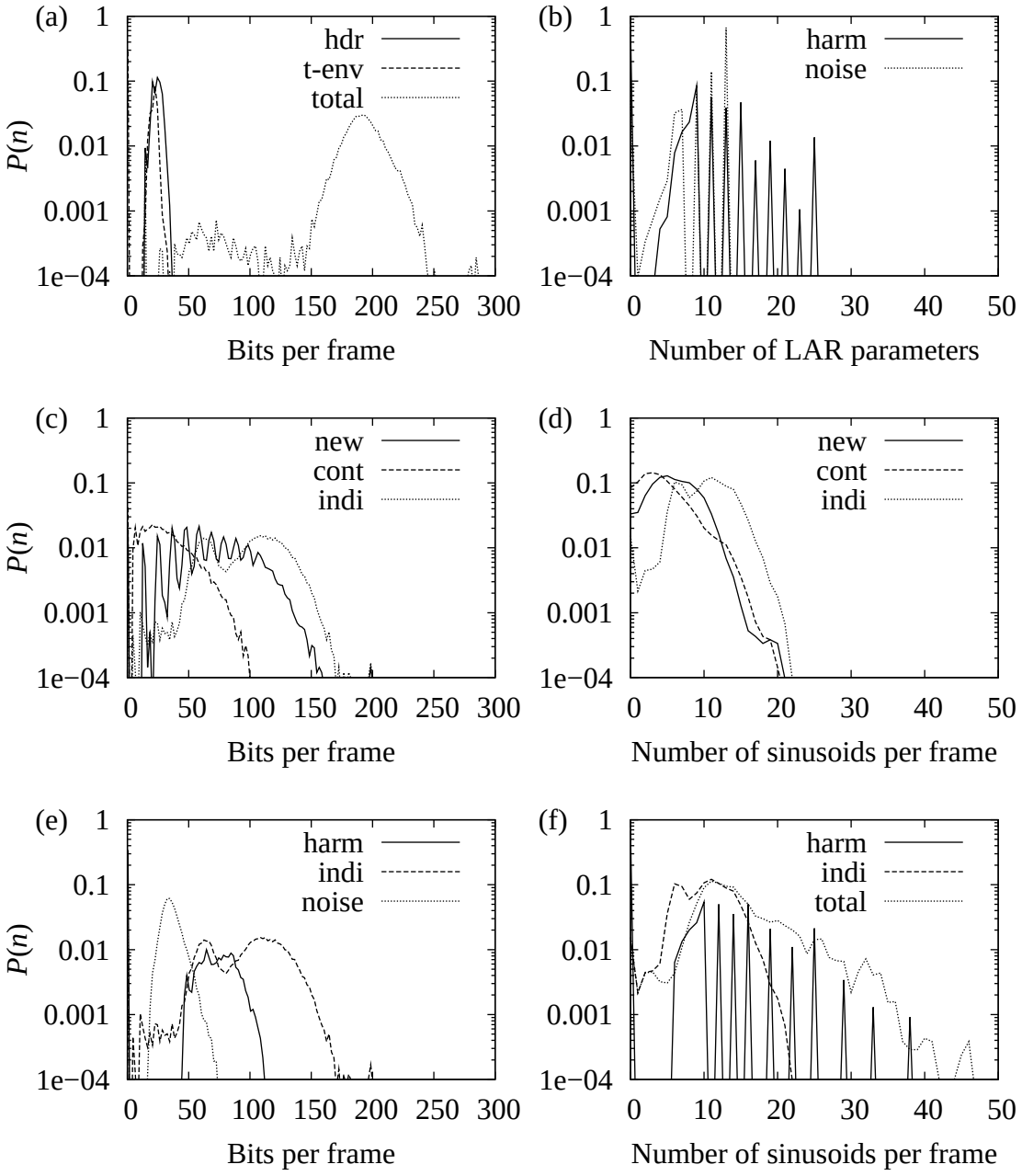

Figure 4.13: Probability distribution of bit allocation for different bit stream elements, of all-pole model order, and of number of individual sinusoids and harmonic tone partials for HILN bit streams encoded at 6 kbit/s with a frame length of $T_f = 32$ ms, measured for 39 audio items.

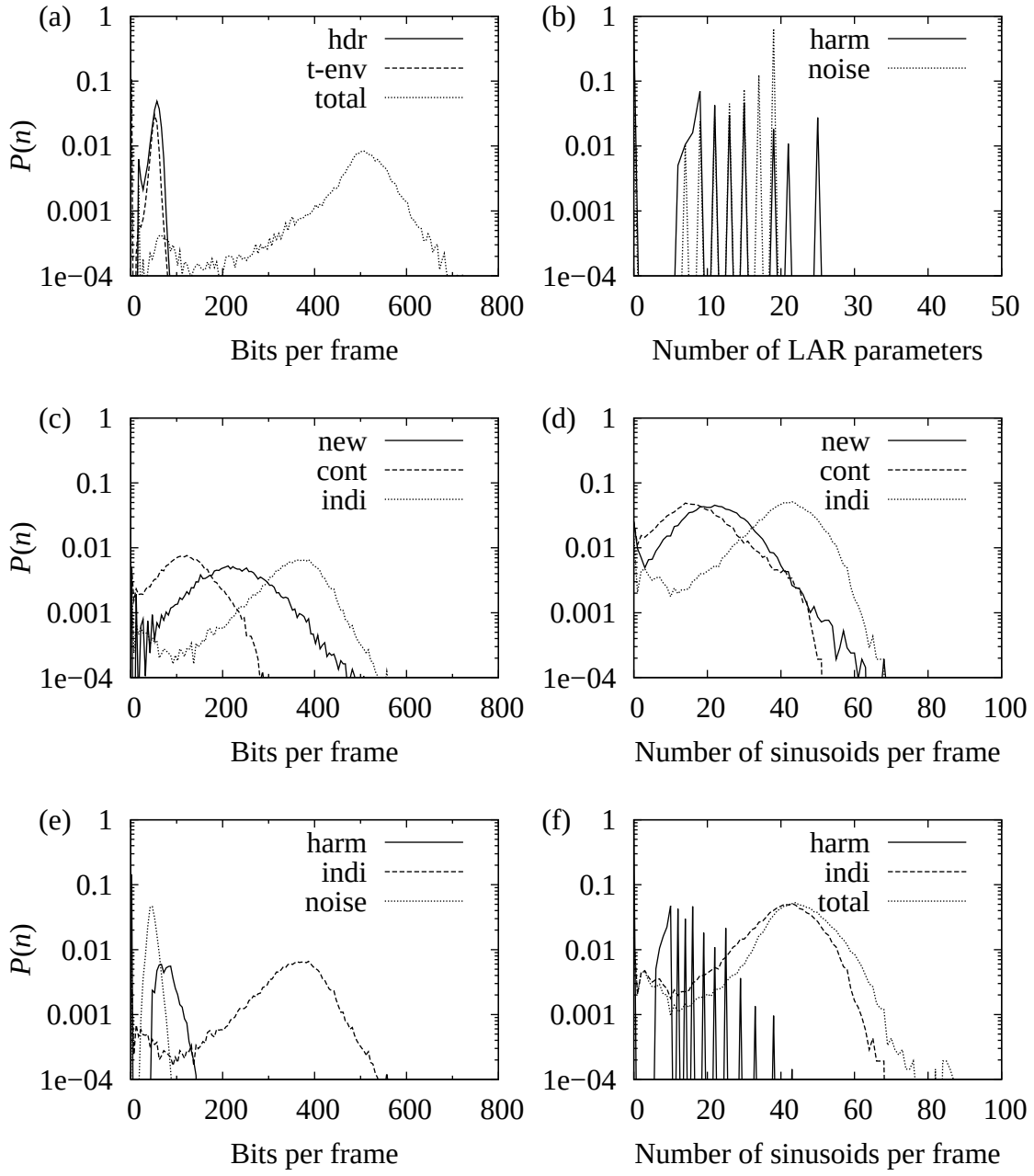

Figure 4.14: Probability distribution of bit allocation for different bit stream elements, of all-pole model order, and of number of individual sinusoids and harmonic tone partials for HILN bit streams encoded at 16 kbit/s with a frame length of $T_\mathrm{f} = 32$ ms, measured for 39 audio items.

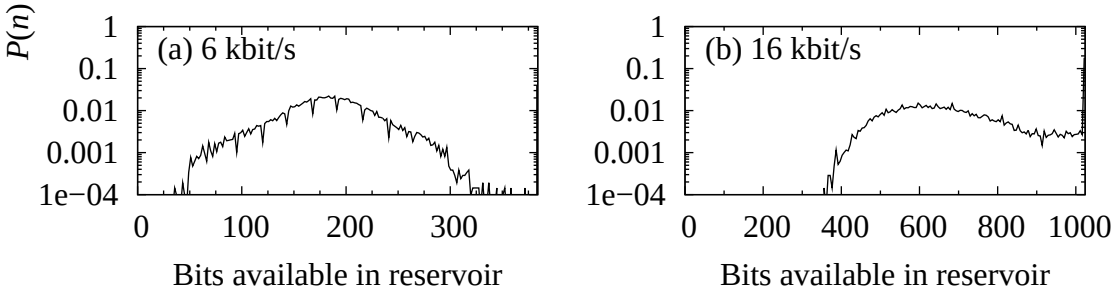

Figure 4.15: Probability distribution of the number of bits available in bit reservoir for HILN bit streams encoded at 6 and 16 kbit/s and a bit reservoir size of 384 bit and 1024 bit, respectively, measured for 39 audio items.

streams encoded at 6 and 16 kbit/s is shown in Figure 4.15. For the bit streams considered here, the size was chosen to be twice the size of a bit stream frame at the target bit rate, i.e., 384 bit for 6 kbit/s and 1024 bit for 16 kbit/s. This causes an additional transmission delay corresponding to 2 frames, i.e., 64 ms. In average there were 191 bit or 731 bit available in the reservoir at the target rated of 6 or 16 kbit/s. The distribution shown in the panel (a) of Figure 4.15 indicates that the bit reservoir control algorithm makes good use of the this reservoir when the encoder is operated at a target rate of 6kbit/s. For higher target bit rates, the performance of the bit reservoir control is less critical. Informal listening for bit streams encoded at a target rate of 6 kbit/s indicates that the availability of a bit reservoir as discussed above avoids audible artifacts that are present if the bit reservoir is completely disabled. These artifacts occur at signal segments with high PE, for example at the onset of complex sounds, and are related to a significant drop in the number of transmitted individual sinusoidal components that occurs in such situations due to the higher bit rate demand of new components compared to continued components.

4.2 Parameter Decoding and Signal Synthesis

The signal synthesis module in the decoder is used to reconstruct the component signals from the parameters that were decoded from the bit stream. By combining these signals, the final decoded audio signal is obtained. This section starts with a description of the parameter decoding process. Then, different signal synthesis techniques and their implementation aspects are discussed and the details and performance of a fast HILN decoder implementation are presented.

4.2.1 Decoding of Model Parameters

The parameter decoding process comprises entropy decoding and parameter dequantization, i.e., the two steps necessary to reconstruct the parameters of the signal components from the information conveyed in the bit stream as shown in the lower part (b) of Figure 2.1. Entropy decoding implements the processes that complement the entropy coding techniques described in Section 4.1.2. Parameter dequantization complements the quantization techniques described in Section 4.1.1, and the corresponding quantized values after dequantization were already given there. Also the prediction process needed to decode the LARs, as shown in Figure 4.1, was already described in Section 4.1.1.2.

In addition to the unambiguous steps for parameter decoding described above, two further aspects need to be considered when reconstructing the complete set of parameters of the hybrid source model required for signal synthesis. These aspects are referred to as *parameter substitution* in lower part (b) of Figure 2.1.

The first aspect is related to the problem that a sinusoidal trajectory could be encoded as a partial of a harmonic tone in one frame and as an individual sinusoid in an adjacent frame, as explained in Section 3.2.3.2. In this case, it is desirable to synthesize the underlying sinusoidal trajectory with a smoothly continuing phase in order to avoid artifacts in the decoded signal during the transition between the two frames. To address this problem, the parameters of all sinusoidal components that were received by the decoder for the current frame are collected in a single merged list. Hence this list includes all individual sinusoidal trajectories that are present in the current frame as well as all partials of a harmonic tone, in case that such a tone is present in the current frame. For each sinusoidal component in this list there is also a pointer that indicates the corresponding entry in the list of the previous frame in case a sinusoidal component is continued from the previous frame. For components that are new in the current frame, this pointer is "null." For both individual sinusoids and harmonic tones, the bit stream explicitly indicates whether these components are continued from the previous frame or are new in the current frame, since this also determines whether the amplitude and frequency parameters were coded differentially with respect to the previous frame or not. In case of a harmonic tone continued from the previous frame, these continuation pointers are set for all partials that exist in both the previous and the current frame (note that the number of partials is transmitted in the bit stream and can vary from frame to frame).

Given these lists of all sinusoidal components for the current and the previous frame, additional continuation pointers can be added which "connect" components in the current frame that have no predecessor with components in the previous frame that have no successor. To find suitable additional connections, the quality measure $q_{k,i}$ for the similarity of frequencies and amplitudes defined in Equation (3.45) is utilized. For each component i in the current frame, the best possible predecessor k is determined, i.e., having the highest quality $q_{k,i}$. If it lies within the maximum permitted frequency ratio $r_{f,\max} = 1.05$ and amplitude ratio $r_{a,\max} = 4$, this connection is added by setting the continuation pointer accordingly. In normal operation, such additional connections are however only permit-

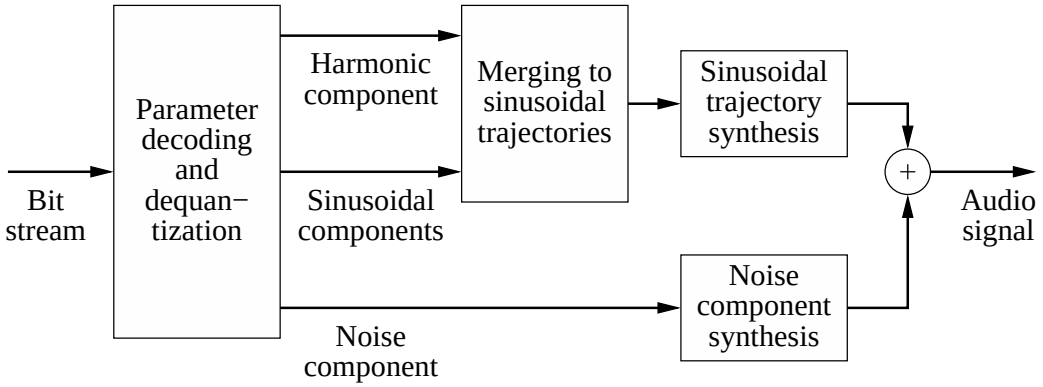

Figure 4.16: General block diagram of the HILN decoder.

ted between individual sinusoids in one frame and partials in the other frame, or between partials in both frames. The latter enables to handle e.g. octave jumps of the fundamental frequency smoothly. Additional connections between individual sinusoids in both frames are only permitted in a special operation mode that is signaled in the bit stream and of interest primarily in case of error-prone transmission channels. In a third special operation mode, no such additional connections are introduced at all.

The second aspect is related to the fact that in typical operation no information about the start phase of new sinusoidal components is transmitted in the bit stream. However, since start phase values are required for the subsequent signal synthesis, the unavailable original start phase parameters are substituted by random values in the decoder, which are taken from a random number generator. These random start phases have a uniform distribution over the full range $0 \leq \varphi < 2\pi$. The use of a fixed start phase (instead of a random one) would typically result in a less naturally sounding signal, in particular for harmonic tones with many partials.

4.2.2 Synthesis Techniques and Implementation Aspects

The general structure of a parametric audio decoder based on the hybrid source model described in Section 3.1 is shown in Figure 4.16. It can be seen as the counterpart of the corresponding HILN encoder structure shown in Figure 3.17. The first two blocks in Figure 4.16 refer to the parameter dequantization and decoding process and to the algorithm that merges all sinusoidal components in a frame into a single list of sinusoidal trajectories that are to be synthesized, as described in Section 4.2.1. In this way, the sinusoidal trajectory synthesis block handles both the individual sinusoidal components and the partials of a harmonic tone. Finally, the noise component is synthesized by a separate synthesis block, and the synthesized noise and sinusoidal signals are added to obtain the decoded audio signal. If the parameters of an additional temporal amplitude envelope are conveyed in the bit stream, the corresponding envelope is reconstructed and applied dur-

ing synthesis of the respective components as indicated in the bit stream. Hence there is no separate block needed for the synthesis of transient components described by means of such a temporal amplitude envelope. The synthesis processes for sinusoidal trajectories and noise components will be discussed in more detail in the following two sections.

4.2.2.1 Synthesis of Sinusoidal Trajectories

The signal of each sinusoidal trajectory can be synthesized by means of a sine wave generator according to Equation (3.1). For trajectories continuing from one frame to the next, phase continuity is maintained and the time-variant control parameters for amplitude and frequency are derived by linear interpolation. At the start or end of a trajectory, the left or right half of a Hann window with 50% overlap, respectively, is used instead of linear amplitude interpolation in order to achieve a smooth fade-in or fade-out of the trajectory. The sinusoid's frequency remains constant during this fade-in or fade-out.

When an additional temporal amplitude envelope with steep onset or steep end is applied to a sinusoidal trajectory, it is questionable whether this envelope should be combined with the linear amplitude interpolation between frames or the 50% overlap windows for fade-in or fade-out. Such a combination can distort the shape of the resulting signal envelope. In case of a short transient impulse, this combination can also affect the resulting amplitude of the impulse. To avoid these problems, an almost rectangular low overlap window $w_1^2(t)$, Equation (3.18), with $\lambda = 1/8$ is used instead of the normal 50% overlap window with $\lambda = 1$. Similarly, the amplitude parameter interpolation in case of a continued trajectory is now done within $1/8$ of a frame, while the old and new value are kept constant before and after this shortened transition, respectively. This shortened fade-in, interpolation, or fade-out is used in case the additional amplitude envelope has an attack or decay with a steep slope $r_{\text{atk}} \geq 5$ or $r_{\text{dec}} \geq 5$ within the synthesized signal segment. The complete definition of these rules is given in Subclause 7.5.1.4.3.4 of the MPEG-4 Audio standard [46].

The synthesis of sinusoidal trajectories is the computationally most complex part of a parametric audio decoder. Hence, it is desirable to use efficient algorithms for sine wave generation. Because sinusoidal components are highly localized in the frequency domain, a frequency-domain approach can be of interest [28], [33, Section 2.5] for a computationally efficient synthesis of many simultaneous sinusoids. However, for the HILN decoder considered here, direct synthesis in the time-domain was found to be more appropriate. The simplest time-domain implementation of a sine wave generator is directly based on Equation (3.1), using a floating point hardware or library implementation of the transcendental $\sin()$ function on the CPU of the target platform. Alternatively, e.g. a table look-up combined with linear interpolation can be used to approximate the $\sin()$ function.

A computational very efficient algorithms for synthesis of sinusoidal trajectories in an HILN decoder can be found in [73]. There, a sine wave is generated by means of an actual oscillator, e.g. a second order real-valued all-pole IIR filter with a pole pair located on the unity circle [40]. Furthermore, optimizations are presented that take into account

the internal data pipeline behavior of common workstation CPUs.

4.2.2.2 Synthesis of Noise Components

The all-pole model used to describe the spectral envelope of a noise component can be used directly in a simple yet efficient approach to synthesize the noise component in the decoder. For this, a white noise signal obtained from a random number generator is filtered with the IIR all-pole filter defined by the spectral envelope parameters in order to shape the spectrum of the noise signal accordingly. The amplitude of the filtered noise is adjusted according to the parameter a_n. Smooth transitions between frames are achieved by a windowed overlap/add scheme, Equation (3.17), using a low overlap window $w_1(t)$ with $\lambda = 1/4$, as described in Section 3.1.4. If indicated in the bit stream, an additional temporal amplitude envelope is applied to the noise component of a frame prior to the overlap-add process.

From a perceptual point of view, there are only weak requirements for the random number generator producing the white noise filter input signal. A simple pseudo-random sequence with uniform distribution in the interval $[-1,1]$ and without audible periodicities has shown to be sufficient for this application.

4.2.3 Complexity of HILN Decoder Implementations

The various approaches for the synthesis of sinusoidal trajectories discussed in Section 4.2.2.1 all produce the same audio signal (if implemented correctly and with appropriate numerical accuracy) and only differ in their computational and implementation complexity. However, it is of interest to note that the synthesis of sinusoidal trajectories and noise components in an HILN decoder is, in general, not deterministic. This is due to the random start phases typically used for sinusoidal trajectories, and because of a random noise generator used as basis for noise component synthesis. This non-deterministic behavior needs to be taken in to account when testing or comparing the output signals of parametric audio decoders.

It is desirable that an HILN decoder has very low computational complexity, which can be achieved by the efficient synthesis techniques outlined in Section 4.2.2.1. In order to obtain an approximate measure for the computational complexity required to decode a bit stream, the MPEG-4 Audio standard [46] introduced the concept of *processor complexity units* (PCU), measured in million operations per second (MOPS). For an HILN decoder, the computational complexity depends primarily on the number n_s of sinusoidal components that are synthesized simultaneously, and on the sampling rate f_s of the output audio signal. According to Subclause 1.5.2.2 of the MPEG-4 Audio standard [46], the PCU value is calculated as

$$\mathrm{PCU} = (1 + 0.15 n_s) f_s / 16\,\mathrm{kHz}, \tag{4.5}$$

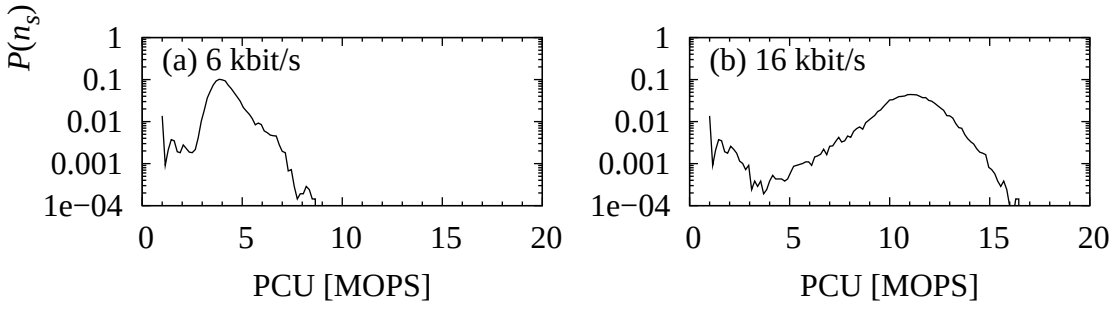

Figure 4.17: Probability distribution of the computational complexity (PCU value) derived from the number of simultaneous sinusoids n_s synthesized for each frame, measured for HILN bit streams encoded at 6 and 16 kbit/s using 39 audio items.

	6 kbit/s		16 kbit/s	
	Max	**Mean**	**Max**	**Mean**
Estimated complexity				
Simultaneous sinusoids (n_s)	55.00	20.72	111.00	63.61
PCU [MOPS] ($1+0.15n_s$)	9.25	4.10	17.65	10.54
Measured CPU load [MHz]				
Pentium MMX, 200 MHz		19		41
Pentium4 Mobile, 1600 MHz		15		33
Athlon64 4000+, 2400 MHz		7		15

Table 4.10: Computational complexity of HILN decoding estimated as PCU value based on the number of simultaneously synthesized sinusoids n_s, and measured for a complete fast decoder implementation (see [73]) on different workstations for HILN bit streams encoded at 6 and 16 kbit/s with frame length $T_f = 32$ ms and sampling rate $f_s = 16$ kHz using 39 audio items.

where the constant of 1 reflects the complexity of parameter decoding and of the synthesis of the noise component.

Figure 4.17 shows the distribution of the per-frame computational complexity (i.e., the PCU value calculated from n_s for each synthesized frame) for HILN bit streams encoded at 6 and 16 kbit/s, measured using 39 audio items. The observed maximum and mean values for n_s and the PCU are given in the upper part of Table 4.10. It can be seen that the peak complexity (maximum PCU) is approximately twice as high as the average complexity. Such complexity peaks are typically caused by harmonic tone components with a high number of partials.

The lower part of Table 4.10 shows the average CPU load in MHz needed to decode HILN bit streams at 6 and 16 kbit/s for three different workstations. For these measurements, the computationally efficient HILN decoder described above was used [73]. It

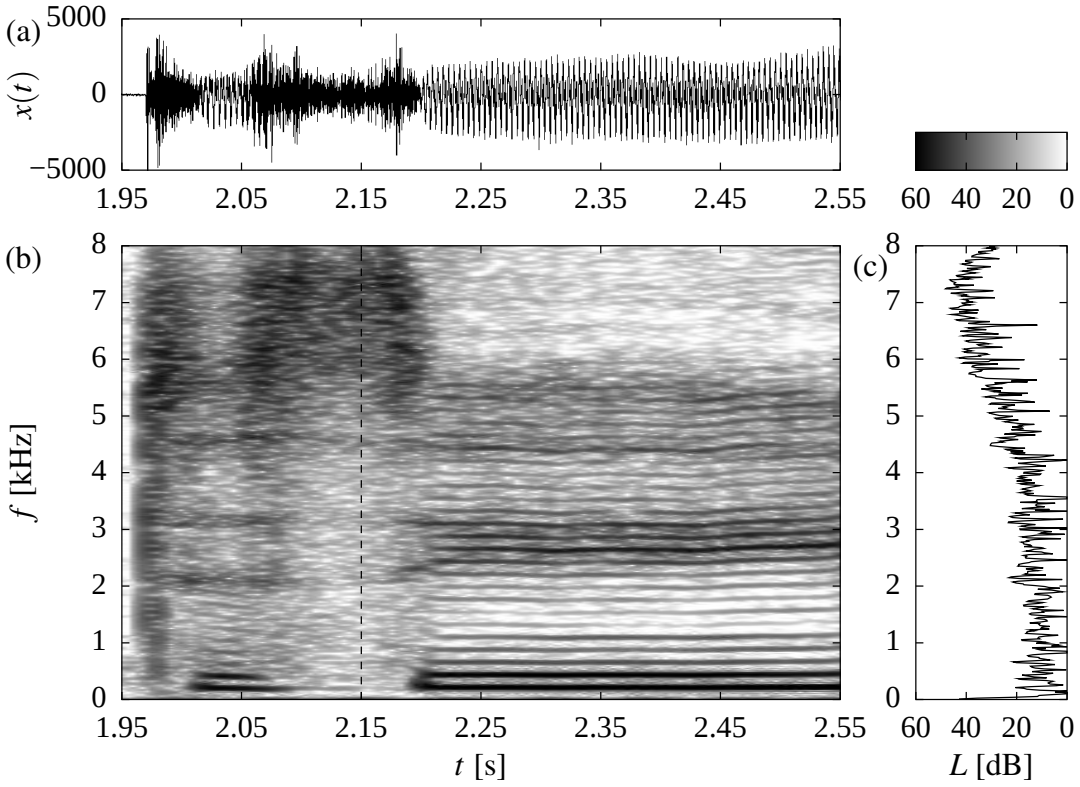

Figure 4.18: Time-domain signal (a) and spectrogram (b) of words "to see" in original speech signal *Suzanne Vega*. The spectrum (c) of the signal at $t = 2.15$ s (as marked by the dashed line in the spectrogram) shows the noise-like sound of unvoiced fricative /s/ in the frequency band from 5 to 8 kHz.

can be seen that decoding of 16 kbit/s bit streams is on average about twice as complex as decoding of 6 kbit/s bit streams. Using computationally efficient sinusoidal synthesis algorithms, the complete decoding and synthesis process for HILN bit streams encoded at a bit rate of 6 to 16 kbit/s accounts for a CPU load of approximately 10 to 20 MHz on today's personal computers. This decoder complexity is comparable to that of most other state-of-the-art audio coding techniques.

4.2.4 Example of HILN Coding and Signal Reconstruction

The reconstruction of an audio signal by means of signal synthesis in an HILN decoder is illustrated using a short segment of a speech signal, the words "to see" in signal *Suzanne Vega*. Figure 4.18 shows the spectrogram (b) of the original speech signal that was sent as input signal (a) to the HILN encoder. Furthermore, the spectrum (c) of this signal at $t = 2.15$ s is given, showing the noise-like sound of unvoiced fricative /s/ in the frequency band from 5 to 8 kHz.

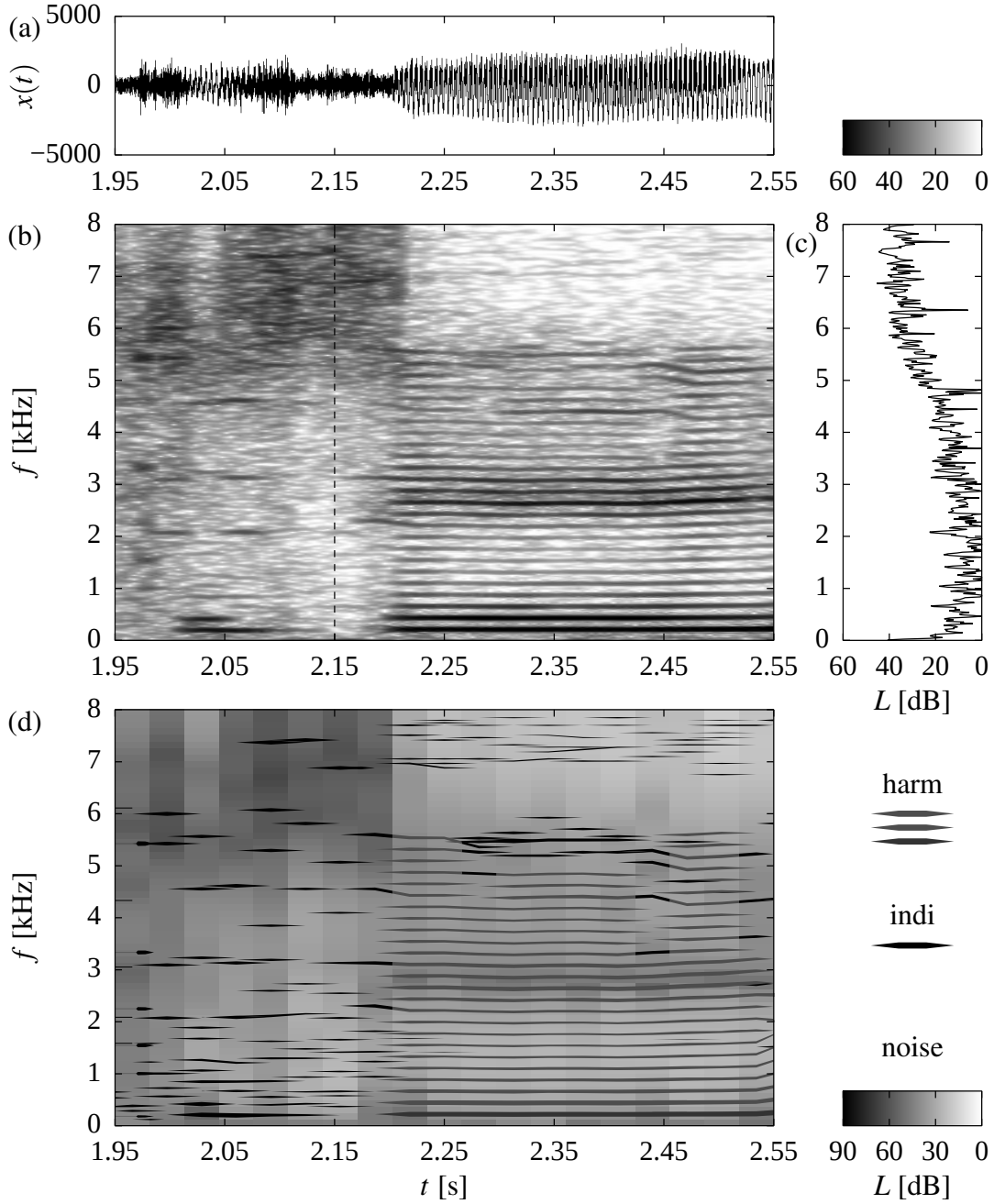

Figure 4.19: Time-domain signal (a) and spectrogram (b) of synthesized decoder output signal for words "to see" in signal *Suzanne Vega* and spectrum (c) of this signal at $t = 2.15$ s. Parametric representation (d) conveyed at 6 kbit/s in an HILN bit stream, where the amplitude of sinusoidal trajectories is represented by the line width (incl. fade-in/fade-out), individual sinusoids are shown black, the fundamental frequency component of a harmonic tone is shown dark gray, and the higher partials middle gray, and where the noise component is shown as spectrogram in background.

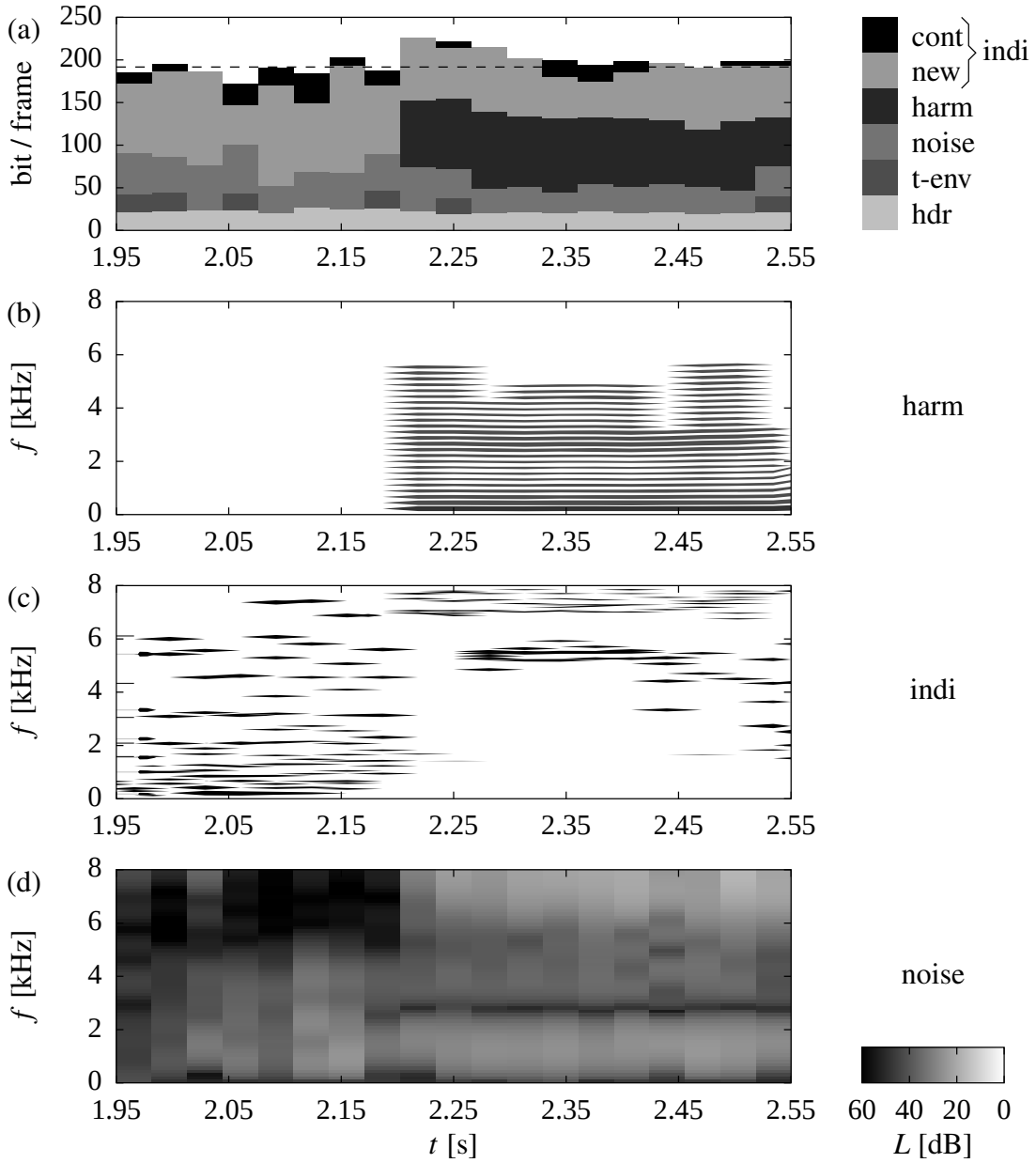

Figure 4.20: Bit allocation (a) for 6 kbit/s HILN bit stream conveying words "to see" in signal *Suzanne Vega* (as shown in Figure 4.19), showing the size of the different bit stream elements conveying the parameters of the signal components. Separated spectrogram-like representation of harmonic tone partials (b), individual sinusoids (c) and noise component (d), as shown together in panel (d) of Figure 4.19.

Figure 4.19 shows in panel (b) the spectrogram of the final output audio signal generated in an HILN decoder based on the model parameters conveyed in the bit stream for a short signal segment encoded at 6 kbit/s. Comparing this spectrogram with the spectrogram of the original signal shown in Figure 4.18, it can be seen that the major features of all tonal and noise-like signal components are maintained fairly good in the decoder output signal.

Panel (d) of Figure 4.19 shows a spectrogram-like graphical representation of the parameters of the signal components that were conveyed in the bit stream. Sinusoidal trajectories are shown as lines, where the line width is proportional to the amplitude of the component on a logarithmic scale. The line width also reflects the effect of an additional temporal amplitude envelope, which e.g. can be seen at 1.97 s. For simplicity, the exact shape of the windows used during fade-in and fade-out at the start and end of a trajectory are not shown here, and a simple triangle-like visualization is used instead. Trajectories of individual sinusoids are shown in black, while the fundamental frequency of a harmonic tone is shown in dark gray, and the higher partials in middle gray. The spectral envelope of the noise component is shown as spectrogram in background. The effect of the overlap-add windows used for noise synthesis is not reflected in this graphical representation, such that the vertical-bar like structure of the noise component spectrogram clearly indicates the frame structure of the parametric description with a frame length of $T_\mathrm{f} = 32$ ms.

Using the same graphical representation, the different components of the decoder output signal are shown separately in Figure 4.20. Here, panel (b) shows the fundamental and higher partials of the harmonic tone component, panel (c) shows the individual sinusoidal components, and panel (d) shows the noise components. It should be noted that the additional connections between individual sinusoids and harmonic tone partials that were added during parameter decoding (as described in Section 4.2.1) are only shown in Figure 4.19, but not in Figure 4.20.

4.3 Extensions for Additional Functionalities

The framework of a parametric audio coding system can easily be extended to include additional functionalities. The following sections discuss the extensions necessary to provide time-scaling and pitch-shifting in the decoder, to enable bit rate scalability in the coding system, and to improve the robustness in case of transmission errors.

4.3.1 Time-Scaling and Pitch-Shifting

The parametric representation of the audio signal that is available in the decoder prior to the synthesis block allows to apply signal modifications directly in the parametric domain. In particular, independent time-scaling and pitch-shifting can be performed in a computationally very efficient way. Both are signal modifications that are difficult and complex to implement for signals represented as time-domain waveform.

Time-scaling is achieved by changing the synthesized frame length T_f without changing the component parameters, thus modifying the playback speed of the signal without affecting the pitch of the signal. In principle, also time-reversed playback or continuous playback of the instantaneous sound of the signal at a single point of time is possible. In these cases, however, the parametric signal representation needs to be re-sampled prior to synthesis such that the actual signal synthesis is carried out for frames of typical length.

Pitch-shifting is achieved by multiplying all frequency parameters of the sinusoidal trajectories by the desired factor prior to synthesis, thus altering the pitch of the signal without affecting the playback speed. In this way, pitch and formants of a speech signal are modified in the same manner. If the pitch is shifted up, care must be taken in order to avoid aliasing for sinusoidal trajectories whose frequencies now exceed $f_\mathrm{s}/2$ of the synthesized signal. This is achieved by muting trajectories during synthesis if their instantaneous frequency is equal to or higher than $f_\mathrm{s}/2$.

For noise component synthesis, however, pitch-shifting is less straight-forward, since the all-pole filter parameterization of the spectral envelope is based on the sampling rate of the original audio signal, i.e., $f_{\mathrm{s,SEM}} = f_\mathrm{s}$. In an HILN decoder, pitch-shifting for noise signals is achieved by first synthesizing the noise component at the original sampling rate and then re-sampling the synthesized waveform according to the desired pitch-shift factor.

4.3.2 Bit Rate Scalability

In several application scenarios, the bit rate available on a transmission channel is unknown at the time of encoding. In these circumstances, it is desirable that a subset of the encoded original bit stream with a lower bit rate not exceeding the actual channel capacity can be extracted easily. This functionality is referred to as *bit rate scalability* or as *hierarchical embedded coding* and is usually realized by a layered structure of the original bit stream, comprising a base layer and one or more additional enhancement layers.

The base layer bit stream of the bit rate scalable HILN system has the same format as for a non-scalable configuration. To simplify system design, the enhancement layers can only carry the parameters of additional individual sinusoidal components. In this way, the perceptually most salient components, including noise components and harmonic tones (if present in the signal), are always conveyed in the base layer. This ensures a meaningful minimum perceptual quality in case that only the base layer is available to the decoder.

The structure of one frame of the enhancement layer bit stream is similar to that of a base layer frame shown in Figure 4.12, but without the parameters describing temporal amplitude envelopes, noise components, or harmonic tones. The frame header indicates the number of additional sinusoids conveyed in this enhancement layer. Sinusoidal trajectories can be continued from the base layer to the enhancement layer, but not the other way around, since that would make the base layer dependent upon the enhancement layer. The enhancement layer format permits to stack several enhancement layers on top of each other, thus enabling a finer granularity of the bit rate scalability.

Bit allocation in an HILN encoder providing bit rate scalability is quite similar to the

non-scalable case described in Section 4.1.4. Bit allocation for the base layer is done in the same way as in a non-scalable system, with the only exception that temporal amplitude envelope data must already be conveyed in the base layer even if only sinusoids in an enhancement layer make use of this data. Bit allocation for enhancement layers adds further sinusoidal components (in addition to those conveyed in the base layer) in the order of perceptual relevance that was determined during signal decomposition.

Rate distortion optimization in a bit rate scalable system is more difficult than for the non-scalable case described in Section 3.4.1 and requires compromises. An example for this is extraction of the noise component, which in the present approach depends on the number of extracted sinusoidal components. And this, in turn, depends on the target bit rate. As result, the noise component in the base layer of a scalable configuration is somewhat weaker (i.e., has less energy) than in a non-scalable bit stream with the same bit rate as the base layer. In general, compromises like this are part of the cost that is associated with enabling the functionality of bit rate scalability.

4.3.3 Error Robustness

In a scenario where the bit stream conveying the coded audio signal is transmitted over error-prone transmission channels, good error robustness of the coding system is desirable. Different adaptions and extensions were integrated into the basic HILN parametric coding system in order to achieve this functionality.

- **Unequal Error Protection** It case of channels with bit errors, it is common to employ error correcting codes that add redundancy in order to enable forward error correction (FEC), i.e., correction of transmission errors in the decoder. However, the impact of a bit error on the decoded audio signal is not the same for the different bits in a bit stream frame. Therefore, it is advantageous to assign the bits in a frame to appropriate error-sensitivity categories (ESC). This allows for unequal error protection (UEP), i.e., to add more redundancy for better error correction to those categories of bits where errors would have a higher impact on the decoded signal. For the HILN, a total of 5 ESCs are defined, and a more detailed discussion can be found in [108].
- **Encoding for Reduced Error Propagation** The time-differential or predictive coding used for the parameters of continued components is prone to error propagation from frame to frame. If transmission of the bit stream over an error prone channel is expected, the encoder can be configured to take precautions that limit the effect of possible error propagation. The general idea is to restrict the maximum duration of the use of differential coding. A correspondingly configured encoder limits the maximum number of consecutive frames for which a component is continued. When reaching this limit, the encoder is forced to re-start the component again in the next frame using absolute coding of parameters. An alternative approach is to re-start all components periodically every Qth frame, similar to an intra-frame (I-frame) as known from video

coding. In both cases, a reduced risk of error propagation comes at the price of an increased bit rate.

The periodic re-start of sinusoidal components would lead to phase discontinuities for long trajectories if no start phase information is conveyed and a random start phase is used. To avoid this problem and enable phase-continuous synthesis of re-started sinusoids, a special operation mode of the decoder module for additional "connections" between sinusoidal components is available, as described in Section 4.2.1.

- **Error Concealment in Decoder** If a transmission error or the loss of a data packet is detected, the decoder can attempt to mitigate the effect of the error on the decoded signal by means of error concealment techniques. For example, the complete signal or affected components can be muted, or missing parameters can be extrapolated from previous data. In the HILN parametric coding systems, error concealment techniques can be implemented in a simple and efficient manner directly in the parameter domain prior to signal synthesis. A more detailed discussion of HILN error concealment and evaluation of its performance can be found in [108].

5 MPEG-4 HILN Experimental Results

The parametric audio coding system developed in this thesis has been adopted as a part of the MPEG-4 standard, where it is referred to as *Harmonic and Individual Lines plus Noise* (HILN) coding. This chapter provides a short overview of the MPEG-4 Audio standard and the development of HILN as part of this standard. To assess the audio quality achieved by HILN, different listening tests were carried out. Their results are reported here, with focus on the verification test that concluded the standardization process.

5.1 MPEG-4 Audio and HILN Parametric Audio Coding

The MPEG-4 standard, formally referred to as *ISO/IEC 14496 – Coding of audio-visual objects*, was developed by ISO's moving picture experts group (MPEG) [75] and an overview of the standardization process can be found in [94]. MPEG-4 comprises a large set of tools for coding of audio and video objects and for the description of complete scenes composed of such objects. HILN is one of the available audio coding tools, and is defined in the Part 3, *Audio*, of the standard [46].

The objective of the MPEG-4 Audio standardization work was to define a system that enables the efficient representation of complete audio scenes containing one or more natural or synthetic audio objects [107]. In this context, a *natural* object represents an audio signal encoded from an input waveform, whereas a *synthetic* object represents an audio signal described at a higher level, e.g., as musical score or as written text. Furthermore, MPEG-4 Audio distinguishes between *speech* objects and *audio* objects in order to utilize the different characteristics of these classes of signals.

For *natural speech* objects, traditional CELP coding at approximately 4 to 24 kbit/s as well as parametric speech coding by means of harmonic vector excitation coding (HVXC) [88] at approximately 2 to 4 kbit/s is available [89]. For *natural audio* objects, traditional transform coding as well as parametric audio coding by means of HILN is available [38]. The transform coder is based on the MPEG-2 AAC framework [9] and was extended in various ways. For low bit rates in the range of approximately 6 to 16 kbit/s/ch, TwinVQ [52] can be used as an alternative quantization and coding tool. The HILN parametric audio coder is subject of this thesis and can be used for low bit rates up to approximately 16 kbit/s. For *synthetic speech* objects, a text-to-speech interface (TTSI) is defined that allows to control a suitable Text-to-Speech synthesizer as part of an MPEG-4 Audio decoder [123]. For *synthetic audio* objects, the structured audio (SA) tool-set is available [123]. It is basically a computer music language similar to MUSIC V or Csound and comprises both a score language (SASL) and an orchestra language (SAOL) [124]. The final

audio scene presented to the listener can be composed of different audio objects, including possible effects processing by means of the SA tool-set. The storage or transmission multiplex carrying the bit streams of the different audio objects in a scene and the details of the scene description are defined in Part 1, *Systems*, of the standard [43].

5.2 Assessment of Performance of HILN

In order to assess the audio quality achieved by the HILN parametric audio coder, various subjective listening tests were carried out during the development and standardization of the coder. These tests followed the guidelines described in ITU-R recommendation BS.1284 [49]. Two different types of tests can be distinguished.

In the first type, two alternative coders, A and B, are compared to each other. A short test item, typically not longer than 20 s, is presented as sequence R, A, B to the subject, where R is the original signal played as reference, and A and B are the two coded versions. The subject is asked to grade the difference between A and B on a continuous 7-point comparison scale ranging from +3 (A much better than B) over 0 (A the same as B) to -3 (A much worse than B), as shown in Figure 5.1. The mean grade over all subjects is referred to as comparison mean opinion score (CMOS). In addition to the mean grade, also its 95% confidence interval (CI) is reported in order to assess the consistency or variation of the grades given by the subjects.

In the second type of tests, the absolute audio quality of a coder A is assessed. The test item is presented as R, A (or, with one repetition, as R, A, R, A), and the subject is asked to grade the audio quality of A on a continuous 5-point quality scale ranging from 5 (Excellent) to 1 (Bad), as shown in Figure 5.2. The original signal R is to be used as reference by the subjects, and they are instructed to give the grade 5 (Excellent) if A sounds the same as R. The mean grade over all subjects is referred to as mean opinion score (MOS), and also its 95% CI is reported.

For all CMOS and MOS tests reported in the following, mono signals were used and played to the subjects over headphones. The original signals used as reference in the test have the same sampling rate as the input and output signal of the coder under test, and thus have a limited audio bandwidth. All tests were conducted as double-blind tests with a randomized presentation order of the items and coders.

5.2.1 Results from MPEG-4 Core Experiment Phase

In the following, the results of the major listening tests carried out during the development of HILN are presented. This development started with the *analysis/synthesis audio codec* (ASAC) [18]. The ASAC parametric audio coder, which supports only individual sinusoidal trajectories, was one of the proposals submitted to MPEG-4 standardization process in response to the initial *MPEG-4 Call for Proposals* [76]. In the following competitive phase, it was selected to become one of the tools in the MPEG-4 Audio tool-set.

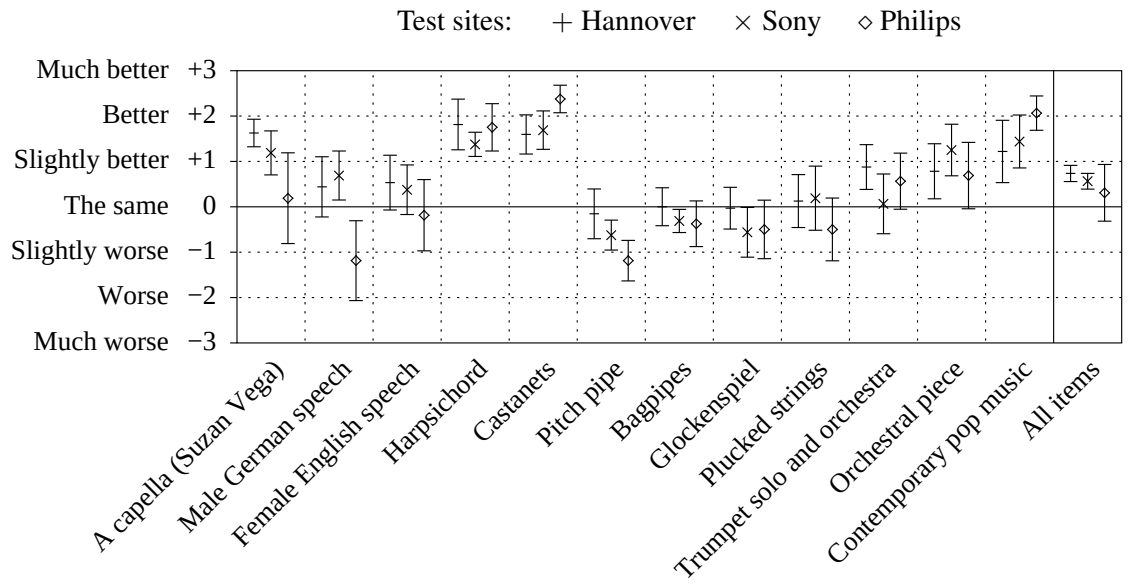

Figure 5.1: Results from the CE on HILN-v1, adding harmonic tone and noise components to the ASAC sinusoidal coder, for three test sites Hannover (8 subjects), Sony (8 subjects), Philips (16 subjects) shown as CMOS with 95% CI. HILN-v1 is compared against ASAC at 6 kbit/s, using original signals with 8 kHz sampling rate and assessing the 12 CE items (after [97]).

Thus, the proposed ASAC system [17] was included in the first draft MPEG-4 Audio verification model [19], [78] in 1996. The development of the MPEG-4 Audio standard then progressed in a collaborative phase based on the core experiment (CE) methodology [79]. The further development of the initial ASAC parametric audio coder was carried out in parallel with the ongoing MPEG-4 standardization process and resulted finally in the HILN system as presented in Chapters 3 and 4.

In a first core experiment [96], the original ASAC system (which supports only individual sinusoids) was extended by harmonic tones and noise components to form a first version of HILN (HILN-v1) [101], [102]. This HILN-v1 coder employs a very simple spectral envelope model (grouping of partials) for the harmonic tone and a simple DCT-based spectral envelope model (i.e., an all-zero model) for the noise components. The results of listening tests that compared ASAC with HILN-v1 are shown in Figure 5.1 (after [97]). Three test sites with a total of 32 subjects participated in this test, and the 12 CE test items listed in Table A.1 in Appendix A were used in this test. Both the old ASAC and the new HILN-v1 coder operated a 6 kbit/s with a sampling rate of 8 kHz, and the CMOS test methodology was used to compare their audio quality. It can be seen from the mean grades and their 95% CIs that the extensions included in HILN-v1 result in a significantly improved quality for 3 to 6 items (depending on the test site) out of the 12 test item. Two test sites also reported a significantly decreased quality for 2 or 3 items. However, the overall improvement of quality is still significant.

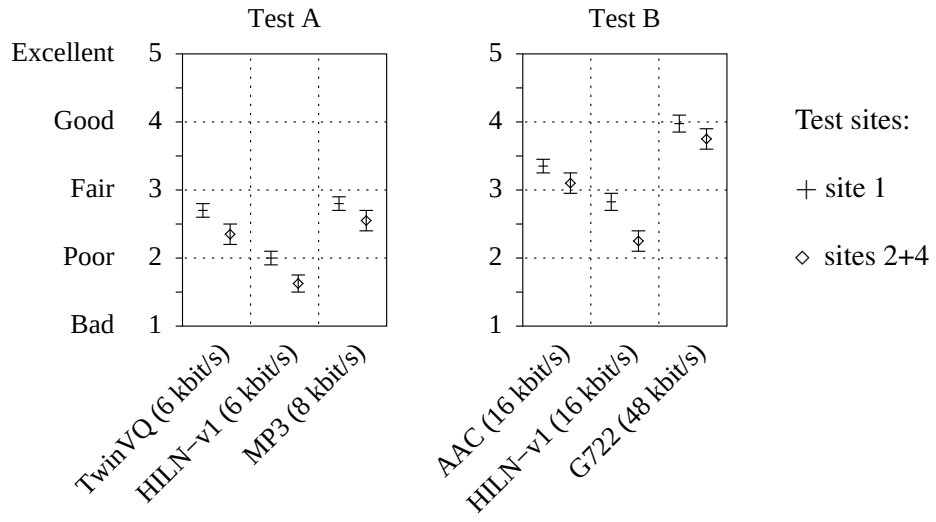

Figure 5.2: Results from the MPEG-4 Audio Version 1 *Audio on Internet* verification tests A (TwinVQ and HILN-v1 at 6 kbit/s) and B (AAC and HILN-v1 at 16 kbit/s) for test site 1 (15 subjects) and test sites 2+4 (21 subjects in total) shown as MOS with 95% CI for the mean grades for 10 items, using original signals with sampling rate of 8 kHz (test A) and 16 kHz (test B). Tests A and B each used their own set of 10 items (after [83]).

The MPEG-4 Audio Version 1 verification test addressing *Audio on Internet* (AOI) applications comprised four parts, referred to as tests A, B, C, and D. The results for the two HILN-related parts, tests A and B, are summarized in Figure 5.2 (after [83]). Test A assessed the performance of the TwinVQ and HILN-v1 at 6 kbit/s, with MP3 at 8 kbit/s as anchor and original signals with 8 kHz sampling rate. Test B assessed the performance of the AAC and HILN-v1 at 16 kbit/s, with G.722 at 48 kbit/s as anchor and originals signals with 16 kHz sampling rate. TwinVQ [52] is the MPEG-4 transform coder intended for bit rates of 6 to 16 kbit/s and was therefore used in test A. AAC [9], on the other hand, is the MPEG-4 transform coder intended for bit rates of 16 kbit/s and above and was therefore used in test B. A total of 36 subjects participated at three test sites. Tests A and B each used their own set of 10 items selected as typical and critical material from a total of 39 test items listed in Table A.2 in Appendix A (see [83] for more details), and the MOS test methodology was used to assess the absolute audio quality of the coders in the test. The mean scores indicate that HILN-v1 performs significantly worse than TwinVQ at 6 kbit/s and significantly worse than AAC at 16 kbit/s. Because of these results, it was decided to exclude HILN-v1 from the first version of the MPEG-4 Audio standard [44] and continue development of HILN for the second version of the standard.

Since the audio quality of HILN-v1 and TwinVQ shows a strong dependency upon the test item characteristics, a further listening test was carried out at two additional test sites for all 39 items coded at 6 kbit/s [98]. The results of this test indicated that the

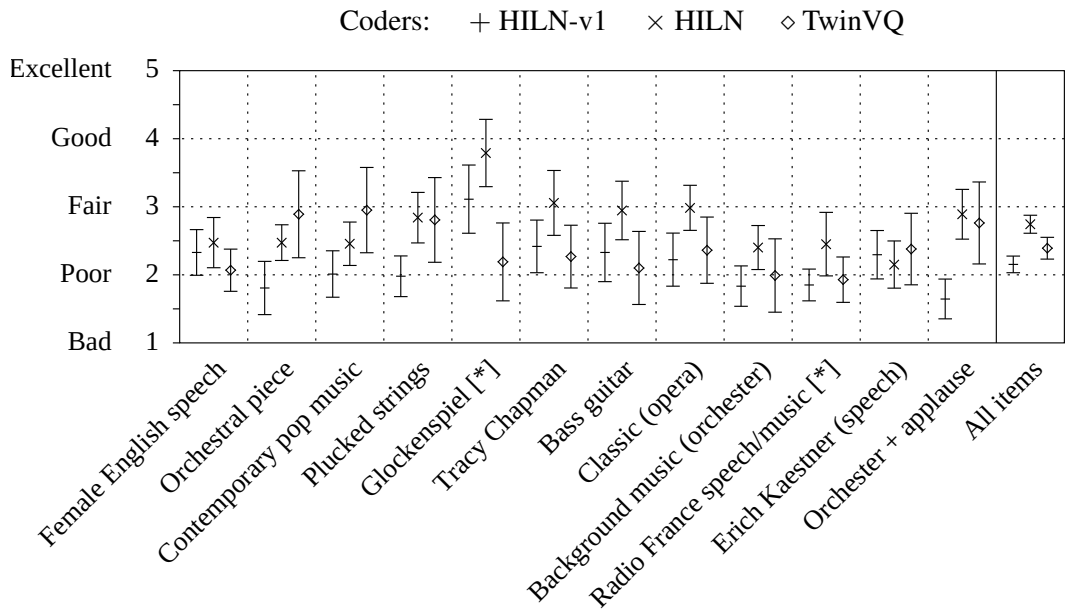

Figure 5.3: Results from the CE on improved quantization, coding, and spectral models for HILN at 6 kbit/s with 9 subjects for HILN-v1, HILN, and TwinVQ shown as MOS with 95% CI, using original signals with 8 kHz sampling rate. The 10 items from AOI test A plus two additional items (marked [*]) were assessed (after [100]).

mean scores for both coders over all 39 items were not significantly different. However, HILN-v1 exhibited a larger variation of the quality over the items than TwinVQ. This explains the inferior mean performance of HILN-v1 when primarily critical test items are used in a listening test.

In a second core experiment [99], the first version of HILN (HILN-v1) was improved further, which resulted in the final HILN coder [106]. In this process, the quantization and coding of the parameters of the individual sinusoids was significantly improved by optimized variable length codes and subdivision coding (SDC) as presented in Section 4.1.2. Furthermore, all-pole spectral envelope models were introduced for the harmonic tone and noise components, using the LAR parameterization and coding as described in Section 4.1.1.2. The results of the listening test from this core experiment are shown in Figure 5.3 (after [100]). In this test, the final HILN coder with improved quantization, coding, and spectral models was compared to HILN-v1 at 6 kbit/s, and TwinVQ was included as anchor. The 10 test items from the AOI test A plus two additional test items were used in this test, with an original sampling rate of 8 kHz. The test was carried out at one test site with 9 subjects, and the MOS test methodology was used. It can be seen that HILN achieves a significantly higher mean audio quality at 6 kbit/s than both HILN-v1 and TwinVQ. For 4 out of the 12 items used in this test, a significant quality improvement of HILN over HILN-v1 was observed, while no item got significantly worse. Comparing HILN with TwinVQ, it can be seen that HILN performs clearly better for single instru-

ment signals, like *glockenspiel* and *bass guitar*, and for mixed speech/music signals, like *Tracy Chapman* (pop music with vocals), *classic* (opera), and *Radio France* (talk with background music). TwinVQ was found to be slightly better than HILN for only two test items with complex orchestral sounds (*orchestral piece* and *contemporary pop music*).

5.2.2 Results of the MPEG-4 Verification Listening Test

For final verification, HILN has been compared to other state-of-the-art audio coding systems by means of a listening test at bit rates of 6 and 16 kbit/s [84]. In addition to non-scalable bit streams at 6 and 16 kbit/s (denoted HILN06 and HILN16), also a scalable HILN bit stream configuration with a 6 kbit/s base layer and a 10 kbit/s enhancement layer was included in this test. The scalable bit streams were decoded at 6 kbit/s (denoted HILN0616BL) and at their total rate of 16 kbit/s (denoted HILN0616EL). The transform coders TwinVQ and AAC were used as anchors for the tests at 6 and 16 kbit/s, respectively, and the original signals for all tests had a sampling rate of 16 kHz. The tests at 6 and 16 kbit/s each used their own set of 7 items selected as typical and critical material from a total of 39 test items as listed in Table A.2 (see [84] for more details). The test was carried out at one test site with 16 subjects, and the MOS test methodology was used. The test results for 6 kbit/s are shown in Figure 5.4. It can be seen that the HILN parametric coder performs comparable to the MPEG-4 transform coder (here: TwinVQ) when the mean score over all items is considered. For single instrument signals (like *glockenspiel* and *percussion*) HILN performs better then TwinVQ, while it performs worse than TwinVQ for complex orchestral sounds (like *orchestra + applause*). For most of the other test items, the performance of HILN and TwinVQ is similar. The test results for 16 kbit/s are shown in Figure 5.5. Also at this bit rate it can be seen that the HILN parametric coder performs comparable to the MPEG-4 transform coder (here: AAC) when looking at the mean score over all items. An analysis of the results for the individual items shows the same tendencies as for 6 kbit/s, namely better performance of HILN for single instrument signals (like *accordion + triangle*) and worse performance for complex orchestral sounds (like *orchestra + applause*). Comparing the audio quality from scalable HILN bit streams (HILN0616) with that from non-scalable HILN bit streams at the same total rates of 6 and 16 kbit/s, only a small reduction in audio quality is observed that is not significant.

Because the HILN coder performed according to expectations in this verification test, it was adopted in the second version of the MPEG-4 Audio standard [45], [103], where it is defined in Subpart 7. Associated conformance test procedures are defined in Part 4 of the MPEG-4 standard [47] and a reference software implementation of the decoder is provided in Part 5 [48]

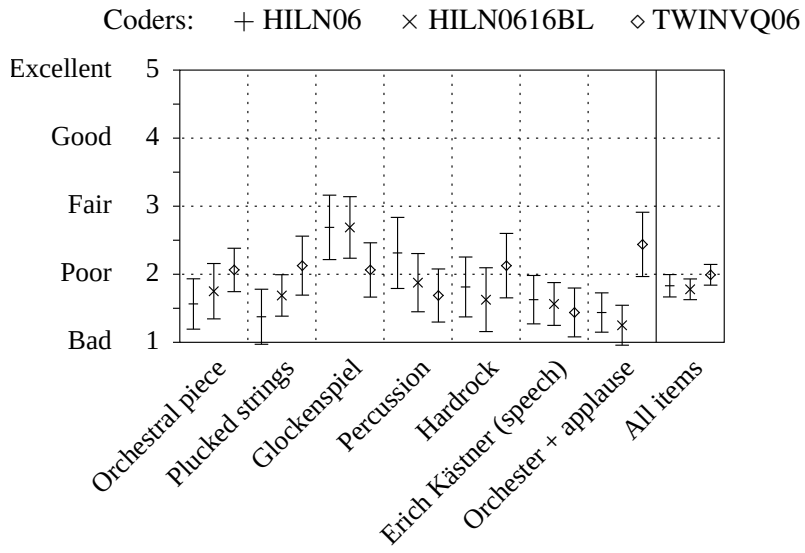

Figure 5.4: Results from the MPEG-4 Audio Version 2 verification test at 6 kbit/s with 16 subjects for HILN06, HILN0616BL, and TWINVQ06 shown as MOS with 95% CI for the 7 assessed items, using original signals with 16 kHz sampling rate (after [84]).

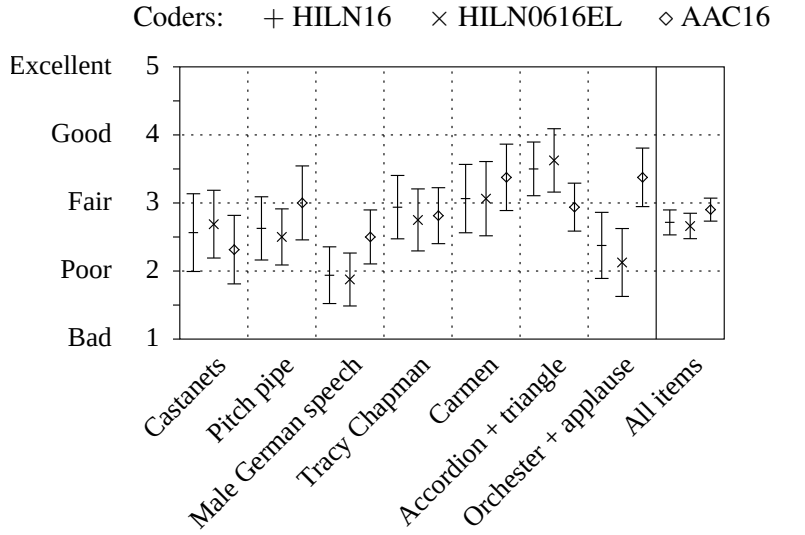

Figure 5.5: Results from the MPEG-4 Audio Version 2 verification test at 16 kbit/s with 16 subjects for HILN16, HILN0616EL, and AAC16 shown as MOS with 95% CI for the 7 assessed items, using original signals with 16 kHz sampling rate (after [84]).

6 Conclusions

The objective of this thesis is the efficient coding of *arbitrary* audio signals at very low bit rates. In order to find a suitable approach to this problem, a general *parametric audio coding* framework has been devised. By combining different source models into a hybrid model, it permits flexible utilization of a broad range of source and perceptual models. Using this framework, a complete parametric audio coding system for bit rates in the range of approximately 6 to 16 kbit/s has been developed. In the course of this development, two major tasks had to be accomplished.

- The *first task* comprises the design of an appropriate hybrid source model and the development of algorithms for parameter estimation and signal decomposition which also consider a perceptual model. It was divided into four steps.
- The *second task* comprises the design of a suitable parameter quantization, coding, and bit allocation scheme as well as the development of efficient synthesis algorithms for the decoder and extensions of the coding system that enable additional functionalities. It was divided into three steps.

Parameter Estimation and Signal Decomposition

As a **first step**, a suitable hybrid source model is required in order to obtain a compact parametric description of an audio signal. To assess the efficiency of such a source model, it is necessary to employ it in the context of a complete coding system. Hence, the design of the hybrid source model has been carried out in parallel with the development of the coding system itself. While known hybrid source models combine sinusoidal, noise, and transient model components, the developed hybrid source model also permits to describe harmonic tones that coexist simultaneously with all other components.

Already a pure sinusoidal source model would permit to describe arbitrary signals exactly, similar to a Fourier representation. It is, however, obvious that this representation is very inefficient for noise-like signals. This reveals the need to include a noise component in the hybrid model. By grouping a set of sinusoidal trajectories as a harmonic tone component, the number of required parameters can be reduced, thus obtaining a more compact representation.

A *sinusoidal trajectory* is characterized by its frequency and amplitude and an initial phase at the onset time. A *harmonic tone* is characterized by its fundamental frequency and the amplitudes and initial phases of its partials. To accommodate for slight inharmonicities, as for example observed for stiff strings, an additional stretching parameter

has been introduced which allows to describe the increasing spacing of partials towards higher frequencies. A more compact parameterization of the amplitudes of the partials is possible when they are described by the spectral envelope of the harmonic tone in combination with an overall amplitude parameter. Different schemes for a compact parameterization of the spectral envelope were investigated. Similar to schemes based on linear predictive coding (LPC), the magnitude response of an all-pole filter of appropriate order has been chosen for this purpose. The filter parameters are represented as reflection coefficients to facilitate easy reduction of the filter order if a less accurate spectral envelope is sufficient. A *noise component* is characterized by its overall amplitude and spectral envelope. Also here, the magnitude response of an all-pole filter of appropriate order is used to describe the spectral envelope.

Because the components' frequency, amplitude, and spectral envelope parameters may vary slowly over time, they are estimated at regular intervals and interpolated during synthesis. This frame-based approach means that the audio signal is handled as a sequence of frames with a fixed frame length (stride) T_f which represent overlapping signal segments. Typically, a frame length T_f of 32 ms optimizes the performance of the coding system when operated at its target bit rate range. This frame length, however, is too long to describe the fast variation of amplitude parameters which occur at *signal transients*. Therefore, additional parameters have been introduced to model the temporal amplitude envelope of components more precisely at signal transients. The additional parameters are the attack and decay rates and the temporal location of the maximum within a frame.

To limit the complexity of the source model and to simplify signal decomposition, only a single harmonic tone component and a single noise component in combination with multiple sinusoidal trajectories are permitted by the hybrid source model at any time.

As a **second step**, accurate and robust parameter estimation for signal components is important in order to allow perceptually equivalent resynthesis of components and to facilitate analysis-by-synthesis-based signal decomposition.

The component parameters are estimated once per frame and the estimation is based on a signal segment obtained using a temporal window centered around the midpoint of the frame. The windows for consecutive segments typically overlap by 50%.

For *sinusoidal components*, accurate estimation of the frequency is vital. For the simple case of a single sinusoid with constant frequency in white Gaussian noise, the maximum likelihood estimator is basically given by the location of the maximum in the periodogram, i.e., the squared magnitude of the Fourier transform. Once a frequency estimate is available, amplitude and phase can be found by correlation with a complex sinusoid having the estimated frequency.

In order to reduce interference from neighboring sinusoidal components and to meet the assumption of a single sinusoid in noise, a bandpass filter with a passband centered at an initial estimate of the sinusoid's frequency has been introduced. The initial estimate, usually the location of a peak in the discrete Fourier transform of the signal segment, is provided by a decomposition algorithm. In combination with the above mentioned temporal window, the estimation thus evaluates only a section of the time-frequency plane.

The filter bandwidth has been chosen carefully to achieve a good trade-off between maximum attenuation of neighboring components and minimum attenuation of the sinusoid to be estimated, which may be located off-center due to the error of the initial estimate. Typically, a bandwidth of approximately 30 Hz is used.

To implement bandpass filtering and frequency estimation, the signal is subjected to a frequency shift determined by the initial estimate, followed by a lowpass filter. Frequency shifting is performed by heterodyning with a complex local oscillator signal having the initially estimated frequency. The frequency of the resulting complex baseband signal is then estimated accurately as the slope of a linear approximation of its phase over time. Finally, the initial estimate is added to obtain an accurate frequency estimate for the original signal.

To accommodate for the time-varying frequency of a sinusoidal trajectory in case of vibrato or portamento, the heterodyne-based frequency estimator has been extended to permit also estimation of the sweep rate of linearly changing frequencies. For this purpose, the filter bandwidth had to be increased to cover the frequency range traversed during the duration of the signal segment.

High sweep rates, as for example observed for the higher partials of a singing voice, cannot be recovered this way because of increasing interference from neighboring partials. To address this problem, algorithms for building sinusoidal trajectories from a time-series of frequency and amplitude parameter estimates have been investigated. A simple trajectory-building approach is based on finding the best matches between frequency and amplitude parameters of the sinusoids estimated independently in consecutive segments. Because of the mentioned problems, the results are not reliable in case of high sweep rates. More reliable results can be obtained if the signal of a sinusoidal component is actually tracked between consecutive segments. Hence, the sweep estimator has been extended to take into account the frequency and phase parameters in the previous segment in order to provide phase-locked tracking of a supposed sinusoidal trajectory. The difference between the frequency in the previous segment and initial frequency estimate for the current segment provides also an initial estimate of the sweep rate. A correspondingly sweeping local oscillator signal permits to extract a slanted section of the time-frequency plane for the accurate frequency and sweep rate estimation.

Reliable estimation of *harmonic tone* parameters is essential because detecting an erroneous fundamental frequency and forcing sinusoidal signal components onto the corresponding incorrect harmonic grid can result in very annoying artifacts. The problem here is to search for potential fundamental frequencies with partials that match well a subset of the sinusoidal trajectories estimated in the current signal segment. First, a coarse search is performed over the full range of permissible fundamental frequencies, typically from 30 Hz to 1 kHz. Then, a refined search in the neighborhood of the first match is carried out to find the accurate fundamental frequency and stretching parameter. If the best match fulfills minimum requirements, the corresponding sinusoids are taken as partials of the harmonic tone.

To improve modeling of *transient signals*, an additional temporal amplitude envelope

can be applied to sinusoidal trajectories, harmonic tones, and noise components. Estimation of the envelope is based on the magnitude of an analytic signal derived from the current signal segment with help of a Hilbert transform. The envelope parameters are found by fitting the triangular attack/decay model envelope to the signal's magnitude over time using weighted regression techniques. The envelope reconstructed from the estimated parameters has then to be taken into account during amplitude estimation for the components in question.

The parameter estimation for a *noise component* assumes a noise-like input signal. Since the spectral envelope of the noise component is described by an all-pole filter magnitude response, the same techniques as used in LPC-based coding schemes are applied to find the filter coefficients. The noise amplitude parameter is calculated directly from the signal's variance. If non-noise components are present in this input signal, the estimated noise parameters may become inaccurate.

As a **third step**, the problem of an appropriate signal decomposition algorithm has to be addressed. Such an algorithm has to determine the different components that constitute the input signal and initiate parameter estimation for these components. For this purpose, deterministic and stochastic components have to be distinguished. The parameters of a deterministic component describe a well-defined signal, whereas the parameters of a stochastic component characterize an underlying stochastic process.

Deterministic components permit subtractive signal decomposition. This enables an iterative analysis-by-synthesis approach. Starting from the original input signal segment, in each step of the iteration a dominant deterministic component in the current residual is extracted. The extraction is realized by estimating the component parameters and then resynthesizing and subtracting the component to calculate a new residual. In contrast, stochastic components do not allow for subtractive decomposition. However, it can be assumed that, if all deterministic components have been extracted properly, only stochastic components are left over in the residual.

Deterministic components, like two sinusoids closely spaced in frequency, can be non-orthogonal due to the limited duration of the analyzed signal segment. In this case, the greedy nature of the iterative analysis-by-synthesis approach can lead to an increased parameter estimation error. This effect, however, has been alleviated sufficiently by using robust parameter estimation and component tracking as described above.

The decomposition algorithm has to ensure that determination of the type of a component is reliable and robust. In order to avoid modeling noise-like signals as sinusoids, estimated sinusoidal parameters are discarded if the resynthesized sinusoid does not lead to a sufficient reduction of the residual in a narrow spectral band centered at the estimated frequency. On the other hand, modeling sinusoidal components as noise should also be avoided. Hence, all significant sinusoidal components have to be extracted from the input signal even if not all of them will be transmitted to the decoder.

Discrimination of noise and sinusoidal components during signal decomposition has been improved further by incorporating a perceptual model that, based on the spectral flatness measure, indicates whether a given spectral band is perceived as a tonal or noise-

like signal. Thus, extraction of sinusoids is avoided for spectral bands perceived as noise.

In view of the application to very low bit rate audio coding, component selection becomes important in order to ensure that the perceptually most relevant components are conveyed in the bit stream. For this purpose, the components are ordered according to their perceptual relevance and the first I components on this list are selected for transmission, where I is adapted dynamically to comply with bit rate constraints. An appropriate perceptual model is required in order to assess the relevancy of a component. Different perceptual models and reordering strategies have been investigated. Generally, strategies based on the masked threshold as well as strategies that seek to approximate the auditory excitation level were found suitable. Differences in performance were mainly observed for the case of a very small number I of selected components, where excitation-based strategies performed advantageous.

As a **fourth step**, in order to optimize the coding system in a rate distortion sense, constraints imposed by the available bit rate have to be considered during signal decomposition and component selection. If, for example, only a very small number of sinusoidal components can be conveyed due to bit rate constraints, it can be perceptually advantageous to include some of the energy of the discarded sinusoids in the noise component. Such aspects were included empirically in the signal decomposition and component selection algorithms. In addition to the quality-optimized encoder described here, also an alternative encoder implementation with significantly reduced computational complexity was studied.

Parameter Coding and Signal Synthesis

As a **first step** when building a complete audio coding system based on the parametric signal representation introduced here, procedures for efficient parameter coding are needed, comprising parameter quantization and entropy coding of the quantized parameters. For each signal segment, the coded parameters are conveyed as one frame of the bit stream.

Amplitude and frequency parameters of sinusoidal trajectories are quantized non-uniformly, with quantization step sizes approximating the just-noticeable differences for amplitude and frequency changes as known from psychoacoustics. Tests in the context of a complete coding system have shown that, for amplitude parameters, a somewhat coarser quantization is permissible, especially at the onset of trajectories. Hence, amplitudes are quantized uniformly on a logarithmic scale with a step size of 1.5 dB, or 3 dB at onsets, while frequencies are quantized uniformly on a critical-band rate scale with $1/32$ Bark step size. It was also found that, if start phases of trajectories are not transmitted and random start phases are used for synthesis instead, the subjective quality of the decoded signal is generally not degraded. However, if deterministic decoder behavior is needed, start phases are quantized uniformly with $\pi/16$ step size.

The fundamental frequency of a harmonic tone is quantized uniformly on a logarithmic scale with a step size of 1:1.0026 (4.5 cent). The parameter for the overall amplitude of a harmonic tone, which is calculated as the root of the summed squares of the partials'

amplitudes, is quantized uniformly on a logarithmic scale with a step size of 1.5 dB.

To quantize and code the all-pole filter parameters describing the spectral envelope of a harmonic tone, a logarithmic area ratio (LAR) representation of the reflection coefficients is adopted. Unlike the line spectral frequency (LSF) representation commonly used in speech coding, LARs allow changes of the filter order from frame to frame while still permitting efficient inter-frame prediction of filter parameters. This inter-frame prediction uses a predictor coefficient of 0.75 or 0.5 after subtracting the mean LARs of the average lowpass spectrum. The LARs are quantized uniformly with empirically determined quantizer step sizes of approximately 0.1. Variable length codes are employed to achieve entropy coding.

To encode the quantized parameters of new sinusoidal trajectories, i.e., those with an onset in the current frame, a sophisticated technique has been devised that is referred to as *subdivision coding* (SDC). It is motivated by the fact that it is not necessary to convey the set of frequency/amplitude parameter pairs of new sinusoidal trajectories in a specific order. By allowing an arbitrary permutation of this set of N parameter pairs, a redundancy of approximately $\log_2(N!)$ bit per frame can be exploited. SDC successfully utilizes this effect by arranging the parameter pairs in an order of increasing frequency. Furthermore, it takes also advantage of the non-uniform probability distribution of the frequency and amplitude parameters.

For all frequency and amplitude parameters of components continued from the previous frame, differential coding of the quantized parameters is employed to implement inter-frame prediction. To exploit the non-uniform probability distribution of the parameter differences, variable length codes are applied for entropy coding.

The spectral envelope and amplitude parameters of a noise component are encoded in the same way as for a harmonic tone. The only difference is that the LARs are quantized uniformly with a step size of approximately 0.3.

Using these coding techniques in combination with an optimized bit allocation strategy that takes into account the rate distortion considerations described above, it is possible to convey the parameters of approximately 10 to 20 simultaneous sinusoidal trajectories at a bit rate of 6 kbit/s. A noise component typically requires 0.5 to 1.5 kbit/s, while a harmonic tone, if present, typically requires 3 kbit/s or more, reducing the number of conveyed sinusoidal trajectories correspondingly.

As a **second step**, the decoding process has to be defined. The signal synthesis in the decoder reconstructs the component signals from the decoded parameters. By combining these signals, the final audio signal is obtained.

All sinusoids, including the partials of a harmonic tone, are synthesized using sine wave generators. Overlapping synthesis windows are used to obtain smooth fade-in and fade-out of sinusoids that begin or end in the current frame. For a sinusoid continuing from one frame to the next, the amplitude and frequency parameters are linearly interpolated between the two frames during synthesis, maintaining phase continuity of the trajectory.

The noise component is synthesized by filtering white noise from a random number

generator with an IIR all-pole filter (i.e., an LPC synthesis filter) to shape the spectrum according to the spectral envelope parameters. A smooth transition between frames is achieved by a windowed overlap/add scheme.

If the parameters of an additional temporal amplitude envelope are conveyed in the bit stream, the corresponding envelope is reconstructed and applied to the selected components as indicated in the bit stream.

Using computationally efficient sinusoidal synthesis algorithms, the complete decoding and synthesis process for a bit rate of 6 to 16 kbit/s accounts for a CPU load of approximately 10 to 20 MHz clock rate on today's personal computers.

As a **third step**, to provide additional functionalities, the parametric coding system can be extended in different ways.

Time-scaling has been implemented by varying the length of the synthesized time frames, so that the playback speed of the signal can be modified in the decoder without affecting the pitch of the signal. *Pitch-shifting* has been implemented by multiplying all frequency parameters by a given factor prior to synthesis to alter the pitch of the decoded signal without affecting the playback speed.

Bit rate scalability has been achieved by transmitting the parameters of the perceptually most important signal components in a base layer bit stream and the parameters of further signal components in additional enhancement layer bit streams. In the case of limited transmission bandwidth, only the base layer is received and decoded. If more bandwidth is available, one or more enhancement layers are received as well. Together with the base layer, the audio signal can then be reconstructed at a higher quality.

Improved *error robustness* for operation over error-prone transmission channels has been achieved by unequal error protection. Furthermore, techniques to minimize error propagation and to achieve error concealment were included in the encoder and decoder, respectively.

Discussion and Directions for Future Work

This thesis studies the problem of efficient parametric audio coding at very low bit rates and provides novel contributions addressing four major problems. An optimized hybrid source model was designed which allows a harmonic tone to be present simultaneously with individual sinusoidal components, transients, and noise. Robustness of signal decomposition and parameter estimation was improved by tracking all sinusoidal components, including the partials of a harmonic tone, over time to reliably build sinusoidal trajectories. Perceptual models were incorporated into both signal decomposition and component selection algorithms to enable efficient operation of the coding system at very low bit rates. Last but not least, the joint coding of the set of parameter of individual sinusoidal components by a novel subdivision coding technique enabled a very high coding efficiency.

The parametric audio coding system developed in the thesis has been adopted as a part of the MPEG-4 standard, where it is referred to as *Harmonic and Individual Lines plus*

Noise (HILN) coding. For final verification, HILN has been compared to state-of-the-art transform coding systems by means of a listening test at bit rates of 6 and 16 kbit/s. This test has shown that HILN performs comparable to TwinVQ when operated at 6 kbit/s and comparable to AAC when operated at 16 kbit/s. A detailed analysis reveals that HILN performs advantageously for certain types of audio signals, like mixed speech/music signals and single instruments, while transform coding performs advantageously for other types of audio signals, like complex orchestral sounds. Clean speech signals, however, are still coded most efficiently by a dedicated speech coder.

Future work should address further optimization of the encoder as well as extensions of the hybrid parametric source model needed in order to overcome limitations observed for some types of audio signals. At a later stage, the combination of speech, transform, and parametric coding techniques in an integrated coding system should be considered.

A Listening Test Items

This appendix provides an overview of the audio test items used during the development of the HILN parametric audio coding system. The item's name and duration are given together with a short description and information about the signal type (*speech*, *solo instrument*, *music*, or *complex sound*) as used during the MPEG-4 Audio standardization process. Table A.1 lists the 12 test items used in the MPEG-4 Audio core experiment procedure. Table A.2 lists the 39 test items used for the final item selection for the MPEG-4 Audio HILN verification tests at 6 and 16 kbit/s. The complete set of these 39 items was also used to measure the parameter statistics reported in Chapter 4.

Item	Length [s]	Source	Description	Signal Type
es01	10.734	mp4_02	A capella (Suzan Vega)	speech
es02	8.600	mp4_05	Male German speech	speech
es03	7.604	mp4_06	Female English speech	speech
si01	7.995	mp4_01	Harpsichord	instrument
si02	7.725	mp4_04	Castanets	instrument
si03	27.887	tk6 (nbc)	Pitch pipe	instrument
sm01	11.149	tk4_m	Bagpipes	music
sm02	10.095	s6_m	Glockenspiel	music
sm03	13.986	s2_m	Plucked strings	music
sc01	10.969	mp4_03	Trumpet solo and orchestra	complex
sc02	12.732	msinger_m	Orchestral piece	complex
sc03	11.552	spot1_m	Contemporary pop music	complex

Table A.1: List of the 12 test items (total duration: 141.028 s) used in the MPEG-4 Audio core experiment procedure.

Item	Length [s]	Filename	Description	6 kbit/s	16 kbit/s	Type
01	7.608	es03	Female English speech			spch
02	8.016	si01	Harpsichord			inst
03	7.728	si02	Castanets		test	inst
04	27.888	si03	Pitch pipe		test	inst
05	11.160	sm01	Bagpipes			inst
06	10.992	sc01	Trumpet solo + orchestra			musi
07	12.744	sc02	Orchestral piece	test		musi
08	11.568	sc03	Contemporary pop music			musi
09	12.549	uhd2	Mari Boine			cplx
10	19.032	te1	Dorita			musi
11	16.776	te2	Plucked strings	test		inst
12	21.024	te6	Glockenspiel	test		inst
13	19.464	te7	Male German speech	trng	test	spch
14	19.968	te8	Suzanne Vega		trng	musi
15	18.576	te9	Tracy Chapman		test	cplx
16	19.416	te13	Bass guitar			inst
17	17.640	te14	Hayden trumpet concert		trng	inst
18	20.016	te15	Carmen		test	musi
19	17.016	te16	Accordion + triangle		test	musi
20	18.024	te18	Percussion	test		musi
21	17.184	te20	George Duke			musi
22	12.912	te21	Asa Jinder			musi
23	17.952	te23	Dalarnas Spelmansforbund			musi
24	17.568	te25	Stravinsky			musi
25	16.968	te30	aimai (acoustic)			musi
26	18.072	te32	Palmtop boogie (acoustic)			musi
27	16.080	te36	O1 (acoustic)			cplx
28	16.272	te42	Kids Drive Dance (acoustic)			musi
29	19.375	track76	Hardrock	test		cplx
30	19.375	track78	Asian folklore			musi
31	19.562	track82	Classic (opera)	trng	trng	cplx
32	19.375	track84	Classic (orchestra)			musi
33	20.016	hexagon	Background music (orchestra)			cplx
34	20.016	radiofr1	Radio France speech/music			cplx
35	20.016	rfi1	Radio France Int. speech/music			cplx
36	19.250	app_guit	Complex sound + applause			cplx
37	20.000	jazzdrum	Complex sound (jazz)	trng		musi
38	20.000	kaest_mal	Erich Kaestner (speech)	test		spch
39	20.000	mussorg	Orchestra + applause	test	test	cplx

Table A.2: List of the 39 test items (total duration: 667.198 s) used in the MPEG-4 Audio verification tests. The items selected for the Version 2 HILN verification tests at 6 and 16 kbit/s are marked as “test” (test item) or “trng” (training item).

B Subdivision Coding Algorithm

The remainder of this appendix provides the definition of the decoding algorithm for the subdivision code including the necessary tables, as it is given in subclause 7.3.2.4 “HILN SubDivisionCode (SDC)” of the MPEG-4 Audio standard ISO/IEC 14496-3:2001 [46].

The SubDivisionCode (SDC) is an algorithmically generated variable length code, based on a given table and a given number of different codewords. The decoding process is defined below.

The idea behind this coding scheme is the subdivision of the probability density function into two parts which represent an equal probability. One bit is transmitted that determines the part the value to be coded is located. This subdivision is repeated until the width of the part is one and then its position is equal to the value being coded. The positions of the boundaries are taken out of a table of 32 quantized, fixed point values. Besides this table (parameter `tab`) the number of different codewords (parameter `k`) is needed too.

The following C function `SDCDecode(k, tab)` together with the 9 tables `sdcILATable[32]` and `sdcILFTable[8][32]` describe the decoding. The function `GetBit()` returns the next bit in the stream.

```
int sdcILATable[32] = {
    0,  13,  27,  41,  54,  68,  82,  96, 110, 124, 138, 152, 166, 180, 195, 210,
  225, 240, 255, 271, 288, 305, 323, 342, 361, 383, 406, 431, 460, 494, 538, 602 };

int sdcILFTable[8][32] = {
{   0,  53,  87, 118, 150, 181, 212, 243, 275, 306, 337, 368, 399, 431, 462, 493,
  524, 555, 587, 618, 649, 680, 711, 743, 774, 805, 836, 867, 899, 930, 961, 992 },
{   0,  34,  53,  71,  89, 106, 123, 141, 159, 177, 195, 214, 234, 254, 274, 296,
  317, 340, 363, 387, 412, 438, 465, 494, 524, 556, 591, 629, 670, 718, 774, 847 },
{   0,  26,  41,  54,  66,  78,  91, 103, 116, 128, 142, 155, 169, 184, 199, 214,
  231, 247, 265, 284, 303, 324, 346, 369, 394, 422, 452, 485, 524, 570, 627, 709 },
{   0,  23,  35,  45,  55,  65,  75,  85,  96, 106, 117, 128, 139, 151, 164, 177,
  190, 204, 219, 235, 252, 270, 290, 311, 334, 360, 389, 422, 461, 508, 571, 665 },
{   0,  20,  30,  39,  48,  56,  64,  73,  81,  90,  99, 108, 118, 127, 138, 149,
  160, 172, 185, 198, 213, 228, 245, 263, 284, 306, 332, 362, 398, 444, 507, 608 },
{   0,  18,  27,  35,  43,  50,  57,  65,  72,  79,  87,  95, 104, 112, 121, 131,
  141, 151, 162, 174, 187, 201, 216, 233, 251, 272, 296, 324, 357, 401, 460, 558 },
{   0,  16,  24,  31,  38,  45,  51,  57,  64,  70,  77,  84,  91,  99, 107, 115,
  123, 132, 142, 152, 163, 175, 188, 203, 219, 237, 257, 282, 311, 349, 403, 493 },
{   0,  12,  19,  25,  30,  35,  41,  46,  51,  56,  62,  67,  73,  79,  85,  92,
   99, 106, 114, 122, 132, 142, 153, 165, 179, 195, 213, 236, 264, 301, 355, 452 }
};

int SDCDecode (int k, int *tab)
{
   int *pp;
   int g,dp,min,max;

   min=0;
   max=k-1;
   pp=tab+16;
   dp=16;

   while ( min!=max )
   {
      if ( dp ) g=(k*(*pp))>>10; else g=(max+min)>>1;
      dp>>=1;
      if ( GetBit()==0 ) { pp-=dp; max=g; } else { pp+=dp; min=g+1; }
   }
   return max;
}
```

Bibliography

[1] L. B. Almeida and J. M. Tribolet, "Harmonic coding: A low bit-rate, good-quality speech coding technique," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, 1982, pp. 1664–1667.

[2] L. B. Almeida and J. M. Tribolet, "Nonstationary spectral modeling of voiced speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, no. 3, pp. 664–678, June 1983.

[3] B. S. Atal and M. R. Schroeder, "Stochastic coding of speech signals at very low bit rates," in *Proc. IEEE Int. Conf. on Communications*, Amsterdam, NL, May 1984, pp. 1610–1613.

[4] F. Baumgarte, C. Ferekidis, and H. Fuchs, "A nonlinear psychoacoustic model applied to the ISO MPEG layer 3 coder," in *AES 99th Convention*, New York, NY, US, Oct. 1995, Preprint 4087.

[5] F. Baumgarte, *Ein psychophysiologisches Gehörmodell zur Nachbildung von Wahrnehmungsschwellen für die Audiocodierung*, Dissertation, Universität Hannover, DE, 2000.

[6] T. Berger, *Rate Distortion Theory*, Prentice-Hall, Englewood Cliffs, NJ, US, 1971.

[7] B. Boashash, "Estimation and interpreting the instantaneous frequency of a signal – part 1: Fundamentals / part 2: Algorithms and applications," *Proc. IEEE*, vol. 80, no. 4, pp. 519–568, Apr. 1992.

[8] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking," in *Proceedings of the Symposium on Time Series Analysis*, M. Rosenblatt, Ed., chapter 15, pp. 209–243. John Wiley & Sons, New York, NY, US, 1963.

[9] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 Advanced Audio Coding," *J. Audio Eng. Soc.*, vol. 45, no. 10, pp. 789–814, Oct. 1997.

[10] K. Brandenburg, "Perceptual coding of high quality digital audio," in *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds., chapter 2, pp. 39–83. Kluwer, Boston, MA, US, 1998.

[11] K. Brandenburg, "mp3 and AAC explained," in *Proc. AES 17th Int. Conf. (High-Quality Audio Coding)*, Florence, IT, Sept. 1999, pp. 99–110.

[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, DE, May 2002, Preprint 5554.

[13] J. C. Brown, "Musical fundamental frequency tracking using a patter recognition method," *J. Acoust. Soc. Amer.*, vol. 92, no. 3, pp. 1394–1402, Sept. 1992.

[14] L. Contin, B. Edler, D. Meares, and P. Schreiner, "Tests on MPEG-4 audio codec proposals," *Image Communication*, vol. 9, pp. 327–342, 1997.

[15] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, Winter 1986.

[16] H. Dudley, "The vocoder," *Bell Labs Rec*, vol. 17, pp. 122, 1939, overview available: `http://www.bell-labs.com/org/1133/Heritage/Vocoder/`, accessed 2007-12-31.

[17] B. Edler, H. Purnhagen, and C. Ferekidis, "Detailed technical description of the MPEG-4 audio coding proposal from University of Hannover and Deutsche Telekom AG," ISO/IEC JTC1/SC29/WG11 M0843, Mar. 1996.

[18] B. Edler, H. Purnhagen, and C. Ferekidis, "ASAC – analysis/synthesis audio codec for very low bit rates," in *AES 100th Convention*, Copenhagen, DK, May 1996, Preprint 4179.

[19] B. Edler, "Current status of the MPEG-4 audio verification model development," in *AES 101st Convention*, Los Angeles, CA, US, Nov. 1996, Preprint 4376.

[20] B. Edler and H. Purnhagen, "Concepts for hybrid audio coding schemes based on parametric techniques," in *AES 105th Convention*, San Francisco, CA, US, Sept. 1998, Preprint 4808.

[21] B. Edler, "Speech coding in MPEG-4," *Int. J. Speech Technology*, vol. 2, no. 4, pp. 289–303, May 1999.

[22] B. Edler and H. Purnhagen, "Parametric audio coding," Beijing, CN, Aug. 2000.

[23] A. El-Jaroudi and J. Makhoul, "Discrete all-pole modeling," *IEEE Trans. Signal Processing*, vol. 39, no. 2, pp. 411–423, Feb. 1991.

[24] ETSI, "GSM 06.60 enhanced full rate (EFR) speech transcoding," European Telecommunications Standards Institute, Nov. 1996.

[25] C. Ferekidis, "Kriterien für die Auswahl von Spektralkomponenten in einem Analyse/Synthese-Audiocoder," Diplomarbeit, Universität Hannover, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, DE, Mar. 1997.

[26] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, pp. 1493–1509, Nov. 1966.

[27] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer, New York, NY, US, 2nd edition, 1998.

[28] A. Freed, X. Rodet, and P. Depalle, "Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware," in *Proc. Int. Conf. Signal Processing Applications & Technology*, Santa Clara, CA, US, 1993.

[29] E. B. George and M. J. T. Smith, "Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones," *J. Audio Eng. Soc.*, vol. 40, no. 6, pp. 497–516, June 1992.

[30] E. B. George and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," *IEEE Trans. Speech Audio Processing*, vol. 5, no. 5, pp. 389–406, Sept. 1997.

[31] M. Goodwin, "Residual modeling in music analysis-synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Atlanta, GA, US, May 1996, pp. 1005–1008.

[32] M. Goodwin, "Matching pursuit with damped sinusoids," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Munich, DE, Apr. 1997, pp. 2037–2040.

[33] M. Goodwin, *Adaptive Signal Models: Theory, Algorithms, and Audio Applications*, Kluwer, Boston, MA, US, 1998.

[34] M. Goodwin, "Multiresolution sinusoidal modeling using adaptive segmentation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Seattle, WA, US, 1998, pp. 1525–1528.

[35] K. N. Hamdy, M. Ali, and A. H. Tewfik, "Low bit rate high quality audio coding with combined harmonic and wavelet representations," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Atlanta, GA, US, May 1996, pp. 1045–1048.

[36] P. Hedelin, "A tone-oriented voice-excited vocoder," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Atlanta, GA, US, Apr. 1981, pp. 205–208.

[37] J. Herre and D. Schulz, "Extending the MPEG-4 AAC codec by perceptual noise substitution," in *AES 104th Convention*, Amsterdam, NL, May 1998, Preprint 4720.

[38] J. Herre and H. Purnhagen, "General audio coding," in *The MPEG-4 Book*, F. Pereira and T. Ebrahimi, Eds., chapter 11. Prentice-Hall, Englewood Cliffs, NJ, US, 2002.

[39] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modelling of audio and speech using psychoacoustic-adaptive matching pursuits," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Salt Lake City, UT, US, May 2001, pp. 3281–3284.

[40] T. Hodes and A. Freed, "Second-order recursive oscillators for musical additive synthesis applications on SIMD and VLIW processors," in *Proc. Int. Computer Music Conf. (ICMC)*, Beijing, CN, 1999.

[41] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, no. 10, pp. 1098–1101, Sept. 1952.

[42] ISO/IEC, "Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)," ISO/IEC Int. Std. 13818-7:1997, 1997.

[43] ISO/IEC, "Coding of audio-visual objects – Part 1: Systems (MPEG-4 Systems, 2nd edition)," ISO/IEC Int. Std. 14496-1:2001, 2001.

[44] ISO/IEC, "Coding of audio-visual objects – Part 3: Audio (MPEG-4 Audio Version 1)," ISO/IEC Int. Std. 14496-3:1999, 1999.

[45] ISO/IEC, "Coding of audio-visual objects – Part 3: Audio, AMENDMENT 1: Audio extensions (MPEG-4 Audio Version 2)," ISO/IEC Int. Std. 14496-3:1999/Amd.1:2000, 2000.

[46] ISO/IEC, "Coding of audio-visual objects – Part 3: Audio (MPEG-4 Audio, 2nd edition)," ISO/IEC Int. Std. 14496-3:2001, 2001.

[47] ISO/IEC, "Coding of audio-visual objects – Part 4: Conformance testing (MPEG-4 Conformance, 2nd edition)," ISO/IEC Int. Std. 14496-4:2003, 2003.

[48] ISO/IEC, "Coding of audio-visual objects – Part 5: Reference Software (MPEG-4 Reference software, 2nd edition)," ISO/IEC Int. Std. 14496-5:2001, 2001.

[49] ITU, "Methods for the subjective assessment of sound quality – general requirements," ITU-R Recommend. BS.1284, 1997.

[50] ITU, "Method for objective measurements of perceived audio quality (PEAQ)," ITU-R Recommend. BS.1387, 1998.

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

[52] N. Iwakami and T. Moriya, "Transform domain weighted interleave vector quantization (TwinVQ)," in *AES 101st Convention*, Los Angeles, CA, US, 1996, Preprint 4377.

[53] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, NJ, US, 1984.

[54] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, no. 10, pp. 1385–1422, Oct. 1993.

[55] J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, New York, NY, US, Apr. 1988, vol. 5, pp. 2524–2527.

[56] S. M. Kay, *Modern Spectral Estimation: Theory & Application*, Prentice-Hall, Englewood Cliffs, NJ, US, 1988.

[57] S. Kay, "A fast and accurate single frequency estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 1987–1989, Dec. 1989.

[58] W. B. Kleijn and K. K. Paliwal, Eds., *Speech Coding and Synthesis*, Elsevier, Amsterdam, NL, 1995.

[59] A. Kodate, "Fundamental frequency estimation for the analysis/synthesis low bit rate audio coder," interner Abschlußbericht, Universität Hannover, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, DE, Oct. 1995.

[60] P. Kroon, E. F. Deprettere, and R. J. Sluyter, "Regular-pulse excitation—a novel approach to effective and efficient multipulse coding of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 5, pp. 1054–1063, Oct. 1986.

[61] S. N. Levine and J. O. Smith, "A sines+transients+noise audio representation for data compression and time/pitch scale modifications," in *AES 105th Convention*, San Francisco, CA, US, Sept. 1998, Preprint 4781.

[62] S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, Ph.D. thesis, CCRMA, Stanford University, CA, US, Dec. 1998.

[63] S. N. Levine and J. O. Smith, "A switched parametric & transform audio coder," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Phoenix, AZ, US, Mar. 1999.

[64] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[65] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[66] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[67] P. Masri, *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*, Ph.D. thesis, University of Bristol, UK, 1996.

[68] M. V. Mathews, "The digital computer as a musical instrument," *Science*, vol. 142, no. 3591, pp. 553–557, 1963.

[69] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. IT-6, pp. 7–12, Mar. 1960.

[70] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 4, pp. 744–754, Aug. 1986.

[71] N. Meine, "LPC-Spektralmodelle für einen Analyse/Synthese Audio Coder," Diplomarbeit, Universität Hannover, Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, DE, Jan. 1999.

[72] N. Meine and H. Purnhagen, "Vorrichtung zur Codierung/Decodierung," Patent DE19914645, 1999.

[73] N. Meine and H. Purnhagen, "Fast sinusoid synthesis for MPEG-4 HILN parametric audio decoding," in *Proc. Digital Audio Effects Workshop (DAFX)*, Hamburg, DE, Sept. 2002.

[74] R. A. Moog, "A brief introduction to electronic music synthesizers," *BYTE*, pp. 278–286, Dec. 1982.

[75] MPEG, "Official MPEG Home Page," available: `http://www.chiariglione.org/mpeg/`, accessed 2007-12-31.

[76] MPEG, "MPEG-4 call for proposals," ISO/IEC JTC1/SC29/WG11 N0997, July 1995.

[77] MPEG, "MPEG-4 audio test results (MOS test)," ISO/IEC JTC1/SC29/WG11 N1144, Jan. 1996.

[78] MPEG, "Draft MPEG-4 audio verification model," ISO/IEC JTC1/SC29/WG11 N1214, Mar. 1996.

[79] MPEG, "MPEG-4 audio core experiment test methodology," ISO/IEC JTC1/SC29/WG11 N1748, July 1997.

[80] MPEG, "Report on the MPEG-2 AAC stereo verification tests," ISO/IEC JTC1/SC29/WG11 N2006, Feb. 1998.

[81] MPEG, "Report on the MPEG-4 audio NADIB verification tests," ISO/IEC JTC1/SC29/WG11 N2276, July 1998.

[82] MPEG, "Report on the MPEG-4 speech codec verification tests," ISO/IEC JTC1/SC29/WG11 N2424, Oct. 1998.

[83] MPEG, "MPEG-4 audio verification test results: Audio on internet," ISO/IEC JTC1/SC29/WG11 N2425, Oct. 1998.

[84] MPEG, "Report on the MPEG-4 audio version 2 verification test," ISO/IEC JTC1/SC29/WG11 N3075, Dec. 1999.

[85] M. Mummert, *Sprachcodierung durch Konturierung eines gehörangepaßten Spektrogramms und ihre Anwendung zur Datenreduktion*, Dissertation, Technische Universität München, DE, 1997.

[86] H. G. Musmann, "A layered coding system for very low bit rate video coding," *Image Communication*, vol. 7, pp. 267–278, 1995.

[87] J. Nieuwenhuijse, R. Heusdens, and E. F. Deprettere, "Robust exponential modeling of audio signals," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Seattle, WA, US, May 1998, vol. 6, pp. 3581–3584.

[88] M. Nishiguchi, K. Iijima, A. Inoue, Y. Maeda, and J. Matsumoto, "Harmonic vector excitation coding of speech at 2.0-4.0 kbps," in *Proc. ICCE*, June 1998.

[89] M. Nishiguchi and B. Edler, "Speech coding," in *The MPEG-4 Book*, F. Pereira and T. Ebrahimi, Eds., chapter 10. Prentice-Hall, Englewood Cliffs, NJ, US, 2002.

[90] A. W. J. Oomen and A. C. den Brinker, "Sinusoids plus noise modelling for audio signals," in *Proc. AES 17th Int. Conf. (High-Quality Audio Coding)*, Florence, IT, Sept. 1999, pp. 226–232.

[91] A. V. Oppenheim and R. W. Schafer, "From frequency to quefrency: A history of the cepstrum," *IEEE Signal Processing Magazine*, pp. 95–106, Sept. 2004.

[92] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–513, Apr. 2000.

[93] T. Painter and A. Spanias, "Perceptual segmentation and component selection in compact sinusoidal representations of audio," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Salt Lake City, UT, US, May 2001, pp. 3289–3292.

[94] F. Pereira, "Context, objectives, and process," in *The MPEG-4 Book*, F. Pereira and T. Ebrahimi, Eds., chapter 1. Prentice-Hall, Englewood Cliffs, NJ, US, 2002.

[95] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, Prentice-Hall, Upper Saddle River, NJ, US, 3rd edition, 1996.

[96] H. Purnhagen, B. Edler, and C. Ferekidis, "Proposal of a core experiment for extended 'harmonic and individual lines plus noise' tools for the parametric audio coder core," ISO/IEC JTC1/SC29/WG11 M2480, July 1997.

[97] H. Purnhagen and B. Edler, "Check phase results of core experiment on extended 'harmonic and individual lines plus noise'," ISO/IEC JTC1/SC29/WG11 M2795, Oct. 1997.

[98] H. Purnhagen and B. Edler, "On HILN and TwinVQ performance in the audioonin-ternet verification test (revised)," ISO/IEC JTC1/SC29/WG11 M4087R, Oct. 1998.

[99] H. Purnhagen and N. Meine, "Core experiment proposal on improved parametric audio coding," ISO/IEC JTC1/SC29/WG11 M4492, Mar. 1999.

[100] H. Purnhagen and N. Meine, "Pre-screening results for CE on improved parametric audio coding," ISO/IEC JTC1/SC29/WG11 M4493, Mar. 1999.

[101] H. Purnhagen, B. Edler, and C. Ferekidis, "Object-based analysis/synthesis audio coder for very low bit rates," in *AES 104th Convention*, Amsterdam, NL, May 1998, Preprint 4747.

[102] H. Purnhagen and B. Edler, "Objektbasierter Analyse/Synthese Audio Coder für sehr niedrige Datenraten," in *ITG-Fachtagung 'Codierung für Quelle, Kanal und Übertragung'*, Aachen, DE, Mar. 1998, pp. 35–40.

[103] H. Purnhagen, "An overview of MPEG-4 audio version 2," in *Proc. AES 17th Int. Conf. (High-Quality Audio Coding)*, Florence, IT, Sept. 1999, pp. 157–168.

[104] H. Purnhagen, "Advances in parametric audio coding," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Mohonk, New Paltz, NY, US, Oct. 1999, pp. 31–34.

[105] H. Purnhagen, N. Meine, and B. Edler, "Speeding up HILN – MPEG-4 parametric audio encoding with reduced complexity," in *AES 109th Convention*, Los Angeles, CA, US, Sept. 2000, Preprint 5177.

[106] H. Purnhagen and N. Meine, "HILN – the MPEG-4 parametric audio coding tools," in *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Geneva, CH, May 2000, pp. III–201 – III–204.

[107] H. Purnhagen, "Der MPEG-4 Audio Standard – ein Überblick," *Rundfunktechnische Mitteilungen (RTM)*, vol. 44, no. 2, Apr. 2000.

[108] H. Purnhagen, B. Edler, and N. Meine, "Error protection and concealment for HILN MPEG-4 parametric audio coding," in *AES 110th Convention*, Amsterdam, NL, 2001, Preprint 5300.

[109] H. Purnhagen, N. Meine, and B. Edler, "Sinusoidal coding using loudness-based component selection," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Orlando, FL, US, May 2002, pp. II–1817 – II–1820.

[110] H. Purnhagen, "Parameter estimation and tracking for time-varying sinusoids," in *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, Leuven, BE, Nov. 2002, pp. 5–8.

[111] J.-C. Risset and M. V. Mathews, "Analysis of musical instrument tones," *Physics Today*, vol. 22, pp. 22–30, Feb. 1969.

[112] D. Schulz, *Kompression qualitativ hochwertiger digitaler Audiosignale durch Rauschextraktion*, Dissertation, Technische Hochschule Darmstadt, DE, 1997.

[113] X. Serra, *A System For Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*, Ph.D. thesis, Stanford University, CA, US, 1989.

[114] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[115] X. Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, C. Roads et al., Eds. Swets & Zeitlinger Publishers, Lisse, NL, 1997.

[116] J. O. Smith and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proc. Int. Computer Music Conf. (ICMC)*, Champaign-Urbana, IL, US, 1987, annotated original full version available: `http://www-ccrma.stanford.edu/~jos/parshl/`, accessed 2007-12-31.

[117] J. O. Smith, "Principles of digital waveguide models of musical instruments," in *Applications of Digital Signal Processing to Audio and Acoustics*, M. Kahrs and K. Brandenburg, Eds., chapter 10, pp. 417–466. Kluwer, Boston, MA, US, 1998.

[118] 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, Mar. 1998.

[119] A. S. Spanias, "Speech coding: A tutorial review," *Proc. IEEE*, vol. 82, no. 10, pp. 1541–1582, Oct. 1994.

[120] T. Thiede, *Perceptual Audio Quality Assessment using a Non-Linear Filter Bank*, Dissertation, Technische Universität Berlin, DE, 1999.

[121] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, and C. Colomes, "PEAQ – the ITU standard for objective measurement of perceived audio quality," *J. Audio Eng. Soc.*, vol. 48, no. 1/2, Jan./Feb. 2000.

[122] S. A. Tretter, "Estimating the frequency of a noisy sinusoid by linear regression," *IEEE Trans. Inform. Theory*, vol. 31, no. 6, pp. 832–835, Nov. 1985.

[123] R. Väänänen and J. Huopaniemi, "SNHC audio and audio composition," in *The MPEG-4 Book*, F. Pereira and T. Ebrahimi, Eds., chapter 12. Prentice-Hall, Englewood Cliffs, NJ, US, 2002.

[124] B. L. Vercoe, W. G. Gardner, and E. D. Scheirer, "Structured audio: Creation, transmission, and rendering of parametric sound representations," *Proc. IEEE*, vol. 85, no. 5, pp. 922–940, May 1998.

[125] T. S. Verma, S. N. Levine, and T. H. Y. Meng, "Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals," in *Proc. Int. Computer Music Conf. (ICMC)*, Thessaloniki, GR, Sept. 1997.

[126] T. S. Verma and T. H. Y. Meng, "An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Seattle, WA, US, May 1998.

[127] T. S. Verma and T. H. Y. Meng, "Sinusoidal modeling using frame-based perceptually weighted matching pursuits," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Phoenix, AZ, US, Mar. 1999.

[128] T. S. Verma and T. H. Y. Meng, "A 6kbps to 85kbps scalable audio coder," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Istanbul, TR, June 2000.

[129] T. Verma, *A Perceptually Based Audio Signal Model With Application to Scalable Audio Compression*, Ph.D. thesis, Stanford University, CA, US, 2000.

[130] E. Zwicker and H. Fastl, *Psychoacoustics – Facts and Models*, Springer, Berlin, DE, 2nd edition, 1999.

Lebenslauf

Heiko Purnhagen

02. 04. 1969	geboren in Bremen als Sohn von Heinz Purnhagen, Hochschullehrer, und Karin Purnhagen, geb. Bröhland, Fremdsprachensekretärin
1975 – 1987	Schulausbildung in Bremen
03. 06. 1987	Abitur am Gymnasium Horn, Bremen
10/1987 – 09/1994	Studium der Elektrotechnik an der Universität Hannover, Studienschwerpunkt Nachrichtenverarbeitung
	Industriepraktika:
07/1987 – 10/1987	Krupp Atlas Elektronik, Bremen
02/1989 – 03/1989	Marine Electronic, Bremen
08/1991 – 10/1991	British Telecom Laboratories, Martlesham Heath, England
18. 09. 1989	Vordiplom Elektrotechnik (mit Auszeichnung)
04/1990 – 05/1994	Stipendiat der Studienstiftung des deutschen Volkes
09/1993 – 05/1994	Diplomarbeit an der Norwegischen Technische Hochschule, Trondheim
30. 05. 1994	Diplom Elektrotechnik (mit Auszeichnung)
10/1994 – 12/1995	Zivildienst in einer Wohngruppe für Körperbehinderte, Hannover
01/1996 – 10/2002	Wissenschaftlicher Mitarbeiter im Laboratorium für Informationstechnologie der Universität Hannover, Abteilung Systemtechnik
seit 11/2002	Senior Research Engineer Coding Technologies AB, Stockholm, Schweden (seit 11/2007: Sr Member Technical Staff, Dolby Sweden AB)
seit 1996	Mitarbeit im ISO/IEC Standardisierungsausschuß “MPEG” Leitung der Arbeitsgruppe “MPEG-4 Audio Reference Software” Co-Editor der Standards ISO/IEC 14496-3, -4, -5 und 23003-1, -2